Multi-channel target speech enhancement based on ERB-scaled spatial coherence features

Yicheng Hsu; Yonghan Lee; Mingsian R. Bai

1 Department of Power Mechanical Engineering, National Tsing Hua University, Taiwan
2 Electrical Engineering, National Tsing Hua University, Taiwan

ABSTRACT
Recently, speech enhancement technologies that are based on deep learning have received considerable research attention. If the spatial information in microphone signals is exploited, microphone arrays can be advantageous under some adverse acoustic conditions compared with single-microphone systems. However, multichannel speech enhancement is often performed in the short-time Fourier transform (STFT) domain, which renders the enhancement approach computationally expensive. To remedy this problem, we propose a novel equivalent rectangular bandwidth (ERB)-scaled spatial coherence feature that is dependent on the target speaker activity between two ERB bands. Experiments conducted using a four-microphone array in a reverberant environment, which involved speech interference, demonstrated the efficacy of the proposed system. This study also demonstrated that a network that was trained with the ERB-scaled spatial feature was robust against variations in the geometry and number of the microphones in the array.

Keywords: multi-channel target speech extraction, spatial features, microphone array

1. INTRODUCTION
Recently, monaural speech enhancement technologies that are based on deep learning have demonstrated promising results compared with traditional signal processing methods. Most advanced approaches operate in the short-time Fourier transform (STFT) domain, and they estimate real-valued masks [1, 2] or complex masks [3] using a deep neural network. However, time–frequency-masking-based approaches are ineffective in canceling out noise between speech harmonics, and they consume considerable computational resource. To solve these problems, PercepNet [4] uses a triangular equivalent rectangular bandwidth (ERB) filter band, and it applies a comb filter for the finer enhancement of periodic components of speech. DeepFilterNet [5], which is a two-stage speech enhancement framework, uses ERB-scaled gains to enhance the speech envelope, and it employs deep filtering to enhance periodic components. Moreover, target speech enhancement methods [6–8] that utilize auxiliary speaker information to address privacy issues and handle overlapped speech conditions are increasingly being developed. The auxiliary information can be obtained from pre-enrolled utterances of the target speaker [9–11], video imagery of the target speaker [12], electroencephalogram signals [13], and speech activities of the target speaker [14].

Although the aforementioned monaural approaches are effective in extracting close-talking speech, the signal received at the microphone is heavily distorted in far-field applications such as hands-free teleconferencing and smart speakers. Speech enhancement performance can be improved by the use of additional spatial information, which can be provided by a microphone array. For instance, the direction-aware SpeakerBeam presented in [15] combines an attention mechanism with beamforming. The neural spatial filter proposed in [16] uses directional information while extracting the target speech. The time-domain SpeakerBeam (TD-SpeakerBeam) presented in [17] employs interchannel phase differences as additional input features to increase the speaker separation capability. Instead of ad hoc spatial features [18, 19], suitable spatial feature estimates can be obtained from multichannel microphone signals by using a trainable spatial encoder. Although these methods can utilize spatial information, the training models can be used only for a microphone array that is identical to the...
training set. Therefore, a previous study introduced an array-geometry-agnostic personal speech enhancement model [20] that works regardless of the number of microphones that are used or the type of array configuration that is applied. Moreover, we recently proposed a target speech lifting network [21] that is based on a long-short-term spatial coherence (LSTSC) feature; target speaker enrollment data were used to demonstrated the effectiveness of the network in speech enhancement, and it remains robust when changes are made to the array geometry and number of microphones used. However, to the best of our knowledge, no study has investigated ERB-scaled spatial information.

To address the aforementioned research gap, the present study developed a geometry-agnostic multichannel target speech enhancement system, which utilizes spatial features and pre-enrolled utterances as inputs, based on DeepFilterNet [5]. In this system, a novel ERB-scaled LSTSC feature is computed as a spatial feature in relation to the speaker activity pertaining to each ERB band, which is derived from our previous work [21]. Spatially varying source signals are extracted by the ERB encoder layers, and speaker-dependent information is used to further extract the target speaker signals from the mixture. To assess the effectiveness of the proposed system, we explored its robustness against unseen array geometries and determined the influence of the number of microphones, including a single-microphone setup with no spatial information, on its performance. Short-time objective intelligibility (STOI) [22] and perceptual evaluation of speech quality (PESQ) [23] were employed as performance metrics in the experiments.

2. MULTI-CHANNEL TARGET SPEECH ENHANCEMENT SYSTEM

2.1 LSTSC feature

Consider one static interference source and one target speaker in a reverberant room. The signals are received by a microphone array containing \( M \) elements and are analyzed in the STFT domain. Assume that the target speaker and a spatially stationary interference source coexist in a room. The problem can be formulated in the STFT domain, with \( l \) denoting the time index and \( f \) denoting the frequency index. The signal captured by the \( m \)th microphone can be expressed as

\[
Y^m(l, f) = \sum_{j=1}^{J} A^m_j(f) S_j(l, f) + V^m(l, f),
\]

where \( m \in \{1, \ldots, M\} \) denotes a given microphone, \( l \in \{1, \ldots, L\} \) denotes the time frame, \( f \in \{1, \ldots, F\} \) denotes the frequency bin, \( Y^m(l, f) = A^m_j(f) S_j(l, f) \) denotes the signal of the \( j \)th source measured by the \( m \)th microphone, \( A^m_j(f) \) denotes the acoustic transfer function relating the \( j \)th source and the \( m \)th microphone, \( S_j(l, f) \) denotes the signal of the \( j \)th source, and \( V^m(l, f) \) denotes the nondirectional noise measured by the \( m \)th microphone.

For each TF bin, the short-term relative transfer function (RTF) between the \( m \)th microphone and reference microphone 1 can be estimated by averaging \( (R + 1) \) frames:

\[
\tilde{R}^m(l, f) = \frac{\hat{\Phi}_{x^m,y^m}}{\hat{\Phi}_{y^m,y^m}} = \frac{\sum_{l=1}^{R+1} Y^m(n, f)Y^m(n, f)}{\sum_{l=1}^{R+1} Y^m(n, f)Y^m(n, f)},
\]

where * denotes the complex conjugate operation, \( \hat{\Phi}_{x^m,y^m} \) denotes the short-time cross-spectral density estimate between channels \( m \) and 1, and \( \hat{\Phi}_{y^m,y^m} \) denotes the short-time autospectral density of the reference microphone. A “whitened” feature vector \( r(l, f) \in \mathbb{R}^{M-1} \) pertaining to each TF bin can be calculated as follows:

\[
r(l, f) = \left[ \frac{\tilde{R}^m(l, f)}{|\tilde{R}^m(l, f)|} \ldots \frac{\tilde{R}^1(l, f)}{|\tilde{R}^1(l, f)|} \right],
\]

where \(|*|\) is the complex modulus.

For a spatially stationary interference source, the following long-term RTF (which is computed
through recursive averaging) can be used to approximate the expectation of time-average of the feature vector:

\[ \bar{r}^m(l, f) = \lambda \bar{r}^m(l-1, f) + (1-\lambda)r^m(l, f), \quad m = 2, \ldots, M, \]  

(4)

where \( \lambda \) is the forgetting factor that facilitates the tuning between the global view (large \( \lambda \)) and the local view (small \( \lambda \)) of time frames. The feature vector \( \bar{r}(l, f) \) is also whitened after each recursive step:

\[ \bar{r}(l, f) = \left[ \frac{\bar{r}^2(l, f)}{\|\bar{r}^2(l, f)\|}, \ldots, \frac{\bar{r}^M(l, f)}{\|\bar{r}^M(l, f)\|} \right]^T. \]  

(5)

To fully exploit the temporal–spatial information conveyed by the whitened RTF, [21] we can calculate the LSTSC, \( \gamma_{lf}(l, f) \), between the short-term whitened feature vector \( \bar{r}(l, f) \) and the long-term whitened feature vector \( \bar{r}(l, f) \) as follows:

\[ \gamma(l, f) \approx \frac{1}{M-1} \sum_{m=2}^{M} \text{Re}\left\{ \bar{R}^m(l, f)\bar{r}^m(l, f)^* \right\} \approx \frac{1}{M-1} \text{Re}\left\{ \bar{r}^H(l, f)\bar{r}(l, f) \right\}, \]  

(6)

where \( \text{Re}\{\cdot\} \) denotes the real-part operator. The Euclidean angle [24] is adopted in the LSTSC definition to ensure sign sensitivity. The LSTSC is an indicator of the spatial correlation between the short-term RTF and long-term RTF, which are associated with the spatially stationary interference. LSTSC values with a large \( \lambda \) (global LSTSC) are used to sift out TF bins that either correspond to the active target or correspond to both the target and the interference, which are rendered inactive. By contrast, LSTSC values with a small \( \lambda \) (local LSTSC) are used to identify TF bins that correspond to the directional sources.

### 2.2 ERB-scaled LSTSC

In this section, we present the novel ERB-scaled LSTSC feature derived from the aforementioned LSTSC feature. On the basis of the equivalent rectangular bandwidth (ERB) for human hearing [25], the dimensions of a noisy signal can be reduced to 16 bands:

\[ Y_{ERB}(l, b) = \sum_{f=0}^{F_{b}} w_b(f)\|Y(l, f)\|^2, \quad b \in \{0,1,\ldots,16\}, \]  

(7)

where \( w_b(f) \) is the weight of the frequency bins for band \( b \) and \( F_{b} \) is the number of frequency bins for band \( b \). Therefore, the dimensions of the LSTSC feature can also be reduced to 16 bands of an ERB-scaled spatial coherence feature:

\[ \gamma_{ERB}(l, b) = \frac{1}{\pi_b} \sum_{f=0}^{F_{b}} w_b(f)\|\gamma(0, f)\|, \quad b \in \{0,1,\ldots,16\}, \]  

(8)

where \( \pi_b \) denotes a weight normalization and is expressed as

\[ \pi_b = \sum_{f=0}^{F_{b}} w_b(f). \]  

(9)

### 2.3 Speaker encoder

The speaker encoder generates a speaker embedding vector from pre-enrolled utterances of the target speaker. A speaker embedding can be extracted using a speaker encoder that is trained using the target speech model or a pretrained model for the extraction of speaker information such as the i-vector [26], x-vector [27], or d-vector [28]. In this study, we used the d-vector, which has been successfully used in various applications such as speaker diarization, speech synthesis, personal voice activity detection, and source separation. The proposed model, containing a three-layer long–short-term memory network followed by a projection layer, was trained using a generalized end-to-end loss function [28], and the speaker encoder was trained using the VoxCeleb2 data set [29]. The model yields embeddings
in sliding windows. The resulting aggregated embedding, which is known as the d-vector, encodes the target speaker’s voice characteristics.

2.4 ERB-based multi-channel target speech sifting network

DeepFilterNet [5] is a two-stage deep filtering approach that uses a complex filtering instead of a point-wise multiplication with a mask. In this study, DeepFilterNet was extended yield to multichannel personalized DeepFilterNet. As illustrated in Fig. 1, this network has four inputs: the real and imaginary parts of complex spectrogram features, ERB-scaled spectral feature computed for the reference microphone, d-vector of the target speaker generated by the speaker encoder, and ERB-scaled spatial coherence calculated from the array signal. In this network, an ERB encoder/decoder architecture is used to predict ERB-scaled gains, which can suppress the spatially stationary persistent interference. The d-vector is concatenated to the middle layer to sift the latent features pertaining to the target speaker. To further enhance the periodic components, DeepFilterNet predicts per-bin filter coefficients.

Figure 1 – ERB-based multi-channel target speech sifting architecture

3. EXPERIMENTAL STUDY

3.1 Data preparation

To validate the proposed multichannel target speech enhancement system, we trained our model using simulated room impulse responses (RIRs) and tested it using measured RIRs. The measured RIRs were based on the recently proposed Tampere University Rotated Circular Array Dataset [30], which contains 777 RIRs. The simulated RIRs were generated using the image-source method [31], with reverberation time (T60) being set to 0.32, 0.48, and 0.60 s. The microphone array was placed at the center of the room, and the target speaker and interference were randomly positioned around the array (within 0.5–2.5 m) under the assumption that the target speaker is always close to the array. In the training and validation sets, a four-element uniform circular array (UCA) of a radius of 4 cm was chosen. In the test set, three array configurations were employed to evaluate the robustness of the proposed enhancement system. The array geometry is illustrated in Fig. 2.

We used data from publicly available data sets and convolved them with the measured and simulated RIRs. Clean utterances that are necessary for training and testing were selected from the train-clean-360 and dev-clean subsets of the LibriSpeech corpus [32], which contain utterances from 921 and 40 speakers, respectively. We generated noisy training and test data using the VoxConverse data set [33], from which 74-h human-conversation clips from YouTube were chosen. The audio contained noise of various types, such as background noise, music, laughter, and applause.

In the training phase, a reference speech signal, which was derived from the utterances of the target speaker and different from the clean signal, was randomly selected for enrollment. Noisy audio signals in the form of 8-s clips were prepared by mixing clean target speech signals and interferences to yield signal-to-noise ratios (SNRs) of −5, 0, 5, and 10 dB. The sampling rate for all utterances was
16 kHz. Furthermore, the sample size of the training, validation, and test sets were 50,000, 5,000, and 5,000, respectively. The STFT settings were a 32-ms window length, a 16-ms hop size, and a 512-point fast-Fourier transform. The 16-dimensional ERB spectral feature was calculated from the noisy signal that was captured by the reference microphone and the proposed ERB-scaled spatial coherence feature was calculated on the basis of the ERB-scaled LSTSC in Eq. (8). In this experiment, the forgetting factors were set at $\lambda = 0.999$ and 0.01 to calculate the global ERB-scaled spatial coherence (ERB-G-LSTSC) feature and the local ERB-scaled (ERB-L-LSTSC) feature.

Figure 2 – Array geometries that were utilized in the training and test sets.

3.2 Results and discussions

Figure 3 presents the performance of the proposed model for different array geometries at different SNRs. We observed a considerable improvement in the performance of the proposed model when array geometries that were identical to the training set were applied. Furthermore, the proposed model was effective when array geometries that were not used during training were applied. These results suggest that the multichannel target speech enhancement system based on the proposed ERB-scaled LSTSC feature is robust against array geometry variations, which is a desirable property in real-world applications.

![Figure 3 – PESQ and STOI scores for different array geometries.](image)

To determine the influence of the number of microphones on the proposed system, four different configurations with one to four elements (Fig. 4) were used to assess the robustness of the system. For the proposed ERB-scaled spatial coherence feature, a change in the number of microphones engenders a change in only the feature vector dimensions in Eqs. (3) and (5) but not the input dimension in Eq. (8). As indicated in Table 2, increasing the number of microphones was advantageous to enhancement performance at the various SNRs. The monaural speech enhancement system (representing the approach with no spatial information) yielded the lowest performance, especially for scenarios with low SNRs. This result demonstrates that the proposed ERB-scaled LSTSC feature can make the model independent of the array configurations. Without the need to change the model architecture for a specific array configuration, a single model with the proposed ERB-scaled LSTSC feature can be shared by multiple arrays with different array geometries and numbers of microphones.

![Figure 4 – Array geometries with different numbers of microphones.](image)
4. CONCLUSIONS
We propose a multichannel target speech enhancement system that is based on ERB-scaled spatial coherence features along with speaker embedding. Our results demonstrate that the novel ERB-scaled spatial feature is useful for target speaker speech enhancement as well as for system robustness against unseen RIRs, unseen array geometries, and changes in the number of microphones used, which is highly desirable for real-world applications. The use of the ERB-scaled LSTSC feature can effectively reduce the computational resource of the proposed speech enhancement system, rendering it compatible with embedded devices.

ACKNOWLEDGEMENTS
This work was supported by the Ministry of Science and Technology (MOST), Taiwan, under the project number 110-2221-E-007-027-MY3.

REFERENCES
11. Z. Zhang, B. He, and Z. Zhang, “X-tasnet: Robust and accurate time-domain speaker extraction network,”
Statistical beamforming based on AuxIVA with distortionless and null constraints for robust speech recognition

Ui-Hyeop SHIN(1), Hyung-Min PARK(1)

(1)Department of Electronic Engineering, Sogang University, Seoul 04107, Republic of Korea

ABSTRACT

In this paper, we present a generalized beamforming algorithm based on auxiliary-function-based independent vector analysis (AuxIVA) for robust speech recognition. The beamforming is derived by imposing distortionless and null constraints on target speech at the target output and noises at the other outputs of the cost of AuxIVA, respectively. The filter for the target output results in a beamformer with the weighted covariance matrix by a weighting function calculated by a target speech model, which may consider various models used in the conventional independent component analysis (ICA)/IVA, and may include the conventional beamformers such as minimum-power distortionless response (MPDR) and maximum-likelihood distortionless response (MLDR) beamformers as special cases. Based on the generalization using AuxIVA, various statistical beamformers are compared depending on various source models used in ICA/IVA including Independent low-rank matrix analysis (ILRMA) as well as MPDR and MLDR. Consequently, a generalized beamforming algorithm is proposed with a frequency-independent source model considering the sparsity of speech with source variances. Various beamformers derived from AuxIVA are experimentally compared on the CHiME-4 challenge dataset and proposed source model shows the effectiveness as a pre-processing method for robust speech recognition.

Keywords: Independent component analysis, beamforming, robust speech recognition

1 INTRODUCTION

Noise robustness has been an important issue for automatic speech recognition (ASR) [1, 2]. If multi-microphone data are available, beamforming methods have achieved significant performance improvement as a pre-processing step for noise-robust ASR by enhancing target speech [3, 4]. Especially, the minimum-variance distortionless response (MVDR) beamformer has been a popular choice because it efficiently enhances the target speech without distortion while minimizing a noise power [5]. Because the noise covariance matrix (NCM) is hard to estimate accurately with active target speech for MVDR beamforming, the minimum-power distortionless response (MPDR) beamformer minimizing the whole power instead of the noise is practically adopted [4]. Recently, the maximum-likelihood distortionless response (MLDR) beamformer was proposed based on the maximum-likelihood estimation of a beamforming filter assuming that its beamformed output follows a complex Gaussian distribution with time-varying variances (TVVs) [6]. The MLDR beamformer utilizes weighted covariance matrix (WCM) to estimate NCM. Although these beamformers require an accurate steering vector for a target speaker to achieve successful target enhancement, steering vector estimation (SVE) can be successfully performed by modeling multi-channel input data by a complex Gaussian mixture model (CGMM) [7].

On the other hand, many methods based on independent component analysis (ICA) can be considered as pre-processing for noise-robust ASR because of their successful applications to the blind source separation (BSS) problem (e.g. [8, 9, 10]). However, because it has a problem that separated sources are randomly permuted at every frequency bins, independent vector analysis (IVA) and independent low-rank matrix analysis (ILRMA) and is proposed to solve the permutation problem by introducing a source model assuming the dependency of frequency components [11, 12, 13, 14]. In terms of optimization, based on auxiliary function techniques, a fast and robust update rule was developed, AuxIVA [15]. From the conventional BSS approaches, a class of blind source extraction (BSE) has been studied to obtain target speech at a specified output for ASR. Many BSE
approaches are based on imposing spatial constraints on steering vectors for sources [16, 17, 18] or modelling the target speech and noises distinctly [19, 20]. However, the most BSE approaches are based on IVA exploiting the strict source assumptions, which does not necessarily help with ASR pre-processing.

A target speech extraction method was developed by estimating weights to extract target speech that is as independent as possible from the other noise outputs obtained by forming a directional null to the target speaker [8]. In this paper, based on AuxIVA with a distortionless constraint on the target speaker in addition to the null constraint for the other noise outputs, we present a statistical beamforming algorithm. The resulting beamformer is derived from the WCM with a weighting function characterized by a target source model, which may consider various models used in the conventional ICA/IVA [14], and may include the conventional beam-formers as special cases. In the beamforming framework where beamforming parameters are obtained based on the steering vector, the target speech extraction may be improved by using a frequency-independent source model considering the sparsity of speech. As a result, an efficient statistical beamforming algorithm is proposed.

2 CONVENTIONAL BEAMFORMING

In real-world environments with ambient background noises, $M$ noisy speech observations at frequency bin $k$ and frame $\tau$, $x(k, \tau)$, can be expressed as $x(k, \tau) = h(k)S(k, \tau) + n(k, \tau)$, where $S(k, \tau)$ and $h(k)$ denote the time-frequency segment of target speech and its steering vector containing acoustic transfer functions to the microphones from the target source, respectively. $n(k, \tau)$ is noise components in the observations $x(k, \tau)$. In the conventional beamforming, a linear filter $w(k)$ is applied to the observations to obtain an enhanced speech signal: $Y(k, \tau) = w^H(k)x(k, \tau)$, where $H$ denotes the Hermitian operation on the vector.

Under the distortionless constraint of $w^H(k)h(k) = 1$ that keeps the gain of the target direction at unity to preserve the target source signal at the output, the MPDR beamformer minimizes the power of filtered output $\frac{1}{T} \sum_{\tau=1}^{T} |w^H(k)x(k, \tau)|^2$, which can be formulated by the following constrained optimization problem:

$$w(k) = \arg \min_{w(k)} \frac{1}{T} \sum_{\tau=1}^{T} |w^H(k)x(k, \tau)|^2, \text{ subject to } w^H(k)h(k) = 1,$$

where $T$ denotes the number of frames. Using the Lagrange multiplier method, the beamforming weight vector can be given by [4]

$$w(k) = \frac{V_p^{-1}(k)h(k)}{h^H(k) V_p^{-1}(k)h(k)},$$

where $V_p(k)$ is defined by $V_p(k) = \sum_{\tau=1}^{T} x(k, \tau)x^H(k, \tau)$.

However, because the MPDR beamformers cannot exploit the statistical distribution of target speech different from noises, the MLDR beamformer is presented based on the probabilistic modeling the target speech by a complex Gaussian distribution with TVVs [6] expressed as

$$q(Y(k, \tau)) \propto \frac{1}{\lambda(k, \tau)} \exp\left(-\frac{|Y(k, \tau)|^2}{\lambda(k, \tau)}\right).$$

where $\lambda(k, \tau)$ denotes a TVV. To maximize the log-likelihood function of (3) under the distortionless constraint, optimal parameter values of the variance $\lambda(k, \tau)$ and the MLDR beamforming weight vector $w(k)$ can be obtained by iterative update: [6]

$$w(k) = \frac{V_L^{-1}(k)h(k)}{h^H(k) V_L^{-1}(k)h(k)},$$

where $V_L(k)$ is the WCM calculated by $V_L(k) = \sum_{\tau=1}^{T} x(k, \tau)x^H(k, \tau)/\lambda(k, \tau)$. The TVVs can be directly estimated from its own output powers. Furthermore, estimation of the TVVs can be improved by introducing the moving average at adjacent frames to enhance temporal continuity as $\lambda(k, \tau) = \frac{1}{2n_0+1} \sum_{\tau'=\tau-n_0}^{\tau+n_0} |Y(k, \tau')|^2$, where $2n_0+1$ is the number of adjacent frames to be averaged [6]. If the TVVs are fixed to one, the $V_L(k)$ becomes the input covariance matrix $V_p(k)$ in the MPDR beamformer. Therefore, the MLDR beamformer can be viewed as a generalized version of the MPDR beamformer with varying source variances.
3 CONVENTIONAL BSE METHODS

While the beamforming methods require a steering vector for target speech, BSE based on ICA/IVA can be considered to extract the target speech regardless of availability of the steering vector. In the BSE formulation, $M$ noisy speech observations at frequency bin $k$ and frame $\tau$, $x(k,\tau)$, can be expressed as $x(k,\tau) = A(k)S(k,\tau)n^T(k,\tau)^T$, where $n(k,\tau)$ denotes a source vector to generate the noise vector $\tilde{n}(k,\tau)$, and $A(k)$ represents the mixing matrix at frequency bin $k$. The mixing condition is assumed to be determined for simplicity. Then, let us consider the demixing model of $y(k,\tau) = W(k)x(k,\tau)$ where $y(k,\tau) = [Y_1(k,\tau), \cdots, Y_M(k,\tau)]^T$ denotes a vector of the time-frequency segments of demixed outputs, and $W(k) = [w_1(k) \cdots w_M(k)]^H$ is a demixing matrix at frequency bin $k$. Without loss of generality, let us assume that $Y_1(k,\tau) = S(k,\tau)$ whereas the others represent noise outputs. $W(k)$ can be estimated by minimizing the auxiliary function as [15]

$$Q_k = \frac{1}{2} \sum_{m=1}^{M} w_m^H(k) V_m(k) w_m(k) - \log |\det W(k)|. \quad (5)$$

$V_m(k)$ denotes the WCM computed by $V_m(k) = \frac{1}{T} \sum_{\tau=1}^{T} \phi_m(k,\tau)x(k,\tau)x^H(k,\tau)$, where $\phi_m(k,\tau)$ is a weighting function characterized by the source model [12]. The iterative projection algorithm to optimize (5) is well known as [12, 14, 15]

$$w_m(k) = (W(k)V_m(k))^{-1} e_m. \quad (6)$$

$$w_m(k) = w_m(k) / \sqrt{w_m^H(k)V_m(k)w_m(k)}. \quad (7)$$

Generally, the Laplace distribution has been adopted in the conventional BSE methods [8, 10], expressed as $q(Y_1(k,\tau)) \propto \exp(-|Y_1(k,\tau)|\lambda(k,\tau))$. Then, the weighting function for the source is given as $\phi_1(k,\tau) = 1/(2|Y_1(k,\tau)|\lambda(k,\tau))$. Instead of the stationary Laplace distribution, non-stationarity of speech can be modeled by the Gaussian distribution with TVVs of (3) and its weighting function is $\phi_1(k,\tau) = 1/\lambda(k,\tau)$. On the other hand, noises are usually modeled by a stationary Gaussian distribution given as $q(Y_2(k,\tau),\ldots,Y_M(k,\tau)) \propto \exp(-\sum_{m=2}^{M}|Y_m(k,\tau)|^2)$. Since the weighting functions $\phi_m(k,\tau), 2 \leq m \leq M$ for the noises become one, the WCM is calculated as $V_m(k) = \frac{1}{T} \sum_{\tau=1}^{T} x(k,\tau)x^H(k,\tau)$. Moreover, if the target steering vector $h(k)$ is available, the geometric constraint to form a directional null on the target source can be simply imposed by $V_m(k) = \frac{1}{T} \sum_{\tau=1}^{T} x(k,\tau)x^H(k,\tau) + p_n h(k) h^H(k)$ where $p_n$ denotes a parameter weighting the importance of the constraint [17, 18].

However, the frequency-domain ICA suffers from the problem of random permutation on frequency bins. The problem can be overcome by IVA adopting a multivariate Gaussian distribution with a variance shared for all the frequency bins for the source model to impose inter-frequency dependency: [12, 19, 20]

$$q(\tilde{Y}_1(\tau)) \propto \frac{1}{L^K(\tau)} \exp \left\{ -\frac{\sum_{k=1}^{K} |Y_1(k,\tau)|^2}{\lambda(\tau)} \right\}, \quad (8)$$

where $\tilde{Y}_1(\tau) = [Y_1(1,\tau),\ldots,Y_1(K,\tau)]^T$ and the variance can be estimated by $\lambda(\tau) = \frac{1}{T} \sum_{k=1}^{K} |Y_1(k,\tau)|^2$. Then, the weighting function for $V_1(k)$ is given by $\phi_1(k,\tau) = 1/\lambda(\tau)$. Although its promising performance by stably resolving the permutation problem makes the distribution widely adopted, it may have inherent limitations due to the too strict assumption that the variance is shared for all the frequency bins.

To alleviate the strict assumption in the conventional IVA, ILRMA considers the low-rankness of source variances decomposed into spectral bases and their time activations [13]. The target source can be modeled as

$$q(\tilde{Y}_1(\tau)) \propto \frac{1}{\prod_{k=1}^{K} \sum_{l=1}^{l} t(k,l)v(l,\tau)} \exp \left\{ -\sum_{k=1}^{K} \frac{|Y_1(k,\tau)|^2}{\sum_{l=1}^{l} t(k,l)v(l,\tau)} \right\}, \quad (9)$$

where $t(k,l)$ and $v(l,\tau)$ denote the spectral basis and the corresponding time activation at the $l$-th rank index. As a result, the weighting functions are calculated as $\phi_1(k,\tau) = 1/\sum_{l=1}^{l} t(k,l)v(l,\tau)$, where the parameters can be updated by non-negative matrix factorization (NMF) [13]. Compared to the IVA, the ILRMA adopts a more flexible source model, but still strict for pertaining inter-frequency dependency.
4 GENERALIZED STATISTICAL BEAMFORMING

4.1 Statistical beamforming based on the auxiliary function with distortionless and null constraints

In the BSE formulation of Section 3, let us assume that the mixing matrix is decomposed into \( \mathbf{A}(k) = [\mathbf{h}(k) \mid \mathbf{D}(k)] \) to get \( \mathbf{x}(k, \tau) = \mathbf{h}(k) \mathbf{S}(k, \tau) + \mathbf{D}(k) \mathbf{n}(k, \tau) \), where \( \mathbf{h}(k) \) is the steering vector for target speech as in Section 2 and \( \mathbf{D}(k) \) is an \( M \times (M - 1) \) matrix so that \( \mathbf{h}(k, \tau) = \mathbf{D}(k) \mathbf{n}(k, \tau) \). Then, in the demixing model, distortionless and null constraints can be respectively derived as \( \mathbf{w}_1^H(k) \mathbf{h}(k) = 1 \) and \( \mathbf{w}_m^H(k) \mathbf{h}(k) = 0 \), \( 2 \leq m \leq M \). Therefore, the auxiliary function for the \( k \)-th frequency bin with the constraints is given as

\[
Q_k = \frac{1}{2} \sum_{m=1}^{M} \mathbf{w}_m^H(k) \mathbf{V}_m(k) \mathbf{w}_m(k) - \log |\det \mathbf{W}(k)| + \sum_{m=1}^{M} \alpha_m(k) (\mathbf{w}_m^H(k) \mathbf{h}(k) - \beta_m),
\]

where \( \beta_1 = 1, \beta_m = 0, 2 \leq m \leq M \), and \( \alpha_m(k) \) denotes the Lagrange multiplier for the \( m \)-th constraint.

\( \mathbf{w}_m(k) \) to minimize (10) can be obtained by a solution of the equation given as

\[
\mathbf{V}_m(k) \mathbf{w}_m(k) - \frac{\partial}{\partial \mathbf{w}_m} \log |\det \mathbf{W}(k)| + \alpha_m(k) \mathbf{h}(k) = 0.
\]

Then, (11) can be rearranged to give

\[
\mathbf{W}(k) \mathbf{V}_m(k) \mathbf{w}_m(k) = \mathbf{e}_m - \alpha_m(k) \mathbf{e}_1 \text{ where } \mathbf{e}_m \text{ denotes a unit vector whose } m \text{-th element is unity.}
\]

To find the Lagrange multiplier \( \alpha_m(k) \), pre-multiplying \( \mathbf{h}_1^H(k) (\mathbf{W}(k) \mathbf{V}_m(k))^{-1} \) to both the sides of the equation gives

\[
\alpha_m(k) = \frac{\mathbf{h}_1^H(k) (\mathbf{W}(k) \mathbf{V}_m(k))^{-1} \mathbf{e}_m - \beta_m}{\mathbf{h}_1^H(k) (\mathbf{W}(k) \mathbf{V}_m(k))^{-1} \mathbf{e}_1}.
\]

When \( m = 1 \), \( \mathbf{w}_1(k) \) for the target output is simply obtained by replacing \( \beta_1 \) to unity. Since \( \mathbf{A}(k) \mathbf{e}_1 = \mathbf{W}^{-1}(k) \mathbf{e}_1 = \mathbf{h}(k) \), it is reduced to

\[
\mathbf{w}_1(k) = \frac{\mathbf{V}_1^{-1}(k) \mathbf{h}(k)}{\mathbf{h}_1^H(k) \mathbf{V}_1^{-1}(k) \mathbf{h}(k)}.
\]

which has the same form as (2) and (4). Actually, the “statistical beamforming” filter based on the auxiliary function with distortionless and null constraints, given by (13), may provide a generalized beamformer by choosing a specific source distribution. Note that (13) becomes the exactly same as (2) and (4) when \( \phi_1(k, \tau) \) is unity and the reciprocal value of the variance, respectively. Furthermore, although most of the BSE methods for robust ASR require a projection-back technique or normalization of demixing matrices by the minimal distortion principle (MDP) to avoid the scale indeterminacy of ICA/IVA \cite{8, 21}, the target output \( \mathbf{Y}_1(k, \tau) = \mathbf{w}_1^H(k) \mathbf{x}(k, \tau) \) by the generalized beamformer can be directly used for ASR because of the distortionless constraint included in the auxiliary function, similar to the conventional beamformers. On the other hand, the update of output weights \( \mathbf{w}_m(k), m = 2, \ldots, M \) for noises can be omitted because \( \mathbf{w}_1(k) \) from (13) does not affected.

4.2 Generalized AuxIIVA- and ILRMA-based beamformers

Since statistical beamforming based on the auxiliary function with spatial constraints results in a generalized beamformer in which various source models can be adopted, we may consider not only conventional source models commonly used in BSE but also super-Gaussian distributions with TVVs to exploit both non-stationarity and sparsity of speech.

A generalized source model of (8) can be expressed as

\[
q(\tilde{y}_1(\tau)) \propto \frac{1}{\hat{\lambda}(\tau)} \exp \left\{ -\frac{\sum_{k=1}^{K} |\mathbf{V}_1(k, \tau)|^{2\gamma}}{\hat{\lambda}(\tau)} \right\},
\]

where \( \gamma \) denotes the shape parameter. The shared TVV is directly estimated as \( \hat{\lambda}(\tau) = \left( \frac{1}{K} \sum_{k=1}^{K} |\mathbf{V}_1(k, \tau)|^{2\gamma} \right)^{1/\gamma} \). Similar to \cite{12, 14}, the weighting function in \( \mathbf{V}_1(k) \) is given by \( \phi_1(k, \tau) = \gamma |\mathbf{V}_1(k, \tau)|^{2\gamma-2\hat{\lambda}(\tau)} \). Since the
conventional IVA source model of (8) is simply obtained when \( \gamma = 1 \), we will call the beamformer for (8) “IVA beamformer”. On the other hand, with \( \gamma = 1/2 \) to impose source sparsity, we can obtain the weighting function of \( \phi_1(k, \tau) = 1/(2\sqrt{\lambda(k, \tau)}|Y_1(k, \tau)|) \), which results in “IVA-S beamformer”. This function reflects the absolute value output at the corresponding frequency bin while the shared TVS is still contained.

For ILRMA-based beamformers, the source can be modeled as

\[
q(\hat{y}_i(\tau)) \propto \frac{1}{\prod_{k=1}^{K} \sum_{l} t(k, l) v(l, \tau)} \exp \left\{ -\frac{1}{2} \sum_{k=1}^{K} \frac{|Y_1(k, \tau)|^2}{\sum_{l} t(k, l) v(l, \tau)} \right\}. \tag{15}
\]

Thus, similar to [13, 14], the NMF update rules for \( t(k, l) \) and \( v(l, \tau) \) are obtained as

\[
t(k, l) = t(k, l) \frac{\sum_{T=1}^{T} \gamma |Y_1(k, \tau)|^2 v(l, \tau) (\sum_{l} t(k, l) v(l, \tau))^{-1}}{\sum_{l} t(k, l) v(l, \tau) (\sum_{l} t(k, l) v(l, \tau))^{-1}}, \tag{16}
\]

\[
v(l, \tau) = v(l, \tau) \frac{\sum_{k=1}^{K} \gamma |Y_1(k, \tau)|^2 t(k, l) (\sum_{l} t(k, l) v(l, \tau))^{-1}}{\sum_{k=1}^{K} t(k, l) (\sum_{l} t(k, l) v(l, \tau))^{-1}}. \tag{17}
\]

Then, we can calculate the weighting function as \( \phi_1(k, \tau) = \gamma |Y_1(k, \tau)|^2 \sum_{l} t(k, l) v(l, \tau) \). “ILRMA beamformer” based on the conventional ILRMA formulation can be obtained with \( \gamma = 1 \) while we can impose sparsity on the target source by \( \gamma = 1/2 \) to get “ILRMA-S beamformer” whose weighting function is calculated as \( \phi_1(k, \tau) = 1/(2\sqrt{\sum_{l} t(k, l) v(l, \tau)}|Y_1(k, \tau)|) \).

### 4.3 Generalized ICA-based beamformers

Although source models assuming the dependency of frequency components are considered in the previous section, the spatial constraints on the steering vector may avoid the permutation problem of the frequency-domain ICA. In particular, independent distributions at every frequency bins can provide more flexible modeling of sources than source models for IVA. As a source model, a generalized super-Gaussian distribution with TVVs can be considered as follows:

\[
q(Y_1(k, \tau)) \propto \frac{1}{\lambda(k, \tau)} \exp \left\{ -\frac{|Y_1(k, \tau)|^{2\gamma}}{\lambda(k, \tau)} \right\}. \tag{18}
\]

For the distribution, the weighting function \( \phi_1(k, \tau) \) is determined by \( \phi_1(k, \tau) = \gamma |Y_1(k, \tau)|^{2\gamma - 2} \lambda^{-\gamma}(k, \tau) \). Source variances can be directly estimated as \( \lambda(k, \tau) = \gamma^2 |Y_1(k, \tau)|^2 \) in the maximum-likelihood sense, and robust estimation of \( \lambda(k, \tau) \) can be improved by introducing the moving average at adjacent frames, similarly to MLDR beamformer in Section 2.

When source variances are set to unity and \( \gamma = 1 \), the distribution becomes stationary Gaussian given as \( q(Y_1(k, \tau)) \propto \exp\{-|Y_1(k, \tau)|^2\} \), and the weighting function is unity. Therefore, the WCM becomes the input covariance matrix, and \( w_1(k) \) in (13) corresponds to the MPDR beamforming filter [4]. When source variances are assumed to be time-varying and \( \gamma = 1 \), the source model is expressed as (3). The weighting function is given by \( \phi_1(k, \tau) = 1/\lambda(k, \tau) \), which makes \( V_1(k) = \frac{1}{\lambda} V_L(k) \). Then, \( w_1(k) \) in (13) corresponds to the MLDR beamforming filter [6].

Considering the sparsity of the speech, the Laplacian distribution has been adopted in the conventional ICA [8, 10], expressed as \( q(Y_1(k, \tau)) \propto \exp\{-|Y_1(k, \tau)|\} \), which can be obtained when source variances are set to unity and \( \gamma = 1/2 \). Then, the weighting function is given by \( \phi_1(k, \tau) = 1/(2\sqrt{\lambda(k, \tau)}|Y_1(k, \tau)|) \). In this case, we refer to as “ICA beamformer”. Not only the sparsity, but also considering the non-stationarity of speech as in MLDR beamformer, the source can be modeled as

\[
q(Y_1(k, \tau)) \propto \frac{1}{\sqrt{\lambda(k, \tau)}} \exp\{-|Y_1(k, \tau)|/\sqrt{\lambda(k, \tau)}\}. \tag{19}
\]

when source variances are assumed to be time-varying and \( \gamma = 1/2 \). Then, the weighting function is given by \( \phi_1(k, \tau) = 1/(2\sqrt{\lambda(k, \tau)}|Y_1(k, \tau)|) \), which we will call as “ICA-TVV beamformer”.
Table 1. WERs (%) on the CHiME-4 dataset for enhancement by statistical beamformers (BFs) using different source models (IVA, IVA-S, ILRMA, ILRMA-S, MPDR, ICA, MLDR, and ICA-TVV) with steering vectors estimated by CGMMs. The sparsity denotes whether the source models regarded the sparsity of sources or not, which respectively corresponded to whether the distributions were Laplacian or Gaussian.

<table>
<thead>
<tr>
<th>Beamformer</th>
<th>Sparsity</th>
<th>Dev. Test</th>
<th>Average</th>
<th>Beamformer</th>
<th>Sparsity</th>
<th>Dev. Test</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVA BF</td>
<td>X</td>
<td>3.03</td>
<td>3.52</td>
<td>4.77</td>
<td>6.03</td>
<td>4.34</td>
<td></td>
</tr>
<tr>
<td>ILRMA BF</td>
<td>X</td>
<td>2.73</td>
<td>3.57</td>
<td>4.62</td>
<td>5.90</td>
<td>4.21</td>
<td></td>
</tr>
<tr>
<td>MPDR BF</td>
<td>X</td>
<td>2.94</td>
<td>3.17</td>
<td>4.78</td>
<td>5.63</td>
<td>4.13</td>
<td></td>
</tr>
<tr>
<td>MLDR BF</td>
<td>X</td>
<td>2.72</td>
<td>3.48</td>
<td>4.39</td>
<td>5.36</td>
<td>3.99</td>
<td></td>
</tr>
<tr>
<td>IVA-S BF</td>
<td>O</td>
<td>2.73</td>
<td>3.32</td>
<td>4.31</td>
<td>5.56</td>
<td>4.07</td>
<td>3.99</td>
</tr>
<tr>
<td>ILRMA-S BF</td>
<td>O</td>
<td>2.76</td>
<td>3.49</td>
<td>4.38</td>
<td>5.65</td>
<td>4.07</td>
<td></td>
</tr>
<tr>
<td>ICA BF</td>
<td>O</td>
<td>2.82</td>
<td>3.16</td>
<td>4.33</td>
<td>5.42</td>
<td>3.93</td>
<td></td>
</tr>
<tr>
<td>ICA-TVV BF</td>
<td>O</td>
<td>2.68</td>
<td>3.34</td>
<td>4.32</td>
<td>5.34</td>
<td>3.92</td>
<td></td>
</tr>
</tbody>
</table>

5 EXPERIMENTAL EVALUATION

The presented algorithms were evaluated by ASR experiments on the CHiME-4 challenge dataset [22]. The ASR system was constructed same as in [6]. A Hanning window was used as the analysis window of the short-time Fourier transform with a 1024-sample length and 256-sample shift on the data of a sampling rate of 16 kHz. For all experimented methods using the WCM for target sources \( V_1(k) \), \( \phi_1(k, \tau) \) is clipped by \( \phi_0 \) for the robust estimation, which was experimentally set to \( \phi_0 = 10^6 \).

5.1 Comparison of beamformers using various source models with the CGMM-based SVE

Various statistical beamformers (IVA, IVA-S, ILRMA, ILRMA-S, MPDR, ICA, MLDR, and ICA-TVV) were compared in Table 1. The steering vectors were estimated by CGMMs. For the MLDR and ICA-TVV beamformers, \( \tau_0 \) for TVV estimations were respectively set to 1 with the steering vectors fixed by the CGMMs. For the ILRMA and ILRMA-S beamformers, the number of bases was set to 50, and the number of iterations was also set to 50 for sufficient convergences of bases and activations which were randomly initialized. The other beamformers used 10 iterations.

The performances of the IVA/ILRMA-based beamformers imposing the inter-frequency dependency were generally inferior to those of the ICA-based beamformers (MPDR, ICA, MLDR, and ICA-TVV). This is because frequency-independent source modeling provided a higher degree of freedom to accurately model speech and the permutation problem could be avoided by the steering vectors in addition to smoothing along frequency bins to estimate source variances. In addition, the beamformers using the Laplacian distribution or TVVs generally showed lower WERs than those using the Gaussian distribution or constant variances, respectively, because speech is sparse and non-stationary. Finally, the proposed ICA-TVV beamformer with the source model considering both sparsity and non-stationarity of speech outperformed the other beamformers.

5.2 Comparison of BSE, semi-BSE, and beamforming methods as pre-processing for ASR

In Table 2, we compared several BSE, semi-BSE, and beamforming methods in terms of the WERs. The number of iterations was commonly set to 10. As a BSE method, Over-determined IVA (OverIVA) [20] was performed. We considered geometrically constrained IVA (GCIVA) [17] and direction-of-arrival constrained ICA (DCICA) [8] as semi-BSE methods utilizing pre-determined steering vectors. For fair comparison, the steering vectors \( h(k) \) were estimated by the CGMMs. The weighting parameters \( p_n \) was set to 10, and the OverIVA can be considered as a special case of the GCIVA with the weighting parameter of zero. \( W(k) \) of BSE methods is initialized by \( [h(k)|e_2, ..., e_M]^{-1} \). In the DCICA using a frequency-independent source model, only the weights for the target speech output were updated after \( W(k) \) was initialized to \( [h(k)|e_2, ..., e_M]^{-1} \). In addition, the DCICA method was repeated with a frequency-dependent source model of (14), simply named DCIVA. The DCIVA method was also considered because it can be viewed as a special case of the GCIVA with the weighting parameter of infinity. Since the semi-BSE methods did not use the distortionless constraint, their outputs were normalized by the MDP [21]. As beamforming methods, MPDR, MLDR, ICA, and ICA-TVV beamformers were compared using the same CGMM-based steering vectors as in the semi-BSE methods.

The OverIVA method showed very unstable results to obtain an average WER higher than the no processed
Table 2. WERs (%) on the CHiME-4 dataset for the baseline without any processing for input data acquired at the fifth microphone (similar to [6]) and enhancement by OverIVA, GCIVA, DCIVA, and DCICA as semi-BSE methods, MPDR, MLDR, ICA, and ICA-TVV beamformers (BFs) as beamforming methods. The steering vectors estimated by CGMM are commonly used for all the methods.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>5.51</td>
<td>6.52</td>
<td>6.71</td>
<td>11.63</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OverIVA</td>
<td>5.27</td>
<td>10.92</td>
<td>8.68</td>
<td>21.09</td>
<td>11.49</td>
<td>MPDR BF</td>
<td>2.94</td>
</tr>
<tr>
<td>GCIVA(10)</td>
<td>3.14</td>
<td>4.71</td>
<td>4.99</td>
<td>9.59</td>
<td>5.61</td>
<td>MLDR BF</td>
<td>2.72</td>
</tr>
<tr>
<td>DCIVA</td>
<td>2.88</td>
<td>4.17</td>
<td>4.72</td>
<td>7.14</td>
<td>4.72</td>
<td>ICA BF</td>
<td>2.82</td>
</tr>
<tr>
<td>DCICA</td>
<td>2.76</td>
<td>3.87</td>
<td>4.24</td>
<td>6.50</td>
<td>4.34</td>
<td>ICA-TVV BF</td>
<td>2.68</td>
</tr>
</tbody>
</table>

input despite $W(k)$ initialized by the CGMM. As the geometric constraint to form a directional null on the target source was imposed, the recognition rates of the GCIVA were improved, reaching an average WER of 4.72% in the DCIV corresponding to the case of the weighting parameter of infinity. By using a frequency-independent source model to provide a higher degree of freedom given fixed steering vectors, the DCICA showed better recognition performance than the DCIV. However, these semi-BSE methods that impose the null constraint on noise outputs followed by MDP normalization showed inferior recognition performance than the beamforming methods that directly use the distortionless constraint on target speech.

6 CONCLUSIONS

In this paper, we presented a statistical beamforming algorithm based on AuxIVA with distortionless and null constraints, used as a pre-processing step for robust ASR. The resulting beamformer may include the conventional beamformers such as MPDR and MLDR beamformers by considering generalized source models. Experimental results showed that the proposed beamformer outperformed the conventional beamformers and methods based on BSE by considering both sparsity and non-stationarity of speech.

ACKNOWLEDGEMENTS

This work was partly supported by Institute of Information & communications Technology Planning & evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00989, Development of artificial intelligence technology for multi-speaker dialog modeling) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2020R1A2B5B01002398).

REFERENCES


Music classification using automatic generation of artificial emotional EEG characteristics corresponding to auditory stimuli

Hyoung-Gook KIM\textsuperscript{1}; Dong-Ki JEONG\textsuperscript{1}; Jin Young KIM\textsuperscript{2}; Jörn FISCHER\textsuperscript{3}

\textsuperscript{1} Kwangwoon University, Republic of Korea
\textsuperscript{2} Chonnam National University, Republic of Korea
\textsuperscript{3} Mannheim University of Applied Sciences, Germany

ABSTRACT

Attempts to transplant the core elements of human intelligence into artificial intelligence algorithms so that machines can think like human brains are expanding into research and development to infuse ‘emotions’ into machines. Music is the medium that affects people emotionally. In this paper, we propose a method to extract emotional characteristics from human brain waves that appear when listening to music, and to automatically classify music as the listener feels by learning these human emotional characteristics by machine. In actual application, the proposed system classifies music by automatically generating artificial brainwave features that match the characteristics of the input music without the need to measure human brain waves. The experimental results show that the proposed method automatically classified the input music with high accuracy as felt by the person who gave emotional characteristics.

Keywords: Music Classification, EEG, Deep Neural Network, Attention-Mechanism

1. INTRODUCTION

With the creation of a mobile digital environment where music can be listened to without time and space constraints, the need for a music search and recommendation system is rapidly increasing. Accordingly, a technique that reflects appropriate search terms such as singer, lyrics, and album name, learning of the user's music selection tendency, and similarity of feature patterns extracted from music songs has been developed and applied. However, most music search and recommendation systems have been developed from a system-centric point of view rather than a user-centric system, and studies on the emotions or expressions of users who listen to music are still lacking. If a user's physiological signals that occur naturally when listening to music are applied to a music classification and recommendation system using deep learning techniques, which have recently received attention in the field of artificial intelligence, it is possible to maximize the satisfaction of users who listen to music in various situations. However, since it is inconvenient to measure physiological signals through wearable physiological sensors and devices whenever a user needs a recommendation, an alternative method is needed.

2. PROPOSED METHOD

Fig. 1 illustrates the process flow of the proposed emotion-based music classification system; it consists of two parts: the case where the EEG sensor is applied and that where it is not applied.

In the case where EEG sensor is applied, the signals obtained from the EEG sensor worn by music listeners are input to a sample-level inner attention-mechanism-based bidirectional gated recurrent unit (BGRU) encoder to extract emotional features and send them to a smartphone. In the smartphone, the extracted features are applied to a BGRU decoder with segment-level inner attention mechanism models to classify the music by emotion.

\textsuperscript{1} hkim@kw.ac.kr; jdklist@kw.ac.kr
\textsuperscript{2} beyond@jnu.ac.kr
\textsuperscript{3} j.fisher@hs-mannheim.de
In the case where EEG sensor is not applied, regression target features are selected among the previously generated EEG features, which are extracted from the sample-level inner attention-mechanism-based BGRU encoder. The correlation model between the selected regression target features and musical features is then generated by the multi-layer perceptron (MLP)-based regression training on the server and sent to the smartphone. During the application phase on the smartphone, musical features of the input music signal are entered into MLP-based regressors to automatically generate emotion features of EEG. Then, the EEG features are input into segment-level inner attention-mechanism-based BGRU decoder for emotion-based music classification.

3. EXPERIMENT AND RESULTS

To evaluate the performance of the proposed method, we selected 15 students and constructed a dataset by collecting EEG data at a sampling frequency of 128 Hz through a 32-channel EEG sensor while each subject listened to 80 songs. And a ten-fold cross-validation evaluation was performed for four emotion classifications on this dataset. Table I presents the experimental results of the proposed method, as well as comparisons with two methods, including MSP-CBGRU (a method of converting a music signal into a spectrum and applying it to a convolutional BGRU classifier) and EEG-HABGRU (a method applying multi-channel EEG signal to hierarchical BGRU with attention).

4. CONCLUSIONS

In this paper, we proposed a method of automatically classifying input music into four human emotional responses by generating artificial EEG features corresponding to audio stimuli. As a future study, we will develop a deep learning approach that more emotionally classifies music by utilizing various physiological signals along with multi-channel EEG signals.

REFERENCES

Enhancement of waveform reconstruction for variational autoencoder-based neural audio synthesis with pitch information and automatic music transcription

Seokjin LEE¹, Minhan KIM², Seunghyeon SHIN², Daeho LEE², Inseon JANG³, Wootaek LIM³
¹ School of Electronics Engineering, Kyungpook National University, Republic of Korea
² School of Electronic and Electrical Engineering, Kyungpook National University, Republic of Korea
³ Electronics and Telecommunications Research Institute, Republic of Korea

ABSTRACT
In recent audio signal processing techniques, analysis and synthesis models based on deep generative models have been applied for various reasons, such as audio signal compression. Particularly, some recently developed structures such as vector-quantized variational autoencoders can compress speech signals. However, extending these techniques to compress audio and music signals is challenging. Recently, a real-time audio variational autoencoder (RAVE) method for high-quality audio waveform synthesis was developed. The RAVE method synthesizes audio waveforms better than conventional methods; however, it still encounters certain challenges, such as missing low-pitched notes or generating irrelevant pitches. Therefore, to be applied to audio reconstruction problems such as audio signal compression, the reconstruction performance should be improved. Thus, we propose an enhanced structure of RAVE based on a conditional variational autoencoder (CVAE) structure and automatic music transcription model to improve the reconstruction performance of music signal waveforms.

Keywords: Audio Synthesis, Generation Model, Variational Autoencoder

1. INTRODUCTION
Various analysis/synthesis methods for sound signals have been developed in recent machine-learning-based acoustic signal processing techniques. To generate audio signals using deep learning models, several generation models based on generative adversarial networks (GANs) and variational autoencoders (VAE) have been studied, such as SynthGAN [1], HiFi-GAN [2], differentiable digital signal processing (DDSP) [3], and realtime audio variational autoencoder (RAVE) [4]. The models are widely used for audio signal generation and timber transfer.

The recently developed DDSP method [3], a VAE-based sound generation model, can successfully generate audio signals. Despite the excellent reconstruction quality of DDSP, the system consists of several complex models; therefore, the overall model is quite bulky. Moreover, the DDSP model focuses on reconstructing monophonic music signals rather than polyphonic music signals. The RAVE [4] method can reconstruct polyphonic audio signals with relatively simple networks; however, the polyphonic reconstruction performance is still inadequate.

In this paper, we propose an enhanced structure of the RAVE method with an automatic music transcription (AMT) model for polyphonic music signals. The proposed structure is based on the conventional RAVE model, and a pre-trained AMT model is used to provide activated pitch information.

2. Proposed Method
2.1 Structure
In this study, we improve the conventional RAVE model based on the conditional VAE (CVAE)
method [5]. CVAE uses conditional distributions for both input and conditional information instead of the distribution with the given input data in the original VAE model. The distributions of the target signal to be reconstructed and latent vector may depend on activated pitch data. Therefore, we added an AMT model to provide the activated pitch information and modified the RAVE model to manage the estimated pitch information.

As shown in Figure 1, the AMT model estimates the pitch information, and the estimated pitch information is concatenated to a multiband input signal. The concatenated signal is fed to the modified RAVE encoder, where the encoder structure is same as that of the encoder model in the original RAVE [3]; however, the number of input channels is increased to manage the pitch information. The estimated pitch information is fed to the fully connected layer, which is added to combine the latent vectors and pitch information, and the output of the fully connected layer is used to reconstruct the waveform using the RAVE decoder. The dimensions of the output data of the fully connected layer are the same as those of the input dimensions of the original RAVE decoder; therefore, the RAVE decoder in Figure 1 is the same as the original RAVE decoder [3]. The estimated pitch information of each frame is provided as a one-hot encoded vector with a length equal to the number of candidate pitches (e.g., 88 for piano sounds).

2.2 Training Procedure

The training procedure consists of three steps:
1) AMT learning step: the AMT model is trained with the ground truth pitch information.
2) Representation learning step: the RAVE encoder, fully connected layer, and RAVE decoder are trained.
3) Adversarial fine-tuning step: the RAVE decoder and discriminator are trained.

The modules trained in steps 1, 2, and 3 are shown in Figure 1 in green, blue, and yellow, respectively. The discriminator module operates only in the third step.

3. Experiment

3.1 Settings

Training. As mentioned in Section 2.2, the AMT model was trained for the AMT learning step. In this study, the onset and frames [5] model was used as the AMT model to provide the estimated pitch information. The AMT model was trained for 100k iterations using the Adam [6] optimizer with a learning rate of 0.0006. The learning rate decayed for 10k iterations with a rate of 0.98. After the AMT model was trained, the modified RAVE model was trained with the same optimizer with a learning rate of $10^{-4}$ where $\beta_1$ and $\beta_2$ were 0.5 and 0.9, respectively. The number of iterations for the representation learning and adversarial fine-tuning steps were set at 1M and 2M, respectively. The other training parameters were the same as those in the original studies for the onset and frames [5] and RAVE [4].

Datasets. Our model was trained using the MAESTRO dataset [6], comprising piano music clips and MIDI data. For the AMT learning step, all clips and MIDI data from the MAESTRO dataset except for “2013” were used because the “2013” data of the MAESTRO dataset was used to evaluate the reconstruction performance. To train the AMT model, the input signal was transformed into Mel-
spectrogram with 229 Mel-frequency bins and 32 ms frames. For the representation learning and adversarial fine-tuning steps, “2014,” “2015,” “2017,” and “2018” data were used to train our model, and 5% of the training data were randomly selected and used as the validation data. All audio data were down-sampled to 16 kHz.

To evaluate the effect of the AMT performance on the reconstruction performance, the reconstruction results were further evaluated after using the MAPS dataset [7], which was not used in the reconstruction model to train the AMT model. Table 1 lists the comparison results of the AMT performances of the MAESTRO- and MAPS-trained AMT models.

<table>
<thead>
<tr>
<th>AMT Performance</th>
<th>Precision Frame Note(Onset)</th>
<th>Recall Frame Note(Onset)</th>
<th>F1-Score Frame Note(Onset)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAESTRO-trained</td>
<td>91.2 %</td>
<td>87.0 %</td>
<td>88.9 %</td>
</tr>
<tr>
<td>MAPS-trained</td>
<td>82.5 %</td>
<td>43.8 %</td>
<td>55.4 %</td>
</tr>
</tbody>
</table>

### 3.2 Results

Figure 2 shows the examples of ground-truth spectrograms and reconstructed signals. Because the output of our model is a time-domain waveform, not a spectrum, it is expressed in the time-frequency domain to compare reconstruction results. As shown in Figure 2, the reconstruction results of our models are better than those of the conventional RAVE model, particularly in the low- and high-frequency ranges. The result with the MAPS-trained AMT model is slightly degraded owing to the low transcription performance; however, it is still better than the reconstruction result of the conventional RAVE model.

Figure 2 – Examples of the spectrograms for (a) target waveform, reconstructed waveforms using (b) the conventional RAVE, (c) proposed with MAPS-trained AMT model, and (d) proposed with MAESTRO-trained AMT model.
4. Conclusion

In this paper, an enhanced RAVE model is proposed to improve the reconstruction performance of audio waveforms. The proposed model enhances the conventional RAVE model using the estimated pitch information provided by the AMT model. The estimated pitch information is used as the condition value for the CVAE structure. To evaluate the performance of the proposed modification relative to the conventional RAVE model, experiments with the MAESTRO and MAPS (only to train the AMT model) datasets were performed. According to the results, the proposed model reconstructed the target audio signal better than the conventional RAVE model, although the AMT model degraded the transcription performance.

ACKNOWLEDGEMENTS

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [22ZH1200, The research of the basic media-contents technologies]

REFERENCES

A two-stage full-band speech enhancement model with effective spectral compression mapping

Zhongshu HOU\textsuperscript{1,2,3}; Qinwen HU\textsuperscript{1,2,3}; Kai CHEN\textsuperscript{1,2,3}; Jing LU\textsuperscript{1,2,3}

\textsuperscript{1}Key Laboratory of Modern Acoustics, Nanjing University, Nanjing 210093, China
\textsuperscript{2}NJU-Horizon Intelligent Audio Lab, Horizon Robotics, Beijing 100094, China
\textsuperscript{3}Nanjing Institute of Advanced Artificial Intelligence, Nanjing 210014, China

ABSTRACT
The direct expansion of deep neural network (DNN) based wide-band speech enhancement (SE) to full-band processing faces the challenge of low frequency resolution in low frequency range, which would highly likely lead to deteriorated performance of the model. In this paper, we propose a learnable spectral compression mapping (SCM) to effectively compress the high frequency components so that they can be processed in a more efficient manner. By doing so, the model can pay more attention to low and middle frequency range, where most of the speech power is concentrated. Instead of suppressing noise in a single network structure, we first estimate a spectral magnitude mask, converting the speech to a high signal-to-ratio (SNR) state, and then utilize a subsequent model to further optimize the real and imaginary mask of the pre-enhanced signal. We conduct comprehensive experiments to validate the efficacy of the proposed method.

Keywords: Full-band speech enhancement, Deep learning, Spectral compression

1. INTRODUCTION
Speech enhancement (SE) plays an important role in front-end processing for many applications such as speech communication, automatic speech recognition (ASR) and digital hearing aids, aiming at improving the overall perceptual quality and intelligibility of speech signals distorted by background disturbances. In the last decade, data-driven deep neural network (DNN) based SE methods have achieved significantly better performance over traditional rule-based signal processing SE methods especially in adverse environments with non-stationary interference (1). However, most of the DNN-based methods are designed for signals with 16 kHz sampling rate, and the quality of speech signal at this sampling rate is still inferior to the full band speech with 48 kHz sampling rate.

Most DNN based SE methods operate in the time-frequency (T-F) domain to estimate a mask between clean and noisy spectrum (2, 3) or directly predict the real and imaginary parts of the target clean complex spectrum from the noisy speech (4, 5). There are also methods directly estimating the raw-waveform of the clean signals in the time domain (6). Recently, convolutional recurrent network (CRN) was proposed (7) and has been proven an efficient network structure for SE in T-F domain with low computational complexity. It utilizes the convolution neural network (CNN) to capture the local features in spectrogram and the recurrent neural network (RNN) to exploit the temporal correlation between consecutive frames. The dual-path recurrent neural network (DPRNN) (8), originally designed to overcome the deficiency of conventional RNN in modeling long sequence in temporal dimension, can also be applied in the frequency dimension with the inherent advantage of making full use of the harmonic spectral structure of speech. We have combined the benefits of CRN and DPRNN and designed a model called dual-path convolutional recurrent network (DPCRN) (9).

Self-attention mechanism, first introduced in (10), is broadly utilized in sequence-to-sequence tasks. It can more efficiently model long-term dependencies than RNNs (11) and temporal reasoning.

* This work was supported by National Science Foundation of China with Grant No. 11874219.
convolutional networks (TCNs) (12, 13) with more efficient parallelization capability. Recent studies have shown that the attention mechanism also performs well in SE tasks (12, 14).

Most of the SE models are basically employed in wide band (16 kHz) scenarios, while their performance has not been validated in full band (48 kHz) processing. The direct expansion of these models to full band is not reasonable. The threefold computational complexity will hinder its real-time implementation. More importantly, uniform processing in the frequency domain will comparatively weaken the network’s modeling capacity in low and middle frequency range where the majority of speech power concentrates. The perceptual band focusing on spectral envelope of speech (15) is a possible solution, but the rough resolution of spectrum features may drop important information and degrade the recovered speech quality.

Inspired by the chain optimization framework (5), we propose a two-stage full band SE model called MHA-DPCRN in this paper. In the first stage, a multi-head attention network (MHAN) is trained to estimate the amplitude mask between the noisy spectrogram and the clean spectrogram, aiming at converting the noisy speech to a relatively high SNR state. In the second stage, a DPCRN structure is jointly trained with MHAN to further optimize the real and imaginary parts of the complex spectrum. In both stages, a learnable spectral compression mapping (SCM) is designed to keep vital spectrum features intact while smoothly compressing the high frequency bands. Accordingly, a learnable inverse spectral compression mapping (iSCM) is used to reconstruct the full-band spectrum and compensate the information loss during the compressing procedure. We conduct comprehensive experiments on different open datasets to demonstrate the advantage of the SCM/iSCM structure and the efficacy of the proposed model.

2. MHA-DPCRN

2.1 Problem Formulation

Let $S(n, k)$ and $X(n, k)$ denote clean and noisy speech in the T-F domain at time frame index $n$ and frequency bin index $k$. For simplicity, indexes $n$ and $k$ are omitted hereafter. To recover the clean speech from the degraded signal, an MHAN, denoted as $F_{MHA}$, is first applied to estimate the spectral magnitude mask (SMM) (16) $M$ and the operation can be expressed as

$$M = F_{MHA}(X)$$

(1)

where $\odot$ denotes element-wise multiplication and $\tilde{S}_{MHA}$ is the enhanced spectrogram by MHAN. To further suppress the residual noise and retrieve the phase magnitude details, a successive network on DPCRN is employed to directly estimate the real and imaginary parts of the complex spectrogram. Then the enhancement process can be represented as:

$$\tilde{S}_{DPCRN} = F_{DPCRN}(\tilde{S}_{MHA})$$

(3)

where $F_{DPCRN}$ and $\tilde{S}_{DPCRN}$ denote the operation of DPCRN and the final enhanced speech spectrogram respectively.

2.2 Model Architecture

As shown in Figure 1(a), the proposed two-stage network consists of two principal parts, namely MHAN and DPCRN, and the network details are as follows.

2.2.1 Learnable Spectral Compression Mapping

Directly expanding the bandwidth of a wide-band network to full-band with the same spectral resolution will allocate two thirds of the computational resources to the less informative high band (8 kHz - 24 kHz), resulting in significant increase of both the computational burden and the learning difficulty. Therefore, it is necessary to introduce a spectral information mapping strategy to effectively compress the high band. Mel-scale filter banks are widely used to extract Mel-frequency cepstral coefficients in speaker recognition (17) due to their analogy to the behavior of human auditory system. Similar to the Mel-scale, our converted scale is defined with a logarithmic function. To further retain the information in the critical low band, we warp the spectrum by keeping frequencies below 5 kHz intact while only transforming frequencies above 5 kHz logarithmically, with the mapping curve defined as
\[ f_c = \begin{cases} 
2500 \left[ \ln \left( \frac{f}{2500} \right) + 2 \right] , & 0 \leq f \leq 5kHz \\
, & 5k < f \leq 24kHz 
\end{cases} \] 

Based on the mapping in equation (4), the compression operation is designed. Suppose the \( F \)-dimensional spectrum is compressed into an \( F_c \)-dimensional spectral representation, with the first \( K \) dimensions denoting the uncompressed low frequencies. The compression can be realized using a matrix multiplication between a compression matrix of size \( F_c \times F \) and the original spectrum. The \( K \times F \) low band mapping portion of the compression matrix keeps low frequency bands unchanged with a \( K \times K \) identity submatrix, while the \( (F_c - K) \times F \) high-band part initialized with a series of triangular filter banks compresses higher bands logarithmically (18). Note that the mapping of equation (4) roots from the human auditory system and cannot effectively match the sparse distribution of speech spectrum in high frequency range. Therefore, the direct implementation of this compression pattern would lead to considerable residual noise.

To more effectively exploit the sparse distribution of speech in high frequency range, we propose to set the compression matrix partially learnable. In network implementation, a dense layer with no bias is initialized with weights of the compression matrix. The low band mapping of layer is fixed and the high-band part is learned by the network. Correspondingly, the inverse spectral compression mapping (iSCM) is also through a learnable dense layer, but it is randomly initialized. It is observed that the inversion pattern is closely related to the compression pattern.

**2.2.2 Multi-Head Attention Network**

As shown in Figure 1(b), MHAN resembles the multi-head attention network structure used in (12). It first employs the aforementioned SCM to compress frequency bands in each frame to an \( F_c \)-dimension vector, along with a frame-wise normalization and a ReLU activation layer. \( B \) repetitive MHA blocks follow afterwards and no positional encoding is applied. An MHA block consists of a masked MHA module and a two-layer feed-forward module, with a residual connection (19) and a frame-wise layer normalization applied after each module. The detailed description on MHA can be found in (12, 13). Finally, the processed information goes through an iSCM and is mapped into an interval between 0 and 1 by a sigmoid function, generating the SMM. The product of the estimated SMM and input STFT spectrogram is obtained in an element-wise manner, and then fed to the DPCRN.

**2.2.2 Dual-Path Convolutional Recurrent Network**

DPCRN consists of an encoder, a dual-path RNN (DPRNN) module and two decoders, as shown in Figure 1(c). The real and imaginary parts of the noisy spectrogram are mapped by an SCM before sending to the encoder. The encoder utilizes the 2-dimensional convolutional (Conv-2D) layers to extract local patterns from noisy spectrogram. The decoders use the transposed 2-dimensional convolutional layers (TransConv-2D) symmetric with that in the encoder to refactor low-resolution features to the original shape. There are skip connections between the encoder and the decoders for information interaction. Every convolutional layer is followed by a batch normalization layer and a PReLU activation function. In the DPRNN module, inter-chunk processing uses long short-term memory (LSTM) unit to model the temporal dependence. As for the intra-chunk processing, bidirectional LSTM (BiLSTM) is employed to model spectral patterns in a single frame. The LSTM

![Diagram](image-url)
and BiLSTM are followed by a dense layer and an instance normalization (IN). A residual connection is then applied between the input of RNN and the output of IN to further mitigate the gradient vanishing problem.

2.2.3 Training Targets and Loss Function

Power compress loss functions (20) are utilized to better process the information in low power T-F points:

\[ L_{RI}(\tilde{S}, S) = \left\| \mathcal{S}_\text{real}^c - \tilde{S}_\text{real}^c \right\|_F^2 + \left\| \mathcal{S}_\text{imag}^c - \tilde{S}_\text{imag}^c \right\|_F^2, \]
\[ L_{Mag}(\tilde{S}, S) = \left\| |S| \gamma - |\tilde{S}| \gamma \right\|_F^2 \]  

(5)

\[ S^c = |S| \cos \theta_S, \quad S^e = |S| \sin \theta_S, \]  

(6)

where \( \theta_S \) denotes the phase angle of complex spectrogram, \( \gamma \) refers to the compression parameter, superscript \( c \) denotes the power compressed pattern, and \( \left\| \cdot \right\|_F \) refers to the Frobenius norm of the matrix. MHAN is pre-trained based on the loss function below to estimate an SMM:

\[ L_1 = L_{Mag}(\tilde{S}_{MHA}, S). \]  

(7)

In the joint training procedure, MHAN is initialized with checkpoint pre-trained and the parameters of MHAN and DPCRN are optimized simultaneously with the loss function

\[ L_2 = L_{Mag}(\tilde{S}_{MHA}, S) + L_{Mag}(\tilde{S}_{DPCRN}, S) + L_{RI}(\tilde{S}_{DPCRN}, S). \]  

(8)

3. Experiments

3.1 Datasets

To demonstrate the efficiency of the SCM/iSCM structure, we first train our model on simulated datasets where clean speech clips are mainly generated from VCTK (21) and SIWI (22), and noise recordings are from DEMAND (23) and QUT-NOISE (24). 16000 clips of clean speech (around 45 h) are generated with 8% for validation. Audios are convolved with room impulse responses randomly selected from openSLR26 and openSLR28 (25) to simulate reverberant environments. The SNR of noisy speech ranges from 15 to -5dB. In the testing stage, we select clean speech from DAPS (26) and noise from Saki (27, 28), to create simulated noisy speech. The SNR range of the test noisy speech is the same as the training set. All the audio clips used is sampled at 48 kHz.

For evaluation on real acoustic environment, we train our model with dataset provided by ICASSP 2022 DNS4 challenge (29). The training configures are the same as simulated datasets. To compare our model with previous state-of-the-art (SOtA) full-band and super-wideband SE methods, we also train and test our model on the classic open VCTK-DEMAND dataset (30).

3.2 Parameter Setup and Training Strategy

The window length and hop size are 25ms and 12.5ms respectively. The FFT length is 1200 and the hanning window is used. The SCMs map the 601-dimension spectrum to a 256-dimension feature. For MAHN part, we set parameters \( B = 5 \) and the number of heads is 8. In DPCRN, the output channel of the Conv-2D in the encoder is \([16, 32, 48, 64, 80]\). The kernel size and the stride are respectively set to \([5, 2), (3, 2), (3, 2), (3, 2), (2, 1)]\) and \([2, 1), (1, 1), (1, 1), (1, 1), (1, 1)]\) in frequency and time dimension. We use DPRNN module with 1 intra chunk and 1 inter chunk, and the hidden size is set to 127. There are two ways to train the SCMs on simulated datasets, with one setting the whole parameters learnable while another setting the high-band part learnable, labeled as High Learn.

Warmup strategy (12) is critical in training MAHN, where the learning rate \( \alpha \) is updated with the rule:

\[ \alpha = \frac{1}{\sqrt{t}} \times \min \left( \frac{1}{\sqrt{\psi}}, \frac{\varphi}{\sqrt{\varphi}} \right), \]  

(13)

where \( C = 128 \), warmup steps \( \psi = 10000 \) and \( \varphi \) denoting the training step. We train the model by the warmup-based Adam optimizer with \( \beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-6} \). The compression parameter \( \gamma \) is \( \frac{1}{\sqrt{\gamma}} \). The total parameter of the whole framework is 5.00 million and the computation complexity is 7.82 GMACs per second.

3.3 Baselines and Evaluation Metrics

On the simulated test set, we compare our model with the models in DNS challenge, including NSNet2 (31), DPCRN (9). To prove the validity of the proposed SCM, we intentionally add the SCM/iSCM structure to the conventional DPCRN, marked as SCM-DPCRN. On VCTK-DEMAND test set, we compare our model with previous SOtA systems, including RNNoise (32), PercepNet (15),
DCCRN (33), DCCRN+ (34), DeepFilterNet (35), S-DCCRN (36) and DMF-Net (37). On the DNS4 blind test set, we use the NSNet2 as baseline.

We use DNS-MOS P.835 (38) to evaluate results on simulated test set. It provides simulated subjective evaluation scores namely speech quality (SIG), background noise quality (BAK), and overall audio quality (OVRL) based on a deep learning model. We also use a common objective metric called signal to distortion ratio (SDR) (39) for full-band evaluation. On the VCTK-DEMAND test set, we use the perceptual evaluation of speech quality (PESQ) (40), short-time objective intelligibility (STOI) (41) to evaluate the SE performance compared with previous SOTA methods. On the DNS4 blind test set, ITU-T P.835 (42) based subjective MOS is employed. It should be noted that the audios evaluated with PESQ, STOI and DNS-MOS metrics are all down-sampled to 16 kHz.

3.4 Result and Analysis

The performance on simulated test set is presented in Table 1.

<table>
<thead>
<tr>
<th>Models</th>
<th>High Learn</th>
<th>BAK</th>
<th>SIG</th>
<th>OVLR</th>
<th>SDR(dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td></td>
<td>3.175</td>
<td>4.302</td>
<td>3.289</td>
<td>1.32</td>
</tr>
<tr>
<td>NSNet2</td>
<td></td>
<td>4.312</td>
<td>3.952</td>
<td>3.600</td>
<td>11.37</td>
</tr>
<tr>
<td>DPCRN</td>
<td></td>
<td>4.400</td>
<td>4.104</td>
<td>3.692</td>
<td>13.70</td>
</tr>
<tr>
<td>SCM-DPCRN-1</td>
<td>No</td>
<td>4.414</td>
<td>4.126</td>
<td>3.735</td>
<td>14.46</td>
</tr>
<tr>
<td>SCM-DPCRN-2</td>
<td>Yes</td>
<td>4.466</td>
<td>4.184</td>
<td>3.811</td>
<td>14.73</td>
</tr>
<tr>
<td>MHA-DPCRN-1</td>
<td>No</td>
<td>4.502</td>
<td>4.232</td>
<td>3.874</td>
<td>15.44</td>
</tr>
<tr>
<td>MHA-DPCRN-2</td>
<td>Yes</td>
<td>4.507</td>
<td>4.237</td>
<td>3.884</td>
<td>15.74</td>
</tr>
</tbody>
</table>

It can be seen that DPCRN with SCM structure performs better than the original DPCRN in terms of all metrics. It can be attributed to the fact that SCM makes the harmonics feature in low and middle frequency range more discriminative, as shown in Figure 2, which facilitates more effective attenuation of noise. Comparing two SCM-DPCRN networks, we can see that the one trained with High Learn strategy achieves better results. We present the learnt SCM parameters of both in Figure 3, from which we can see that the learnt SCM parameters of model SCM-DPCRN-1 will map the high-frequency information to the low-frequency range, resulting in additional distortion to major speech components. While SCM-DPCRN-2 keeps the spectrum below 5 kHz completely intact and only higher frequency bands are adjusted. The proposed two-stage MHA-DPCRN models show significant improvements over baselines. It can be seen from Figure 4 that a prior estimated SMM can effectively reduce major noise components and a subsequent network refines the spectrogram details.

![Figure 2 – Outputs comparison of DPCRN and SCM-DPCRN](image1)

![Figure 3 – Comparison of the learnt SCMs](image2)
Figure 4 – Illustrating spectrograms of the enhancement process.

Results on VCTK-DEMAND test set is shown in Table 2, and the proposed method achieves higher scores in terms of both metrics compared with other SOTA full-band models. On average, no less than 0.05 increase in PESQ is achieved. Compared with RNNoise and PercepNet, MHA-DPCRN yields significant improvement in PESQ and STOI, indicating that the proposed SCM/iSCM structure better improves speech quality and intelligibility by keeping the resolution of low frequency bands unchanged.

Table 2 – Results on VCTK-DEMAND test set

<table>
<thead>
<tr>
<th>Models</th>
<th>year</th>
<th>Para.(M)</th>
<th>PESQ</th>
<th>STOI(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td>-</td>
<td>1.97</td>
<td>92.1</td>
<td></td>
</tr>
<tr>
<td>RNNoise</td>
<td>2020</td>
<td>0.06</td>
<td>2.34</td>
<td>92.2</td>
</tr>
<tr>
<td>PercepNet</td>
<td>2020</td>
<td>8.00</td>
<td>2.73</td>
<td>-</td>
</tr>
<tr>
<td>DCCRN</td>
<td>2020</td>
<td>3.70</td>
<td>2.54</td>
<td>93.8</td>
</tr>
<tr>
<td>DCCRN+</td>
<td>2021</td>
<td>3.30</td>
<td>2.84</td>
<td>-</td>
</tr>
<tr>
<td>DeepFilterNet</td>
<td>2021</td>
<td>1.80</td>
<td>2.81</td>
<td>-</td>
</tr>
<tr>
<td>S-DCCRN</td>
<td>2022</td>
<td>2.34</td>
<td>2.84</td>
<td>94.0</td>
</tr>
<tr>
<td>DMF-Net</td>
<td>2022</td>
<td>7.84</td>
<td>2.97</td>
<td>94.4</td>
</tr>
<tr>
<td>MHA-DPCRN(Pro.)</td>
<td>2022</td>
<td>5.00</td>
<td><strong>3.02</strong></td>
<td><strong>94.4</strong></td>
</tr>
</tbody>
</table>

The P.835 based subjective MOS on the DNS challenge blind test set is shown in Table 3, including subjective speech (SIG), background noise (BAK), overall MOS (OVRL) scores. It shows that our model still achieves remarkable performance in real acoustic scenarios.

Table 3 – Results on DNS4 challenge blind test set

<table>
<thead>
<tr>
<th>Models</th>
<th>BAK</th>
<th>SIG</th>
<th>OVRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td>2.15</td>
<td><strong>4.29</strong></td>
<td>2.63</td>
</tr>
<tr>
<td>NSNet2</td>
<td>3.93</td>
<td>3.62</td>
<td>3.26</td>
</tr>
<tr>
<td>MHA-DPCRN(Pro.)</td>
<td><strong>4.42</strong></td>
<td>3.97</td>
<td><strong>3.72</strong></td>
</tr>
</tbody>
</table>

4. Conclusions

We design a compression structure in spectrum called SCM for full-band SE task, which has the benefit of efficiently compressing the high frequency components while guaranteeing the effective modeling capacity in low and middle frequency range where the majority of speech power concentrates. We propose a two-stage full band speech enhancement model with SCM in the T-F domain, named MHA-DPCRN. It suppresses major disturbances with an SMM estimated by MHAN and further clear residual noise by DPCRN. With 5.0M parameters, our model achieves competitive results on various datasets.

REFERENCES


32. Valin J-M, editor A hybrid DSP/deep learning approach to real-time full-band speech enhancement. 2018 IEEE 20th international workshop on multimedia signal processing (MMSP); 2018: IEEE.


ABSTRACT
In this study, we propose differentiating positive and negative patients with COVID-19 based on the coughing sound. Cough is an acoustic signal that differs from speech. The difference is in the aperiodic component indicated by the carrier-to-noise ratio. Meanwhile, between positive and negative COVID-19 coughs, non-harmonic components can be analyzed based on a spectrogram, where this feature can use to characterize COVID-19 cough. At the initial stage, experiments carried out the MFCC extracted features in coughing sounds which then entered the classification process in the form of a training and validation process using the LSTM algorithm and obtained an accuracy of 82.2% and a UAR of 71.3% with a training time of 7 minutes 27 seconds. Then after reviewing the imbalance classification data, we used the synthetic minority oversampling technique to synthesize the sample from the minority class in the training dataset before obtaining the classification model. This process can balance the distribution of classes but does not provide any additional information to the model. Then the classification results increased to 91.3% UAR, 90.9% accuracy, with a training time of 1 minute 24 seconds.

Keywords: Cough, Covid-19, synthetic minority oversampling technique, long-short term memory

1. INTRODUCTION
SARS-CoV-2 is the virus that causes coronavirus disease by COVID-19, which has been transmitted and spread in a short time since its first report in China (1). Recent research on the diagnosis of covid 19 either with CCT or CXR has reasonably high accuracy as with the CNN method (Kedia et al., 2021) so that it can help overcome the sensitivity of PCR tests that are less than optimal (3). However, although these two approaches do not reduce the burden on radiologists to make a diagnosis, they still require complete clinical facilities. As a result, this approach also has drawbacks mainly due to the inability to test most of the population quickly and safely because it requires interaction with patients, is cost-effective, and precisely tracks the actual spread to withstand the pandemic (4).

The study’s results (5) stated that collecting speech data (audio) could help diagnose and detect COVID-19 through three basic things. First, coughing sounds help detect positive cases of COVID-19 by applying ML techniques(4,6). Second, the respiratory rate can detect from speech results in COVID-19 screening in a person(7). Third, stress detection techniques from speech can use to detect people who have indications of mental health disorders and the severity of COVID-19 symptoms(5). This speech-based COVID-19 diagnosis technique can be activated with a smartphone app or remote medical care via telemedicine, thus reducing direct contact with patients.

Previous studies have shown that coughs from different respiratory syndromes have different latent features(8,9). These different features can extract with proper signal processing and mathematical transformation of the coughing sound. The feature can then train an AI machine that performs an initial diagnosis based solely on coughing. The experts have used coughing sounds to perform differential diagnoses among several respiratory conditions such as pneumonia, asthma, COPD, laryngitis, and Tracheitis(10,11). However, This is possible because, in all these diseases, the nature and location of the cause of irritation in the respiratory system are so different

1 Roudhotuljannah525@gmail.com
2 arifianto@gmail.com
that it causes a coughing sound that sounds different (4). However, without assistance, the human ear cannot distinguish the cough caused by covid or non-covid conditions.

Research with an audio approach to detecting and diagnosing covid using several ML methods to produce high-quality diagnoses. The cough is a symptom of more than thirty non-COVID-19 related medical conditions. Study (4) used cough to diagnose and reduce the risk of misdiagnosis using three ML Classifiers: a Deep Learning-based multi-class class, a Classic ML-based binary class classifier, and a Deep Learning-based binary class classifier. The results showed that AI4COVID-19 could distinguish between COVID-19 cough and some types of non-COVID-19 cough with an accuracy of more than 90%. The research (12) collected cough sound datasets through www.covid-19-sounds.org apps. Then the classification was carried out using the Logistic Regression, Gradient Boosting Trees, and SVM methods. The features used are handcraft features (mfcc, F0, Etc) and transfer learning features (VGGish). The study obtained an AUC (area under the curve) above 80% in all classifications.

Cough can use as a classifier of diseases caused by disruption of the respiratory system. The results stated that coughing could use to classify asthma and pneumonia with neural network methods resulting in 80% sensitivity and 100% specificity (13). (14) stated that his research successfully used cough to classify pertussis with logistic regression methods and produced 92% sensitivity and 90% specificity. Cough also has proved to be used as a covid-19 cure as research is currently being (4,12) good accuracy in utilizing cough as a covid-19 classifier. This voice-based COVID-19 diagnosis technique can be activated with a smartphone app or remote medical treatment via telemedicine, reducing direct patient contact. In the contribution of this paper, we will observe the distinguishing features of suspected positive and negative covid patients based on coughing sounds and can diagnose covid-19 with cheap computing and high accuracy.

2. CHARACTERIZATION OF COUGH

2.1 Cough

According to the physiology book, cough consists of three or four phases: (1) the inspiration phase, which consists of profound inspiration; (2) the compressive phase, with the closure of the larynx and forced expiratory efforts; (3) the excursive phase, when the larynx is open and rapid expiratory occurs with the first characteristic coughing sound; and (4) the restorative phase, when the last deep breath take as shown in Figure 1. All phases occur as a characteristic of a normal cough. However, reflex coughs such as those generated by inhalation of irritant substances or those that occur spontaneously in the disease may be very different. There may be a second (or third or even fourth) closure of the larynx during the excursive phase, producing a second (or third or fourth) coughing sound, so it is different for each cough produced (15).

Figure 1 - Changes in several variables during coughing: sound level, lung volume, flow rate, and subglottic pressure. During inspiration, negative flow rates; at glottic closure, the flow rate is zero; and during the expiratory phase, the positive flow rate. The last phase can divide into three parts: growing, constant, and declining (15).
Cessation of cough marks a relaxation of the expiratory muscles. It sometimes happens with the onset of inspiring muscle activity. Alveolar and pleural pressures decrease towards ambient pressure. The neurophysiological and mechanical events that mark the different phases of coughing are illustrated schematically in Figure 2.

Figure 2 - Schematic representation of cough pattern

Figure 2 shows the schematics of the pattern during a single cough, along with the flow at the airway opening (airflow), Subglottal pressure, and sound level. The form of "I, C, and E" indicate phases of inspiration, compression, and expiration characterized by vertical dotted lines. Note that the laryngeal muscles are also active during eucapnic breathing and rhythmically perform abduction (posterior cryoternoid muscles) and adduction (thyroarytenoid muscles) of the vocal cords (16).

2.2 Cough and Speech

Coughing and speech sounds obviously have differences in the process of forming sounds to the sounds emitted. Here is a spectrogram image of coughing and speech sounds.

Figure 3 - Cough And Speech

The spectrogram determines the difference between two sounds through the characteristics indicated by the dB level against the sound frequency along the sound data. In this case, the spectrogram uses color differences to indicate the dB level, the lowest in blue and the highest in red. Areas that are blue or have deficient dB levels are also called unvoice regions, which are areas that do not have a sound or periodic component. Figure 3 shows the spectrogram shape of the two types of sounds, namely coughing sounds and speech sounds. In this case, the cough used is a regular (non-covid) and a speech sound with the pronunciation "Saya." Figure 3 shows spectrogram results of coughing sounds and speech sounds at low frequencies (0-4000 Hz). The picture shows that the
difference pronounces in the spectrogram of the speech voice is indicated by a black circumference. There is a wave-like voice area, the energy that occurs due to the sound of reciting a sentence. Although the coughing sounds like saying "uhuk" in the shape of the spectrogram, it does not look the same as the sound of speech. The voice area of the coughing voice has a strong-weak-strong periodic region along its frequency, as indicated by a dB level that is brighter than the rest of the section.

Figure 4 - Aperiodicity Analysis of Cough and Speech

Aperiodicity is a measure of comparing harmonious information to non-harmonious information in the frequency domain. It expresses the relative energy distribution of disharmonious components indicated by the carrier to noise ratio (C/N) value. The difference between the sound of coughing and speech is very clearly visible from the aperiodicity of the two voices. Speech sounds certainly have harmonic information shown on the C/N panel graph, which is higher or equal to 20 dB (17) compared to coughing sounds that do not have harmonic information, as seen in figure 4.

2.3 Covid and Healthy Cough

Every cough caused by a different condition is difficult to distinguish from the normal hearing because the symptoms in the form of a cough seem similar. Therefore, an analysis of the difference in cough between covid-positive and non-covid patients needs to do. The following are samples of positive and negative coughing sounds taken from Virufy data. Cough consisting of explosive, intermediate, and voice phases appear different in both forms of cough waveforms. The intermediate and voiced phases in covid positive cough are longer than in non-covid (healthy). The covid positive voice phase amplitude is higher than non-covid, as seen in figure 5 on the left. When viewed from the form of the spectrogram (right fig.5), the difference between the two cough spectrograms appears that the non-covid has periodic and aperiodic regions alternating along the frequency. In contrast, the covid cough spectrogram form is flat with the highest dB level indicated in red, and the periodic area is wider against time compared to non-covid cough.

Statistical analysis of this data determines the distribution of cough data used based on sound features in the form of MFCC. The MFCC feature is a feature used as a covid cough classifier. This feature chooses because it can represent the difference between a positive and negative covid cough. When viewed from the Mel frequency cepstogram (MFCC) form of the two coughs, the difference is a little difficult to analyze through the eyes. Therefore, here is the distribution of MFCC features that can illustrate the existence of differences between the following two types of cough.
The distribution below extracts the MFCC feature from each coughing sound in the dataset. Then the feature, which is multidimensional, is flattened to be one dimension through the mean of the feature. The results in cough distribution between positive and negative coughs get from the value of SSE (Sum Square Error). SSE is used to measure a cluster's variation (20). The SSE value shows that each cough cluster categorizes as a form of distribution of each cough. However, this shows that positive and negative covid coughs can be distinguished through the cough MFCC feature without confusion, as seen from the difference in the distribution in figure 6.

![Figure 6 - MFCC Feature Distribution of non-covid cough (left) and covid cough (right)](image)

The sound of coughing is like a burst, which causes the cough to tend to contain an aperiodic component. Therefore, the distribution model of cough sounds is close to the normal distribution. However, in the picture above, the covid distribution is not normal; this indicates that a positive cough for COVID differs from the usual cough.

### 3. MATERIALS AND METHOD

#### 3.1 Dataset

This study uses datasets from Virufy, Coswara, ComParE2021, and KDD (Knowledge Discovery and Data Mining) (18,21,22). The dataset used is in the form of a secondary dataset obtained with a description of many cough data as follows:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Positive Cough</th>
<th>Negative Cough</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virufy (Pakistan Hospital)</td>
<td>46</td>
<td>72</td>
</tr>
</tbody>
</table>
An aperiodicity analysis using STRAIGHT analyzes the difference between coughing and speech (17). While positive and negative covid cough analyzes from non-harmonic components based on a spectrogram, the feature uses as a feature of covid coughing. In the early stages, the preprocessing process is in the form of cutting, labeling, and normalizing cough sound data. Furthermore, the extraction process acoustics features on coughing sound, which is then continued with data augmentation to overcome imbalance classification using SMOTE (23). Here are some sample data from the virufy dataset.

### Table 2 – Sample Data [virufy]

<table>
<thead>
<tr>
<th>Sample</th>
<th>Corona test</th>
<th>Age</th>
<th>Gender</th>
<th>Medical history</th>
<th>Reported symptoms</th>
<th>Cough file name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Negative</td>
<td>53</td>
<td>Male</td>
<td>None</td>
<td>None</td>
<td>neg-0421-083-cough-m-53.mp3</td>
</tr>
<tr>
<td>2</td>
<td>Positive</td>
<td>50</td>
<td>Male</td>
<td>Congestive heart failure</td>
<td>Shortness of breath</td>
<td>pos-0421-084-cough-m-50.mp3</td>
</tr>
<tr>
<td>3</td>
<td>Negative</td>
<td>43</td>
<td>Male</td>
<td>None</td>
<td>Sore throat</td>
<td>neg-0421-085-cough-m-43.mp3</td>
</tr>
<tr>
<td>4</td>
<td>Positive</td>
<td>65</td>
<td>Male</td>
<td>Asthma/chronic lung disease</td>
<td>Shortness of breath, new or worsening cough</td>
<td>pos-0421-086-cough-m-65.mp3</td>
</tr>
<tr>
<td>5</td>
<td>Positive</td>
<td>40</td>
<td>Female</td>
<td>None</td>
<td>Sore throat, loss of taste, loss of smell</td>
<td>pos-0421-087-cough-f-40.mp3</td>
</tr>
</tbody>
</table>

The cough sound feature has few harmonic components (different from speech sounds). The LLD (Low-Level Descriptors) feature consists of 40 MFCCs, 12 chroma vectors, 128 mel-scaled spectrograms, seven spectral contrast features, and six tonal centroid features to identify the characteristics of covid or not in the coughing sound. This feature will simulate a program that creates so that the most effective acoustic features can be obtained and used as a COVID-19 symptom in coughing. The following is an explanation of each feature, as seen in the table below:

### Table 3 - Acoustic Feature

<table>
<thead>
<tr>
<th>Feature</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromagram</td>
<td>In the analysis of audio files, an audio file can consist of 12 different tone classes. This pitch class profile analyzes an audio file that represents a tone below the audio file. Chromagram uses different scales to classify tone classes under audio files. The colors in the chromagram chart show different tone classes between 1 to 12. Pitch is a property of any sound or signal that allows sequencing files based on a frequency scale in the form of sound quality measurements that help assess sound as higher, lower, and medium.</td>
</tr>
<tr>
<td>Mel spectrograms</td>
<td>Mel scales are scales listeners can perceive with the exact distance between each other. For example, a listener can identify the difference between 10000 Hz and 15000 Hz audio if the audio sources are in the same distance and atmosphere. The representation of frequencies into the Mel scale produces a Mel spectrogram. The frequency can be...</td>
</tr>
</tbody>
</table>
Mel-frequency cepstral coefficients (MFCCs) converted into a Mel scale using the Fourier transform. The MFCC represents sound better because its frequency bands distribute on the Mel scale, which is close to the response of the human auditory system closer. The MFCC derive by mapping the Fourier transformation signal to the mel scale using a triangular or cosine overlap window. The log of the Mel frequencies and discrete cosine transformation of the power of the Mel log gives the amplitude of the spectrum. Its amplitude value is MFCC.

Spectral contrast In audio signals, spectral contrast measures the frequency energy at each timestamp. A large part of an audio file contains frequencies whose energy changes over time. So it becomes difficult to measure energy levels. Spectral contrast is a way to measure those energy variations. The high contrast value generally corresponds to a clear narrowband signal, while the low contrast value corresponds to the wideband noise. The energy contrast value compares the average energy in the peak energy frame with the energy below or valley.

tonal centroid in the chroma feature, the audio file can consist of 12 pitch classes. So tonnetz assumes that audio files divide into six pitch classes by combining several.

3.2 Architecture Model

The classification algorithm in this study used model LSTM with the following architecture:

![Figure 7 - Differences Between The Usual LSTM Architecture and The Purpose LSTM Model](image)

**Table 4 - LSTM Architecture**

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output (Dimensions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch_normalization_1</td>
<td>(Batch (None, 1, 20))</td>
</tr>
<tr>
<td>lstm_1 (LSTM)</td>
<td>(None, 1, 128)</td>
</tr>
<tr>
<td>lstm_2 (LSTM)</td>
<td>(None, 1, 256)</td>
</tr>
<tr>
<td>lstm_3 (LSTM)</td>
<td>(None, 1, 512)</td>
</tr>
</tbody>
</table>
From figure 7 and table 4, show the LSTM architecture used in the COVID infection classification program through coughing sounds. There are seven layers in the form of an input layer, 3 LSTM layers, a flatten layer, a dropout layer, and a dense layer or output layer. In the input layer, which is the first layer, a batch normalization layer uses to automatically adjust the number of dimensions as input nodes on the classification network. In this case, the input layer has dimensions of 1x20 in the form of many MFCC features used as program inputs. The subsequent three layers use an LSTM layer with dimension variations arranged into layers with multiple dimensions (1x128, 1x256, 1x512), the selection of LSTM layer dimensions is a condition that produces the best accuracy in try and error attempts. The flattening layer uses to convert multi-dimension into a single dimension. A dropout layer uses to trim neurons to prevent overfitting and speed up computation. A dense layer is the last layer or output layer containing classification program decisions in the form of "negative COVID" or "COVID positive" status. The COVID Classification Program shows in the diagram below:

3.3 Cough Detection System

A cough detection program is needed to filter out junk data in the training process. The cough detection program builds with a groove scheme such as the following:

The data used for training the cough detection program in the form of CompE, KDD, and Fena speech data amounted to 3410. So this process takes quite a lot of time in the preprocessing process. Then, after all the data goes through the VAD process, it is continued with the process of extracting features and classification using the LSTM binary classification. The accuracy obtained from the validation process is 100%; this is possible because the analysis of aperiodicity shows a striking difference between coughing and speech sounds, so the classification program easily distinguishes the two.

3.4 Data Augmentation

The problem with imbalance classification is that too few minority class samples use to make decisions in the classification model. One way to solve this problem is to add oversampling to minority
classes by duplicating/synthesizing samples from minority classes in the training dataset before installing the model. This process can balance the distribution of classes but does not provide any additional information to the model. SMOTE (Synthetic Minority Oversampling Technique), described by Nitesh Chawla et al. in 2002, is one of the Techniques for synthesizing new samples.

Figure 11 - Data Distribution Before And After Augmentation Process

Figure 11 is a scatter plot of the data distribution before and after the augmentation process. After the feature extraction process, the sound features of positive and negative covid coughs extract from MFCC features in the data augmentation process. The blue dot represents non-covid cough, while the orange represents covid cough data. Before the augmentation process, the data distribution was 2065 for non-covid data and 607 for covid data. After the data augmentation process, the data have a balance distribution.

4. RESULT AND DISCUSSION

4.1 Program Performance

Program performance can be seen from the accuracy value, UAR (Unweighted Average Recall), and computational length. Accuracy values from the convolution matrix, which has several properties such as True Positives (TP), False Negatives (FN), True Negatives (TN), and False Positives (FP). The number of covid detected as True Positives (TP), covid detected incorrectly as False Negatives (FN), non-covid that are correct but rejected as True Negatives (TN), and non-covid that are received incorrectly as False Positives (FP). So that the accuracy value with the following equation:

\[
\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]  

(1)

While UAR (Unweighted Average Recall) is the accuracy used in the balance classification, which is from the sensitivity and specificity values of the program with the following equation:

\[
\text{UAR} = \text{sensitivity} \times 0.5 + \text{specificity} \times 0.5
\]  

(2)

Then the following performance parameter is in the form of the length of computational time required during the training program.

4.2 Simulation of Variation of Feature Type

The audio feature extracted from the coughing sound during previous simulations has not obtained a high degree of accuracy. This experiment simulates varying types of audio features extracted from the coughing sound to get the highest accuracy value. The dataset used is the Coswara dataset (cough heavy and cough shallow). The model used is an LSTM model with an algorithm as in figure 7 obtained the following results:

<table>
<thead>
<tr>
<th>No</th>
<th>Feature</th>
<th>UAR</th>
<th>Acc</th>
<th>sensitivity</th>
<th>specificity</th>
<th>time training</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>baseline</td>
<td>68.47%</td>
<td>80.60%</td>
<td>87.36%</td>
<td>49.47%</td>
<td>4m 3s</td>
</tr>
</tbody>
</table>
Table 5 shows that the valuable feature used by the extraction process pad of the cough sound feature affects the accuracy obtained. The maximum accuracy obtained when the sound feature uses in the form of MFCC is 83.1%, higher than when all the sound features use simultaneously.

### 4.3 Simulation Result of Purpose Model

The research results showed that the MFCC feature is the most effective sound feature compared to other handcraft features as a covid-19 classifier. The addition of the data augmentation process shows an increase in the data training accuracy rate. After the data augmentation process, data distribution becomes balanced at 1:1 for both class data. It increases in UAR results to 91.3%, accuracy of 90.9%, with a shorter training time of 1 minute 24 seconds.

![Convolution Matrix](image)

#### 4.4 Simulation Result with Other Model

This experiment determines the accuracy of results obtained by comparing the deep learning architecture models. The dataset used is the Coswara dataset (cough heavy and cough shallow). Meanwhile, the extraction feature process in this experiment uses all the features of MFCC. The purpose model (LSTM with SMOTE) performance is superior to the baseline and other models, as seen from the table below:

<table>
<thead>
<tr>
<th>No</th>
<th>Model</th>
<th>UAR</th>
<th>Accuracy</th>
<th>sensitivity</th>
<th>specificity</th>
<th>time training</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>baseline</td>
<td>68.47%</td>
<td>80.60%</td>
<td>87.36%</td>
<td>49.47%</td>
<td>4m 3s</td>
</tr>
<tr>
<td>2</td>
<td>LSTM</td>
<td>71.31%</td>
<td>82.25%</td>
<td>87.58%</td>
<td>55.04%</td>
<td>7m 27s</td>
</tr>
<tr>
<td>3</td>
<td>GRU</td>
<td>67.88%</td>
<td>80.60%</td>
<td>85.76%</td>
<td>50%</td>
<td>5m 47s</td>
</tr>
<tr>
<td>4</td>
<td>SVM</td>
<td>66.75%</td>
<td>79.54%</td>
<td>86.49%</td>
<td>47%</td>
<td>0m 46s</td>
</tr>
<tr>
<td>5</td>
<td>LSTM (with SMOTE)</td>
<td>91.33%</td>
<td>90.9%</td>
<td>95.6%</td>
<td>87.05%</td>
<td>1m 24s</td>
</tr>
</tbody>
</table>

### 5. CONCLUSIONS

Coughs distinguish from other sounds through aperiodic analysis indicated by the value carrier-to-noise ratio (C/N) exceeding 20dB. At the same time, covid and non-covid coughs can be distinguished based on the power spectral density of the cough sound. The difference between the two coughs shows
by the distribution model of the MFCC feature of the two different coughs indicated by the value of SSE (sum square of error). From the research results, The MFCC feature is the most effective sound feature compared to other handcraft features as a covid-19 classifier. The covid-19 infection classification system design successfully distinguishes the sound of coughing infected with covid with a UAR rate of 91.33% and an accuracy of 90.9% in a training time of 1 minute 24 seconds.

ACKNOWLEDGEMENTS

The author would like to thank the Vibration and Acoustic Laboratory of the Physics Engineering Department FTI-RS ITS for lending the lab room and helping the entire process of collecting research data. And, not to forget, the author would also like to express his deepest gratitude to Badan Riset Inovasi Nasional (BRIN) Indonesia providing financial support for the whole research activities.

REFERENCES


18. Virufy Dataset.


Extended abstract

Deep Learning Based Source Localization On A Piping System

Dukyeon Kim\textsuperscript{1}; Haesang Yang\textsuperscript{1}; Woojae Seong\textsuperscript{1,2}

\textsuperscript{1} Naval Architecture and Ocean Engineering, Seoul National University, Korea

\textsuperscript{2} Research Institute of Marine Systems Engineering, Seoul National University, Korea

ABSTRACT

Piping systems are widely used in many industrial fields. Noise source localization is one of the current issues in these piping systems, but it has some limitations such as systems' complexity and inaccessibility. Conventional localization methods using time difference of arrival fail in complex systems. In this paper, we propose a deep learning-based noise source localization method. We used pre-trained convolutional neural network (CNN) models for classification and regression concerning the location of the noise source. To acquire the dataset, experiments are conducted on a pipe in steady state according to conditions such as various types of sources and configuration of pipes, with and without the presence of water in the pipe. Vibration signals are collected via accelerometers starting from the exciting points with 0.5m distance. The acquired signals from one sensor are pre-processed as normalized Log-mel spectrogram in order to train the CNN model. After the training, good validation performance is observed. The results show that the proposed deep learning based method is applicable in localizing the noise source in various piping systems.

Keywords: Noise Source Localization, Deep Learning, Convolutional Neural Network

1. INTRODUCTION

Noise on piping systems become a serious problem in various industrial fields. This noise can be transmitted to exterior structures as well as workspaces and accommodations, which leads to negative effect on workers and residents. In order to control this pipe noise, localization of noise source is the most fundamental solution. Most researches on this area are related to acoustic emission (AE) to localize pipe leakages and failures. There are many conventional methods for pipe noise source localization such as time difference of arrival (TDOA) based methods, matched-field processing (MFP) based methods, and beamforming based methods. Most of the conventional methods are based on TDOA with cross-correlation methods using two or more sensors. Pan et al. (2013) estimated the source location by using characteristics of signal amplitude attenuation and TDOA based on cross-correlation [1]. Choi et al. (2017) introduced the maximum likelihood (ML) pre-filter to improve the TDOA performance in various types of pipes [2]. One of the drawbacks of these TDOA methods is highly sensitive to noise. In the last decade, the deep learning-based methods have shown promising results in this problem. Heng et al. (2018) architected the convolutional neural network (CNN) model and utilized the cross-correlation map of two signals' short time Fourier transform (STFT) from each sensor [3].

No accelerometer-based applied to noise source localization on piping systems have been found in open literature. In this study, accelerometers are used to measure structure-borne noise which is difficult to be dealt with in advance. Furthermore, accelerometers are more effective method than other sensors because of relatively low-cost and non-intrusive measurements without modification of the existing structures. While other methods need more than two sensors, this method use a single accelerometer to localize the noise source not only on 1-D but also on 2-D pipe. Signal from AE sources is transient, however, the structure-borne noise is also continuous and periodic signal generated from main equipment including engines, coolers, and pumps and water flow. We suggest the localization methods based on both CNN classification and regression distinct from other studies, which indicates that accessible and even inaccessible points can be localized. Representative conditions are considered to validate possibilities of application on noise source
localization in piping systems in terms of source types, configuration of pipes, and with and without the presence of water in the pipe.

2. Experiments

In Figure 1, experiments are carried out on a pipe in steady state. There are 3 conditions which can be easily seen in general piping systems:

- Condition 1: source types - impulse, continuous, and broadband sources. Impulse source is simulated with impact hammer and other sources are experimented with shaker and function generator in the straight pipe.
- Condition 2: configuration of pipes - straight, elbow, and branch pipe.
- Condition 3: with and without the existence of water in the pipe - the straight pipe full of water and the pipe with no water.

Each excitation point has 0.5m distance starting from left end where the accelerometer exists. For example, the total number of the excitation points is 25 in 12m straight pipe. In all conditions except for impulse and broadband source, the number of data is 100 per each excitation frequency of 100, 200, 500Hz from every excitation points. On the other hand, in each impulse test and broadband test (100~1000Hz), the number of data are 50 and 100 from every excitation points. All signals were 5-second duration and acquired by data acquisition (DAQ) system through one accelerometer with sampling frequency of 32,768 Hz.

3. Training and Validation

3.1 Input Feature Extraction

The collected raw vibration signal in time domain is converted into the Log-mel spectrogram focusing on dominant lower-frequency band and making the spectrogram become more visible. The Log-mel spectrogram is used as CNN input, size of which is 224×224×3 for being fed to VGG16 and ResNet50. For the better training of CNN model, the Log-mel spectrogram is normalized.

3.2 Training Strategies

CNN models such as VGG16 and ResNet50 have shown good performance in deep learning-methods with image [4-5]. Building and training complex CNN models and acquiring big data are difficult because of temporal and economical costs, therefore, pre-trained VGG16 and ResNet50 are applied with fine-tuning for solving this paper's problems.

Each input is labelled according to the distance from position of the used accelerometer, thus, the number of labels is 25 for straight pipe, 17 for elbow and 33 for branch. To avoid over-fitting of the CNN models, 5-fold cross validation is used. VGG16 for classification and ResNet50 for
regression use pre-trained weight and adaptive the top layer for fine-tuning.

4. Results and Discussion

<table>
<thead>
<tr>
<th>Validation loss</th>
<th>Impulse</th>
<th>Continuous</th>
<th>Broadband</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean accuracy</td>
<td>0.1415</td>
<td>0.1379</td>
<td>0.1091</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Validation loss</th>
<th>Without water</th>
<th>With water</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean accuracy</td>
<td>0.1940</td>
<td>0.1379</td>
</tr>
<tr>
<td></td>
<td>98.0%</td>
<td>97.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification</th>
<th>Straight</th>
<th>Elbow</th>
<th>Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation loss</td>
<td>0.1379</td>
<td>0.1369</td>
<td>0.0480</td>
</tr>
<tr>
<td>Mean accuracy</td>
<td>97.4%</td>
<td>95.1%</td>
<td>98.3%</td>
</tr>
</tbody>
</table>

Table 2 – Classification results of the straight pipe without and with water

Table 3 – Classification and regression results of straight, elbow, and branch pipe

Each Table 1-3 indicates the mean classification accuracy of cross-validation according to Condition 1-3 and the regression validation loss in terms of the configuration of pipes. In condition 1, each classification accuracy of impulse, continuous, modulation sources is 98.7%, 97.4%, and 96.4%. This shows that noise source localization on a pipe regardless of source types which are able to exist in piping systems.

In condition 2, both classification accuracy of with and without water presence are over 97%. Most pipes being used in industries are filled with gas or water, which means that this localization method can be applicable in pipes with water or gas.

In condition 3, each classification accuracy is more than 95% according to configuration of pipes. Moreover, each the validation loss of regression is under 0.1415. These indicate that although there are many configurations of pipes in piping systems, this method can be applied to even unseen data.

ACKNOWLEDGEMENTS

This research is funded by Daewoo Shipbuilding & Marine Engineering (DSME) under Contract No. 0690-20210008

REFERENCES

Domestic sound event detection by shift consistency mean-teacher training and adversarial domain adaptation

Fang-Ching Chen, Kuan-Dar Chen, Yi-Wen Liu

1National Tsing Hua University, Taiwan {fumchin@gmail.com, Jerrychen445@gmail.com, ywliu@ee.nthu.edu.tw}

ABSTRACT
Semi-supervised learning and domain adaptation techniques have drawn increasing attention in the field of domestic sound event detection thanks to the availability of large amounts of unlabeled data and the relative ease to generate synthetic strongly-labeled data. In a previous work, several semi-supervised learning strategies were designed to boost the performance of a mean-teacher model. Namely, these strategies include shift consistency training (SCT), interpolation consistency training (ICT), and pseudo-labeling. However, adversarial domain adaptation (ADA) did not seem to improve the event detection accuracy further when we attempt to compensate for the domain gap between synthetic and real data. In this research, we empirically found that ICT tends to pull apart the distributions of synthetic and real data in t-SNE plots. Therefore, ICT is abandoned while SCT, in contrast, is applied to train both the student and the teacher models. With these modifications, the system successfully integrates with an ADA network, and we achieve 47.2% in the F1 score on the DCASE 2020 task 4 dataset, which is 2.1% higher than what was reported in the previous work.

Keywords: Sound event detection, adversarial domain adaptation, semi-supervised learning

1 INTRODUCTION
In sound event detection (SED), it is desired for a model to determine the onset and offset of the events precisely. However, marking the time stamps of events in an audio clip is a time-consuming and labor-intensive task when anyone attempts to provide such strongly-labeled data to train a sound event classifier. Due to the growing attention to SED, a competition called Detection and Classification of Acoustic Scenes and Events (DCASE) has been held annually since 2017. Currently, there are two urgent issues to address in this field; the first is how to utilize unlabeled data effectively, and the second is how to utilize a large amount of strong-labeled synthetic data to real-world environment. For the first problem, several semi-supervised learning strategies (SSL) have been utilized in the past, including pseudo-labeling (1), mean-teaching (2), interpolation consistency training (ICT) (3), and Mixmatch (4); these strategies have been proven to help the SED system learn from the unlabeled data efficiently. Beside semi-supervised training strategies, the architecture of the backbone of the SED model is also worth investigating. For neural network-based backbones, convolutional neural network (CNN) (5), convolutional recurrent neural network (CRNN) (6), residual convolutional recurrent neural network (RCRNN) (7) are commonly used.

To tackle the second problem (scarcity of strongly-labeled data), one may contemplate using synthetic mixture of clips of known sound events with background audio recordings. Compared with unlabeled and weakly-labeled real data, such synthetic data are appealing in that the accurate time stamps of sound events can be automatically created. However, the synthetic mixtures may have acoustic mismatch to the real-world recordings. Such mismatch needs to be regarded as domain differences, and various domain adaptation techniques (8, 9, 10) have been proposed to align the distributions of synthetic and real audio samples in the feature space and alleviate the domain-gap problem.

In this research, we studied an existing mean teacher-based SED model (11) (from our research group) and found that its semi-supervised learning strategies somehow conflict with adversarial domain adaptation (ADA). Some performance analysis and visualization of data distributions in the feature space led us to believe that ICT...
should be abandoned while SCT could be emphasized by deploying it to both teacher and student models. We call this new arrangement *shift consistency mean-teacher* (SCMT) training. Thereby, we successfully integrated SCMT strategies with ADA and improved the SED performance in the DCASE 2020 task 4 dataset. The rest of this paper is organized as follows: in Sec. 2, we give an overview of the baseline model and our newly proposed methods. Then we introduce the dataset used in the experiments. In Sec. 3, the parameters of data pre-processing and experiment setup will be described. Finally, in Sec. 4, we discuss and compare the performance of different models and strategies, also outlining the mutual incompatibility between ICT and ADA with both features visualization and quantitative analysis.

2 METHODS

The existing mean-teacher model is hereafter referred to as *our previous work* (11) and briefly reviewed in section 2.1. In sections 2.2 and 2.3, the changes we made in this research will be described.

2.1 Baseline

In our previous work, a novel backbone and several effective mean-teacher based semi-supervised strategies were proposed and integrated. A convolutional recurrent neural network which contained feature-pyramid components (FP-CRNN) was used as the backbone; it was comprised of CNN blocks, bidirectional gated recurrent (GRU) (12) unit cells, an attention part to generate the clip-level and frame-level output, and most importantly, feature-pyramid (13) components which were originally used in object detection task in computer vision. With the help of the feature-pyramid components, the model could utilize multi-scale features. Figure 1 shows the architecture of FP-CRNN.

The semi-supervised learning strategies that were utilized in our previous work included the mean-teacher technique, weakly pseudo-labeling, ICT, and SCT. Generally speaking, mean-teacher approaches contain a teacher model and a student model that share the same network architecture but have different weights. For the student model, the weights are learned from back propagation, while the weights of the teacher model are updated by the exponential moving average of the parameters of the student model. The loss function is given as follows,

\[
L = L_{w,BCE} + L_{s,BCE} + r(t)(L'_{w,MSE} + L'_{s,MSE}),
\]

where \(t\) denotes the current step of training,

\[
 r(t) = \exp\left[-5 \left(1 - \frac{1}{T} t\right)^2\right]
\]

is a ramp-up function, and \(T\) denotes the ramp-up length; the subscripts \(w\) and \(s\) denote clip-level output and frame-level output, respectively, \(L, L'\) denote the loss between outputs of the student model and the ground truth and the loss between the outputs of the teacher model and the student model, respectively. Finally, MSE and BCE stands for mean squared error and binary cross entropy.

Next, in our previous work, a weakly pseudo-labeling (PL) technique was adopted to infer unlabeled data into weakly-labeled data by an audio tagging system which was pre-trained with weakly-labeled and strongly-labeled data. Generally speaking, compared with the accuracy in frame-level sound event detection, the accuracy in clip-level sound event classification is easier to improve by adopting a deeper neural network (11). By generating reliable weak pseudo-labels, the performance of the sound event detection system could also be improved. Another semi-supervised learning strategy that was utilized is interpolation consistency training (ICT). It encouraged the prediction of the interpolation of unlabeled data to be consistent with the interpolation of predictions of the unlabeled data in the following manner; we desired that

\[
F_{\theta}(\text{Mix}_\lambda(u_j, u_k)) \approx \text{Mix}_\lambda(F_{\theta'}(u_j), F_{\theta'}(u_k)),
\]

where \(\text{Mix}_\lambda(a, b) = \lambda a + (1 - \lambda) b\), \(u_j, u_k\) denote unlabeled data, \(F_{\theta}\) denotes a student model and \(F_{\theta'}\) denotes a teacher model. In our previous work, \(\lambda\) was randomly sampled from Beta distribution.

Though ICT helps the model to be consistent when facing ambiguous data, we still remove the ICT in this paper, and the reason will be given in section 4.3. Last but not the least, our previous work proposed *shift*
consistency training (SCT), which was inspired by ICT. It encouraged the predictions of time-shifted and pitch-shifted data to be consistent with time-shifted and pitch-shifted predictions. The loss function was designed as follows,

$$L_{SCT} = L_{w,f,BCE} + L_{s,f,BCE} + L_{st,BCE} + r(t)L_{st,MSE},$$

where subscripts $w$ and $t$ denote clip-level outputs (weak) and frame-level outputs (strong), respectively, and $f$ and $t$ denote frequency shift and time shift, respectively.

2.2 Shift consistency mean-teacher training

SCT successfully helped the previous model to be more consistent by randomly shifting the spectrograms in time and frequency; however, it was only applied to the student model during training. In this paper, we propose a new version of SCT called shift consistency mean-teacher training (SCMT) by strengthening the relationship between the mean-teacher approach and SCT. For the implementation of the SCMT, we deploy SCT to both the student model and the teacher model. Figure 2 shows the concept and architecture of SCMT, and the loss function is devised as below,

$$L_{SCMT} = L_{SCT} + r(t)(L'_{w,f,MSE} + L'_{w,s,MSE} + L'_{s,f,MSE} + L'_{s,t,MSE}),$$

where the subscript $w, s, f$ and $t$ have the same meanings as in Eq. (4).

2.3 Adversarial domain adaptation

Adversarial domain adaptation (ADA) (14) refers to the broad idea to generate domain-invariant embedded features by fooling a domain discriminator $D$ with a generator $G_f$. During its training, a gradient reversal layer (GRL) (15) acts as an identity transform during forward propagation, but changes the sign of the gradient before passing to the preceding layers during back-propagation. By placing GRL between $G_f$ and $D$, $G_f$ and $D$ are optimized in a reverse direction, while the label predictor $G_s$ will also be trained. The following shows the learning objective functions:

$$E(\theta_f, \theta_s, \theta_d) = L_s(\theta_f, \theta_s) + L_d(\theta_f, \theta_s) - \lambda_d \cdot L_{d,BCE}(\theta_f, \theta_s, \theta_d),$$

where $\theta_f$, $\theta_s$ and $\theta_d$ denote the parameters of $G_f$, $G_s$ and $D$, respectively. $\lambda_d$ denotes the adversarial loss weight of the domain loss. $L_d$ denotes the frame-level domain loss. The subscript $w, s$ have the same meanings as in Eq. (1). The following equations describe the adversarial training process,

$$\hat{\theta}_f = \arg\min_{\theta_f} E(\theta_f, \theta_s, \hat{\theta}_d),$$
Figure 2. Network architecture of SCMT. \( \eta \sim \mathcal{N}(0, 0.5) \) denotes the Gaussian noise added to the input of the teacher model. \( t \) and \( f \) denote time and frequency, respectively; \( \tau \) and \( \nu \) denote the time and frequency shift, respectively. \( F^\theta_0 \) denotes a student model, \( F^{\theta'}_0 \) denotes a teacher model and the superscripts \( w \) and \( s \) denote clip-level and frame-level outputs from the model, respectively.

\[
\hat{\theta}_d = \arg\max_{\theta_d} E(\hat{\theta}_f, \hat{\theta}_s, \theta_d).
\]  

In the present research, we adopt the concept of two-stage (9) domain adaptation. In the first stage, we train the model that contains FP-CRNN and the semi-supervised learning strategies mentioned above to ensure the ability to detect the sound event classes. In the second stage, we add adversarial domain adaptation into the training to bridge the domain gap between real and synthetic data.

3 EXPERIMENTS

DCASE 2020 domestic environment sound event detection (DESED) dataset is used in the experiments. It provides ten different domestic sound event classes. The dataset consists of 10-second audio clips either recorded in domestic environments or in synthesized soundscapes. Each audio clip contains at least one sound event. For the training set, 1,578 weakly-labeled, 14,412 unlabeled real soundscapes, and 2,584 strongly-labeled synthetic soundscapes are provided. We evaluated the performance on the validation set which has 1,168 strongly-labeled real recordings.

Log-mel spectrograms were adopted as the input to the present model. To generate the spectrograms, all audio clips were first resampled to 16 kHz, while setting window size, hop length, and the number of mel bins to 2048, 255 and 128, respectively. Finally, for each audio clip, a \( 648 \times 128 \) mel-log spectorgram was generated.

The amount of time shift and frequency shift for SCT and SCMT was normally distributed between \(-2\) to \(2\) sec and \(-4\) to \(4\) mel bins, respectively. As for weakly pseudo-labeling, we fine-tuned a pre-trained ResNet18 and used it to produce weak labels from unlabeled data.

4 RESULTS AND DISCUSSION

All experiments adopted the pseudo-labeling strategy and the mean-teacher approach during training.

4.1 Evaluation of shift consistency mean-teacher model and domain adaptation

Table 1 shows that adversarial domain adaptation consistently improved the F1-score of all the models. Meanwhile, FP-CRNN with SCMT outperformed the FP-CRNN with SCT from our previous work by 2.0% before ADA, and 3.5% after ADA. Not only does this show that SCMT performs better than SCT, it also indicates that the compatibility between SCMT and ADA is better than between SCT and ADA. Also, Table 2 shows the model we propose in this paper performed 2.1% better than the highest F1-score that was reported in our previous work.
Table 3, however, shows that the performance of the models dropped when integrating both ICT and SCT with ADA. Thus, mutual compatibility between SSL strategies and ADA is a problem which needs to be further investigated. To this end, we applied t-SNE and silhouette analysis to visualize and quantify what happened in the embedded feature space when various combinations of learning techniques were applied.

**Table 1. F1 score(%) comparison of different models before and after domain adaptation**

<table>
<thead>
<tr>
<th>Model</th>
<th>Before ADA</th>
<th>After ADA</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRNN</td>
<td>38.2</td>
<td>40.3</td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>40.8</td>
<td>42.1</td>
</tr>
<tr>
<td>FP-CRNN + SCT</td>
<td>42.7</td>
<td>43.7</td>
</tr>
<tr>
<td>FP-CRNN + SCMT</td>
<td>44.7</td>
<td><strong>47.2</strong></td>
</tr>
</tbody>
</table>

**Table 2. Comparison with the best-performing models from our previous work**

<table>
<thead>
<tr>
<th>Architecture</th>
<th>SSL Strategies</th>
<th>ADA</th>
<th>F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRNN</td>
<td>v SCT</td>
<td>-</td>
<td><strong>45.1</strong></td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>v SCT</td>
<td>-</td>
<td>44.5</td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>- SCMT</td>
<td>v</td>
<td><strong>47.2</strong></td>
</tr>
</tbody>
</table>

**Table 3. Incompatibility between ADA and semi-supervised techniques. The F1 score(%) of several architecture and strategies before and after domain adaptation is listed.**

<table>
<thead>
<tr>
<th>Architecture</th>
<th>SSL Strategies</th>
<th>ADA</th>
<th>F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRNN</td>
<td>ICT+SCT</td>
<td>-</td>
<td><strong>45.1</strong></td>
</tr>
<tr>
<td>CRNN</td>
<td>ICT+SCT</td>
<td>v</td>
<td><strong>44.2</strong></td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>ICT+SCT</td>
<td>-</td>
<td><strong>44.7</strong></td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>ICT+SCT</td>
<td>v</td>
<td><strong>43.0</strong></td>
</tr>
</tbody>
</table>

**4.2 Discussion on domain adaptation and its compatibility to semi-supervised strategies**

First, to verify the effectiveness of domain adaptation, we visualize the embedded features before and after domain adaptation using t-SNE. By inspecting the distributions of real and synthetic domain in Figure 3, one may be inclined to conclude that domain adaptation has ability to narrow the domain gap. Further, we conducted the embedded feature analysis on our training strategies using t-SNE individually. Figure 4, 5 and 6 show embedded features distribution using different training techniques. In Figure 4, We observe that training with ICT tended to widen the gap between synthetic and real data, which is opposite to what domain adaptation did in Figure 3. Moreover, the domain gap can still be clearly recognized after applying ADA. In contrast, though the difference between the distributions of real and synthetic audio data is noticeable when simply using FP,
combining FP with SCT or SCMT, the situation seems not as extreme as the results produced by training with ICT.

To quantitatively justify the observations above, we used silhouette analysis (16) to evaluate the level of dispersion between the feature distributions of real and synthetic data. For each data entry $i$, the silhouette coefficient $s_i$ is defined as follows,

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)},$$  \hspace{1cm} (9)

where $a_i$ denotes the average distance between data point $i$ and all other data within the same cluster, and $b_i$ denotes the lowest average distance between data point $i$ and all the points in any other cluster.

![Figure 3. t-SNE visualization of embedded features](image)

The value of $s_i$ ranges from $-1$ to $1$. A value close to $1$ indicates that the sample is far away from neighboring clusters. If the silhouette coefficient is close to $0$, it indicates that the sample is close to the decision boundary between two neighboring clusters. Finally, a negative value indicates that the sample might be in a wrong cluster.

The average of silhouette coefficient over all samples is called the silhouette score. By calculating the silhouette score of the t-SNE feature maps clustered by the domain labels (real or synthetic), we can quantify the extent to which the model pulls the domains apart. The results are shown in Table 4; note that the scores of FP-CRNN with ICT are higher than other SSL strategies combinations, which means it has the tendency to separate real and synthetic domains. Also note that, by applying ADA to the model, the silhouette score can be decreased, which means that ADA successfully encourages the feature extractor to produce domain-invariant embedded features.
By investigating the implementation of ICT in our previous work, we noticed that Eq. (3), which only uses real unlabeled data, boosts the performance by generating and training on ambiguous data in the real domain itself. We suspect that this process concentrates the distribution of the real domain, while pushing the distributions of real and synthetic audio data away from each other. Consequently, it leads to the consistency within synthetic or real domains alone, but not the consistency in the entire dataset. This being said, however, ICT and ADA has been successfully integrated in the field of computer vision (17). We believe that the mutual compatibility of ADA and ICT is a topic worth deeper investigation in sound event detection in the future.
Table 4. F1 (%) and Silhouette score comparison of different SSL strategies before and after domain adaptation

<table>
<thead>
<tr>
<th>Architecture</th>
<th>SSL Strategies</th>
<th>ADA</th>
<th>F1 (%)</th>
<th>Silhouette score</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP-CRNN</td>
<td>-</td>
<td>-</td>
<td>40.8</td>
<td>0.11</td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>- v</td>
<td></td>
<td>42.1</td>
<td>0.05</td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>ICT</td>
<td>-</td>
<td>43.2</td>
<td><strong>0.23</strong></td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>ICT v</td>
<td></td>
<td>43.9</td>
<td><strong>0.24</strong></td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>SCT</td>
<td>-</td>
<td>42.7</td>
<td>0.09</td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>SCT v</td>
<td></td>
<td>43.7</td>
<td>0.04</td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>SCMT</td>
<td>-</td>
<td>44.7</td>
<td>0.05</td>
</tr>
<tr>
<td>FP-CRNN</td>
<td>SCMT v</td>
<td></td>
<td>47.2</td>
<td>0.03</td>
</tr>
</tbody>
</table>

5 CONCLUSIONS
In this work, we demonstrate a novel combination of learning strategies for SED by (1) replacing SCT with SCMT and (2) removing the ICT, which was shown to conflict with domain adaptation according to t-SNE visualization and silhouette analysis. Hence, the proposed model surpassed the best-performing model from our previous work by 2.1% in terms of F1-score on the DCASE 2020 task 4 dataset.

ACKNOWLEDGEMENTS
This research is supported by the Ministry of Science and Technology of Taiwan (Grant No. 110-2634-F-002-050).

REFERENCES


ABS-0610

**Information Geometric Approach to Source Localization and Tracking**

Jaehyoun Ha; Dongsoo Seo; Joonki Min

Agency for Defense Development, Korea

**ABSTRACT**

According to the conventional approach to an estimation of source location with passive sensor arrays, a set of parameters to be solved are assumed to construct Euclidian space in a geometric sense. With information geometric points of view, the space of probability distribution of parameters is regarded as a statistical manifold. Therefore, the differential geometric terms such as Riemannian metric tensor and geodesics are more appropriate on a curved manifold as a metric rather than applying Euclidian distances between parameters. Examples of information geometric application on a source localization with sensor arrays and their intuitive results are suggested on this research.

Keywords: Information Geometry, Riemannian Geometry, Source Localization

1. INTRODUCTION

In 1945, Rao (1) suggested that the parameterized probability distributions construct a differential manifold in which distances between two distributions could be measured by geometric metrics. Those differential geometric approach on statistics enlightened to many scientific fields which deals with stochastic data. Information geometry by Amari (2) is influenced by Rao’s achievement. Differential geometry is a mathematical discipline that elaborates the infinitely differentiable curved space and quantify its metric with mathematical techniques such as differential calculus. The set of probability distributions defined by the parameters to estimate constructs a statistical manifold and Fisher Information Matrix (FIM) is a role of a metric tensor in differential geometry. With a metric tensor, a distance between two points on a curved manifold, so called a geodesic, can be derived.

In this study, we apply the Iterative extended Kalman Filter (IKF) incorporated with Natural Gradient Decent (NGD) algorithm to the Sullivan’s target tracking model (3).

2. TARGET TRACKING WITH NATURAL GRADIENT DECENT

2.1 Information Geometry

On an n-dimensional parameter estimation problem, probability distribution function of the sample $x$ is given as $p(x|\theta)$, a set with the given probability distribution is $S = \{p(x|\theta)\}$, and $S$ is n-dimensional statistical manifold. The parameter $\theta$ rolls as coordinates of the manifold, and its FIM works as a metric tensor on a Riemannian manifold. If all the metric tensors on a manifold are positive definite, the manifold is defined as Riemannian manifold, and Riemannian metric tensors are used to derive the length of geodesics inner product of two vectors. On an Information geometric view, a set of probability distribution with sample $x$ is defined as Riemannian manifold, and its FIM of the parameters are used as a Riemannian metric tensor.

2.2 Problem Formulation and Measurement Model

In this study, we introduced the Sullivan’s bearing estimation model as shown in Figure 1. Given a hydrophone moving at a constant speed $v$ in $+x$ direction, estimation of the incidence angle in each time stamp $t$ is a Bayesian model. The acoustic pressure $P$ measured by hydrophone on each time stamp is given by

$$P(x, t) = e^{i(\omega t + k(x+vt)\sin\theta)}.$$ 

Since the measurement model is non-linear with respect to the unknown parameter $\theta$, Bayesian
estimators including Extended Kalman Filter (EKF) and IKF are applicable on this problem.

Figure 1 – Incidence angle estimation with a moving hydrophone.

2.3 Natural Gradient Descent Filtering

For an optimization problem, gradient descent is widely used to estimate the local minimum on an objective function that parameters formulate the Euclidean manifold. On the other hand, NGD is applicable to the curved manifold since it incorporates the Riemannian metric tensor. Applying NGD on a multi variable estimation is more general because it covers both flat and curved manifold. According to Li’s NGDF (4), NGD is incorporated in a conventional IKF and gives us following iterative estimation of the state

\[ \hat{\theta}_{k+1} = \hat{\theta}_k - \eta_t(G^{-1}(\hat{\theta}_k))\nabla_{\theta_k} L(\hat{\theta}_k), \]

where \( \hat{\theta}_{k+1} \) is a parameter to estimate on k time stamp, \( \eta_t \) is a step size of each iteration, \( G^{-1}(\hat{\theta}_k) \) is a metric tensor on the parameter \( \hat{\theta}_k \), \( \nabla_{\theta_k} L(\hat{\theta}_k) \) is partial derivative of log likelihood of posterior distribution. Estimation for the incidence angle 45 degree begging from 0 degree with EKF, IKF and NGDF results are given Figure 2.

Figure 2 – Simulation Results

3. CONCLUSIONS

According to the study, incorporation of the NGD in Bayesian estimation enhances the performance of single target tracking with a moving sensor. But NGDF requires more computation for each step due to its complexity and additional step size search. In the given example, we estimate 2-dimensional parameter with its angle and angular velocity. Further researches on high dimension parameter estimation and more complex system model are desired in the future to prove that the application of the Riemannian metric tensor is more effective on the curved statistical manifold.

REFERENCES
ABSTRACT

Recently, the acoustic scene classification (ASC) algorithm, which classifies the environment audio signals into acoustic scenes, such as shopping malls and trains, mainly uses machine learning to achieve its goal. Various techniques, such as subsystem ensembles, network dropout, and data augmentation, have been employed to overcome the overfitting problem and improve the performance of machine-learning systems. In particular, data augmentation is easy to use because it modifies the input data only without changing the predefined model’s size or structure. However, existing data augmentation methods sometimes generate meaningless data that cannot improve ASC performance. Therefore, we tried to maximize the augmentation of meaningful data using the GridMask augmentation method, which was developed for image processing systems. Because the temporal axis and the frequency axis differ in the audio spectrogram while the axes are the same in the image signal, we propose to use checked pattern masking to change the data and to prevent loss of important data. The proposed masking strategy is promising, as it does not mask a large part of data at once. The performance of the proposed method was evaluated using ASC data from the Detection and Classification of Acoustic Scenes and Events (DCASE) 2021 Task 1A dataset.

Keywords: acoustic scene classification, machine learning, data augmentation

1. INTRODUCTION

Acoustic scene classification (ASC) deals with the recognition of an audio environment mixed with background noise and sound events. It can help determine the context of a situation and how a system behaves across various smart or IoT devices. Recent ASCs rely heavily on machine learning. In particular, after feature extraction from input data, convolutional neural network (CNN) [1] models are often used. Among the models used for ASC, we used the ResNet [2] model, a widely used CNN model. In the learning process, overfitting is one of the factors hindering machine-learning performance. Many techniques have been used to overcome this problem and to improve performance. Therefore, we tried to solve overfitting using an aggregation approach. Common techniques include Mix-up [3], Temporal crop, Time shift, and SpecAugment [4]. Using the CNN-based model, we made available the augmentation techniques used for images. Among the techniques used for images, grid augmentation masks the data of a grid pattern.

Inspired by this, we used it for augmentation of acoustic data. However, existing studies truncate a large amount of data or generate zero data, causing a loss of much data. Unlike images, we consider this issue because the meaning of data along an axis differs. In this study, data is masked to the smallest data unit to prevent the loss of important data that can affect decision making.

2. BACKGROUND

2.1 Acoustic Scene Classification

ASC is the science of listening to sounds captured by various actual and virtual equipment and

1 baek7350@knu.ac.kr
2 sjlee6@knu.ac.kr
matching them to various recorded locations. The objective is to separate the device's mixed-in noise from feature signals and to derive general ASC performance while considering various acoustic characteristic inputs from real and simulated devices. A brief diagram of an acoustic scene classifier is shown in Figure 1.

![Figure 1 – Overview of ASC](image)

2.2 Preliminary Augmentation Method for Image Signals

Regularization is a common method for training machine-learning models. Regularization methods include Dropout [5], Mix-up, data augmentation, and other approaches. GridMask [6], from which we took our cue, is primarily utilized in images and corresponds to data augmentation. GridMask creates transformations in the form of a grid on the input image and trains the model. The GridMask description is shown in Figure 2.

![Figure 2 – Description of GridMask](image)

3. OUR APPROACHES

Machine learning is the most widely used method in recent ASC problems. We divided the training and evaluation datasets to train a CNN model, which serves as the classifier's model. Figure 3 shows the overall learning and evaluation processes.
The acoustic scene classifier’s performance depends on the degree of the dataset to be trained, feature extraction parameters, and the model’s size and type. By considering these factors, the dataset was augmented to improve the classifier’s performance.

We tried to improve overall performance by solving the overfitting problem of the learning model using the given data. Sufficient data are required to improve the model’s performance. Therefore, the augmented data was made to have similarities to the data to be predicted.

As mentioned in 2.2, we tried to apply GridMask used in image signals to acoustic scene recognition. Figure 2 shows that the horizontal and vertical axes of the image are the same in the unit of pixels. However, considering the acoustic scene data we want to apply, the horizontal and vertical axes have completely different meanings. We used a Mel-spectrogram as training data for feature extraction. In this case, the horizontal axis of the input data is the time axis and the vertical axis is the frequency axis. Because of this difference, when techniques commonly used in image signals are applied to acoustic scene recognition, performance may not improve. Later, we will examine the effects of this through an experiment.

4. EXPERIMENTS

4.1 Settings

We used the Detection and Classification of Acoustic Scenes and Events (DCASE) 2021 Task 1A dataset [7] for evaluation. We utilized SGD with 0.9 momentum in the dual ResNet as the optimizer and categorical cross-entropy as the loss function. To prevent local minima and locate a deeper optimal point, the learning rate was modified using a cosine annealing learning rate scheduler with a restart. In addition, 2, 6, 14, 32, 60, 100, 130, 180, 210, 220, 250, 270, 290, 310, and 340 were chosen as the restart epochs. The
initial learning rate was set to 0.5 in the dual ResNet model. Moreover, the learning rate was reduced to 0.0004. With each restart, the restart learning rate decreased by 10%.

4.1.1 Feature Extraction
A Mel-spectrogram was used to transform the input audio data into the CNN model’s input. The specifications are shown in Table 1.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Mel</td>
<td>256</td>
</tr>
<tr>
<td>Number of FFT</td>
<td>2048</td>
</tr>
<tr>
<td>Sample rate</td>
<td>44100</td>
</tr>
</tbody>
</table>

4.1.2 Model Architecture
We used the dual ResNet [8] structure, inspired by [9, 10]. Table 2 shows the detailed parameters of the dual ResNet model. After feature extraction, the input was divided into 1–128, 129–256 bins, and the ResNet model was concatenated into two paths to compose the entire model. By doing this, it became possible to learn by dividing low and high frequencies. Figure 4 shows the overall structure of the dual ResNet model.

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low frequency (1–128 bins)</td>
<td>High frequency (129–256 bins)</td>
</tr>
<tr>
<td>CNN (1 × 1,32)</td>
<td>CNN (1 × 1,32)</td>
</tr>
<tr>
<td>Batch normalization</td>
<td>Batch normalization</td>
</tr>
<tr>
<td>ReLU activation</td>
<td>ReLU activation</td>
</tr>
<tr>
<td>Residual block (32) × 2</td>
<td>Residual block (32) × 2</td>
</tr>
<tr>
<td>Max pooling (2 × 2)</td>
<td>Max pooling (2 × 2)</td>
</tr>
<tr>
<td>Residual block (64) × 2</td>
<td>Residual block (64) × 2</td>
</tr>
<tr>
<td>Max pooling (2 × 2)</td>
<td>Max pooling (2 × 2)</td>
</tr>
<tr>
<td>Residual block (64) × 2</td>
<td>Residual block (64) × 2</td>
</tr>
<tr>
<td>Max pooling (2 × 2)</td>
<td>Max pooling (2 × 2)</td>
</tr>
<tr>
<td>CNN (1 × 1,32)</td>
<td>CNN (1 × 1,32)</td>
</tr>
<tr>
<td>CNN (1 × 1,64)</td>
<td>CNN (1 × 1,64)</td>
</tr>
<tr>
<td>Batch normalization</td>
<td>Batch normalization</td>
</tr>
</tbody>
</table>

Concatenation along the frequency axis

CNN (1 × 1,10)
Batch normalization
Global max pooling
Softmax activation
4.2 Checked Pattern Mask

As shown in Figure 5, after feature extraction, the checked pattern mask (CPM) was applied to the input data shown in the Mel-spectrogram. Unlike other techniques, the adjacent areas were not significantly removed. The parameter $\delta_x$ and $\delta_y$ were randomly set based on $r =$ pin size, $w =$ width, and $h =$ height. $N_x =$ the pin number of time axis wise, $N_y =$ the pin number of the frequency axis wise.

$$r = \text{random}(r_{\text{min}}, r_{\text{max}}), \quad w = \text{random}(w_{\text{min}}, w_{\text{max}})$$ \hspace{1cm} (1)

We randomly selected an integer pin size $r$ between $r_{\text{min}}$ and $r_{\text{max}}$. $w$ was also randomly selected from $w_{\text{min}}$ and $w_{\text{max}}$ in the same way. $r_{\text{min}}$ and $w_{\text{min}}$ were set to 1, and $r_{\text{max}}$ and $w_{\text{max}}$ were set to 7. $r$ is an integer selected from a uniform distribution between $r_{\text{min}}$ and $r_{\text{max}}$. In addition, $w$ is an integer selected from a uniform distribution between $w_{\text{min}}$ and $w_{\text{max}}$. $r_{\text{max}}$ and $w_{\text{max}}$ were an experimentally chosen
hyperparameter where 7 was used as the maximum value. The smallest masking unit in a square shape was $r$.

$$x_{\text{limit}} = \max(\delta_x) - N_x \times r, \quad y_{\text{limit}} = \max(\delta_y) - N_y \times r$$

$x_{\text{limit}}$ is the maximum value of the area to be masked, which was reduced by the product of pin size $r$ and the number of pin $N_x$. $y_{\text{limit}}$ was also similarly limited.

$$\delta_x = \text{random}(0, x_{\text{limit}}), \quad \delta_y = \text{random}(0, y_{\text{limit}})$$

The parameters $\delta_x$ and $\delta_y$ were the lower left position values of the pins to be masked and were determined randomly.

4.3 Results

Table 3 shows the log loss and F1-score for each augmentation technique.

<table>
<thead>
<tr>
<th>Augmentation</th>
<th>Log loss</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanilla</td>
<td>1.162</td>
<td>61%</td>
</tr>
<tr>
<td>Mix-up</td>
<td>1.068</td>
<td>64%</td>
</tr>
<tr>
<td>SpecAugment</td>
<td>1.080</td>
<td>62%</td>
</tr>
<tr>
<td>CPM</td>
<td>1.028</td>
<td>62%</td>
</tr>
<tr>
<td>SpecAugment + Mix-up</td>
<td>1.022</td>
<td>66%</td>
</tr>
<tr>
<td>CPM + Mix-up</td>
<td>1.022</td>
<td>67%</td>
</tr>
</tbody>
</table>

As shown in Table 3, the vanilla model without any augmentation technique had the lowest F1-score and log loss. For Mix-up, the F1-score exceeded that of the proposed CPM, but it lagged behind the log loss by 0.04. SpecAugment also had a similar F1-score but lagged behind the log loss by about 0.052. Both SpecAugment and CPM had improved performance in log loss and F1-score when used with Mix-up.

5. CONCLUSION

We studied the model’s performance improvement in ASC tasks. Our proposed CPM brought about a 1% performance improvement in the F1-score compared to SpecAugment, which resulted from the distinction between masking with a checked pattern and masking the full time and frequency axes. Although the single augmentation technique showed a bit low performance, the Mix-up technique showed better results. In addition, applying other augmentation techniques by masking a smaller area is easier than applying SpecAugment.

ACKNOWLEDGMENTS

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1I1A3069162)

REFERENCES

Normalizing Flow based Audio Coding

Inseon JANG; Wootaek LIM; Seungkwon BEACK; Taejin LEE
Electronics and Telecommunications Research Institute (ETRI), Republic of Korea

ABSTRACT
Recently, deep neural network-based audio compression has been widely studied. Some of them include attempts at generative models, but most of them are based on autoencoder networks and focus on improving the decoded sound quality through adversarial training of generative adversarial network (GAN). In this paper, we present an audio compression using a flow based-generative model. With the benefit of the normalizing flow factor-out structure, the proposed method improves the compression efficiency while maintaining the encoding quality. To verify the performance of the proposed scheme, the objective evaluation using the signal-to-distortion ratio (SDR) and perceptual evaluation of speech quality (PESQ) as metrics are performed and compared with the autoencoder-based approach. The result of objective assessment shows the outperformance of the proposed method.

Keywords: Audio Coding, Normalizing Flow

1. INTRODUCTION
As one of the methods for machine learning, deep neural networks have led to remarkable performance improvements in various fields of signal processing in recent years. A generative model using a deep neural network is trained to model the distribution of data through a large amount of data, and it has reached the point of generating data that is almost indistinguishable from actual data in the field of video and speech synthesis. In the field of image compression, various studies utilizing the generative model have been made for a long time to obtain an image close to the original from a small amount of data [1, 2, 3], while research on generative models has been recently conducted in the speech and audio compression. Most of the studies are based on an autoencoder (AE) network, and focused on improving the decoded sound quality through adversarial training of generative adversarial network (GAN) [4, 5].

This paper addresses the audio compression using a flow based-generative model. Since the flow-based generative model explicitly learns the distribution of data through a sequence of invertible transformations on the probability distribution of the latent, it has less instability for model training and is possible to infer the latent, compared to other generative models such as variational autoencoder (VAE) and GAN.

There are several works that have explored the normalizing flow for image compression. In [6], the integer discrete flow for lossless compression was studied, which introduces the entropy coder based on discrete distribution to the conventional flow of learning the continuous distribution. Lossy image compression based on the normalizing flow was proposed in [7], which takes advantage of the factor-out structure to sample un-transmitted latent values in learned conditional distribution during the decoding process. Previous studies have shown that the flow-based approach is able to provide low bitrate to lossless for image compression, but it has been limited to image compression.

In this paper, we present flow-based audio coding. With the benefit of the factor-out structure, the latent is extracted for each level of the normalizing flow, and quantized differently according to their entropy so that the proposed scheme improves the compression efficiency while maintaining the encoding quality. We also verify the lossy scheme proposed in [7] in the audio compression and compare with end-to-end optimized AE-based speech coding called K-Net [8]. For the implementation
of flow, WaveGlow [9] model structure is used. WaveGlow is a flow-based neural vocoder, which it generates speech samples based on the conditional spectrogram. It has been used as a base model for a flow-based approach to speech enhancement [10]. In this work, an affine coupling layer and an invertible 1x1 convolution layer of WaveGlow are introduced with some modifications such as excluding the upsampling layer for condition input and modifying the non-causal convolution in WaveNet-like blocks in the affine coupling layer. We evaluate the suggested approach using the signal-to-distortion ratio (SDR) and perceptual evaluation of speech quality (PESQ).

2. PROPOSED METHODS

2.1 Network Structure

In this work, the flow-based image compression architecture proposed in [7] was adopted as the backbone model. As shown in Figure 1, it is composed of two-level flows of mapping audio to its latent representation bijective, and the additional neural network \( \phi \) to estimate the parameters of the conditional distributions \( p(z_1|z_0) \). For the factor-out, the latent of the first flow level is divided into two equally sized partitions, then only one of them passes to the next level. We used a fully factorized model as prior \( p(z_0) \) and a discrete logistic distribution as conditional distribution \( p(z_1|z_0) \).

Latents are quantized and entropy coded according to the learned probability distribution through normalizing flows. Utilizing the factor-out in the normalizing flows, it allows storing detailed and coarse information in the upper layer and in the bottom layer, respectively. That is, since the latent extracted at each flow level has a different amount of information as well as different temporal resolutions, bitrate efficiency can be obtained by adjusting the quantization step size for each latent. In order to further reduce the bitrate, only a part of the latent is transmitted, as shown in \( z_0 \) in Figure 1, and the missing values are sampled from \( p(z_1|z_0) \) whose parameters are estimated in the parameterized network \( \phi \).

We used a WaveGlow-based architecture to implement the normalizing flow. The WaveGlow is a flow-based generative model for the neural vocoder, which generates speech samples based on the conditional spectrogram. Since conditions such as the spectrogram are not required in the proposed audio coding scheme, only an affine coupling layer and an invertible 1x1 convolution layer are introduced and no additional operation is performed on the upsampling layer. In this implementation, each flow level consisted of 6 coupling layers and 6 invertible 1x1 convolutions layers. The WaveNet-like blocks in the affine coupling layer, so-called ‘WN’ had 8-layers of dilated convolutions with gated-tanh nonlinearities, 256 channels used as residual connections, and 128 channels in the skip connections. In order to reduce the latency for audio encoding, its convolution with 3 taps was modified to be causal.

![Figure 1 – Overview of the proposed flow-based audio coding](image-url)
2.2 Loss Function

To train the proposed model, the modified rate-distortion loss proposed in [7] was used, which extends the rate-distortion loss to include the deviation of the reconstructed audio $\hat{x}$ through the sampling path as follows:

$$L(x; \theta) = -\log p_\theta(\hat{x}) + \lambda(d(x,\hat{x}) + d(x,\hat{x}))$$

where $\hat{x}$ is obtained from the rounded latent $z_0$ and $z_1$, i.e., $\hat{x} = f_0(\hat{z}_0,\hat{z}_1)$. Since the rounding operation is not differentiable, we use noise channel modelling, where every element of the latent space is shifted by the same random sample $u \sim U(-\Delta/2, \Delta/2)$. The mean-squared error (MSE) was used to minimize the difference between the original $x$ and the reconstructed signal $\hat{x}$ (and $\hat{x}$). To verify the proposed method, we trained a single model using a fixed $\lambda = 100$.

3. EXPERIMENTS

3.1 Training Details

We used commercially released audio clips for training the proposed neural audio codec, whose total length is 14 hours. They were obtained from a number of DVDs in 13 genres, which contain various sound types such as music, speech and mixed. For training data, the audio was mono down-mixed and resampled at 16 kHz. The model was trained using the Adam optimizer [11] with batch size 100, and the learning rate was set to a constant value of 1e-4.

3.2 Experimental Results

For testing, 12 audio files for the MPEG-D USAC (unified speech and audio coding) verification test [12] were used. They covered three categories: speech, music, and mixed, whose duration ranges between 8 and 17 seconds. To measure the proposed scheme, two metrics were used: SDR and PESQ in the recommended wideband version in ITU-T P.862.2 [13]. We evaluated PESQ score only for speech test items because it performs the objective assessment of the perceptual speech quality. We also included the K-Net which is an end-to-end optimized AE-based speech compression method [8] to verify the lossy scheme with sampling path. In this experiment, the configuration of K-Net was set the same as discussed in [8].

Table 1 presents the objective test results. For the proposed system at 96 kbps, latents were quantized with step size 1, then entropy-coded. At 56.48 kbps, different quantization step sizes were used for each latent, i.e., $\Delta = 1$ and 32 were used for $z_0$ and $z_1$, respectively. Comparing the performance for 96 kbps and 56.48 kbps, sound quality is not significantly degraded – reduction of SDR 2.93 dB and PESQ 0.24 - even though the bitrate is reduced by more than 40% through coarse quantization of $z_1$. It is caused by lower entropy of $z_1$ than that of $z_0$. At 32.64 kbps, only $z_0$ was decoded with sampling path. As shown in Table 1, the proposed method for the lossy scheme provided better performance to K-Net in SDR, but it showed a lower PESQ score than K-Net. The perceptual loss measured using Mel-frequency cepstral coefficients (MFCC) was included in the loss function of K-Net to improve the perceptual quality of the reconstructed audio. Since the proposed method does not consider the perceptual loss, verification of the perceptual quality of the proposed model has not been performed yet.

<table>
<thead>
<tr>
<th>System</th>
<th>Bitrate [kbps]</th>
<th>SDR [dB]</th>
<th>WB-PESQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>96</td>
<td>29.07</td>
<td>4.45</td>
</tr>
<tr>
<td></td>
<td>56.48</td>
<td>26.14</td>
<td>4.21</td>
</tr>
<tr>
<td></td>
<td>32.64</td>
<td>16.21</td>
<td>3.15</td>
</tr>
<tr>
<td>K-Net</td>
<td>32</td>
<td>15.31</td>
<td>3.45</td>
</tr>
</tbody>
</table>
In addition, some audio decoded with sampled latent using the lossy scheme showed aliasing observed as a horizontal line near 3 kHz (and some 4 kHz) as seen in the upper and middle spectrograms in Figure 2. However, it does not observe in decoded audio using all of the quantized latent, regardless of the quantization step size as shown in the lower spectrogram in Figure 2.

4. CONCLUSIONS

In this paper, we present an audio compression method using a flow based-generative model. The hierarchical model structure for factor-out allows for improving the compression efficiency while maintaining the encoding quality and saving the bitrate by decoding with only a part of the latent. We verified the effectiveness of the proposed system through objective assessments, and in particular, its superiority of reconstruction performance was demonstrated compared to K-Net for the lossy scheme. Future research will include the exploration of different loss functions and sampling methods to improve perceptual audio quality.

ACKNOWLEDGEMENTS

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [22ZH1200, The research of the basic media contents technologies]

REFERENCES

Deep Learning-Based Multi-Frame Filtering for Binaural Speech Enhancement

Marvin TAMMEN, Simon DOCLO

Dept. of Medical Physics and Acoustics and Cluster of Excellence Hearing4all, University of Oldenburg, Germany,
{marvin.tammen, simon.doclo}@uni-oldenburg.de

ABSTRACT
In many speech communication scenarios, head-mounted assistive listening devices capture not only the target speaker but also interfering noise sources, resulting in a degradation of speech quality and speech intelligibility. To alleviate this issue, several binaural speech enhancement algorithms such as the binaural minimum variance distortionless response (MVDR) beamformer have been proposed, which exploit spatial correlations of both the speech and noise components. Furthermore, for single-microphone scenarios it has been proposed to exploit the fact that speech is highly correlated over time, resulting in the multi-frame MVDR (MFMVDR) filter. In this contribution, we consider a binaural extension of the MFMVDR filter, which exploits both spatial as well as temporal correlations. The binaural MFMVDR filter is embedded into an end-to-end deep learning framework, where the required parameters are estimated by temporal convolutional networks (TCNs) that are trained by minimizing the mean spectral absolute error loss function. Simulation results comprising measured binaural room impulses and diverse noise sources at signal-to-noise ratios in [-5, 20] dB demonstrate the advantage of utilizing the binaural MFMVDR filter structure instead of directly estimating the multi-frame filter coefficients with TCNs.

Keywords: binaural speech enhancement, multi-frame filtering, supervised learning

1 INTRODUCTION
In many speech communication scenarios, head-mounted assistive listening devices such as binaural hearing aids capture not only the target speaker, but also interfering noise sources, resulting in a degradation of speech quality and speech intelligibility. Hence, several binaural speech enhancement algorithms have been proposed, which typically assume that adjacent short-time Fourier transform (STFT) coefficients are uncorrelated over time. When considering sufficiently long frames and a small frame overlap, this assumption is suitable, and the speech STFT coefficients can be estimated by applying single-frame binaural filters to the microphone signals, thus exploiting spatial correlations. However, when considering shorter frames (and a larger frame overlap), as is typically required for low-delay processing such as in hearing aids, the correlation between adjacent STFT coefficients increases. To exploit this temporal correlation, multi-frame algorithms have been proposed for both single- and microphone speech enhancement, which apply (complex-valued) multi-frame filters to the most recent noisy STFT coefficients of each microphone. Several approaches have been proposed to estimate these multi-frame filters, which can be categorized into statistical model-based approaches (e.g., [1], [2]) and, more recently, deep learning-based approaches (e.g., [3]–[7]). Focusing on the deep learning-based approaches, on the one hand, it is possible to estimate the filter coefficients directly using deep neural networks (DNNs) [3]. On the other hand, a certain structure can be imposed upon the filter coefficients. As an example, by using the DNNs to estimate not the filter coefficients directly, but the spatio-temporal correlations of the speech and noise components, in [7] it was proposed to impose a binaural multi-frame minimum variance distortionless response (MFMVDR) structure upon the filter coefficients. The DNNs were trained by embedding the (fully differentiable) binaural MFMVDR filter into an end-to-end deep learning framework and minimizing the mean spectral absolute error loss function [8].
In this contribution, we consider the deep binaural MFMVDR filter proposed in [7]. Simulation results using measured binaural room impulse responses from [9] as well as clean speech and noise from the third Deep Noise Suppression Challenge (DNS3) [10] at signal-to-noise-ratios (SNRs) from −5 dB to 20 dB show that the considered deep binaural MFMVDR filter outperforms directly estimating the binaural multi-frame filter coefficients using DNNs, i.e., without imposing the structure of the binaural MFMVDR filter.

2 DEEP BINAURAL MULTI-FRAME MVDR FILTER

In this section, the deep binaural MFMVDR filter proposed in [7] is reviewed.

2.1 Signal Model

We consider an acoustic scenario with a single localized speech source and a single localized noise source, both located in a reverberant room, recorded by binaural hearing aids with $M$ microphones. In the STFT domain, the noisy microphone signals $y_{m,f,t}$ are given by

$$y_{m,f,t} = x_{m,f,t} + n_{m,f,t}$$  (1)

where $x_{m,f,t}$ and $n_{m,f,t}$ denote the speech and noise components, respectively, at the $m$-th microphone, the $f$-th frequency bin, and the $t$-th time frame. Since all frequency bins are processed independently, the index $f$ will be omitted in the remainder of this paper.

In single-microphone multi-frame speech enhancement algorithms [1], [5], the noisy multi-frame vector $\bar{y}_{m,t} \in \mathbb{C}^N$ is defined as

$$\bar{y}_{m,t} = \begin{bmatrix} y_{m,t} & \cdots & y_{m,t-N+1} \end{bmatrix}^T,$$  (2)

with $\cdot^T$ denoting the transpose operator, such that (1) can be written as $\bar{y}_{m,t} = \bar{x}_{m,t} + \bar{n}_{m,t}$. In this case, using a complex-valued multi-frame filter $\bar{w}_{m,t} \in \mathbb{C}^N$, the speech component $x_{m,t}$ is estimated as

$$\bar{x}_{m,t} = \bar{w}_{m,t}^H \bar{y}_{m,t},$$  (3)

where $\cdot^H$ denotes the conjugate transpose operator.

In multi-microphone multi-frame speech enhancement algorithms [2], [6], the noisy multi-microphone multi-frame vector $\bar{y}_{m,t} \in \mathbb{C}^{NM}$ is defined as

$$\bar{y}_t = \begin{bmatrix} \bar{y}_{1,t}^T & \cdots & \bar{y}_{M,t}^T \end{bmatrix}^T,$$  (4)

such that (1) can be written as $\bar{y}_t = \bar{x}_t + \bar{n}_t$. Without loss of generality, in this paper we consider the case $M = 2$, with one hearing aid per side and one microphone per hearing aid, i.e., $m \in \{L,R\}$, where $L$ and $R$ denote the left and right side, respectively. In this case, using (complex-valued) binaural multi-frame filters $\bar{w}_{m,t} \in \mathbb{C}^{2N}$ with $2N$ taps each, the binaural speech components are estimated as

$$\bar{x}_{m,t} = \bar{w}_{m,t}^H \bar{y}_t.$$  (5)

Assuming that the speech and noise components are uncorrelated, the noisy spatio-temporal covariance matrix (STCM) $\Phi_{y,t} = \mathcal{E}\{\bar{y}_t \bar{y}_t^H\} \in \mathbb{C}^{2N \times 2N}$, with $\mathcal{E}\{\cdot\}$ the expectation operator, can be written as

$$\Phi_{y,t} = \Phi_{x,t} + \Phi_{n,t},$$  (6)

where $\Phi_{x,t}$ and $\Phi_{n,t}$ are defined similarly as $\Phi_{x,t}$.

In order to exploit speech correlations across successive time frames, it has been proposed in [1] to decompose the single-microphone multi-frame speech vector into a temporally correlated and a temporally uncorrelated component. Similarly, the binaural multi-frame speech vector $\bar{x}_t$ can be decomposed into a spatio-temporally correlated and a spatio-temporally uncorrelated component w.r.t. the left or right speech STFT coefficient $x_{m,t}$:

$$\bar{x}_t = \begin{bmatrix} \gamma_{L,t} x_{L,t} + \bar{x}_{L,t}^T \cr \gamma_{R,t} x_{R,t} + \bar{x}_{R,t}^T \end{bmatrix} \begin{bmatrix} \text{correlated} \cr \text{uncorrelated} \end{bmatrix} \begin{bmatrix} \text{correlated} \cr \text{uncorrelated} \end{bmatrix}.$$  (7)
The highly time-varying left or right speech spatio-temporal correlation vector (STCV) \( \gamma_{x,m,t} \in \mathbb{C}^{2N} \) describes the correlation between the \( N \) most recent left and right speech STFT coefficients and the current left or the right speech STFT coefficient \( x_{m,t} \), and it is defined as

\[
\gamma_{x,m,t} = E\{x_t x^*_m,t\} / E\{|x_m,t|^2\},
\]

where \( \circ^* \) denotes the conjugate operator and with \( e_L^T \gamma_{x,L,t} = e_R^T \gamma_{x,R,t} = 1 \). Here, \( e_L \) and \( e_R \) denote selection vectors with their first or \( N + 1 \)-th element equal to 1, respectively, and the other elements equal to 0.

### 2.2 Filter Definition

Aiming at minimizing the output noise power spectral densities while leaving the left and right spatio-temporally correlated speech components undistorted, the binaural MFMVDR optimization problems are defined as [1], [7]

\[
\arg\min_{w_{m,t}} w_{m,t}^H \Phi_{n,t}^{-1} w_{m,t} \quad \text{s.t.} \quad w_{m,t}^H \gamma_{x,m,t} = 1.
\]

Solving this optimization problem, the binaural MFMVDR filters are given by

\[
w_{m,t}^{\text{MFMVDR}} = \frac{\Phi_{n,t}^{-1} \gamma_{x,m,t}}{\gamma_{x,m,t}^H \Phi_{n,t}^{-1} \gamma_{x,m,t}}
\]

### 2.3 Parameter Estimation

As has been shown for the single-microphone MFMVDR filter, the performance of the MFMVDR filter strongly depends on how well the required parameters, i.e., the inverse noise STCM \( \Phi_{n,t}^{-1} \) as well as the speech STCVs \( \gamma_{x,m,t} \), are estimated from the noisy STFT coefficients [11]. In contrast to using statistical model-based estimators similar to [12], the binaural MFMVDR filter is embedded in an end-to-end deep learning framework [7]. Within this framework, separate DNNs or, more specifically, temporal convolutional networks (TCNs) are used to map features computed from the noisy microphone signals to the required inverse noise STCM and speech STCV estimates \( \Phi_{n,t}^{-1} \) and \( \gamma_{x,m,t} \), respectively (see Fig. 1). Using these estimates, the binaural MFMVDR filters in (10)
are computed and applied to the noisy binaural multi-frame vector $y$ to yield the binaural speech estimates. The TCNs are trained by minimizing a speech enhancement loss function computed at the output of the deep binaural MFMVDR filter instead of providing explicit parameter labels. Furthermore, a-priori knowledge about the properties of the estimated parameters is exploited during the estimation process. More explicitly, it is ensured that the estimated inverse noise STCM is Hermitian positive-definite and that $e_{L}^{T}y_{s,L} = e_{R}^{T}y_{s,R} = 1$. For more details on how this is achieved, it is referred to [7].

3 MATERIALS AND METHODS

3.1 Baseline Algorithm
To allow investigating the effect of not imposing vs. imposing the binaural MFMVDR structure, in the baseline algorithm the real and imaginary parts of the binaural multi-frame filter coefficients are estimated using a TCN directly, instead of first estimating the inverse noise STCM and speech STCV and afterwards computing the filter coefficients using (10) (see Figure 1). Similarly to [3], the real and imaginary parts of the filter coefficients are bounded to $[-1, 1]$ using a hyperbolic tangent activation function.

3.2 Dataset
To train and validate the considered algorithms, we combined 6000 simulated binaural room impulse responses (BRIRs) from the training subset of the first Clarity Enhancement Challenge (CEC1) dataset [13] with clean speech and noise from the training subset of the third Deep Noise Suppression Challenge (DNS3) dataset [10]. These BRIRs were simulated by considering a randomly positioned directed speech source and an omnidirectional noise point source captured by binaural behind-the-ear hearing aids in randomly sized rooms with “low to moderate” reverberation, i.e., around 0.2 s to 0.4 s. Clean speech and noise were convolved with randomly chosen BRIRs before being mixed at better ear SNRs from 0 dB to 15 dB. In total, the training and validation datasets have a length of 80 h and 20 h, respectively.

To evaluate the considered algorithms, we used measured BRIRs from the dataset proposed in [9] as well as clean speech and noise from the deep noise suppression (DNS) test dataset [14]. The dataset in [9] comprises BRIRs measured with the same binaural behind-the-ear hearing aids as in the training and validation datasets “for multiple, realistic head and sound-source positions in four natural environments reflecting daily-life communication situations with different reverberation times”. Clean speech and noise were convolved with the BRIRs before being mixed at better ear SNRs from $-5$ dB to 20 dB. In total, 100 utterances of length 10 s were considered in the evaluation. All datasets were used at a sampling frequency of 16 kHz.

3.3 Settings
For the STFT, $\sqrt{\text{Hann}}$ windows with a relatively small frame length of 8 ms and 75% overlap were used in order to increase speech interframe correlations. As features for the TCNs, we used a concatenation of the logarithmic magnitude, the cosine of the phase, and the sine of the phase, of the noisy left and right STFT coefficients. The algorithms use $N = 5$ frames, resulting in the capability of exploiting temporal correlations within 16 ms, and a minimum gain of $-20$ dB is included.

To estimate the required parameters of the deep binaural MFMVDR filter or the real and imaginary parts of the baseline binaural multi-frame filter coefficients, we used causal TCNs, with their hyperparameters fixed to 2 stacks of 6 layers with 516 hidden dimensions, yielding a temporal receptive field size of 512 ms.

As loss function, the mean spectral absolute error proposed in [8] was used, where the loss was averaged across the left and right output signals, the frequency bins, and time frames, i.e.,

$$L_{m,f,t} = \beta |x_{m,f,t} - \hat{x}_{m,f,t}| + (1 - \beta) \left| |x_{m,f,t}|- |\hat{x}_{m,f,t}| \right| \quad (11)$$

where $F$ and $T$ denote the number of frequency bins and time frames in an utterance, and $\beta = 0.4$ [8]. The binaural reverberant speech signals were chosen as the target in (11).

The TCNs were implemented based on the official Conv-TasNet implementation1, and they were trained for a maximum of 150 epochs with early stopping using the AdamW optimizer [15]. The learning rate was

1https://github.com/naplab/Conv-TasNet
4 RESULTS AND DISCUSSION
Figure 2 depicts the improvements in terms of PESQ and HASPI w.r.t. the noisy microphone signals on the evaluation dataset. First, a considerable improvement in terms of PESQ can be observed for both algorithms, with the deep binaural MFMVDR filter outperforming the filter with no imposed structure. Second, the improvement in terms of HASPI is rather small for the considered evaluation dataset. To investigate this result more closely, Figure 3 depicts the improvements vs. the input metrics for both PESQ and HASPI achieved by the deep binaural MFMVDR filter. It can be observed that the speech quality as estimated by the PESQ metric is often low within the evaluation dataset, thus leaving room for improvement. In contrast, the speech intelligibility as estimated by the HASPI metric is close to the maximum value 1 for most of the utterances within the evaluation dataset, thus leaving only little room for improvement. However, for those utterances that do exhibit a low HASPI score, the improvement tends to be considerable. Audio examples for the compared algorithms are available online.

5 CONCLUSIONS
In this paper we considered two deep learning-based multi-frame algorithms for binaural speech enhancement, which are capable of utilizing both spatial and temporal correlations of the noisy microphone signals. In the first algorithm, the binaural multi-frame filter coefficients are estimated by the DNN directly, i.e., no structure is imposed upon the filter coefficients. On the contrary, in the second algorithm the DNN is used to estimate the parameters required by the binaural MFMVDR filter. Comparing these two algorithms, simulations

\[https://uol.de/en/sigproc/research/audio-demos/22ica-tammen\]
Figure 3. Scatter plot of the PESQ and HASPI improvements vs. the input PESQ and HASPI values obtained on the evaluation dataset. PESQ scores typically range from 1 MOS to 4.5 MOS, while HASPI scores can range from 0 to 1.

comprising measured binaural room impulse responses as well as diverse noise sources at SNRs from \(-5\) dB to 20 dB demonstrate the advantage of imposing the binaural MFMVDR structure in terms of binaural speech enhancement performance.

ACKNOWLEDGMENTS

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC 2177/1 - Project ID 390895286.

REFERENCES


Outlier-guided contrastive representation learning for unsupervised anomalous sound detection

You-Siang Chen; Yi-Hsuan Lin; Mingsian Bai
Power Mechanical Engineering, National Tsing Hua University, Taiwan

ABSTRACT
In this paper, a contrastive learning approach is proposed for unsupervised anomalous sound detection (ASD) with enhanced latent-space features. One of the key issues in deep neural network (DNN)-based ASD is how to obtain a reliable representation of machine sounds with only data in the normal condition in the training stage. The network architecture consists of an autoencoder by which the latent features of normal sound are extracted so as for the network output to reconstruct the input signals. The autoencoder is trained with outliers labeled with machine identifications (IDs) associated with the recording subsets within the same machine type. The deep clustering (DC) loss is used in the optimization to attract the latent features with the same section ID of the target machine, while repelling those with different IDs. The contrastive learning-based training framework is validated for the dataset of malfunctioning industrial machine investigation and inspection (MIMII dataset). The proposed ASD approach is compared with several baselines in terms of the area under the receiver operating characteristic curve (AUC) and partial-AUC (pAUC).

Keywords: anomalous sound detection, contrastive learning

1. INTRODUCTION
Anomaly sound detection (ASD) has been an active area of research in machine learning community. In the last decades, significant progress has been made in deep neural network (DNN). ASD methods can be categorized as supervised and unsupervised anomalous detection. If the data of the normal and abnormal sounds are available, DNN techniques can be performed via supervised learning. However, the scarcity of anomalous data often renders the supervised learning approach infeasible. Therefore, it is most desirable in real-world applications to develop an unsupervised ASD technique when only normal data are available for the network training in the early stage of machine condition monitoring.

Conventionally, various unsupervised ASD were developed on the basis of the specific mechanical structure. (1)(2) With the prior knowledge of the target machine, a rule-based monitoring system can be successfully built according to the possible features appearing in the recorded signals. With the aid of the DNN, a great amount of methods without such knowledge are proposed. (3)(4)(5)(6) Recently, the challenge in the IEEE audio and acoustic signal processing society’s detection and classification of acoustic scenes and events (DCASE) (7) aims to perform the unsupervised ASD where the monitoring system should detect the anomalies outside the given normal data. Previous studies show that the reconstruction-based approaches were frequently used in the unsupervised ASD problem. These methods attempt to extract the representative latent vectors and reconstruct the original feature using the autoencoder (AE) and further assume that the large reconstruction error indicates the out-of-distribution anomalies. (8)(9) In DCASE2020, several methods of the self-supervised classifier (10), (11), (12), (13) were used to deal with the unsupervised ASD problem. In this case, the samples that are neither normal nor abnormal are considered as the anomalies in the training stage. Because the provided dataset defines the machine type and the corresponding machine identification (ID), the data that is not belong to the target machine can be treated as the outliers. Therefore, the ASD task can be reformulated into a supervised classification problem. This framework has retrieved promising results and still been widely used in the following DCASE challenges. (14)

With the great success of the above-mentioned approaches, we believe that the performance of the unsupervised ASD network can be further improved if the representative latent vectors for the target machine can be obtained. Self-supervised learning (SSL) is known for its superiority on the
unsupervised tasks due to its ability to learn representative features. (15),(16) (17) Some studies also seek to utilize the SSL methods to improve the ASD performance. (18) In this paper, we build the network on the basis of an autoencoder in which the latent features of normal sound are extracted. By imposing an additional loss function which aims to perform the deep clustering (DC) (19) with respect to the machine IDs, the latent vectors of the machine sound in different IDs can be properly separated and be further used for the calculation of the anomaly score. Based on the results of AUC and pAUC, the proposed framework shows better performance compared to the baseline approaches.

The arrangement of this paper is as follows: In the section 2, the related work which illustrates the reconstruction-based approach, outlier classifier, and a contrastive semi-supervised framework is described. In the section 3, the proposed learning approach along with the calculation of the anomaly score is presented. Section 4 includes the data that is applied in the experiment and the detailed network parameters. The results and discussion are shown in section 5 and the conclusion is provided in section 6.

2. RELATED WORK

Three approaches to be used as the baselines are reviewed in this section. In addition to the reconstruction-based method (7) and the outlier-exposure classifier (11), a recent study named contrastive semi-supervised framework (18) is also considered.

2.1 Baselines 1 – Autoencoder

The AE structure as depicted in Fig. 1 was built upon the baseline used in DCASE2020 (7). The input feature \( x \in \mathbb{R}^{F \times T} \) is passed to the encoder that consists of four-layer fully-connected network (FNN) along with the batch normalization and the rectified linear unit (ReLU). Followed by the bottleneck layer that extracts the representative vector \( v_{\text{AE}} \in \mathbb{R}^{D} \) using the leaky ReLU function, the input feature is reconstructed as \( \hat{x} \in \mathbb{R}^{F \times T} \) with the decoder that also comprises of four-layer FNN components.

![Figure 1 – The AE structure that consists of a four-layer encoder and decoder. A bottleneck layer is in the middle of the network to extract the important feature of the normal data. The goal of the AE model is to reconstruct the input feature of the normal data.](image)

The mean-square-error (MSE) is used as AE’s objective function \( L_{\text{AE}} \) to minimize the distance of the reconstruction. In the inference stage, the anomaly score \( A_{\text{AE}} \) is calculated as

\[
A_{\text{AE}}(x) = \| x - \hat{x} \|_F,
\]

where \( \| \cdot \|_F \) denotes the Frobenius norm. It is assumed that the reconstruction error would be large if the data is out of distribution from the training set.

2.2 Baselines 2 – Outlier classifier

Since DCASE2020, the outlier classifier has shown its efficacy in the the ASD task. In Fig. 2, the classifier was constructed based on the convolutional neural network (CNN). We utilized six-layer two-dimensional CNN with each layer doubled its output channel to reduce the dimension of the input feature \( x \). Then, two-layer FNN is used to further predict the class \( \hat{y} \) of the \( K \) machine IDs. Each layer contains the batch normalization and the leaky ReLU function, while the last layer is the softmax function to predict the probability of each class.
Figure 2 – The structure of the outlier classifier comprising of six-layer CNN and two-layer FNN. Each layer contains the batch normalization and the leaky ReLU function and the output layer with softmax function predicts the probability of the \( K \) machine IDs.

For the outlier classifier, the objective function is the cross-entropy function \( \mathcal{L}_{AE} \) described as

\[
\mathcal{L}_{CE} = - \frac{1}{N} \sum_{i} y_i \log(\hat{y}_i),
\]

where \( N \) is the number of the samples. The anomaly score \( \mathcal{A}_i \) is defined as follows with \( \hat{y}_i \) being the probability of the target machine ID.

\[
\mathcal{A}_i(x) = \log(\frac{1 - \hat{y}_i}{\hat{y}_i}).
\]

2.3 Baselines 3 – Contrastive semi-supervised framework

In the recent years, the SSL technique gains much attention owing to its capability of representing unlabeled data for the unsupervised learning. From the above widely used ASD approaches, researchers seek to find the valuable latent feature by constructing a joint network that adopts the contrastive learning method. (18) Figure 3 illustrates the network structure consisting of the AE and the outlier classifier. The latent feature of the two models \( \mathbf{v}_{AE} \) and \( \mathbf{v}_o \) are projected to the same dimension \( p \) and \( u \in C \) through the linear transformation, followed by the contrastive loss as described below:

\[
\mathcal{L}_{\text{contrastive}} = -\sum_{i} \log \frac{\exp(<\mathbf{u}, \mathbf{p}_i>/\tau)}{\sum_{i,j} \exp(<\mathbf{u}, \mathbf{p}_j>/\tau)},
\]

where \(<,>\) denotes the inner product and \( \tau \) is the hyperparameter of the temperature. The optimization of the network is performed with the summation of the three losses in Eq. 5. During the inferencing stage, we only used the classifier for the calculation of the anomalous score as described in Eq. 3.

\[
\mathcal{L}_{\text{total}} = \mathcal{L}_{AE} + \mathcal{L}_{CE} + \mathcal{L}_{\text{contrastive}}
\]
of the two models are projected to the same dimension through the linear transformation.

3. METHOD

The key issue of the unsupervised ASD task is to obtain a reliable representation in the normal machine data. The contrastive loss in (18) implies the following idea: pull together a latent vector encoded by the AE and the corresponding vector generated by the classifier, while push apart all the other latent vectors. However, in the ASD task, we wish to attract the latent vectors encoded from the same machine ID, and repel those vectors from the outliers. Therefore, the DC loss (19) was adopted to guide the arrangement of the latent vectors in the AE model.

3.1 DNN architecture and the objective functions

In our proposed method, the AE model is built upon the baseline 1 described in Sec. 2. To make the AE encode ordered latent representation associated with the machine IDs, an additional projection layer comprising of an FNN layer and a Tanh activation function is applied to the latent space of the AE model. The projection layer has the same size with the bottleneck layer. Followed by the DC loss in Eq. 6, the projected vectors \( q \) are clustered according to the machine IDs.

![Figure 4 – An additional projection layer comprising of an FNN layer and a Tanh activation function is applied to the latent vector of the AE model. The AE and DC loss are jointly trained during the training stage.](image)

The DC loss (19) is formulated as follows:

\[
\mathcal{L}_{\text{DC}} = \|QQ^T - YY^T\|_F^2 = \|Q^TQ\|_F^2 - 2\|Q^TY\|_F^2 + \|Y^TY\|_F^2.
\]

in which \( Q \) is constituted with the output vectors \([q_1^T, q_2^T, \ldots, q_N^T] \) and \( Y \) is constructed by the one-hot vectors \([y_1^T, y_2^T, \ldots, y_N^T] \) corresponding to the machine IDs. The DC loss performs inner-product to every output embedding vector making the in-class data attract together and out-class data separate apart. In this way, the AE model is forced to generate unique latent features associated with the machine IDs. The total loss of the network is the summation of the loss of the AE and DC as follows:

\[
\mathcal{L}_{\text{total}} = w_{\text{AE}} \mathcal{L}_{\text{AE}} + w_{\text{DC}} \mathcal{L}_{\text{DC}},
\]

where \( w_{\text{AE}} \) and \( w_{\text{DC}} \) are the weighing of two losses which are chosen here to be 0.1 and 1, respectively.

3.2 Anomaly score

During the inferencing stage, the decoder of the AE is discarded, while preserving the projection layer for the calculation of the anomaly score. The following anomaly score is defined to calculate the condition of the machine sound:

\[
\mathcal{A}_{\text{DC}}(x) = 1 - \frac{q^Tq}{\|q\| \|q\|}
\]

where the \( q_c \) is the centroid vector of the target machine ID which is averaged based on output vectors of the target machine in the training set, and \( q_i \) is the \( i \)th output vector of the testing data. Large value implies the degree of the anomaly.
4. EXPERIMENTS

4.1 Dataset

In the experiment, the MIMII dataset (20) in the DCASE2020 task was used to validate the proposed method. The development set contains four machine types which are fan, pump, slide rail, and valve. The machine IDs were labeled according to the individual product model for each machine type. Besides, in the training set, the data samples were all labelled as normal, while in the testing set, normal and abnormal recordings were provided and cannot be used in the training stage. We leverage the arrangement of the outliers in (11) which views the specific machine ID as the negative sample and treats the other IDs as the positive ones.

4.2 Data preprocessing

The sampling rate of the audio recordings is 16 kHz. At first, we performed 1024 points short-time Fourier transform (STFT) with 512 hop size to a 10-second audio recording. Followed by the mel filter bank with 128 filters, the feature of the STFT spectrogram was further extracted with reduced dimensions. Then, the logarithm was applied to the mel-spectrogram to strengthen the input signals.

4.3 Parameter settings

In the training stage, Adam optimizer (21) was exploited with learning rate being 0.001, beta1 = 0.9, and beta 2 = 0.999. For the outlier classifier, four machine IDs were involved during the training of the classification. The batch size for the networks in the paper was set to be 16. The detailed parameters of the proposed network and the baselines are summarized in Table 1.

Table 1 – Parameter setting of the AE and classifier. Unit denotes the output channel of each layer, BN is the batch normalization, \((k_1, k_2)\) is the kernel size, \((s_1, s_2)\) is the stride step, and \((s_1, s_2)\) is the padding size of the CNN classifier.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Unit</th>
<th>BN</th>
<th>Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNN</td>
<td>128</td>
<td>✓</td>
<td>ReLU</td>
</tr>
<tr>
<td>FNN</td>
<td>128</td>
<td>✓</td>
<td>ReLU</td>
</tr>
<tr>
<td>FNN</td>
<td>128</td>
<td>✓</td>
<td>ReLU</td>
</tr>
<tr>
<td>FNN</td>
<td>64</td>
<td>✓</td>
<td>LeakyReLU</td>
</tr>
<tr>
<td>FNN</td>
<td>128</td>
<td>✓</td>
<td>ReLU</td>
</tr>
<tr>
<td>FNN</td>
<td>128</td>
<td>✓</td>
<td>ReLU</td>
</tr>
<tr>
<td>FNN</td>
<td>128</td>
<td>✓</td>
<td>ReLU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer</th>
<th>Unit</th>
<th>((k_1, k_2))</th>
<th>((s_1, s_2))</th>
<th>((p_1, p_2))</th>
<th>BN</th>
<th>Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv2d</td>
<td>2</td>
<td>(4, 4)</td>
<td>(2, 2)</td>
<td>(1, 1)</td>
<td>✓</td>
<td>LeakyReLU</td>
</tr>
<tr>
<td>Conv2d</td>
<td>4</td>
<td>(4, 4)</td>
<td>(2, 2)</td>
<td>(1, 1)</td>
<td>✓</td>
<td>LeakyReLU</td>
</tr>
<tr>
<td>Conv2d</td>
<td>8</td>
<td>(4, 4)</td>
<td>(2, 2)</td>
<td>(1, 1)</td>
<td>✓</td>
<td>LeakyReLU</td>
</tr>
<tr>
<td>Conv2d</td>
<td>16</td>
<td>(4, 4)</td>
<td>(2, 2)</td>
<td>(1, 1)</td>
<td>✓</td>
<td>LeakyReLU</td>
</tr>
<tr>
<td>Conv2d</td>
<td>32</td>
<td>(4, 4)</td>
<td>(2, 2)</td>
<td>(1, 1)</td>
<td>✓</td>
<td>LeakyReLU</td>
</tr>
<tr>
<td>Conv2d</td>
<td>64</td>
<td>(4, 4)</td>
<td>(2, 2)</td>
<td>(0, 1)</td>
<td>✓</td>
<td>LeakyReLU</td>
</tr>
<tr>
<td>FNN</td>
<td>64</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>ReLU</td>
</tr>
<tr>
<td>FNN</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>Softmax</td>
</tr>
</tbody>
</table>

5. RESULTS AND DISCUSSIONS

5.1 Comparative evaluation of the unsupervised ASD networks

We evaluate the performance of the ASD network with the AUC:

\[
\text{AUC} = \frac{1}{N_N N_A} \sum_{i=1}^{N_N} \sum_{j=1}^{N_A} \mathcal{H}(\mathcal{A}(x_i^N) - \mathcal{A}(x_i^A)),
\]

(9)

where \(N_N\) and \(N_A\) are the number of the normal and abnormal samples, \(x_i^N\) and \(x_i^A\) denote the normal and anomalous recordings, and \(\mathcal{H}\) is the indicator function that returns 1 when the input is greater than 0 and 0 otherwise. To evaluate the performance of the ASD method under specific false positive rate (FPR), the pAUC as follows can be adopted:
pAUC = \frac{1}{pN} \left( \frac{1}{N} \sum_{i=1}^{pN} \sum_{j=1}^{\lfloor N \rfloor} \mathcal{H}(\mathcal{A}(x_i^j) - \mathcal{A}(x_i^j)) \right),

\text{where } \lfloor \cdot \rfloor \text{ is the flooring function and } p \text{ is the FPR which is selected as } 0.1.

Table 2 – Comparative results of the proposed method and the baselines. Each machine type is evaluated per machine IDs using the AUC (%) and pAUC (%). The best scores for the respective subset are boldfaced.

(a) Fan

<table>
<thead>
<tr>
<th>ID</th>
<th>AE</th>
<th>Outlier classifier</th>
<th>Contrastive framework</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC (%)</td>
<td>pAUC (%)</td>
<td>AUC (%)</td>
<td>pAUC (%)</td>
</tr>
<tr>
<td>0</td>
<td>55.15</td>
<td>48.73</td>
<td>68.57</td>
<td>60.56</td>
</tr>
<tr>
<td>2</td>
<td>76.70</td>
<td>59.21</td>
<td>84.94</td>
<td>71.91</td>
</tr>
<tr>
<td>4</td>
<td>68.22</td>
<td>53.47</td>
<td>62.50</td>
<td>51.79</td>
</tr>
<tr>
<td>6</td>
<td>85.26</td>
<td>60.00</td>
<td>92.88</td>
<td>87.17</td>
</tr>
<tr>
<td>Avg.</td>
<td>71.33</td>
<td>55.35</td>
<td>77.22</td>
<td>67.85</td>
</tr>
</tbody>
</table>

(b) Pump

<table>
<thead>
<tr>
<th>ID</th>
<th>AE</th>
<th>Outlier classifier</th>
<th>Contrastive framework</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC (%)</td>
<td>pAUC (%)</td>
<td>AUC (%)</td>
<td>pAUC (%)</td>
</tr>
<tr>
<td>0</td>
<td>77.55</td>
<td>67.52</td>
<td>70.15</td>
<td>67.77</td>
</tr>
<tr>
<td>2</td>
<td>69.40</td>
<td>65.94</td>
<td>78.68</td>
<td>67.84</td>
</tr>
<tr>
<td>4</td>
<td>94.44</td>
<td>78.47</td>
<td>96.83</td>
<td>90.82</td>
</tr>
<tr>
<td>6</td>
<td>77.43</td>
<td>61.15</td>
<td>68.84</td>
<td>64.23</td>
</tr>
<tr>
<td>Avg.</td>
<td>79.70</td>
<td>68.27</td>
<td>78.62</td>
<td>72.66</td>
</tr>
</tbody>
</table>

(c) Slide rail

<table>
<thead>
<tr>
<th>ID</th>
<th>AE</th>
<th>Outlier classifier</th>
<th>Contrastive framework</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC (%)</td>
<td>pAUC (%)</td>
<td>AUC (%)</td>
<td>pAUC (%)</td>
</tr>
<tr>
<td>0</td>
<td>99.36</td>
<td>97.21</td>
<td>99.72</td>
<td>98.65</td>
</tr>
<tr>
<td>2</td>
<td>81.42</td>
<td>60.78</td>
<td>92.04</td>
<td>80.79</td>
</tr>
<tr>
<td>4</td>
<td>85.18</td>
<td>60.21</td>
<td>91.44</td>
<td>76.52</td>
</tr>
<tr>
<td>6</td>
<td>58.89</td>
<td>50.92</td>
<td>95.36</td>
<td>78.85</td>
</tr>
<tr>
<td>Avg.</td>
<td>81.21</td>
<td>67.28</td>
<td>94.64</td>
<td>83.70</td>
</tr>
</tbody>
</table>

(d) Valve

<table>
<thead>
<tr>
<th>ID</th>
<th>AE</th>
<th>Outlier classifier</th>
<th>Contrastive framework</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC (%)</td>
<td>pAUC (%)</td>
<td>AUC (%)</td>
<td>pAUC (%)</td>
</tr>
<tr>
<td>0</td>
<td>67.31</td>
<td>53.89</td>
<td>99.93</td>
<td>99.85</td>
</tr>
<tr>
<td>2</td>
<td>65.96</td>
<td>49.63</td>
<td>99.95</td>
<td>99.75</td>
</tr>
<tr>
<td>4</td>
<td>74.68</td>
<td>52.73</td>
<td>99.99</td>
<td>99.95</td>
</tr>
<tr>
<td>6</td>
<td>58.60</td>
<td>48.36</td>
<td>78.96</td>
<td>71.66</td>
</tr>
<tr>
<td>Avg.</td>
<td>66.63</td>
<td>51.15</td>
<td>94.70</td>
<td>92.80</td>
</tr>
</tbody>
</table>
The AUC and pAUC scores for four machine types are summarized in Table 2. The ASD performance for the individual machine ID is considered and averaged for overall evaluation. The comparative results indicate that the overall performance of the proposed method is better than the baseline approaches. Furthermore, when utilizing the cluster-based anomaly calculator, the ASD methods have outstanding efficacy on the non-stationary machine sound such as the slide rail and valve. The average score of the AUC and pAUC in our method can achieve 96.66% and 90.28% for the slide rail, and 98.30% and 94.60% for the valve. For the stationary sound like the fan and pump, the contrastive framework and our method have comparable performance that outperforms the AE and the outlier classifier. This reveals the effectiveness of the latent representation learning.

5.2 Visualization in latent space

To visualize the distribution of the latent space in the AE model, we exploit t-distributed stochastic neighbor embedding (t-SNE) (22) to the latent space of AE. The latent vectors with 64 dimensions are projected to the two-dimensional space as illustrated in Fig. 5. Each color represents the machine ID. The result in Fig. 5(a) shows that the encoded vectors are mixed when the DC loss is not applied. However, through the additional DC loss, the latent vectors encoded by the AE become ordered and can be further be used for the ASD task.

(a) The latent space in AE without DC loss
(b) The latent space in AE with DC loss

Figure 5 – The latent vectors of the machine valve that are projected from 64 to 2 dimensions through the t-SNE approach. Each color represents the machine ID where red is 0, green is 1, blue is 2, and purple is 3 in each machine type.

6. CONCLUSIONS

In this paper, we have presented a contrastive latent representation learning approach by applying the DC loss to the latent space of the AE model. The output of the projection layer seeks to attract the data with the same machine ID, while excluding the target machine with the different IDs that are inferred as the outliers. The joint optimization based on MSE and DC enables the ordered latent representation to be encoded by the AE. In addition, these latent vectors can be used in the calculation of the anomaly score. The results confirm the efficacy in terms of AUC and pAUC obtained using the proposed method for ASD tasks, compared with the baseline approaches.

ACKNOWLEDGEMENTS

The work was supported by the Ministry of Science and Technology (MOST) in Taiwan, under the project number 110-2221-E-007-027-MY3.

REFERENCES

(3) Marchi E, Vesperini F, Eyben F, Squartini S, Schuller B. A Novel Approach for Automatic Acoustic
Novelty Detection using a Denoising Autoencoder with Bidirectional LSTM Neural Networks, 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP); April 2015, 1996-2000.


Normalizing flow based conditional density estimator ensemble for industrial anomaly detection

Jose A. LOPEZ\textsuperscript{1}; Georg STEMMER\textsuperscript{2}; Paulo LOPEZ-MEYER\textsuperscript{3}, Juan A. DEL HOYO ONTIVEROS, Hector A. CORDOURIER
Intel Corp, USA

ABSTRACT
We present a novel architecture for acoustic anomaly detection which uses several normalizing flow based conditional density estimators to assign anomaly scores to audio samples. This model was part of the winning submission to the DCASE2021 Challenge Task 2, an annual competition for anomalous sound detection. The data used for that challenge builds on the data from the previous year by including domain shifts, or variations in the recording setup and/or machine operating conditions. In this paper we analyze and evaluate the capability of our model, which we call NF-CDEE, for the first time on the DCASE2020 development data, which we believe is more suitable to represent the performance of our model because it does not include domain shifts. We show that the model generalizes well to all machine categories and can scale to accommodate resource constraints.

Keywords: Industrial, Anomaly, Detection

1. INTRODUCTION
We consider the problem of determining whether the sound produced by a machine is normal or anomalous using only the normal-state samples. Solving this problem enables one to monitor the condition of machines using audio. The advantages of machine condition monitoring are well understood and include reduced downtime, longer machine use, less product waste, etc. There are well-established approaches like vibration and acoustic emission sensors, but they have disadvantages like being cost inefficient and generally requiring physical contact to the machine. These pain points can be addressed by using microphones, which additionally can theoretically be used to monitor multiple machines using, for example, beamforming (1).

The audio community is aware of these advantages and has been working towards machine learning based solutions, as evidenced by DCASE2020 and DCASE2021 Task 2 Challenges (2,3). Out of the different machine learning anomaly detection approaches, unsupervised methods are the most prized because they are the easiest to use and deploy. In particular, two unsupervised approaches that have been successful in machine learning based anomaly detection for audio are autoencoders and density estimation methods (see, for example, (4) for a more comprehensive list). However, the scientific community has also found that self-supervised techniques can perform well. It is not clear that this is ideal from a practical point of view. Meta-data can change or become unavailable – at the very least, it can be inconvenient to manage. For example, several teams successfully leveraged machine identification numbers for self-supervision (5–9). However, in practical settings machines can be clustered in various physical configurations so that even choosing an enumeration scheme is inconvenient. For these reasons, successful unsupervised approaches are generally the most desirable.

The solution we present here is unsupervised and falls under the density estimation category. The

\textsuperscript{1}jose.a.lopez@intel.com
\textsuperscript{2}georg.stemmer@intel.com
\textsuperscript{3}paulo.lopez.meyer@intel.com
key ingredient is a collection of density estimators that estimate the density of a subset of the frequency bins of a spectrogram, conditioned on the remaining bins. The density estimators make use of the recent technique called normalizing flows (NF) \((10–17)\). Therefore, we call the model NF-CDEE, because it uses normalizing flows, and it consists of a conditional density estimator ensemble.

The main concept of the normalizing flow method is to exploit the change of (random) variables formula by learning invertible transformations to transform a reference, usually normal, distribution into a more complicated one. It is a powerful approach for density estimation, but it is not without numerical difficulties which can be encountered when computing the determinant of the Jacobian of the learned transformations. We observed that divergent training increased with dimensionality \((\text{the number of frequency bins})\) and with data noise. Clearly, the training difficulties increase when training multiple estimators simultaneously, since any of them can derail the training. Therefore, apart from the architecture, our additional contributions are the techniques to side-step training problems.

We are not the first to use NFs for anomalous sound detection \((18,19)\). However, to our knowledge, we are the first to successfully train an ensemble of NFs for this purpose. In \((19)\), the authors demonstrated the effectiveness of adaptive batch normalizations for adapting a trained network to a new domain, without retraining, using simulated anomalies generated from toy car sound data. The resulting NF performed similarly to a fine-tuned NF using much less computational effort.

The work in \((18)\) is more similar to ours; there, the authors also experimented with DCASE2020 data \((20,21)\) and used NFs. However, that work obtained improvements by training NFs in a self-supervised manner. They trained 1 NF for 1 machine using a loss that decreased for inputs from the same machine but increased for inputs from other machines of the same machine type. Thus, if one has data for 10 machines of the same type, one needs to train 10 models. Additionally, using their approach, the input spectrograms were arranged in overlapping frames, so the model needs to accommodate the artificially larger input. In contrast, our work is entirely unsupervised and trains 1 model per machine type. Moreover, the input dimension is the number of Mel bins, and each NF is a conditional density estimator. Finally, our ensembled conditional estimator architecture performs better across all machine categories. Therefore, we believe that for most practical anomaly detection usages the proposed NF-CDEE is a very promising model architecture.

In the following sections we describe our methodology, experiments, and results.

2. METHODOLOGY

The DCASE2020 Task 2 dataset consists of 10-second audio files that include the sound of a target machine and environmental noise \((2)\). There are six machine categories. ToyCar and ToyConveyor are from the ToyADMOS dataset \((20)\), while Valve, Pump, Fan, and Slider are from the MIMII dataset \((21)\). There are several machine IDs within each machine category. However, all the audio files are single-channel and use a 16 kHz sampling rate. We only use the data from the development dataset since the true labels are available \((22)\).

For our experiments, we rely on the PyTorch, nnAudio, and Pyro libraries for Python \((23–25)\). To compute the Mel spectrograms, we used nnAudio in the so-called “stand-alone” mode, i.e., without the trainable kernel option. However, our method does not require a Mel transformation. We do not expect this choice to change the performance significantly or the conclusions of this work.

Our intention is to explore how hyperparameters affect the model performance, and to convince the reader that NF-CDEE is sufficiently general to be worth exploring for their applications, rather than to find an optimal configuration. For all experiments, the models were trained for 50 epochs. For some models, like Fan, the best performance was obtained close to the end of training, which suggests that there may be more performant models past 50 epochs. On the other hand, for some machines like ToyCar, the peak performance was reached near 10 epochs. As in DCASE2020 Task 2, we use area-under-the-curve (AUC) and partial area-under-the-curve (pAUC) metrics to measure performance. (For pAUC, the max false positive rate was set to 0.1.) We use the sum of AUC and pAUC to compare models using a single score.
2.1 NF-CDEE

As previously stated, the key ingredient of this model is the use of a density estimator to fit the distribution of an \( n \)-bin segment of an input \( m \)-bin spectrogram conditioned on the remaining \( m-n \) bins. Clearly, there are \( \binom{m}{n} \) ways of selecting \( n \) bins. Here, we simply take adjacent segments with or without overlap. The decision to use conditional density estimators instead of estimating the full joint distribution was born from the need to counteract training instability due to dimensionality. Figure 1 shows a diagram of the model.

For preprocessing, we standardize along the frequency axis by subtracting the mean and dividing by the standard deviation of each bin. This is conveniently implemented using a batch-norm layer with the trainable parameters disabled. After, we include a mean operation (along the time dimension) to counteract instability due to noisy data. While we found it necessary to include this operation for training with DCASE2020 or DCASE2021 data, it is not clear that the operation is always required or useful (26).

To summarize, each estimator outputs the probability \( p(s_{\mathcal{A}_i} | s_{\mathcal{A}^c_i}) \), where \( s \) is a vector of dimension equal to the number of Mel bins \( m \) that is indexed by the set \( \mathcal{I} = \{1, \ldots, m\} \). \( \mathcal{A} \) is an \( n \)-element subset of \( \mathcal{I} \), and \( \mathcal{A}^c \) is its complement \( \mathcal{I} - \mathcal{A} \). We define the likelihood of the normal state as:

\[
p(\text{normal}) = \prod_i p(s_{\mathcal{A}_i} | s_{\mathcal{A}^c_i})
\]

where \( i \in [1, \ldots, k] \) and \( k \) is the number of estimators in the ensemble. Here, we used \( A_1 = \{1, \ldots, n\}, A_2 = \{n - o + 1, \ldots, 2n - o\}, \ldots, A_k = \{m - n + 1, \ldots, m\} \) where \( o \) is the number of overlapping bins.

To train the model, we minimize the negative logarithm of \( p(\text{normal}) \). Therefore, the output of NF-CDEE is the sum of the individual negative log-likelihoods. For the optimizer, we used the AdamW optimizer with a learning rate of \( 1 \times 10^{-3} \) and weight decay set to \( 1 \times 10^{-4} \). Gradient clipping was also used to limit the norm of the gradients to 0.8. The weight decay and gradient clipping improve training stability. During training, the spectrograms were sampled 192 frames at a time using frame windows as was done in (27), and the following spectrogram settings were fixed: \( f_{\text{min}} = 0 \), \( f_{\text{max}} = 8000 \), hop length=512, and power=1.0. Lastly, each normalizing flow used a single conditional spline with 16 count-bins and hidden layer dimension equal to \( 10n \).
3. EXPERIMENTS

We include Table 1 to put the performance values of NF-CDEE into perspective with the baseline and top scores of the DCASE 2020 competition. However, please note that the top DCASE2020 scores generally include models that are ensembled and some are self-supervised (28). Moreover, the AUC and pAUC scores generally come from different submissions. Thus, the final row of Table 1 may come from different submissions.

Table 1 - DCASE2020 Task 2 baseline and top scores

<table>
<thead>
<tr>
<th></th>
<th>ToyCar</th>
<th>ToyConveyor</th>
<th>Fan</th>
<th>Pump</th>
<th>Slider</th>
<th>Valve</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline AUC</td>
<td>0.788</td>
<td>0.725</td>
<td>0.658</td>
<td>0.729</td>
<td>0.848</td>
<td>0.663</td>
</tr>
<tr>
<td>baseline pAUC</td>
<td>0.676</td>
<td>0.604</td>
<td>0.525</td>
<td>0.600</td>
<td>0.665</td>
<td>0.510</td>
</tr>
<tr>
<td>baseline AUC+pAUC</td>
<td>1.464</td>
<td>1.330</td>
<td>1.183</td>
<td>1.33</td>
<td>1.513</td>
<td>1.173</td>
</tr>
<tr>
<td>top AUC</td>
<td>0.983</td>
<td>0.904</td>
<td>0.941</td>
<td>1.0</td>
<td>1.0</td>
<td>0.999</td>
</tr>
<tr>
<td>top pAUC</td>
<td>0.936</td>
<td>0.815</td>
<td>0.882</td>
<td>1.0</td>
<td>1.0</td>
<td>0.994</td>
</tr>
<tr>
<td>top AUC+pAUC</td>
<td>1.919</td>
<td>1.719</td>
<td>1.824</td>
<td>2.0</td>
<td>1.0</td>
<td>1.993</td>
</tr>
</tbody>
</table>

3.1 Performance vs. FFT Window Length

The FFT window length can have a significant effect on anomaly detection performance for different machine sounds. Therefore, we explore its effect first. In the first experiment we vary the FFT window length for batch sizes 32, 64, and 128. m, n, and o were fixed at 256, 32 and 0, respectively. Table 2 summarizes the best scores relative to FFT window length and batch size. Figure 2 shows the performance for batch size 32. It appears that the FFT window length has the strongest effect for impulsive machine sounds. The Figures for other batch sizes are similar and excluded for space considerations.

Table 2 - Best performance vs. FFT window length

<table>
<thead>
<tr>
<th></th>
<th>ToyCar</th>
<th>ToyConveyor</th>
<th>Fan</th>
<th>Pump</th>
<th>Slider</th>
<th>Valve</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch size</td>
<td>32</td>
<td>32</td>
<td>128</td>
<td>128</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>FFT window length</td>
<td>2048</td>
<td>16384</td>
<td>2048</td>
<td>2048</td>
<td>16384</td>
<td>16384</td>
</tr>
<tr>
<td>AUC</td>
<td>0.978</td>
<td>0.904</td>
<td>0.820</td>
<td>0.932</td>
<td>0.949</td>
<td>0.908</td>
</tr>
<tr>
<td>pAUC</td>
<td>0.909</td>
<td>0.728</td>
<td>0.652</td>
<td>0.823</td>
<td>0.865</td>
<td>0.783</td>
</tr>
<tr>
<td>AUC+pAUC</td>
<td>1.887</td>
<td>1.631</td>
<td>1.472</td>
<td>1.755</td>
<td>1.814</td>
<td>1.691</td>
</tr>
</tbody>
</table>

Figure 2 - Performance vs. FFT window length for batch size 32. For ToyCar, the performance remains fairly constant for window lengths 2048-16384 for all batch sizes. This is also the case for ToyConveyor. The performance of Fan and Pump decreases with FFT window length, while the opposite is true for Valve and Slider.
3.2 Sensitivity of Performance to Variations in Parameters \( (m, n, o) \)

In this experiment we vary the conditional density estimator segment size and overlap parameters for \( m \in \{256, 128, 64\} \). Figures 3 and 4 show that performance is generally less sensitive to overlap parameter, and more sensitive to the number of frequency bins and segment size. Thus, the spectrogram and spectrogram segment size parameters are more important. We exclude the plot for \( m = 128 \) for space considerations. Table 3 summarizes the best performance with respect to \( m, n, \) and \( o \).
### 3.3 Scaling the Model Size by Variation of \((m, n, o)\)

An important consideration in model deployment is the size of the model. Table 4 shows the model size for various \((m, n, o)\) with respect to number of parameters and memory footprint. The smallest model is for the \((m, n, o) = (64, 8, 0)\) configuration which uses about 415K parameters and 1.7 MB of memory. Table 5 shows the performance of the smallest model. The largest model corresponds to the \((m, n, o) = (256, 32, 16)\) and uses about 12.3M parameters and 49.4 MB of memory. Table 6 shows the performance of this model. Thus, the user can scale the model depending on their needs without sacrificing much performance.

For Tables 5 and 6, the batch and FFT sizes are the same as in Table 3.

<table>
<thead>
<tr>
<th>(m, n, o)</th>
<th>AUC</th>
<th>pAUC</th>
<th>AUC+pAUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>32, 32, 0</td>
<td>0.982</td>
<td>0.935</td>
<td>1.917</td>
</tr>
<tr>
<td>256, 16, 8</td>
<td>0.913</td>
<td>0.738</td>
<td>1.651</td>
</tr>
<tr>
<td>256, 16, 0</td>
<td>0.820</td>
<td>0.661</td>
<td>1.480</td>
</tr>
<tr>
<td>256, 32, 0</td>
<td>0.932</td>
<td>0.823</td>
<td>1.755</td>
</tr>
<tr>
<td>256, 16, 0</td>
<td>0.963</td>
<td>0.899</td>
<td>1.862</td>
</tr>
<tr>
<td>256, 32, 0</td>
<td>0.908</td>
<td>0.783</td>
<td>1.691</td>
</tr>
</tbody>
</table>

### Table 4 - Model size for various \((m, n, o)\)

<table>
<thead>
<tr>
<th>(n)</th>
<th>(o)</th>
<th>(m = 256)</th>
<th>(m = 128)</th>
<th>(m = 64)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>0</td>
<td>6.575</td>
<td>3.1236</td>
<td>1.521</td>
</tr>
<tr>
<td>32</td>
<td>16</td>
<td>12.328</td>
<td>5.466</td>
<td>2.281</td>
</tr>
<tr>
<td>32</td>
<td>8</td>
<td>9.040</td>
<td>3.905</td>
<td>2.281</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>3.626</td>
<td>1.649</td>
<td>0.784</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>7.025</td>
<td>3.092</td>
<td>1.371</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>2.151</td>
<td>0.912</td>
<td>0.415</td>
</tr>
</tbody>
</table>

### Table 5 - Performance of the smallest model

<table>
<thead>
<tr>
<th></th>
<th>ToyCar</th>
<th>ToyConveyor</th>
<th>Fan</th>
<th>Pump</th>
<th>Slider</th>
<th>Valve</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>0.964</td>
<td>0.885</td>
<td>0.761</td>
<td>0.885</td>
<td>0.928</td>
<td>0.871</td>
</tr>
<tr>
<td>pAUC</td>
<td>0.876</td>
<td>0.706</td>
<td>0.579</td>
<td>0.755</td>
<td>0.823</td>
<td>0.747</td>
</tr>
<tr>
<td>AUC+pAUC</td>
<td>1.840</td>
<td>1.588</td>
<td>1.340</td>
<td>1.640</td>
<td>1.751</td>
<td>1.620</td>
</tr>
</tbody>
</table>

### Table 6 - Performance of the largest model

<table>
<thead>
<tr>
<th></th>
<th>ToyCar</th>
<th>ToyConveyor</th>
<th>Fan</th>
<th>Pump</th>
<th>Slider</th>
<th>Valve</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>0.980</td>
<td>0.907</td>
<td>0.821</td>
<td>0.930</td>
<td>0.951</td>
<td>0.901</td>
</tr>
<tr>
<td>pAUC</td>
<td>0.918</td>
<td>0.740</td>
<td>0.654</td>
<td>0.815</td>
<td>0.875</td>
<td>0.763</td>
</tr>
<tr>
<td>AUC+pAUC</td>
<td>1.898</td>
<td>1.647</td>
<td>1.475</td>
<td>1.744</td>
<td>1.826</td>
<td>1.664</td>
</tr>
</tbody>
</table>
4. Conclusions and Future Work

We have presented NF-CDEE, described how to train it, and shown how it performs on a well-known dataset. As shown in Table 3, NF-CDEE leads to much better results than the baseline for all machine categories, which suggests it generalizes well to different tasks. Not all models known from the literature can do this for the DCASE2020 data because it originates from two independent datasets. In particular, in DCASE2020 there were submissions that performed well on the categories derived from the MIMII dataset, but struggled with the ToyADMOS categories (5). This includes the NF method from (18) which obtained near baseline level performance for ToyConveyor.

For NF-CDEE, Fan was the only category with less than 0.9 AUC. However, we know from top submissions on Table 1 that Valve performance can also be substantially improved. The first area we plan to investigate is the use of a mean operation to stabilize training. Surely the mean operation hinders the detection of time-sparse events. Does it also hinder detection for more stationary sounds? Looking in this direction may ultimately lead to new ways of implementing NFs.

REFERENCES


On choosing decision thresholds for anomalous sound detection in machine condition monitoring

Kevin WILKINGHOFF\textsuperscript{1}, Alessia CORNAGGIA-URRIGSHARDT\textsuperscript{2}

\textsuperscript{1}Fraunhofer FKIE, Germany, kevin.wilkinghoff@fkie.fraunhofer.de
\textsuperscript{2}Fraunhofer FKIE, Germany, alessia.cornaggia-urrigshardt@fkie.fraunhofer.de

ABSTRACT
Most anomalous sound detection (ASD) systems output a score for each audio sample presented to the system. Ideally, these anomaly scores differ for normal and anomalous samples such that one can determine whether a given sample is normal or anomalous by comparing the scores to predefined thresholds. However, determining these thresholds is non-trivial, especially when no anomalous samples are provided as training data. In this work, several methods for finding such decision thresholds are evaluated and compared to each other when acoustically monitoring the condition of machines in noisy environments. To this end, the state-of-the-art in ASD for machine condition monitoring will be reviewed first. Using a state-of-the-art ASD system, experimental evaluations are conducted on the DCASE 2020 ASD dataset to evaluate differently attained decision thresholds.

Keywords: anomalous sound detection, decision threshold, machine listening

1 INTRODUCTION
Anomaly detection \cite{1, 2} is the task of identifying samples substantially differing from normal samples that are frequently encountered. Collecting these anomalous samples is difficult because by definition anomalies occur only rarely and often are very costly to produce artificially. For example, when acoustically monitoring the condition of machines, creating anomalous samples of machine sounds translates to damaging potentially costly machines in very specific ways whereas recording fully functioning machines is much less costly. Furthermore, for many applications anomalies are very diverse making it practically impossible to sufficiently cover the space of anomalous samples by collecting as many as possible of them. Hence, in many cases anomaly detection takes place in a semi-supervised setting meaning that only normal samples are available for training a system.

Evaluating and comparing different systems for anomaly detection or sound detection should be independent of the choice of decision thresholds to allow for a more objective comparison \cite{1, 3}. Because of this, metrics such as the area under the receiver-operating characteristic curve (ROC-AUC) that do not utilize any decision threshold are usually used. However, when setting up a system for practical applications detection thresholds are still needed to distinguish between normal and anomalous test samples. But without access to anomalous training samples, it is impossible to determine decision thresholds by simply testing multiple values and picking the best-performing one. Hence, estimating these thresholds is highly non-trivial and requires sophisticated techniques.

Anomalous sound detection (ASD) for machine condition monitoring is highly promoted through the annual DCASE challenge \cite{4, 5, 6}. The baseline systems of the ASD tasks utilize the 90th percentile of a gamma distribution estimated from the histogram of the anomaly scores belonging to the normal training samples as a decision threshold. For all systems that participated in the DCASE challenge 2021 either no procedure for automatically estimating the decision thresholds is explicitly mentioned or the same (or a very similar) procedure as used by the baseline system is applied \cite{7, 8, 9, 10, 11, 12}. The most likely reason for a lack of focus on techniques for choosing decision thresholds is that the evaluation of the ASD systems is based on the AUC...
score and thus no decision thresholds are needed. This is done to have an objective comparison between the ASD performance of the systems and to prevent participants from cheating by utilizing anomalous samples of the development set for estimating thresholds. Furthermore, challenges are usually carried out only for research purposes without the goal of obtaining a fully functioning ASD system for a real-world application that would inevitably need a sophisticated technique for determining a decision threshold.

The goal of this work is to investigate multiple techniques for estimating decision thresholds in the context of anomalous sound detection for machine condition monitoring. For this purpose, first the state-of-the-art in ASD including a specific system for experimental evaluations is briefly reviewed. Second, multiple methods for estimating a decision threshold are presented. In experimental evaluations on the DCASE 2020 ASD dataset [4], these techniques are applied and compared to each other.

## 2 STATE-OF-THE-ART OF ANOMALOUS SOUND DETECTION

### 2.1 Review

First, the state-of-the-art in ASD will be reviewed. For this purpose, we will mainly follow [5]. There are two general state-of-the-art ASD paradigms for machine condition monitoring. Both rely on deep learning. The first one consists of using an autoencoder trained on normal data only. Here, it is assumed that the autoencoder can reconstruct normal data better than anomalous data due to deviations from the normal data used for training the model and thus the reconstruction error can be used as an anomaly score. Many different autoencoder architectures have been used for this purpose, e.g. class-conditioned autoencoders [13] or group masked autoencoders [14]. This approach of directly estimating the distribution of normal data is also more generally called inlier modeling (IM).

The second approach is to train a discriminative model to learn meaningful embeddings of the data. Here, it is assumed that the information needed to discriminate among predefined classes also captures the information needed to detect anomalous samples. This approach is called outlier exposure (OE) [15]. In machine condition monitoring, most models are trained to discriminate among different machine types or even finer subdivisions of the data such as different machine states or noise types [11]. To train an OE model, usually angular margin losses such as ArcFace [16] or AdaCos [17] are used [18, 19]. These losses ensure that not only inter-class similarity is minimized but simultaneously maximize intra-class similarity using an angular margin in combination with the cosine distance. Thus after training, embeddings belonging to normal samples of a specific class are concentrated around a learned mean embedding and anomalous samples are expected to have a larger angle than normal samples to this mean enabling the detection of anomalies.

Many state-of-the-art systems utilize both ASD paradigms. As noted in [5], there are two different ways to combine both approaches: a parallel and a sequential approach. The parallel approach is simply an ensemble of multiple OE and IM models [20, 21, 22] and the sequential approach consists of first applying an OE model as a feature extractor and then using an IM model for these features [23, 11]. Compared to a parallel approach, a sequential approach has the advantage that the system consists of fewer hyperparameters. However, when training a discriminative model to extract features some information needed to detect anomalous data may be lost if this information is not important for identifying the pre-defined classes.

### 2.2 Used system

For all experimental evaluations in this work, the system presented in [24] is used. The system is a sequential approach consisting of a neural network trained to extract discriminative audio embeddings from log-mel spectrograms using the sub-cluster AdaCos loss and a GMM for IM. The sub-cluster AdaCos loss is an extension of the AdaCos loss specifically designed for ASD. This means that the loss is also an angular margin loss with an adaptive scale parameter. The major difference to the standard AdaCos loss is that instead of learning a single class center for each class, the loss learns multiple sub-clusters for each class to learn more complex distributions. In this case, the classes are defined as specific machines recorded in noisy environments (see Section 4.1). More details about the sub-cluster AdaCos loss can be found in [24]. For all experiments, 32 sub-clusters for each class are used. When computing the log-mel spectrograms 128 mel bins, a window size of 1024 and a hop size of 512 are used.
Table 1. Modified ResNet architecture used for extracting discriminative embeddings.

<table>
<thead>
<tr>
<th>layer name</th>
<th>structure</th>
<th>output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>-</td>
<td>313 × 128</td>
</tr>
<tr>
<td>2D convolution</td>
<td>7 × 7, stride= 2</td>
<td>157 × 64 × 16</td>
</tr>
<tr>
<td>residual block</td>
<td>3 × 3 × 2, stride= 1</td>
<td>78 × 31 × 16</td>
</tr>
<tr>
<td>residual block</td>
<td>3 × 3 × 2, stride= 1</td>
<td>39 × 16 × 32</td>
</tr>
<tr>
<td>residual block</td>
<td>3 × 3 × 2, stride= 1</td>
<td>20 × 8 × 64</td>
</tr>
<tr>
<td>residual block</td>
<td>3 × 3 × 2, stride= 1</td>
<td>10 × 4 × 128</td>
</tr>
<tr>
<td>max pooling</td>
<td>10 × 1, stride= 1</td>
<td>4 × 128</td>
</tr>
<tr>
<td>flatten</td>
<td>-</td>
<td>512</td>
</tr>
<tr>
<td>dense (representation)</td>
<td>linear</td>
<td></td>
</tr>
<tr>
<td>sub-cluster AdaCos</td>
<td>32 sub-clusters</td>
<td>41</td>
</tr>
</tbody>
</table>

The neural network has a modified ResNet architecture [25] as shown in Table 1 and is implemented in Tensorflow [26]. The model is trained for 400 epochs with a batch size of 64 using Adam [27]. For data augmentation, mixup [28] with a uniformly sampled mixing coefficient is used to randomly generate additional data and prevent overfitting of the model. After training, the model is used to extract embeddings, which are length-normalized by projecting them to the unit sphere. The only exception is the machine type “ToyConveyor” for which the temporal means of the log-mel spectrograms are used instead of the embeddings because these representations yield a much better ASD performance for this machine type [24]. For each pre-defined class i.e. for each specific machine of a given machine type, the distributions of the resulting embeddings are then estimated with Gaussian mixture models (GMMs) with 32 Gaussian components and a full covariance matrix, which is regularized by adding 0.001 to the diagonal as implemented in scikit-learn [29]. To obtain anomaly scores, the corresponding log-likelihood values of the GMMs are used.

3 FINDING DECISION THRESHOLDS

There are many different approaches for finding decision thresholds [30, 31]. For arbitrary anomaly scores obtained with supervised classifiers potentially being biased, these anomaly scores can be calibrated by converting them into (pseudo-)probabilities for which thresholds can be determined [32, 33]. Usually, a probability of 0.5 is used as a threshold for these calibrated scores. Since the anomaly scores of the used system already are log-likelihood values, a mapping to probabilities is not necessary. For multivariate data or scores, a threshold can be determined by using the empirical distribution function of the squared Mahalanobis distance and using a critical value such as a small quantile of the chi-squared distribution, which is the theoretical distribution function, as a threshold. [34]. An extension of this approach uses an adaptive threshold [35]. However, anomaly scores are usually univariate and thus multivariate approaches are not suitable. When continuously monitoring data streams for anomalies, these data streams themselves consist of sequential samples of which most samples are normal and only a few are anomalous. Therefore, thresholds can utilize previously encountered samples and need to adapt to changes occurring in the data stream [36, 37, 38]. This is very different from the ASD setting investigated in this work where individual recordings are either entirely normal or anomalous.

Now, several techniques for automatically estimating decision thresholds using only scores obtained with normal data will be reviewed. The general idea of all methods is to estimate a threshold that separates the extreme values of the training scores from the rest. Most of these methods are based on the assumption that the considered data, i.e. the anomaly scores, follow some specific distribution, typically a normal distribution. As this is not true for anomaly scores in general, the considered methods may work only to some extent.
3.1 Gamma distribution percentile (GP)
The strategy used by the baseline systems of the DCASE challenge [5, 6] is to fit a gamma distribution to the scores obtained with the normal training samples and use the inverse of the 90th percentile of the cumulative distribution function as the decision threshold. Test scores larger than this threshold are marked as anomalous; otherwise they are considered normal.

3.2 Histogram percentile (HP)
One can also directly use the histogram of the scores without fitting a distribution first. Note that this silently assumes a uniform distribution. We used the 90th percentile of the histogram of the scores as the decision threshold as done in [11].

3.3 Standard Deviation (SD)
One of the most commonly used approaches is to fit a normal distribution to the scores. All values exceeding the range \( \mu \pm \alpha \sigma \) are marked as anomalous, where \( \mu \) and \( \sigma \) are the mean and standard deviation of the normal scores. Note, that technically this implies the usage of two thresholds. Since the anomaly scores in this work consist of negative log-likelihoods and thus scores belonging to normal and anomalous samples are assumed to be linearly separable, only the upper threshold is used. To have a consistent evaluation with the previous two approaches, we used \( \alpha = 1.28 \), which approximately corresponds to the 90th percentile.

3.4 Median Absolute Deviation (MAD)
Following the assumption that the median is more robust against outliers than the mean, decision thresholds may be obtained by \( \tilde{x} \pm \alpha \cdot \text{MAD} \), where MAD is given by MAD = \( \beta \cdot \text{median}(x - \tilde{x}) \) and \( \tilde{x} \) is the median value of the score values \( x \). [31] proposes \( \alpha = 3 \) and \( \beta = 1.4826 \) following [39], [30] uses \( \alpha = 2 \), which is also the value we used.

3.5 Interquartile Range (IQR)
This approach is based on the division of the score values \( x \) into subsets by setting Q1 and Q3 such that \( x \geq Q1 \) for 75% and \( x \geq Q3 \) for 25% of \( x \). Then IQR = Q3 – Q1. Values outside the range \( Q1 - \alpha \cdot \text{IQR} \) and \( Q3 + \alpha \cdot \text{IQR} \) are considered anomalous. Typically, \( \alpha = 1.5 \) is assumed [39]. We used, \( \alpha = 0.5 \) as this significantly improved the performance. This approach is also known as boxplot [30].

3.6 One-class support vector machine (OCSVM)
To estimate the support of a distribution, a one-class support vector machine [40], which learns to discriminate between regions of high and low density using a hypersphere in high-dimensional space, can also be used. We used the implementation of scikit-learn [29] with a linear kernel. For the hyperparameter \( \nu \), we used a value of 0.1 i.e. 10% of the normal training scores are treated as anomalous.

3.7 Generalized Extreme Studentized Deviate (GESD)
GESD [41] is an iterative approach based on the Grubbs’s test (GRUBBS) [42]. This statistical test, named after Grubbs, assumes a normal distribution and is calculated on the so-called Grubbs statistic

\[
G = \frac{\max(x) - \mu}{\sigma}
\]  

(1)

with mean \( \mu \) and standard deviation \( \sigma \). \( G \) is evaluated against a critical value of the student’s \( t \)-distribution with a significance level \( \alpha \), set to 0.05 as default, and data size \( N \):

\[
G > N - 1 \sqrt{N} \left[ \frac{t^2_{\alpha/(2N),N-2}}{N - 2 + t^2_{\alpha/(2N),N-2}} \right].
\]  

(2)

GRUBBS only tests for a single anomalous sample. For GESD, GRUBBS is thus repeated iteratively until no further anomalies are detected.
3.8 Clever Standard Deviation (cleverSD)
CleverSD [43] is another iterative approach. The idea is to repeatedly eliminate a single sample with the highest score from the training scores in case it is found to be anomalous by applying SD ($\alpha = 2$). This is done until no additional anomaly is found. We used the last score removed by this approach as the decision threshold.

3.9 Two-stage Thresholding (-x2)
Generalizing cleverSD, [31] suggests yet another iterative anomaly detection method called multi-stage thresholding. The main idea is to simply apply a non-iterative method multiple times to remove anomalies from the training scores. The difference to cleverSD is, that not only one anomaly but all anomalies detected are removed in each iteration. Experiments have shown that two stages are sufficient and thus we only used two iterations. When applying this technique to non-iterative approaches, we use the name of the method with the suffix -x2 to denote its two-stage version.

4 EXPERIMENTS

4.1 Dataset
For all experiments in this work, the DCASE 2020 ASD dataset [4] has been used. It consists of recordings from six different machine types, namely “ToyCar” and “ToyConveyor” from ToyAdmos [44], and “fan”, “pump”, “slider” and “valve” from MIMII [45]. Each recording contains a specific machine sound as well as factory noise and has a length of 10 seconds with a sampling rate of 16 kHz. There are six to seven different machine ids per machine type that correspond to a specific machine of that type and a total of 42 machine ids. These machine ids are used as classes when training the discriminative model described in Section 2.2.

The dataset is divided into a training set, a development set and an evaluation set. The training set consists of approximately 1000 normal sounds for each machine id. The development set consists of 100 to 200 normal sounds and 100 to 200 anomalous sounds for each of one half of the machine ids belonging to each machine type. The evaluation set consists of approximately 400 recordings containing both normal and anomalous sounds for each of the other half of machine ids.

4.2 Comparison of the decision methods
The scores obtained with the normal training data are used to estimate thresholds for ASD scores for each machine type and each machine id individually. These thresholds or the models representing them are evaluated both on the development and the evaluation set by applying them to the corresponding test sets containing a mixture of normal and anomalous samples. F1 scores are then calculated for each machine id individually and the mean is calculated for each machine type. For comparison, the performance obtained with a single optimal threshold is evaluated as an additional method denoted by optimum. These optimal thresholds are calculated by simply trying multiple values as decision thresholds, calculating the corresponding F1-scores and denoting the highest achieved F1-score for each machine type. To have a more robust estimation of all results, the ASD system is trained five times and the whole evaluation procedure is repeated five times for each method.

Then, the mean of the five resulting F1 scores is calculated as the final performance. The final results for development and evaluation set are listed in Table 2 and Table 3, respectively. The best method for estimating decision thresholds per machine type is underlined. Additionally, average F1 scores computed over all machines types are provided. The results show that different threshold detection methods yield varying performances for distinct machine types.

To compare the different methods while reducing the influence of the difference in performance for individual machine types, we used the ratio between F1-score of a method and the best possible F1-score obtained with a single threshold, i.e. the results obtained with optimum, instead of the F1-scores themselves. Therefore, these values show how close an estimated threshold is to the optimal threshold and allows a better comparison between the methods regardless of the actual ASD performance of the used system for different machine types.

The results are depicted in Figure 1. The following observations can be made. First, most approaches result in a very similar ASD performance. The only exceptions are GESD, which performs worse on both the development and evaluation set, and GP/GPx2, which performs slightly worse on the development set, than
Table 2. Mean of F1 scores among all machine ids belonging to single machine types obtained with five independent trials on the development dataset. Highest F1 score among different methods for each machine type is underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>fan</th>
<th>pump</th>
<th>slider</th>
<th>ToyCar</th>
<th>ToyConveyor</th>
<th>valve</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>0.83204</td>
<td>0.82253</td>
<td>0.91109</td>
<td>0.80763</td>
<td>0.61818</td>
<td>0.87869</td>
<td>0.81169</td>
</tr>
<tr>
<td>HP</td>
<td>0.75288</td>
<td>0.83735</td>
<td>0.95056</td>
<td>0.84823</td>
<td>0.67279</td>
<td>0.89828</td>
<td>0.82668</td>
</tr>
<tr>
<td>SD</td>
<td>0.75055</td>
<td>0.83927</td>
<td>0.95147</td>
<td>0.85007</td>
<td>0.67236</td>
<td>0.89985</td>
<td>0.82726</td>
</tr>
<tr>
<td>MAD</td>
<td>0.76191</td>
<td>0.84408</td>
<td>0.95138</td>
<td>0.85412</td>
<td>0.66953</td>
<td>0.90171</td>
<td>0.83046</td>
</tr>
<tr>
<td>IQR</td>
<td>0.78953</td>
<td>0.84310</td>
<td>0.93899</td>
<td>0.83723</td>
<td>0.67722</td>
<td>0.89694</td>
<td>0.83050</td>
</tr>
<tr>
<td>OCSVM</td>
<td>0.75288</td>
<td>0.83735</td>
<td>0.95049</td>
<td>0.84907</td>
<td>0.67265</td>
<td>0.89794</td>
<td>0.82673</td>
</tr>
<tr>
<td>GESD</td>
<td>0.66239</td>
<td>0.81641</td>
<td>0.96752</td>
<td>0.86416</td>
<td>0.59373</td>
<td>0.86991</td>
<td>0.79569</td>
</tr>
<tr>
<td>cleverSD</td>
<td>0.83559</td>
<td>0.83511</td>
<td>0.91850</td>
<td>0.82063</td>
<td>0.66289</td>
<td>0.88529</td>
<td>0.82633</td>
</tr>
<tr>
<td>GPx2</td>
<td>0.86353</td>
<td>0.80954</td>
<td>0.88840</td>
<td>0.77259</td>
<td>0.61148</td>
<td>0.86485</td>
<td>0.80173</td>
</tr>
<tr>
<td>HPx2</td>
<td>0.81633</td>
<td>0.83612</td>
<td>0.92617</td>
<td>0.81426</td>
<td>0.66316</td>
<td>0.88786</td>
<td>0.82398</td>
</tr>
<tr>
<td>SDx2</td>
<td>0.82656</td>
<td>0.83333</td>
<td>0.91861</td>
<td>0.80603</td>
<td>0.65868</td>
<td>0.88201</td>
<td>0.82087</td>
</tr>
<tr>
<td>MADx2</td>
<td>0.79681</td>
<td>0.84887</td>
<td>0.94039</td>
<td>0.84690</td>
<td>0.67556</td>
<td>0.89848</td>
<td>0.83450</td>
</tr>
<tr>
<td>IQRx2</td>
<td>0.83306</td>
<td>0.83445</td>
<td>0.91598</td>
<td>0.80581</td>
<td>0.65856</td>
<td>0.88336</td>
<td>0.82187</td>
</tr>
<tr>
<td>OCSVMx2</td>
<td>0.81573</td>
<td>0.83616</td>
<td>0.92650</td>
<td>0.81474</td>
<td>0.66307</td>
<td>0.88739</td>
<td>0.82393</td>
</tr>
<tr>
<td>optimum</td>
<td>0.92574</td>
<td>0.88895</td>
<td>0.98461</td>
<td>0.89175</td>
<td>0.68857</td>
<td>0.91896</td>
<td>0.88310</td>
</tr>
</tbody>
</table>

Figure 1. Comparison of decision methods based on the mean of the normalized F1 scores taken over all machine types.

all other methods. Second, the two-stage versions of the approaches, yield about the same performance on the development set and a slightly better performance on the evaluation set. Furthermore, the iterative approach cleverSD has approximately the same F1 score as the two-stage approaches. In conclusion, these experiments indicate that one should use a two-stage approach (or cleverSD) when estimating decision thresholds for ASD and the particular choice for the underlying one-stage method is not that important.
Table 3. Mean of F1 scores among all machine ids belonging to single machine types obtained with five independent trials on the evaluation dataset. Highest F1 score among different methods for each machine type is underlined.

<table>
<thead>
<tr>
<th></th>
<th>fan</th>
<th>pump</th>
<th>slider</th>
<th>ToyCar</th>
<th>ToyConveyor</th>
<th>valve</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>0.89956</td>
<td>0.86065</td>
<td>0.88863</td>
<td>0.60121</td>
<td>0.63508</td>
<td>0.67127</td>
<td>0.75940</td>
</tr>
<tr>
<td>HP</td>
<td>0.93845</td>
<td>0.89173</td>
<td>0.92304</td>
<td>0.58433</td>
<td>0.59703</td>
<td>0.68612</td>
<td>0.76512</td>
</tr>
<tr>
<td>SD</td>
<td>0.93675</td>
<td>0.89318</td>
<td>0.92132</td>
<td>0.57726</td>
<td>0.59974</td>
<td>0.65573</td>
<td>0.76399</td>
</tr>
<tr>
<td>MAD</td>
<td>0.93878</td>
<td>0.88953</td>
<td>0.91896</td>
<td>0.57180</td>
<td>0.59806</td>
<td>0.65103</td>
<td>0.76136</td>
</tr>
<tr>
<td>IQR</td>
<td>0.94128</td>
<td>0.89359</td>
<td>0.92027</td>
<td>0.59976</td>
<td>0.61941</td>
<td>0.66482</td>
<td>0.77319</td>
</tr>
<tr>
<td>OCSVM</td>
<td>0.93832</td>
<td>0.89185</td>
<td>0.92321</td>
<td>0.58277</td>
<td>0.59607</td>
<td>0.65587</td>
<td>0.76468</td>
</tr>
<tr>
<td>GESD</td>
<td>0.90994</td>
<td>0.88218</td>
<td>0.89995</td>
<td>0.51314</td>
<td>0.49565</td>
<td>0.61057</td>
<td>0.71857</td>
</tr>
<tr>
<td>cleverSD</td>
<td>0.94124</td>
<td>0.87428</td>
<td>0.90940</td>
<td>0.61826</td>
<td>0.64311</td>
<td>0.67869</td>
<td>0.77749</td>
</tr>
<tr>
<td>GPx2</td>
<td>0.90652</td>
<td>0.85206</td>
<td>0.88491</td>
<td>0.63815</td>
<td>0.66252</td>
<td>0.69189</td>
<td>0.77267</td>
</tr>
<tr>
<td>HPx2</td>
<td>0.94287</td>
<td>0.88472</td>
<td>0.91462</td>
<td>0.61838</td>
<td>0.63550</td>
<td>0.67583</td>
<td>0.77865</td>
</tr>
<tr>
<td>SDx2</td>
<td>0.94289</td>
<td>0.87921</td>
<td>0.91107</td>
<td>0.62399</td>
<td>0.64169</td>
<td>0.68273</td>
<td>0.78026</td>
</tr>
<tr>
<td>MADx2</td>
<td>0.94149</td>
<td>0.89154</td>
<td>0.91883</td>
<td>0.59180</td>
<td>0.61854</td>
<td>0.66447</td>
<td>0.77111</td>
</tr>
<tr>
<td>IQRx2</td>
<td>0.94292</td>
<td>0.87495</td>
<td>0.90981</td>
<td>0.62575</td>
<td>0.64314</td>
<td>0.68402</td>
<td>0.78010</td>
</tr>
<tr>
<td>OCSVMx2</td>
<td>0.94283</td>
<td>0.88521</td>
<td>0.91462</td>
<td>0.61806</td>
<td>0.63553</td>
<td>0.67594</td>
<td>0.77870</td>
</tr>
</tbody>
</table>

Figure 2. Mean of optimal F1 scores among machine types obtained with varying percentage of data samples not used for training the ASD system on the development set.

4.3 Dividing normal samples in disjoint sets for training ASD system and estimating decision threshold

To avoid a degraded ASD performance due to overfitting resulting from using the normal samples for estimating the decision threshold and training the ASD system, a commonly applied strategy is to only use a part of the normal samples for training the model and use the remaining samples for extracting more realistic scores. By using this strategy, the training scores and test scores are more similar and thus the decision threshold is expected to be more accurate. However, it is clear that using less data for training the ASD system also degrades the ASD performance since less information is incorporated into the model. In the following experiments, we investigate whether this strategy actually improves the ASD performance.

First, we evaluated the ASD performance obtained with a single optimal decision threshold for a varying number of data samples used for training the ASD system. The resulting F1 scores can be found in Figure 2. As
expected, the F1 scores decrease when using less data for training. However, the degradation in performance is much less severe than anticipated and is not noticeable before using less than 40% of normal training samples. Even when using only 5% of normal training samples, the F1 scores are only slightly worse than when using all samples. The most likely reason for this is that, ignoring the background noise, the variability of sounds emitted by machines is relatively low and thus their acoustic behavior can be captured with only a few recordings. Since this opens the possibility to train an ASD system for machine condition monitoring with much fewer computational and data resources, this observation is interesting on its own.

![Figure 3](image3.png)

**Figure 3.** Mean of normalized F1 scores among machine types obtained with different methods for estimating decision thresholds and varying percentage of data samples not used for training the ASD system on the development set.

![Figure 4](image4.png)

**Figure 4.** Mean of normalized F1 scores among machine types obtained with different methods for estimating decision thresholds and varying percentage of data samples not used for training the ASD system on the evaluation set.

Next, we evaluated the ASD performance obtained with different methods for estimating a decision threshold for a varying number of data samples used for training the ASD system. The results for the development set and evaluation set can be found in Figure 3 and Figure 4, respectively. It can be seen that using fewer samples for training the system and using these samples to estimate more realistic anomaly scores for estimating the decision threshold does not significantly improve the ASD performance. Moreover, since the absolute F1 score is actually slightly decreasing (see Figure 2) the performance is actually worse. When comparing individual methods, one can see that in general the gaps in performance get wider the less data is used for training
the ASD system. Once more, iterative approaches, namely SDx2, IQRx2, OCSVMx2 and cleverSD, perform best. Furthermore, their relative performance is relatively stable, making them a robust choice for estimating decision thresholds for ASD. Note that OCSVMx2 does not assume an underlying distribution but only linear separability of the anomaly scores. Hence, it appears to be likely that these methods, especially OCSVMx2, also work well in other settings with other ASD systems and different anomaly scores. One noticeable exception is GP, for which GPx2 the results are very noisy and much worse than every other method. Since this degraded performance is not visible to this extend when using all samples for training the ASD system, this indicates that in general the other iterative methods may be preferable.

5 CONCLUSIONS
In this work, multiple techniques for estimating decision thresholds have been reviewed and applied for detecting anomalous sounds in machine condition monitoring. In experiments conducted on the DCASE 2020 dataset, these techniques have been compared using the anomaly scores obtained with a state-of-the-art ASD system. For this particular experimental setup, the following observations have been made: First, most techniques for estimating a decision threshold perform equally well and yield approximately 90% to 95% of the F1 score obtained with an optimally tuned decision threshold. Hence, there is still a gap in performance but this gap is relatively small. Second, iterative approaches such as multi-stage thresholding [31] slightly improve the overall ASD performance and therefore are to be preferred over single-stage techniques. This is especially true when using less data for training the ASD system indicating that iterative approaches are more robust. Last but not least, holding back normal training samples (i.e. not using them for training the ASD system) for the sole purpose of obtaining more realistic anomaly scores from these samples and using the resulting scores when estimating a decision threshold does not improve the ASD performance and thus can be omitted.

Although this work is not and cannot be exhaustive in listing and comparing all methods for automatically estimating decision thresholds, it shall serve as an initial investigation on applying these techniques for practical ASD applications such as machine condition monitoring. For future work, further studies using other ASD systems for calculating the anomaly scores, other ASD datasets and additional techniques for estimating decision thresholds are to be carried out. In addition, it is planned to evaluate all mentioned methods for finding decision thresholds when dealing with domain shifts [5] and when generalizing models for multiple domains [6]. Furthermore, additional investigations on choosing decision thresholds can be led for open-set classification problems such as acoustic scene classification [46] or speaker recognition [47].

REFERENCES


Improvement of anomalous sound detection method considering the distribution of embedding

Ibuki Kuroyanagi¹; Tomoki Hayashi¹²; Kazuya Takeda¹; Tomoki Toda¹
¹Nagoya University, School of Informatics, Nagoya, Japan
²Human Dataware Lab. Co., Ltd., Nagoya, Japan

ABSTRACT
Anomalous sound detection systems detect unknown, atypical sounds using only normal sound data. This study improves the performance of anomalous sound detection systems. We focus on an outlier exposure approach, enabling powerful feature extraction by training decision boundaries between normal and pseudo-anomalous data. First, we develop a method that explicitly considers the distance between the respective class centroid of normal and pseudo-anomalous data in the embedding to separate normal and pseudo-anomalous data. Although this method is helpful for extracting effective features for anomalous sound detection, the detection performance is still limited, especially when 1) normal and pseudo-anomalous data are too similar or different, and 2) normal data distribution in the embedding space does not follow a normal distribution. To address these issues, we propose a novel outlier exposure approach based on a binary classification task of two types to deal with cases in which normal and pseudo-anomalous data are too similar or different. Furthermore, we propose cascading modules to more flexibly model embedding distribution in normal data. We evaluate our method on the DCASE 2021 Task 2 Challenge dataset and demonstrate that our proposed method outperforms AUC's conventional and top-ranked ensembled methods.

Keywords: Anomalous sound detection, Outlier exposure, Inlier modeling

1. INTRODUCTION
Anomalous sound detection (ASD) identifies whether sounds emitted from a target machine are normal or anomalous. However, ASD differs significantly from the general binary classification problem. In general, anomalous sounds rarely occur when factory equipment is operating normally. Furthermore, the types of anomalous sounds are extremely diverse. It is difficult to collect anomalous sounds in advance comprehensively. Therefore, it is necessary to detect unknown anomalous sounds using only normal sounds [1].

There are two main methods for ASD, inlier modeling (IM) and outlier exposure (OE). IM is a method that models the probability distribution of normal data and detects data that do not correspond to the model as anomalous data. IM methods such as autoencoders, k-nearest neighbors (KNN) [2], local outlier factor (LOF) [3], and Gaussian mixture models (GMM) [4] have been used. IM is robust in performance regardless of machine type, but it is difficult to extract effective features [5]. In contrast, OE is a method for learning the decision boundaries of normal data by classifying normal and pseudo-anomalous data. OE methods such as deep semi-supervised anomaly detection [6], and [7] have been used. OE is easy to extract effective features but is not robust in performance [5].

First, we focus on OE's powerful feature extraction to improve ASD performance. We develop a loss function called deep double centroids semi-supervised anomaly detection (DDCSAD) [8], [9] that explicitly considers the distance between the centroids of each class of normal and pseudo-anomalous data in the embedding used to separate normal and pseudo-anomalous data. However, this method has limited performance improvement because it assumes that normal data distribution follows a normal distribution. To solve this problem, we propose a new OE-based feature extractor training method based on two types of binary classification tasks. The method improves the performance of feature extractors by OE. To model normal data more flexibly, we also train anomalous detectors by IM based on the features obtained from the feature extractors and cascade them together. We evaluate the method using data from the source domain of the DCASE 2021 Task 2 Challenge evaluation set [10], [11]. The results show that the method outperforms the state-of-the-art performance of ensembling.
multiple models [12] in a single model.

2. OE-BASED METHOD

We develop a loss function that explicitly considers the distance between the centroids of each class of normal and pseudo-anomalous data to take advantage of OE’s powerful feature extractor [8], [9]. We call this loss function DDCASD. A schematic image of the embedding obtained by the loss function is shown in Figure 1. We assume that anomalous data is distributed outside of normal data, and pseudo-anomalous data is distributed further outside of anomalous data. Based on this assumption, we train normal data to be closer to the centroid of the normal class and further away from the centroid of the pseudo-anomalous class. At the same time, pseudo-anomalous data are trained to be closer to the centroid of the pseudo-anomalous class and further away from the centroid of the normal class. The metric learning loss function minimizes within-class variance and maximizes between-class variance. The final loss function is a multitask learning process of metric learning and binary classification of normal and pseudo-anomalous data. During inference, a weighted average of Euclidean distance from the embedding feature to the normal class's centroid and the binary classification's output probability is computed as the anomaly score.

We evaluate the performance of our method using the data from the DCASE 2020 Task 2 Challenge evaluation set [13]. Here, the datasets [14], [15] contain several machine types such as fan, pump, valve, etc. Each machine type has K product IDs such as ID 0, 1, 2, etc. We compare performance with several systems and found that the highest performance is achieved when learning multitasking binary classification and DDCASD. Since data that differs significantly from normal data moves away from the centroid of the normal class and closer to the centroid of the pseudo-anomalous class, the method is considered robust in performance when data with distributions that differ significantly from normal data. However, when using Euclidean distance, the method assumes that normal data distribution follows a normal distribution, which limits its performance improvement. In particular, the method degrades performance when 1) normal, and pseudo-anomalous data are too similar or different, or 2) the distribution of normal data does not follow a normal distribution. Furthermore, since a model is created for each product ID, performance variability increases, and model development and maintenance costs are complicated [9].

3. PROPOSED METHOD

To solve the DDCASD problems, we propose loss functions for training the feature extractor $f(\cdot)$ with a new OE based on two binary classification tasks. A schematic image of the embedding obtained by the loss functions is shown in Figure 2. The proposed loss functions explicitly deal with cases where normal and pseudo-anomalous data are too similar or different. The first loss function is product ID classification, where each class is a binary classification of which sound is emitted from the product ID of the target machine. The model is trained to discriminate between normal and anomalous data, even when too similar. The second loss function is machine type classification to identify whether the sound is emitted from the target machine or not by using the norm of the embedding, i.e., the distance from the origin in the embedding space. The model is trained to discriminate between normal and anomalous data, even when too different. These loss functions are adjusted by a hyperparameter $\lambda$. Furthermore, the
proposed method uses the obtained features to train an anomalous detector \( h(\cdot) \) by IM. By using IM, the proposed method more flexibly models the embedding distribution in normal data; GMM and LOF are used for IM. The final system is realized by cascading the feature extractor \( f(\cdot) \) with OE and the anomaly detector \( h(\cdot) \) with IM. During inference, if the input sound clip is longer than the one cut out during training, the sound clip is divided into the same segment lengths as used during training. The anomaly scores are aggregated using an aggregator \( A(\cdot) \) such as average and max pooling to obtain the final anomaly score.

Inspired by DDCSAD's training method of collecting pseudo-anomalous data at a single point, the proposed method focuses on the data distribution in the embedding. Binary classification of product IDs distributes anomalous sounds similar to normal sounds around each product ID. Binary classification of machine type by the norm of embedding distributes anomalous sounds with a distribution that differs significantly from normal sounds to the origin in the embedding space, i.e., the center of the hypersphere. The loss functions obtain an embedding suitable for anomalous detection by IM. The proposed method is also easy to develop and maintain because the model is created for each machine type, not for each product ID.

4. EXPERIMENTAL EVALUATION

4.1 Experimental Conditions

To evaluate the performance of the proposed method, we conducted experiments using the data from the DCASE 2021 Task 2 Challenge (MIMII Due [10], ToyADMO2 [11]). The training and evaluation data in the same domain were used. We used sound from seven machine types: fan, gearbox, pump, valve, slider, ToyCar, and ToyTrain. Each machine type has 6 product IDs. For each product ID, 1,000 normal sound samples were used as training data, while 100 normal and 100 anomalous sound samples were used for evaluation data. ID 0, 1, and 2 of the evaluation data were used as validation data to determine the batch size, learning rate, the hyperparameter \( \lambda \) for multitask learning, and the hyperparameter \( p \) of IM, which are the number of components for GMM and the number of neighbors for LOF. ID 3, 4, and 5 were used as test data. When training Encoder \( f(\cdot) \), 90 \% of the training data was randomly selected, and the remaining 10 \% was used for validation. Each recording is a single-channel, 10 sec. segment of audio sampled at 16 kHz.

Each machine’s sound amplitude was standardized during preprocessing to have a mean of 0 and a variance of 1. For each audio input sequence, we extracted a log-compressed Mel-spectrogram with a window size of 128 ms, a hop size of 16 ms, and 224 Mel-spaced frequency bins in the range of 50—7800 Hz, in 2.0 sec. These features were passed to encoder \( f(\cdot) \) using EfficientNet-B0 [16]. The encoder \( f(\cdot) \) applied global average pooling to the last convolutional layer and performs two non-linear transformations to obtain a 128-dimensional embedding. We used AdamW [17], OneCycleLR [18] for training. Learning rate, batch size, \( \lambda \), inlier model \( h(\cdot) \), and the inlier model's hyperparameter \( p \) are shown in Table 1. We used the model with the smallest loss of validation data of the training data after 300 epochs. We trained the inlier model \( h(\cdot) \) using the validation data of the training data for each product ID. We determined the hyperparameters of the inlier model \( h(\cdot) \) using the validation data of the evaluation data. The GMM used the negative log-likelihood as the anomaly score, while the LOF used the outlier score. The aggregator \( A(\cdot) \) was the mean of the anomaly scores above the median for the GMM and the mean of the entire anomaly scores for the LOF.

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>fan</th>
<th>gearbox</th>
<th>pump</th>
<th>valve</th>
<th>slider</th>
<th>ToyCar</th>
<th>ToyTrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h(\cdot) )</td>
<td>GMM</td>
<td>GMM</td>
<td>GMM</td>
<td>GMM</td>
<td>GMM</td>
<td>LOF</td>
<td>LOF</td>
</tr>
<tr>
<td>( p )</td>
<td>16</td>
<td>64</td>
<td>2</td>
<td>32</td>
<td>2</td>
<td>16</td>
<td>8</td>
</tr>
</tbody>
</table>

4.2 Results

Table 2 shows the experimental results. Lopez et al [12], and Morita et al [19], are the first- and second-ranked methods in the DCASE 2021 Task 2 Challenge, respectively. Table 2 shows that for "ALL / Har-mean," the proposed method outperforms all the previous studies. While Lopez et al. [12]
in Table 2 is the result of an ensemble of several different methods, the proposed method outperformed it with a single model. Morita et al. [19]’s method consisted of a feature extractor trained using only a product IDs classification and a KNN-based anomalous detector. The proposed method outperformed Morita et al. [19] by 7% on ToyCar and ToyTrain. We considered that the proposed method obtained more suitable embedding for ASD by adding a loss function for machine type classification using the norm of embedding. Focusing on $h()$, we found that the best performance was obtained when using GMM for MIMII Due and LOF for ToyADMS2, indicating that it is important to use $h()$ differently depending on the data set.

**Table 2:** Performance evaluation results. Values represent the harmonic mean of AUC [%] and pAUC (p = 0.1) [%] for each product ID. “All / Har-mean” column values represent the harmonic mean of AUC and pAUC over all machines and product IDs.

<table>
<thead>
<tr>
<th>Method</th>
<th>fan</th>
<th>Gearbox</th>
<th>pump</th>
<th>valve</th>
<th>slider</th>
<th>ToyCar</th>
<th>Toy Train</th>
<th>All / Har-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lopez et al. [12]</td>
<td>56.11</td>
<td>62.43</td>
<td>85.33</td>
<td>66.68</td>
<td>74.58</td>
<td>68.46</td>
<td>69.04</td>
<td>67.93</td>
</tr>
<tr>
<td>Morita et al. [19]</td>
<td>82.37</td>
<td>66.22</td>
<td>77.21</td>
<td>72.06</td>
<td>78.02</td>
<td>54.72</td>
<td>50.88</td>
<td>66.78</td>
</tr>
<tr>
<td>DDCSAD [8]</td>
<td>69.21</td>
<td>58.13</td>
<td>68.55</td>
<td>75.66</td>
<td>59.56</td>
<td>57.71</td>
<td>57.71</td>
<td>63.12</td>
</tr>
<tr>
<td>Proposed Method</td>
<td><strong>84.35</strong></td>
<td><strong>68.42</strong></td>
<td>71.60</td>
<td>65.03</td>
<td><strong>83.97</strong></td>
<td><strong>62.08</strong></td>
<td><strong>58.81</strong></td>
<td><strong>69.42</strong></td>
</tr>
</tbody>
</table>

5. CONCLUSION

In this paper, we focused on the strong feature extraction of OEs to improve their anomalous detection performance. First, we developed a loss function called DDCSAD that explicitly considers the distance between the respective class centroids of normal and pseudo-anomalous data. However, the loss function had limited performance improvement because it assumed that normal data distribution follows a normal distribution. We proposed a new OE-based feature extractor training method to solve the problem based on two binary classification tasks. Furthermore, we trained an anomalous detector by IM based on the features obtained from the feature extractor and cascaded them. We evaluated the method using data from the source domain of the DCASE 2021 Task 2 Challenge evaluation set. The results showed that the method outperformed the state-of-the-art performance of ensembling multiple models in a single model. More detailed information on the proposed method can be found in [20], and the results of applying the proposed method to the DCASE 2022 Task 2 Challenge [21] can be found in [22]. We will study the method for training data and data from different domains.

ACKNOWLEDGEMENTS

This paper was partly supported by a project, JPNP20006, commissioned by NEDO.

REFERENCES


Abnormal sound detection of small actuator for vehicle using impulsive vibration signal

Hyun-Soo BYUN¹; Sung-Hwan SHIN¹; Hee-Tae JEONG¹; Seung-Eun LEE¹
¹ Kookmin University, Korea

ABSTRACT
The small actuator is used to operate various devices in automotive, and if it is defective due to an error in quality management, abnormal sound may occur during the operation. In this study, a method for detecting abnormal sound of a small actuator for vehicles is proposed. To this end, first, noise and vibration signals of the electric parking brake (EPB) actuators were measured and the noise of each sample were evaluated through subjective hearing tests by some experts who are in charge of quality control at the production site. Second, vibration signal was analyzed by applying several kinds of signal processing methods to classify its pattern. Finally, it was investigated that the abnormality of the actuator could be detected using the impulsive feature of the vibration signal. This result will be used for automated inspection on the production line to improve the quality management of the actuator.

Keywords: Abnormal sound, Degree of modulation, Modified IMPULSE

1. INTRODUCTION
The development of vehicle technology has propelled the progress of NVH (noise, vibration, and harshness) technology. The goal set previously was to reduce vehicle vibration and noise levels; however, researchers are currently focusing on sound quality while considering the emotional satisfaction of consumers (1).

Many DC motors and small actuators are used (exceeding 200 units) in various vehicle devices (2). As such, the abnormal sound generated by numerous DC motors and small actuators reduces the sound quality of the vehicle, and a solution is required accordingly. In a production line, inspectors listen to the operating sounds of actuators to arrive at pass/fail judgments and use the sound pressure levels (SPLs) of the noise signals as an objective index. However, the inspectors' classification includes a subjective component, and noise signals are vulnerable to disturbances (conveyor noise, line operation noise, etc.), which occur frequently during the manufacturing process. Therefore, the reliability of the pass/fail judgments made using the conventional method is questionable.

Basic methods for the noise analysis of actuators have been presented previously (3, 4, 5). However, these methods are unsuitable for abnormal sound evaluation of actuators because it is difficult to quantify the modulation noise and identify the cause of an abnormality. Therefore, we propose an abnormal sound detection method for small actuators used in vehicles based on degree of modulation (MOD) analysis and Modified IMPULSE analysis. To this end, we measure the noise, vibration, and current signals of an electric parking brake (EPB) actuator and conducted subjective listening tests using the magnitude estimation method (MEM) to set abnormal sound discrimination criteria. After determining the noise characteristics of the actuator, we perform abnormal sound analysis. The results of this analysis indicate that abnormal sound detection in case of the target actuator is possible through degree of modulation (MOD) analysis and Modified IMPULSE analysis by using the vibration signal.

¹ soulshin@kookmin.ac.kr
2. SIGNAL MEASUREMENT AND EVALUATION METHOD

The signal measurement of an electric parking brake (EPB) actuator performed in this study is illustrated in Fig. 1, where the noise, vibration, and current signals inside the inspection booth of the production line are measured. The noise signal was measured at 1m vertical from the actuator, the vibration signal was measured by attaching it to the gear housing of the actuator and the current signal was measured by applying the current of the DC power supply for driving the actuator. A servo motor was used to load the actuator. Through this, a situation in which the disc and the brake pad are fastened was realized.

Among the subjective listening test methods, the magnitude estimation method (6), which is suitable for numerically evaluating an evaluator's perceived level of abnormal sound, was used to set the criteria for detecting abnormal actuator sound. The magnitude estimation method is an evaluation method that evaluates the noise level of the gear part and the motor part by playing the sound sources in random order when multiple evaluation sound sources are specified. The evaluation scale is a 5-point scale, and it is set as shown in Fig. 2, such that evaluators can rate noises as very normal (-2 points) to very abnormal (+2 points). In the subjective listening test, six developers and inspectors who can clearly distinguish abnormal sounds of electric parking brake (EPB) actuators with normal hearing assessed the evaluation sounds (7).

![Figure 1 - Set-up for data acquisition](image1.png)

![Figure 2 - Evaluation scale used in subjective listening test](image2.png)
3. NOISE CHARACTERISTICS OF THE ACTUATOR

3.1 Cause of abnormal sound

The abnormal sound from an actuator is primarily caused by gear or motor defects. Gear part defects are caused by local damage to gear teeth or stabbing due to foreign substances. Motor part defects are caused by poor motor alignment, imbalance, or geometric tolerance of the commutator. If there is a defect in an actuator, the relative position of the corresponding portion is time-varying during operation, which results in amplitude modulation (8). This increases signal complexity, and because human hearing is sensitive to modulation noise, the signal is perceived as an abnormal sound.

3.2 Vibration signal for abnormal sound evaluation

Because frequent disturbances (conveyor noise, line operation noise, etc.) occur inside the production line of electric parking brake (EPB) actuators, it is necessary to check whether the signal to noise ratio (SNR) of a measured signal is secured before abnormal sound analysis. Fig. 5 shows the noise and vibration signal analysis results obtained under the actuator ON/OFF operating conditions. In a noise signal, background noise and actuator operating noise levels are similar in the audio frequency range (20 Hz–20 kHz). By contrast, the vibration signal presents significantly lower background vibration levels than the actuator operating vibration levels at frequencies higher than 500 Hz. This means that the vibration signal is robust against disturbances.

In general, the vibration signal contains abundant information regarding the machine state. Even so, coherence analysis was performed to determine whether the vibration signal can be used by replacing the noise signal of the target actuator. Coherence is measure of the linearity between two different signals in the frequency domain. If linearity is high, coherence is close to 100%; otherwise, the value of coherence is close to 0. Fig. 3 shows the result of our coherence analysis of the vibration signal based on the noise signal. A high linearity of more than 70% is found for the main noise components of the motor and gear parts.

Therefore, in Chapter 4, actuator abnormal sound analysis was performed in the band beyond 500 Hz using the vibration signal.

![Figure 3 - Comparison of the effects of disturbance (left) noise signal vs. (right) vibration signal](image)

4. ABNORMAL SOUND ANALYSIS OF THE ACTUATOR

Although the target was a vibration signal, a real-time band filter was applied to the measured noise signal to select the frequency band where abnormal sound occurred primarily, and the main band was selected after listening to the abnormal sound. Therefore abnormal sound analysis of the actuator was performed by applying a bandpass filter to the vibration signal based on the results of the subjective listening test.
4.1 Degree of modulation analysis

In the chapter 3, it was confirmed that abnormal sound is related to the modulation noise. Accordingly, degree of modulation (MOD) analysis was performed to examine the change in the modulation level and quantify it. The degree of modulation (MOD) (9, 10) is an analysis method for expressing a spectrum of the modulation frequency by performing the fast fourier transform on an original signal after it is subjected to a Hilbert transform. Fig. 4 shows the results of degree of modulation (MOD) analysis of the abnormal and normal samples. The abnormal sound components generated due to defects in the gear and motor parts were classified using the motor RPM and gear reduction ratio, and the abnormal sound levels could then be quantified. Therefore, the degree of modulation (MOD) is suitable as an abnormal sound discrimination factor of the target actuator.

![Figure 4 – Degree of modulation analysis of the abnormal (upper) and normal (lower) samples](image)

4.2 Determination of abnormal sound through degree of modulation

To apply the abnormal sound detection factor described in Section 4.1 to the pass/fail decision of the sample, it is necessary to systematize the discrimination. Therefore, a process for detecting abnormal actuator sounds based on the degree of modulation (MOD) was developed, as shown in Fig. 5.

Abnormal sound detection of the gear part was largely divided into two ways. The first is the occurrence of a high degree of modulation (MOD) for the corresponding modulation frequency component and its harmonic components due to major defects in specific gears. The second is the occurrence of a low degree of modulation (MOD) due to minor defects in various gears or foreign objects. In the first case, the abnormal sound detection criterion is set high because only specific components occur significantly. However, in the second case, the criterion is set low because various components occur with weak levels.

Fig. 6 presents examples of typical abnormal and normal samples classified by applying the abnormal sound detection process to a random sample. The results indicate that it is possible to make pass/fail decision for samples by using the abnormal sound detection process based on the degree of modulation (MOD), and the magnitude and location of defects can be identified.

![Figure 5 – Abnormal sound determination flowchart for classifying samples as normal or abnormal](image)
Figure 6 – Classification of normal and abnormal samples after applying abnormal sound determination flowchart

4.3 Modified IMPULSE analysis

The degree of modulation (MOD) analysis of the abnormal sample confirmed that a continuous shock signal pattern appeared in the time signal because of a defect in the gear part or motor part. Fig. 7 shows the time signal of representative abnormal and normal samples, and shows gear abnormal samples, motor abnormal samples, and normal samples in order from No. 1 to No. 3. The modified IMPULSE analysis was performed to quantify the characteristics of the successive impulse pattern. The modified IMPULSE analysis is an improved analysis method that quantifies the shock signal generated by the knocking of diesel vehicles conducted in a previous study, and the calculation procedures involved are as follows: First, the absolute value of the time signal is obtained, and then the top 5% value of the entire section is calculated. Next, data (positive values) exceeding the top 5% value are extracted, and the time exceeding the top 5% value is calculated. Finally, the result obtained by calculating the time average using the data and time exceeding the upper 5% value is the modified IMPULSE value.

Fig. 8 shows the classification result obtained by applying the modified IMPULSE analysis to the abnormal and normal samples, which confirms that the abnormal and normal samples can be classified.

Figure 7 – Time signal analysis of abnormal and normal samples
Figure 8 – Analysis results of modified IMPULSE

5. CONCLUSIONS

Herein, an abnormal sound detection method for a small actuator used in a vehicle was proposed. First, noise, vibration, and current signals were measured in the actuator production line, The 'abnormal-normal' level of each sample was quantified through a subjective hearing test of the measured noise signal. Next, the cause of the actuator abnormal sound was identified, and the robustness of the vibration signal to disturbance was confirmed based on the signal to noise ratio (SNR) and the coherence analysis of the measured signal. Finally, degree of modulation (MOD) analysis and modified IMPULSE analysis of the vibration signal were performed, and it was introduced that samples can be judged as normal or abnormal through analysis.

Currently, due to the development of vehicle technology, customers' expectations for product performance and quality have been leveled upward. If defective products are not identified and shipped due to quality control errors in the manufacturing process, companies and customers are likely to face various problems. Applying the results of this study to guidelines not only to small actuators for vehicles, but also to guidelines for diagnosing abnormal from various rotating machines and algorithms for automated inspection of production lines, it is expected to help objective/quantitative quality management of products.

ACKNOWLEDGEMENTS

This paper was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P017120, The Competency Development Program for Industry Specialist) and BK21 Program (5199990814084) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education

REFERENCES

Acoustic diagnosis and monitoring for high voltage switchgear

Takashi ENDO(1), Harshpramodbhai PUROHIT(2), Youhei KAWAGUCHI(3), Hikaru KOYAMA(4)

(1) Hitachi, Ltd., Japan, takashi.endo.qf@hitachi.com
(2) Hitachi, Ltd., Japan, harsh_pramodbhai.purohit.yf@hitachi.com
(3) Hitachi, Ltd., Japan, yohei.kawaguchi.xk@hitachi.com
(4) Hitachi, Ltd., Japan, hikaru.koyama.fd@hitachi.com

ABSTRACT
In this report, we propose a method for estimating the stroke speed of high-voltage switching gears in the field of electric power transmission and distribution based on the sound they make when they open and close (opening/closing sound) in order to automate their inspection. The proposed method extracts and stores spectrograms of open/close sounds as templates from recorded training data, and estimates the stroke speed based on template matching during inference. Evaluation using a spring-operated 145 kV switching gear suggests that the estimation accuracy is sufficient to estimate a deviation of 10%, which is considered abnormal.

Keywords: acoustic diagnosis, high voltage switch-gear, anomaly sound detection, sound event detection

1 INTRODUCTION
In Japan and other developed countries, number of maintenance workers is decreasing due to the declining birthrate and aging population. Therefore it becomes difficult to keep quality of inspect infrastructure facilities inspection.[1] In developing countries, it is also difficult to secure skilled workers with the experience and intuition necessary for inspection work. Hence, there is a need to automate the inspection of social infrastructure facilities. In the field of electric power transmission and substation, automation of inspections of high-voltage switching gears is also important.

A diagnostic method based on the sound that a switching gear makes when it opens and closes (open-close sound) is considered promising for automating the inspection of high-voltage switching gears.[2, 3]

Diagnostics using acceleration sensors and coil currents have been attempted in the past[4, 5, 6], but these expensive sensors must be fixed in contact with the equipment, which presents challenges in terms of cost and availability.

In contrast, diagnosis based on opening and closing sounds is relatively inexpensive because the sensor used is a microphone, and it is non-contact, it is expected to be possible to retrofit equipment or to use a portable kit for diagnosis.

However, the conventional method[3] quantifies the degree of deviation from normal based on time-series changes in generated sound pressure at each frequency, but it is difficult for users to understand because it cannot directly estimate changes in open/close time.

In this report, we propose a diagnostic method that directly estimates the opening and closing time.

The proposed method directly estimates the stroke speed from the spectrogram of a short time during opening and closing, focusing on the property that the power spectrogram of the opening and closing operation is constant except for the time expansion and contraction according to the stroke speed. The stroke speed estimated by the proposed method is easy for users to handle because it can be judged as normal or abnormal by comparing it with the rated value of opening/closing time. Experimental results show that the stroke speed can be estimated from the opening and closing sounds with diagnostic accuracy.
2  SWITCH GEAR STROKE DIAGNOSIS

2.1 Significance of Sound Diagnostics in Stroke
The switching gears analyzed in this study are spring-loaded type, which are operated by the force of a spring wound by a motor. When inspecting a switching gear in operation, it is necessary to disconnect it from the system for safety before installing a sensor for inspection. Before the inspection, the switching gear is usually in operation after a long period of inactivity, so there may be signs of abnormality. In some cases, the operation for disconnection causes the disappearance of the signs of abnormality and does not reproduce them in the second and subsequent operations. The disconnection operation may miss the opportunity to measure the predictive state of the abnormality because the sensor for inspection is not installed. Since the microphone can make non-contact measurements, it can be installed without disconnecting from the system, and measurements can be made during disconnection from the grid system.

2.2 Definition of switch gear stroke timing
In the operation of a switching gear, it is important that the contacts open and close at the correct time. The movement of the opening and closing of the contacts of a switching gear is called the stroke. To measure the stroke of the open/close movement, an expensive sensor such as a laser displacement transducer must be added. The measurement of the stroke of the open/close movement requires the addition of an expensive sensor such as a laser displacement transducer. In addition, depending on the structure of the enclosure, the measurement itself may be difficult. In addition, the touch sensor, which measures the energizing and de-energizing of the contact, can also be used to measure the point of change in the energized state. The touch sensor measures the change point of the energized state. If it becomes possible to estimate from sound that can be measured with a non-contact, inexpensive microphone, the cost and operational advantages will be great. The system has significant advantages in terms of cost and operation.

Stroke speed $v$ is defined as the distance $\Delta x$ traveled from the position where the contact point is 10% of the total stroke to the position where the contact point changes contact state divided by the elapsed time $\Delta t$ (Equation (1)). If the stroke speed deviates from the rated speed by more than 10%, it is considered abnormal.

$$v = \frac{\Delta x}{\Delta t} \quad (1)$$

Since the stroke length does not change, the variation of stroke velocity $v$ can be viewed as a deviation of the elapsed time $\Delta t$.

2.3 Estimation of stroke travel time from sound
In the following, we will attempt to determine the elapsed time of the stroke from the sound data. Fig. 2 shows the position of each timing of the stroke. The true value of the timing is generated from the sensor signal. The start, 10%, and end times are calculated from the stroke position, touch position is defined as the point where the touch signal exceeds the midpoint of the maximum and minimum values. touch timing is defined as the point where the touch signal crosses the median of the maximum and minimum values. The touch position is defined as the point where the touch signal exceeds the midpoint between the maximum and minimum values. The spectra at the start (0%), 10%, and touch timing were compressed into two dimensions by t-distributed Stochastic Neighbor Embedding (tSNE), and the distributions were plotted. The result of the 2D compression...
and plotting of the distribution is shown in Fig. 3. In the figure, the shapes of the markers indicate differences in timing, and the sound spectra are distributed separately for each of the 0%, 10%, and touch timings, indicating that there is no significant change in the distributions for the first and second halves of the measurement period. The stroke elapsed time to be estimated in this study is the time from start (0%) to touch in the closed motion. The input signal used for the estimation is the time from start (0%) to touch in the closed motion. Only the sound indicated by waveform in Fig. 2 is used as the input signal, and the power and spectrogram are calculated.

The interval and type of closed motion are estimated by power, and the start (0%), 10%, touch, and end timings are estimated from the spectrogram.

The time between the touch and stroke opening timing is estimated from the sound spectrogram (Fig. 4). For the data measured after multiple openings, the time between start (0%), 10%, and touch timing were compressed two-dimensionally by tSNE and the distribution was plotted. The results are shown in Fig. 5. As in the case of the closed operation, the sound spectra are distributed separately for each of the 0%, 10%, and touch timings, indicating that there is no significant change in the distribution for the first and second halves of the measurement.

2.4 Stroke timing estimation method

As data for stroke estimation, the stroke is measured using a laser displacement meter, and the touch timing at which the contacts join is taken as the true value from the contact signal, and the timing is estimated from sound alone.

The procedure for stroke estimation is shown in Fig. 6. As the power of the sound at opening and closing of the switching gear is very high, the approximate starting time of the opening and closing operation can be extracted from the sound power. The power of sound of the switching gear operation is shown in Fig. 7. The
time variation pattern of the sound power is clearly different between the open and close operations, and the open and close operations can be distinguished.

After extracting the open/close operation interval from the power patterning, the various timings of the open/close operation are extracted from the similarity of each timing to the spectral patterning of the open/close operation.

3 Evaluation experiment with spring-operated switching gear

A spring-operated 145 kV switching gear was used. The sound recording during operation was made simultaneously with the stroke measurement. Fig.8 shows the setup of experimental equipment.

To capture the displacement of 10 percent of the switching gear stroke time, it requires a high accuracy of timing on the order of several milliseconds. Since the speed of sound is 331.5 + 0.61t[m/s] (where t is the temperature in degrees Celsius) in dry air at 1 atmospheric pressure, a difference of 1[m] in distance is about 2.8[ms] in time. When training a model for diagnostics, it is important to consider the time synchronization between the sound waveform and other electrical sensors. The distance between a microphone and target objects must be taken into account.

3.1 Examination of features

The sampling rate of all data including sound data was 20 kHz. Short-time FFTs were concatenated for multiple frames as features. For constructing the feature vectors, the number of frames to be concatenated and the size of FFT window have several choices. In FFT, the window length determines the resolutions of temporal domain and frequency domain. If the analysis window length is short, the time resolution is higher and the frequency resolution is lower. A longer window size results in higher frequency resolution and lower time resolution.

To include time variations in the frequency domain in the feature vector, a short-time FFT is concatenated
over multiple frames to create a feature vector.

Templates were created for each of the open and close events: start, 10%, and touch. The timing of events were estimated for 26 open/close operations to evaluate estimation error. The maximum value of the error was used for each condition. The results are shown in Fig. 9. Based on the estimation error of several conditions, the length of FFT window was set to 256 points and the number of frames to be concatenated was set to the value corresponding to 11[ms].

3.2 Stroke travel time estimation results

An example of the estimated closing motion is shown in Fig.10 and a summary of the results in Fig.11. The estimation error of 3σ is equivalent to 2.4% of the stroke elapsed time, indicating that a deviation of 10% can be detected.

An example of the estimation result of the closing motion is shown in Fig.12 and a summary of the result is shown in Fig.13. The estimated error of 3σ is equivalent to 3.1% of the stroke travel time, indicating that a deviation of 10% is detectable.
4 Conclusion

In this study, to automate the inspection of high-voltage switching gears, we proposed a method to estimate the stroke speed of a switching gear from the sound opening and closing sound. The results of the evaluation experiment confirmed that distribution of the estimation error was shown to be 3.1% for $3(\sigma)$, which is sufficient to measure a 10% deviation.

REFERENCES


Anomalous sound detection using objective metrics related to timbral attributes

Yasuji OTA¹; Seigo KURA; Masashi UNOKI²

¹,² School of Information Science, Japan Advanced Institute of Science and Technology
1-1, Asahi-dai, Nomi, Ishikawa, 932-1292 Japan

ABSTRACT

We propose an anomalous sound detection (ASD) method that uses objective metrics related to timbral attributes to identify whether the sound emitted from a target machine is anomalous. This method involves acoustic-feature extraction and classification. In contrast to current ASD methods focused on the development of machine learning techniques, we investigated the acoustic features related to perceptual factors to improve ASD accuracy. The proposed method uses eight timbral metrics as auditory features and a support vector machine for classification. To use our heuristic knowledge to associate a noticeable difference in hearing with malfunctioning mechanisms, the proposed method weights each timbral metric extracted from a machine's sound signals then detects anomalies using the weighted metrics. We conducted an ASD experiment using the MIMII (Malfunctioning Industrial Machine Investigation and Inspection) dataset to evaluate the anomalous-sound classification performance of the proposed method and compare it with a conventional ASD method that maximizes classification performance by selecting the best combination of timbral metrics. The results indicate that the proposed method slightly outperformed the conventional method, suggesting that modifying the metrics in accordance with a noticeable difference in hearing can be comparable to the best classification performance.

Keywords: Anomalous sound detection, timbral attribute, support vector machine

1. INTRODUCTION

Maintenance work is essential for the safe operation of machines in factories. Monitoring on the basis of acoustics, i.e., human hearing, plays an important role in manufacturing stable products. Therefore, anomalous sound detection (ASD) is expected to help inspectors identify whether the sound emitted from a target machine is anomalous.

The basic ASD framework tends to be a combination of acoustic-feature extraction and classification of anomalous sounds. Mel-frequency cepstrum coefficients or log-Mel energies are generally used as acoustic features and various auto-encoders are typically used as machine-learning techniques for classification (1,2).

An ASD study using timbral attributes as an acoustic feature was conducted (3). Although the study successfully demonstrated that objective metrics related to timbral attributes can improve ASD performance, it was not enough to clarify which timbral attribute contributed to the detection of a machine’s anomalous condition.

From an inspector’s point of view, noticeable differences in hearing the sound emitted from a machine is used as an important monitoring criterion. This criterion might be based on the knowledge to associate the noticeable difference in hearing with a machine’s condition. Since humans can obtain information about their surroundings from timbre (4), timbral attributes can be considered as having a strong relation with a malfunctioning machine.

We propose an ASD method that uses objective metrics related to timbral attributes to identify whether the sound emitted from a target machine is anomalous on the basis of our hearing abilities for ASD. The method uses eight timbral metrics (hardness, depth, brightness, roughness, warmth,
sharpness, boominess, and reverberation) as auditory features and a support vector machine (SVM) for classification. To use the heuristic knowledge regarding timbral attributes, the method features a weighting function of objective metrics, i.e., timbral metrics for improving classification performance. The weighting function was designed from investigating the association of timbral attributes with a machine’s malfunctioning from the viewpoint of the noticeable-difference in hearing. We conducted an ASD experiment using the Malfunctioning Industrial Machine Investigation and Inspection (MIMII) dataset to evaluate the anomalous-sound classification performance of the proposed method and compare it with a conventional ASD method that maximizes classification performance by selecting the best combination of timbral metrics.

The rest of this paper is organized as follows. In Section 2, we briefly introduce timbral attributes. In Section 3, we present our proposed method for optimizing acoustic features. After discussing the experiment and results in Section 4, we conclude the paper in Section 5.

2. TIMBRAL ATTRIBUTES

2.1 Overview and their implementation

Humans obtain information of their surroundings from sound, i.e., auditory perception, and many psychoacoustic studies are conducted to determine the relation between signal analysis and timbre, which is a perceptual variable.

Since timbre is multidimensional, it is separated into several attributes corresponding to adjectival phrases, such as sharpness or roughness. Many studies have been conducted on timbral modeling and implementation of each timbral attribute as objective metrics (5, 6). In one seminal study, the University of Surry developed timbral models in the Audio Commons project, and these models are widely used in psychoacoustic research (7).

2.2 Objective metrics of timbral attributes (timbral metrics)

The proposed method uses eight objective metrics of timbral attributes (i.e., timbral metrics) from the timbral models of the Audio Commons project. The metrics are hardness, depth, brightness, roughness, warmth, sharpness, boominess, and reverberation.

Sharpness, which is a metric related to sharp or shrill sensation, has been reported to increase in magnitude with increasing center frequency. From this perspective, Zwicker defined 1 acum as a unit of a narrow-band noise centered at 1 kHz with a loudness level of 60 phon (5). A sharpness model was then constructed on the basis of the acum and expressed as

\[ S = 0.11 \int_0^{24 \text{Bark}} N'(z) g(z) dz / \int_0^{24 \text{Bark}} N'(z) dz \text{ acum}, \]  

where \( S \) is sharpness, \( N'(z) \) is the loudness density in the critical-band rate \( z \), and \( g(z) \) is the weighting factor of \( S \) at \( z \).

Figure 1 – Weighting factor \( g(z) \) of sharpness as function of critical-band rate \( z \) (5)

Roughness, describing buzzing, harsh, raspy sound quality, is strongly related to the change in the modulation frequency of loudness. As a result of studies on a sinusoidal model approach, a roughness calculation model was proposed (8). This model is constructed from three elements. The first is the dependence of roughness on intensity related to the amplitude of two sinusoidal components. The second element is the dependence of roughness on the amplitude-fluctuation degree related to the amplitude of two sinusoidal components. The third element is the dependences of roughness on the
amplitude-fluctuation rate, which is the frequency difference of two sinusoidal components.

Brightness has been extensively studied, and it has been reported that the spectral centroid and ratio of high frequencies to the overall energy correlates with perceived brightness (9). Therefore, a brightness model was developed to incorporate both a spectral centroid variant and spectral energy ratio (10, 11). The spectral centroid was calculated in bandwidths over 2 kHz, and the energy ratio was calculated as the proportion of energy over 2 kHz to energy over 20 Hz.

Depth, defined as a timbral not spatial attribute, has been reported to be related to an emphasized low-frequency component. Therefore, a model was developed for calculating the depth metric by conducting linear regression with elements, such as the spectral centroid in a range of 20 to 2000 Hz, and the ratio of energy between 20 and 500 Hz (12). The other four timbral metrics, i.e., hardness, warmth, boominess, and reverberation, were also implemented (12).

3. PROPOSED METHOD

3.1 Basic configuration

The proposed method is basically configured as timbral-attributes extraction, weighting the timbral metrics and classification, as shown in Fig. 2.

![Figure 2 - Block diagram of proposed method](image)

In timbral-attribute extraction, eight timbral metrics are derived from a machine sound using the timbral models described in Section 2.2. Each metric is then weighted on the basis of the relation to an anomalous machine sound. The weighting function is defined from the knowledge derived from the original investigation of the association between timbral attributes and malfunction mechanisms. The weighted metrics are then fed into the classification component in which training and classification is executed using a machine-learning technique.

3.2 Weighting of metrics and classification of anomalous sound

In the product manufacturing process, machines to cut raw material or move products in a factory, such as press machines or conveyers, generally form the basis of industrial assembly lines. Such machines generate unique sounds from their operational mechanisms. Table 1 lists the four typical machines and their association between malfunction mechanisms and timbral attributes.
Table 1 – Relations between malfunction mechanisms and timbral attributes

<table>
<thead>
<tr>
<th>Machine type</th>
<th>Malfunction mechanisms</th>
<th>Noticeable difference in hearing by malfunction</th>
<th>Candidates of associated timbral attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Attacking machine</td>
<td>Misalignment, Degradation of hammering</td>
<td>Beat noise, Obscure clatter</td>
<td>Brightness, Roughness, Sharpness, Depth</td>
</tr>
<tr>
<td>[ex. Pressing, Valving]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Sliding machine</td>
<td>Friction increase, Flaking</td>
<td>Noisy rasping, Fricative</td>
<td>Brightness, Roughness, Sharpness, Depth</td>
</tr>
<tr>
<td>[ex. Slider, Grinder]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Rotating machine</td>
<td>Misalignment, Friction increase</td>
<td>Rattling, Noisy hurting</td>
<td>Brightness, Roughness, Boominess</td>
</tr>
<tr>
<td>[ex. Fan, Dryer]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Liquid manipulator</td>
<td>Clogged, Unstable flow</td>
<td>Gurgling, Disappearance of splashing</td>
<td>Brightness, Roughness, Sharpness, Boominess</td>
</tr>
<tr>
<td>[ex. Pump, Compressor]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Based on the consideration of these relations, candidates of associated timbral attributes are selected. The value “0” is then assigned to the timbral metric of non-associated attributes. Values ranging from 0.05 to 1.0 in 0.05 steps are assigned to the timbral metrics of associated attributes to provide the best classification performance with the attributes. Thus, the weighting function for all timbral metrics is developed on the basis of each machine type.

An SVM is used for classification. To apply the SVM for binary classification, the weighted timbral metrics with normal/abnormal flags are used to train a model, then classification of anomalous sound is conducted on the basis of the model dedicated for each machine type.

4. EXPERIMENT AND RESULTS

4.1 Experimental setup

We used the sound dataset from the MIMII database (13) for our experiment. This dataset contains four machine types, i.e., fan, pump, slide rail (slider), and valve, which correspond to the machine types listed in Table 1. In the dataset, normal and abnormal sounds were recorded as 10-sec-long sound files at 16-kHz sampling. The total number of sound files was 14,719 for normal condition and 3,300 for abnormal condition. Four models with IDs 00, 02, 04, and 06 were used for all machine types at a signal-to-noise ratio of 0 dB.

4.2 Noticeable difference in hearing and timbral attributes

As described above, a malfunctioning industrial machine sound noticeably different from its normal operation, and this difference corresponds to the degradation mechanism. Figure 3 compares the spectrograms of normal and abnormal sounds of a slider.

Due to the increase in the friction of metal parts, noisy rasping was detected as abnormal sound from the slider. This difference between its normal sound and this abnormal sound can be expressed as the appearance of a higher-frequency area at a rather flattened magnitude, as shown in Fig. 3.

This difference can be subjectively divided into several timbral aspects. First, the sharpness and brightness of sound increases due to the rubbing of metals. Second, the noisy sound caused by creaking generates a roughness sensation. Third, compared with a normal sound from a slider, sound from friction or flaking may become “deep” due to the low frequency of the signal.

From the investigations regarding association of timbral attributes with malfunctioning mechanisms on each machine type, dedicated weighting functions for each timbral metric are derived as knowledge. The investigation results are summarized in Table 1.
4.3 Classification evaluation

By using the knowledge described in Section 4.2, weighted timbral metrics, \( wtm_{\text{att}}[n] \), are calculated by multiplying the factors of each metric obtained from the timbral models as follows:

\[
wtm_{\text{att}}[n] = wcf_{\text{att}}[n] \times tm_{\text{att}}[n], \quad n=1, 2, ..., N,
\]

where \( wcf \) is the weighing coefficient designed for each combination of machine type and timbral attribute, \( tm \) is a timbral metric, \( N \) is the total number of samples of the metrics, \( att \) is a timbral attribute \{hardness, depth, brightness, roughness, warmth, sharpness, boominess, or reverberation\}, and \( mt \) is a machine type \{fan, pump, slider, or valve\}.

The classification model is then trained with the weighted timbral metrics. We used an SVM for training and classification. The MIMII database has a set of normal/abnormal sound samples with a normality flag on each file. By assigning abnormal samples as "true" and normal samples as "false", dedicated binary decision models can be trained using the SVM (14).

Classification results were evaluated in terms of accuracy, false negative rate (FNR), false positive rate (FPR), and \( F\)-measure. These metrics are calculated as follows:

\[
\text{Accuracy} = \frac{TP+TN}{(TP+FP+TN+FN)},
\]

\[
\text{FNR} = \frac{FN}{(TP+FN)},
\]

\[
\text{FPR} = \frac{FP}{(FP+TN)},
\]

\[
\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{(\text{precision} + \text{recall})},
\]

where \( TP \) is the number of true positives, \( TN \) is the number of true negatives, \( FP \) is the number of false positives, and \( FN \) is the number of false negatives. \( F\)-measure is the harmonic mean of precision and recall, which are calculated as follows.

\[
\text{Precision} = \frac{TP}{(TP+FP)},
\]

\[
\text{Recall} = \frac{TP}{(TP+FN)}.
\]

We used \( F\)-measure to evaluate total performance, which is a measure of a test's accuracy in
statistical analysis of binary classification. The closer $F$-measure is to 1.0, the more performance improves.

### 4.4 Results

Table 2 lists results of the anomalous-sound classification performance of the proposed method compared and the conventional method that maximizes classification performance in terms of $F$-measure by selecting the best combination of the eight timbral metrics (3). Four machine types and four IDs make 16 conditions in total. The results indicate that the classification performance of the proposed method with a dedicated weighting function slightly outperformed the conventional method in terms of average $F$-measure.

<table>
<thead>
<tr>
<th>Machine Type</th>
<th>ID</th>
<th>Accuracy</th>
<th>FNR</th>
<th>FPR</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan</td>
<td>00</td>
<td>0.79</td>
<td>0.83</td>
<td>0.64</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>0.91</td>
<td>0.86</td>
<td>0.31</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>0.82</td>
<td>0.87</td>
<td>0.67</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>06</td>
<td>0.98</td>
<td>0.96</td>
<td>0.07</td>
<td>0.14</td>
</tr>
<tr>
<td>Pump</td>
<td>00</td>
<td>0.98</td>
<td>0.97</td>
<td>0.19</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>0.96</td>
<td>0.95</td>
<td>0.41</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>0.99</td>
<td>0.90</td>
<td>0.07</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>06</td>
<td>0.96</td>
<td>0.97</td>
<td>0.42</td>
<td>0.26</td>
</tr>
<tr>
<td>Slider</td>
<td>00</td>
<td>0.99</td>
<td>0.98</td>
<td>0.01</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>0.96</td>
<td>0.94</td>
<td>0.17</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>1.00</td>
<td>0.89</td>
<td>0.00</td>
<td>0.31</td>
</tr>
<tr>
<td></td>
<td>06</td>
<td>0.89</td>
<td>0.92</td>
<td>0.74</td>
<td>0.48</td>
</tr>
<tr>
<td>Valve</td>
<td>00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>1.00</td>
<td>0.98</td>
<td>0.00</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>0.98</td>
<td>0.99</td>
<td>0.17</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>06</td>
<td>0.90</td>
<td>0.91</td>
<td>0.89</td>
<td>0.83</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>0.94</td>
<td>0.93</td>
<td>0.30</td>
<td>0.34</td>
<td>0.01</td>
</tr>
</tbody>
</table>

### 4.5 Discussion

As shown in Table 2, the proposed method exhibited better $F$-measure in terms of anomaly classification performance against that of the conventional method under 10 out of 16 conditions. By carefully listening to the difference between normal and abnormal sounds stored in the database for the slider, for example, the sound generated by scraping metals against each other became rougher and changed to a dull (un-sharp) perception due to its malfunction. These changes in perception are observed as the differences in roughness and sharpness metrics respectively. Therefore, the proposed method has the advantage of controlling the focus point of perception by weighting the timbral metrics, which is the main reason the proposed method can outperform the conventional method, which uses all timbral metrics.

This indicates that introducing timbral attributes as an acoustic feature is effective for anomalous-sound classification and controlling the contribution degree of each attribute by weighting its metric is promising to improve classification performance. The results also suggest that the proposed method may be able to clarify the contribution of each timbral attribute toward anomalous detection quantitatively and identify specific conditions of machines through investigating the weight factor of each metric.

Table 2 also shows that the classification performance of the proposed method is not as good as that of the conventional method under certain ID conditions. This indicates that further investigation regarding associated-timbral-metric selection and weight assignment is necessary.
5. CONCLUSIONS

We proposed an ASD method for industrial inspection, in which weighting timbral metrics are used to improve anomalous-sound-classification performance. To mimic realistic machine inspection by human hearing, the contribution of the metrics was controlled on the basis of our heuristic knowledge of the association between malfunctioning mechanisms and timbral attributes. Eight metrics provided by the Audio Commons project were introduced in correspondence to timbral attributes, and an SVM was used for training and classification for the ASD.

We conducted an experiment using the MIMII database, which contains recorded sound from four types of machines under both normal and abnormal conditions to evaluate the anomalous-sound-classification performance of the proposed method. The results indicate that the performance of the proposed method with a dedicated weighting function slightly outperformed the conventional method in terms of average $F$-measure. The results also indicate that introducing timbral attributes as an acoustic feature is effective for classification and that controlling the contribution degree of each attribute by weighting its metric is promising to improve classification performance.

Although the experimental results suggest that further research is needed to enhance the performance of the proposed method, it may have the potential to identify specific conditions of machines through investigating the weight factor of each metric.

ACKNOWLEDGEMENTS

This work was supported by SCOPE Program of Ministry of Internal Affairs and Communications (Grant Number: 201605002) and the Fund for the Promotion of Joint International Research (Fostering Joint International Research (B)) (20KK0233).

REFERENCES

Multitask learning for acoustic anomaly detection of machine sounds

Jihoon CHOI* and Jung-Woo CHOI†

1,2 School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea

ABSTRACT

Acoustic anomaly detection of machines has been extensively studied for machine condition monitoring, fault diagnosis, and quality control. Due to the lack of abnormal data in the training stage of a deep neural network, a self-supervised learning technique is being popularly employed, wherein only the features of normal data are trained by pre-defined pretext tasks. Various pretext tasks, such as self-reconstruction or classification, have been proposed to efficiently capture the characteristics of normal data. However, individual pretext tasks inevitably have some limitations; the self-reconstruction can be trivial for stationary sound signals, and the classification task largely depends on the quality and type of data labels. To overcome these limitations, we introduce a multitask model trained by multiple pretext tasks. The proposed model employs a multi-head architecture, in which a WaveNet-based feature extraction layer is shared and utilized for both the reconstruction and classification layers. To discriminate anomalous sounds using a unified anomaly score, Mahalanobis distance between multivariate outputs is utilized. The proposed model is trained and tested by sound datasets from six different types of machines, and the results demonstrate that the proposed multitask model outperforms the conventional single-task model.

Keywords: Anomaly Detection, multitask Learning, multiscale reconstruction

1 INTRODUCTION

Anomaly detection is a process of distinguishing the difference between normal data and abnormal data and has been used in various fields such as security (1), factory (2), medical diagnosis (3), and industrial applications (4). Discovering machine faults at an early stage can reduce factory maintenance expenses, increase product yield, and prevent major accidents. Sounds generated by machines can provide useful clues to detect machine faults, so acoustic anomaly detection of machines has been extensively studied for machine condition monitoring (5), fault diagnosis (6), and quality control (1).

Recently, the deep neural network (DNN) is a popular choice as the anomaly detector for its flexible feature extraction and adaptability to various training data. However, the scarcity of abnormal data during the training stage and the diversity of abnormal features make it difficult to train the DNN model in a supervised setting with labeled abnormal data. Recently, self-supervised learning is being popularly adopted for its advantages of training only with normal data (7, 8). In self-supervised learning, a model is trained to learn features of normal data by assigning several pretext tasks, and the ability to perform the task during the test stage determines the normality of test samples.

There are many pretext tasks for self-supervised anomaly detection. One of them is a WaveNet-based self-reconstruction task that learns features through predicting future data with past data (7). The model refers constant length of time frames to reconstruct future frames, and the model can detect abnormal data based on the amount of reconstruction error. The reconstruction task is advantageous in learning the temporal context of data, i.e., the relation between past, present, and future data. However, in the previous studies (9), the temporal context has been identified from the fixed size of the receptive field. The temporal variation of data, however, occurs in different time scales depending on the machines. The machine sound can be transient or stationary,
and temporal variation can be local or global. Therefore, a flexible architecture that can identify temporal context over various time scales is required.

Another popular choice for the pretext task is the classification task (10), in which a DNN model is trained to classify some internal class labels of normal data and utilizes the confidence of predicted probability to classify abnormal data. The classification task extracts feature required to discriminate data of different labels and does not rely on the temporal context only. For example, when the machine sound is very stationary, the future reconstruction task can be simply accomplished by copying the present information to the future. The classification task can avoid such trivial situations. Nevertheless, the classification task suffers from the overfitting problem, in which the trained model determines any subtle changes from normal data as anomalies, due to the overfitting to meaningless normal data features.

To take the advantage of both pretext tasks, we design a multitask model that has a shared common feature extractor but produces the reconstructed data and class probability through two independent heads. The combination of reconstruction and classification was first proposed in (11) as the deep reconstruction-classification network (DRCN) for the domain adaptation problem. We replace the simple convolutional neural network (CNN)-based autoencoder structure of (11) with the WaveNet that aggregates gated outputs from dilated convolution to resolve the checkerboard artifacts of the CNN-based autoencoder. We also propose a multiscale reconstruction task for anomaly detection, which considers multiple outputs produced from the convolution layers with different receptive fields to determine anomalies based on the temporal context of data analyzed at different time scales.

2 MULTITASK MODEL
2.1 Multitask learning
Multitask learning is a way to improve the model performance through simultaneous training with multiple related tasks. It has been extended into various areas of machine learning including meta-learning, transfer learning, and continual learning (12). Several network structures have also been developed for multitask learning, such as the basic structures that share common feature extractors (13), cross-talk structures that have shared features in the middle layers (14), and structures that fuse features for different tasks in terms of convolution layers instead of linear combinations (15).

Combining different loss functions is another important aspect of training multitask models. In (16), multiple objectives were combined in terms of task-dependent weights that are also trained to maximize the multitask likelihoods. Loss weighting based on learning speed is one of the other ways. In (17), weight was adjusted for each task according to the ratio of initial and current losses. Adjusting the learning speeds of different tasks was also attempted (18) by normalizing the gradient of multiple losses at the final layer.

2.2 Proposed model structures
Modified WaveNet (7) has been widely utilized for anomaly detection for its ability to learn the temporal context of spectrogram data and predict future spectrum for the self-reconstruction task. Our proposed model is also based on the WaveNet structure but employs a multi-head architecture, in which a WaveNet-based feature extraction layer is shared but its output is fed into dual DNN layers for the reconstruction and classification tasks. The feature extraction layer follows the design of (7) and begins with a 1×1 causal convolution encoder of kernel size 1, followed by a series of residual blocks. Each residual block has two causal dilated convolution layers with different activation units (gate and value) whose outputs are then multiplied to yield a single stream of gated features. The resultant signal then passes through two individual 1×1 convolution layers to generate the input for the next residual block and skip connections (9). Since all convolution layers are causal, the features are extracted from the past frames of a spectrogram whose temporal lengths correspond to the receptive fields of dilated convolutions. Dilation rates are configured to exponentially increase in the upper residual blocks (1, 2, 4, 8, 16), so upper blocks have wider receptive fields along the time axis than lower blocks. We utilize these differences in the size of the receptive field to implement the multiscale reconstruction task for anomaly detection. In detail, the bottom residual blocks extract local features from the convolution layers with narrow receptive fields, while the upper blocks are responsible for global features. Unlike the conventional model (7), these multiscale features extracted from residual blocks are not summed up for reconstruction. Instead, they are connected in parallel into individual reconstruction blocks consisting of two 1D convolution layers to produce
multiscale reconstruction signals. For the classification layer, multiscale features are stacked along the channel dimension and fed into the ResNet18 (19) classification block. From the multiscale reconstruction, we expect the model can cope with various types of anomalies occurring from the local to global time spans.

Layer normalization is applied before all convolution layers, except for the classification block and the last convolution layer of the reconstruction block. The overall structure of the proposed multitask/multiscale model is shown in Figure 1.

2.3 Loss functions for multiple tasks

To train the multitask model, we combine two different loss functions corresponding to the reconstruction and classification tasks. For the reconstruction task, we use the mean square loss between the input spectrogram \( \mathbf{X} = [x_1, \cdots, x_T] \in \mathbb{R}^{F \times T} \) with \((F, T)\) bins in frequency and time, and reconstructed multiscale spectrogram \( \hat{\mathbf{X}}_k = [\hat{x}_{k1}, \cdots, \hat{x}_{kT}] \) from the \( k \)th branch of the multiscale reconstruction model \((k = \{1, \cdots, K\})\). Denoting the set of multiscale spectrograms as \( \hat{\mathbf{X}} = \{\hat{\mathbf{X}}_1, \cdots, \hat{\mathbf{X}}_K\} \), the mean square loss can be defined as

\[
L_1 = \text{MSE}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{T - d_k} \sum_{t=d_k+1}^{T} \|x_t - \hat{x}_{kt}\|_2^2 ,
\]

(1)

where \( d_k \) denotes the size of the receptive field corresponding to the dilated convolution of the \( k \)th branch. Since the multiscale model generates \( K \) different spectrograms from the convolutions with different receptive fields, the effective reconstruction is made from the time index \( d_k + 1 \) to \( T \).

For the classification task, we use the cross-entropy loss

\[
L_2 = - \sum_{c=1}^{C} p_c \log(\hat{p}_c) ,
\]

(2)

where \( p_c \) is the ground truth (target) probability of \( \mathbf{X} \) for the class \( c \in [1, C] \), which is equal to one only for a true class and is zero otherwise. \( \hat{p}_c \) is the predicted probability from the classification block. The final combined loss function is a weighted sum of the two loss functions

\[
L = \lambda L_1 + (1 - \lambda) L_2.
\]

(3)

Combining different losses may result in training imbalance, so we use the Dynamic Weight Average technique (17) that adjusts the weighting parameter \( \lambda \) using the instantaneous loss reduction rate for each task. Here, the dynamic weight \( \lambda_i \) for the \( i \)th iteration step is given by

\[
\lambda(i) = \frac{w_1(i-1)}{w_1(i-1) + w_2(i-1)}, \quad \text{where } w_1(i-1) = \exp(L_\ell(i-1)/L_\ell(i-2)) \text{ for } \ell = 1 \text{ or } 2.
\]

(4)
3 EXPERIMENTS AND RESULTS

3.1 Datasets
We used the dataset from the Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring Task of the IEEE Audio and Acoustic Signal Processing Society’s 2020 Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge (20). The dataset consists of machine sound data from six types of machines: toy-car, toy-conveyor, fan, pump, slider, and valve, which have different temporal characteristics (stationary/transient). For each machine type, data are further categorized into six to seven classes depending on the machine ID. These machine IDs were used as class labels for the classification task. Details of DCASE 2020 datasets used for train and test are summarized in Table 1.

Table 1. Number of train and test data for each machine type

<table>
<thead>
<tr>
<th>Machine types</th>
<th>Machine id</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>normal</td>
<td>normal</td>
</tr>
<tr>
<td>Toycar</td>
<td>1 to 7</td>
<td>7000</td>
<td>1400</td>
</tr>
<tr>
<td>Toyconveyor</td>
<td>1 to 6</td>
<td>6000</td>
<td>2399</td>
</tr>
<tr>
<td>Fan</td>
<td>0 to 6</td>
<td>6521</td>
<td>400</td>
</tr>
<tr>
<td>Pump</td>
<td>0 to 6</td>
<td>5766</td>
<td>400</td>
</tr>
<tr>
<td>Slider</td>
<td>0 to 6</td>
<td>5174</td>
<td>400</td>
</tr>
<tr>
<td>Valve</td>
<td>0 to 6</td>
<td>5822</td>
<td>400</td>
</tr>
</tbody>
</table>

3.1.1 Data preprocessing
The 10s-long waveforms sampled at 16kHz sampling rate were converted to log-mel spectrograms using the short-time Fourier transform of the window size 2048 and hop size 512. From the conversion to the log mel-spectrogram with 128 mel-filterbanks, we obtained 2D images of size \((F, T) = (128, 313)\). The frequency axis was treated as the channel dimension in all convolution layers. Since most data had emphasized low-frequency components, frequency normalization was applied to balance the signal level across all frequencies, such that each frequency component of log-mel spectrogram has zero mean and unit variance.

To prevent overfitting in the classification task and increase the diversity of data without losing characteristics of normal data, Mixup (21) was applied. In detail, two spectrograms \(X^{(n)}\) and \(X^{(m)}\) with different data indices \(n\) and \(m\) in the train dataset, respectively, were mixed with a randomly-generated ratio populated from a uniform distribution \(\eta \sim U(0, 1)\) to synthesize a new data

\[
X^{mix} = \eta X^{(n)} + (1 - \eta) X^{(m)}. \tag{5}
\]

The target probability of the synthesized data \(X^{mix}\) was then set to

\[
p^{mix} = \eta p^{(n)} + (1 - \eta) p^{(m)}, \tag{6}
\]

for one-hot vectors \(p^{(n)} = [p_1^{(n)}, \ldots, p_C^{(n)}]^T\) and \(p^{(m)}\) containing the target probability of the \(n\)th and \(m\)th data samples in the train dataset, respectively. The model (\(M(\cdot)\)) takes the mixed spectrogram \(X^{mix}\) and target probability \(p^{mix}\) as the input, and then produces the reconstructed multiscale spectrograms as well as the predicted probability \(\hat{p} : \{\hat{X}_1, \ldots, \hat{X}_K, \hat{p}\} = M(X^{mix}, p^{mix})\).

3.2 Combination of different anomaly scores
The proposed multitask/multiscale model determines anomalous data by comparing the distribution of reconstruction and classification scores. In the reconstruction task, the model trained over the normal data in the
Figure 2. Comparison of AUC-ROCs from WaveNet, ResNet18, and the proposed model.

The training dataset can reconstruct the normal data in the test dataset using the learned features. However, the reconstruction error occurs against the anomalous data with unseen features during training. Therefore, the mean squared error between the input and reconstructed spectrogram can be used as an anomaly score to determine anomalous data. To combine different mean squared errors from different blocks, Mahalanobis distance is used. First, we calculate the mean squared error $l_k$ between the input $X$ and $k$th branch output $\hat{X}_k$,

$$l_k = \frac{1}{T-d_k} \sum_{t=d_k+1}^{T} \| x_t - \hat{x}_kt \|_2.$$  

Then, we define a loss vector $l = [l_1, \cdots, l_K]^T$ and derive mean $m$ and covariance $S$ for the training data. For the $n$th input spectrogram $X^{(n)}$ in the test dataset and its loss vector $l^{(n)}$, the reconstruction score is given by

$$\epsilon_1^{(n)} = \sqrt{(l^{(n)} - m)^T S^{-1} (l^{(n)} - m)}.$$  

In the classification task, the trained model can clearly distinguish the class for the normal data but not for abnormal data. Accordingly, anomalies can be detected by measuring similarity between the target and predicted probability vectors:

$$\epsilon_2^{(n)} = 1 - p^{(n)^T} \hat{p}^{(n)},$$  

which is denoted as the classification score in this work.

To combine these two heterogeneous scores into a unified anomaly score, we also calculate the Mahalanobis distance. First, we define a score vector $\epsilon^{(n)} = [\epsilon_1^{(n)}, \epsilon_2^{(n)}]^T$ consisting of the reconstruction and classification scores, and derive its mean $\mu$ and covariance $\Sigma$ for the training data. Then, the Mahalanobis distance $S^{(n)}$ of the test data is calculated to measure the difference from the score distribution for the normal data. That is,

$$S^{(n)} = \sqrt{(\epsilon^{(n)} - \mu)^T \Sigma^{-1} (\epsilon^{(n)} - \mu)}.$$  

### 3.3 Comparison to single-task models

To evaluate the anomaly detection performance of the model, the area under the curve-receiver operating characteristic (AUC-ROC) (22) of the proposed multitask/multiscale model was compared with a single reconstruction model (WaveNet (9)) and a classification model (ResNet18 (19)).

The AUC-ROCs shown in Figure 2 indicate that the anomaly detection performance for six different machine types was improved with the proposed model compared to the single-task models. For the Toycar, Fan, Pump, and Valve datasets, the classification model shows higher performance than the reconstruction model, while the reconstruction model is more beneficial for the Toyconveyor and Slider datasets. The proposed multitask model, however, takes the advantage of both models and can achieve the highest performance for all datasets. To
Figure 3. Distribution of the reconstruction and classification scores from the proposed model.

Figure 4. Comparison of the reconstruction error map for normal data. (a) input spectrogram in a normalized dB scale (zero mean and unit variance) (b) reconstruction error map from WaveNet, (c) reconstruction error map from the third reconstruction layer of the proposed model.

Figure 5. Comparison of the reconstruction error map for anomalous data. (a) input spectrogram in a normalized dB scale (zero mean and unit variance) (b) reconstruction error map from WaveNet, (c) reconstruction error map from the third reconstruction layer of the proposed model.
illustrate how the multitask model can determine anomalies, distributions of normal and anomalous data with respect to the reconstruction and classification scores are presented in Figure 3. In the examples with the Pump and Slider datasets, both the reconstruction and classification scores contribute to distinguishing normal and anomalous data. In contrast, for Valve and Slider datasets, the classification score plays a dominant role rather than the reconstruction score. Likewise, the multitask model using both measures provide more rich information for anomaly detection than the single task models.

Next, we compare the distribution of reconstruction errors in frequency and time to investigate the effectiveness of using multiscale spectrograms. Figure 4(a) shows one exemplary frequency-normalized spectrogram drawn from the normal machine in the Valve dataset. Reconstruction error maps \((X - \hat{X})\) obtained by the conventional WaveNet and the proposed model are shown in Figure 4(b, c), respectively. Among multiscale outputs from the proposed model, the output from the third layer with the receptive field 15 is depicted. The reconstruction errors for the normal data only have a subtle difference between the two models. However, for the same graphs depicted for the anomalous data (Figure 5), we can see that the proposed multiscale model can produce more error patterns corresponding to the local anomalies than those from the WaveNet model. Such detection of both local and global anomalies can improve the anomaly detection performance for various data with different temporal variations and characteristics.

4 CONCLUSIONS
In this work, we proposed a multitask model with multiscale reconstruction and classification blocks to overcome the shortcomings of existing anomaly detection models. Individual pretext tasks have some limitations; the reconstruction task using a fixed length of the receptive field can not cover various machine sounds, the reconstructed output can be trivial for stationary sounds, and the classification task needs high-quality data labels. To solve these limitations, we proposed a multitask model with multiple pretext tasks. Our model consists of a WaveNet-based feature extractor and both reconstruction and classification layers in parallel. The reconstruction layer was designed to produce multiple outputs through the predictions with different lengths of receptive fields, while the classification layer predicts internal class using features extracted with different lengths of receptive fields. The reconstruction and classification scores were then combined in terms of the Mahalanobis distance to distinguish abnormal data using a unified score. Experiments with DCASE 2020 datasets demonstrate that the proposed model outperforms the single-task models in the test for various machine-type data and the weakness of a single task can be resolved by incorporating multiple tasks.

ACKNOWLEDGEMENTS
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (Ministry of Science and ICT) (No. NRF-2020M2C9A1062710) and supported by the BK21 Four program through the National Research Foundation (NRF) funded by the Ministry of Education of Korea.

REFERENCES


On-site Noise Exposure technique for noise-robust machine fault classification

Wonjun YI1; Jung-Woo CHOI2

1,2 School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea

ABSTRACT

In-situ classification of faulty sounds is an important issue in machine health monitoring and diagnosis. However, in a noisy environment such as a factory, machine sound is always mixed up with environmental noises, and noise-only periods can exist when a machine is not in operation. Therefore, a deep neural network (DNN)-based fault classifier has to be able to distinguish noise from machine sound and be robust to mixed noises. To deal with these problems, we investigate on-site noise exposure (ONE) that exposes a DNN model to the noises recorded in the same environment where the machine operates. Like the outlier exposure technique, noise exposure trains a DNN classifier to produce a uniform predicted probability distribution against noise-only data. During inference, the DNN classifier trained by ONE outputs the maximum softmax probability as the noise score and determines the noise-only period. We mix machine sound and noises of the ToyADMOS2 dataset to simulate highly noisy data. A ResNet-based classifier trained by ONE is evaluated and compared with those trained by other out-of-distribution detection techniques. The test results show that exposing a model to on-site noises can make a model more robust than using other noises or detection techniques.

Keywords: fault classification, noise exposure, on-site noise

1. INTRODUCTION

As the manufacturing process is automatized in a smart factory, on-site fault classification based on machine noise is gaining much attention. In a factory, however, noise is always present and the machine produces a sound only during its operation. Consequently, there may be several signal periods where no machine sound is present but only factory noise exists. For robust autonomous fault diagnosis, a fault classifier is required to be robust to a high level of background noise and to distinguish noise-only periods from machine sounds.

Previously, research on out-of-distribution (OOD) detection has been conducted to detect outliers lying outside of the distribution formed by normal data. Such techniques can be utilized to detect noise-only periods in our application. For example, in (1), OOD samples were detected based on the maximum softmax probability predicted from a DNN model trained by a classification task. Another OOD detection algorithm utilizing the free energy function was also studied in (2).

On the other hand, outlier exposure (OE) has been considered an effective way to improve the OOD performance and robustness of the classifier (3, 4). OE utilizes an outlier dataset disjoint from in-distribution and test-time data. In OE, a model is trained to output uniform distributions for outlier data to make maximum softmax probability (MSP) as low as possible. Using the difference in MSPs for the in-distribution and OOD data, the model can detect OOD samples at test time.

In this paper, we report the importance of noise types used for OE. As mentioned, the objective of the training is (a) to classify the fault types and (b) to detect noise-only data. To accomplish these two goals at the same time, we expose a DNN classifier to different types of noises and compare the performance. Like OE, the classifier is trained to produce a uniform probability distribution for noise signals and to predict a one-hot vector for noisy machine sounds. However, we investigate the performance difference when noises used for OE are from similar or dissimilar distribution to those of on-site factory noises embedded in the noisy machine sound signal. We denote the OE with on-site

1 lasscap@kaist.ac.kr
2 jwoo@kaist.ac.kr
factory noises as the on-site noise exposure (ONE) and compare the performance of the model trained by ONE with models exposed to different types of noises. In addition, to verify the effectiveness of ONE, we also compare it with other training techniques, such as mapping noise-only data into an additional class or free energy-based techniques: energy score and energy-bounded learning (2). The ToyADMS02 dataset is used to synthesize machine sounds with factory noises and noise-only data, and the noise-only period detection and fault type classification performances are evaluated. The result shows that ONE outperforms other training techniques, as well as the OE using other types of noises.

2. BACKGROUND

2.1 DNN classifier and softmax score

A DNN classifier $f(\cdot)$ is trained to classify in-distribution data $x_{in} \in D_{in}$ and predicts softmax probability $f(x_{in}) \in \mathbb{R}^K$ for $K$ classes. The training of the classifier is done such that the categorical cross entropy (CCE) between the target probability $y \in \mathbb{R}^K$ given by a one-hot vector and the predicted probability $f(x_{in})$ can be minimized. At inference time, the softmax score $A(x)$ can be calculated by taking the maximum value of the predicted probability $A(x) = \text{Max}(f(x))$. (1)

Here, $\text{Max}(\cdot)$ is the operator taking the maximum value of a vector. In the previous study (1), the softmax score was used as a measure of OOD, i.e., data are regarded as in-distribution if the softmax score is higher than a pre-defined threshold and OOD otherwise.

2.2 Outlier exposure

For outlier exposure, a DNN model is trained by not only the in-distribution dataset $D_{in}$ but also an auxiliary dataset $D_{OE}$ for the outliers. The DNN classifier is trained to classify in-distribution data $x_{in}$ as a ground truth one-hot vector $y$ but to produce uniform distribution $u = [1, \ldots, 1]^T / K$ against the outlier data $x_{OE}$. The total loss function $\mathcal{L}_{\text{total}}$ can be written as

$$\mathcal{L}_{\text{total}} = \mathbb{E}_{(x,y) \sim D_{in}}(\mathcal{H}(y, f(x))) + \mathbb{E}_{(u,x) \sim D_{OE}}(\alpha \mathcal{H}(u, f(x)))) .$$ (2)

$\mathcal{H}(\cdot)$ denotes the cross-entropy, so the first and second terms of Eq. (2) represent CCE of in-distribution and outlier data, respectively. The constant $\alpha$ is a balancing weight for the joint training. After training, the softmax score $A(x)$ will be high for in-distribution data $x_{in}$ but low for $x_{OE}$, yielding the estimation of outliers. Although the outlier dataset $D_{OE}$ is different from the actual OOD data used in the test time, the DNN classifier learns the ability to discriminate OOD data by tightening the decision boundary to exclude the outliers provided for OE.

2.3 Energy score

Energy score is another measure introduced for OOD detection, which is defined as the negative value of the free energy function:

$$-E(x) = T \log \sum_{k=1}^{K} \exp \left( \frac{g_k(x)}{T} \right),$$ (3)

where $g_k(x)$ is the logit value of the classifier for the $k$th class, and $T$ is the temperature parameter (2). It was known that the softmax score is a special case of the energy score (2), where all logits are biased by the maximum logit value. Since the biased scoring function is not desirable for OOD detection, the energy score is claimed to be more advantageous than the softmax score.

2.4 Energy-bounded learning

While the energy score only considers a different scoring of a trained network, the energy-bounded learning (2) utilizes the free energy function for training. The DNN classifier is trained to produce a high energy score for in-distribution data and a low score for outlier data. To this end, the
regularization loss $\mathcal{L}_{\text{energy}}$ is defined using two squared hinge loss terms as below formula:

$$
\mathcal{L}_{\text{energy}} = \mathbb{E}_{(x,y) \sim D_{\text{in}}} (\max(0, E(x_{\text{in}}) - m_{\text{in}}))^2 + \mathbb{E}_{(x,\alpha) \sim D_{\text{out}}} (\max(0, m_{\text{out}} - E(x_{\text{out}})))^2.
$$

(4)

Here, $m_{\text{in}}$ and $m_{\text{out}}$ are margin hyperparameters for in-distribution and outlier data, respectively. The total loss unifying the classification and regularization losses can be described as shown below:

$$
\mathcal{L}_{\text{total}} = \mathbb{E}_{(x,y) \sim D_{\text{in}}} (\mathcal{H}(y, f(x))) + \beta\mathcal{L}_{\text{energy}}.
$$

(5)

Here, $\beta$ is a balancing weight for joint training. During inference, an energy score is used for OOD detection.

3. ON-SITE NOISE EXPOSURE (ONE)

**Figure 1 – Training and inference procedures of ONE**

In-situ data acquired in a factory can be split into signal segments of a fixed length. Each segment can include only background noises, which we want to determine as outliers. To enhance the detection performance, we apply the OE technique using the background noises of the factory measured while the machine is inactive.

Figure 1 illustrates the training and inference procedures of ONE. The dataset to train the DNN model is the combination of in-distribution data $D_{\text{in}}$ with machine sounds contaminated by on-site factory noises and noise-only data $D_{\text{out}}$ measured during noise-only periods. In the training step, two input batches are sampled from $D_{\text{in}}$ and $D_{\text{out}}$. Both batches have the data size $(B,1,T)$, where $B$ represents the batch size, 1 is the number of the channel (mono channel), and $T$ represents the number of samples in time. During training, the classifier predicts the fault type probability $f(x_{\text{in}})$ from the input $x_{\text{in}} \in D_{\text{in}}$, whose target $y_{\text{in}}$ is given by a one-hot vector. Like OE, the classifier is trained to output a uniform distribution for noise-only data $x_{\text{out}} \in D_{\text{out}}$. Similar to Eq. (2), the loss function can be defined as

$$
\mathcal{L}_{\text{NE}} = \mathbb{E}_{(x,y) \sim D_{\text{in}}} (\mathcal{H}(y_{\text{in}}, f(x_{\text{in}}))) + \alpha\mathbb{E}_{x \sim D_{\text{out}}} (\mathcal{H}(u, f(x_{\text{out}}))).
$$

(6)

We denote the second term of Eq. (7) as a noise exposure loss.

At the inference time, we determine noise-only data using the noise score $N(x)$ defined as
The model calculates the noise score for each data at inference time and judges as noise-only data when the noise score is higher than the pre-defined threshold $\eta$. When the noise score is lower than the threshold, the model classifies the fault type by finding the class corresponding to the maximum of $f(x)$. The threshold is determined through the validation step such that the optimal model performance (macro F1 score) can be obtained, and then the fixed threshold is used for the test.

One possible alternative form of noise exposure is to use the free energy function (FE) as the noise score. That is,

$$N_{FE}(x) = -E(x).$$

The energy-bounded learning (EB) mentioned in Eq. (1) can also be utilized for the training with noise exposure. In this case, the loss function can be defined as

$$L_{EB} = \mathbb{E}_{(x,y)\sim D_{M}}(H(y_{M}, f(x_{M}))) + \beta L_{energy}$$

where

$$L_{energy} = \mathbb{E}_{(x,y)\sim D_{M}}(\text{Max}(0, E(x_{M}) - m_{M}))^2 + \mathbb{E}_{x_{N}\sim D_{N}}(\text{Max}(0, m_{N} - E(x_{N})))^2,$$

for margin hyperparameters $m_{M}$ and $m_{N}$ of machine sound and noise-only data, respectively. For energy-bounded learning, the same noise score as Eq. (8) is used.

The last alternative form we examine for noise exposure is to train a model to classify the noise-only data into a separate class (additional class; AC). The loss function for this objective can be written as

$$L_{AC} = \mathbb{E}_{(x,y)\sim D_{M}}(H(y_{M}, f(x_{M}))) + \mathbb{E}_{x_{N}\sim D_{N}}(H(y_{N}, f(x_{N}))),$$

where the target probability vector $y_N \in \mathbb{R}^{K+1}$ is equal to one only for the $(K+1)$th class and zeroes otherwise.

4. EXPERIMENT

4.1 Dataset

Table 1 – The number of data of each class for the toy car dataset

<table>
<thead>
<tr>
<th></th>
<th>Normal</th>
<th>Fault</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>200</td>
<td>600 (50 per each fault type)</td>
<td>800</td>
</tr>
<tr>
<td>Validation</td>
<td>100</td>
<td>300 (25 per each fault type)</td>
<td>400</td>
</tr>
<tr>
<td>Test</td>
<td>100</td>
<td>300 (25 per each fault type)</td>
<td>400</td>
</tr>
</tbody>
</table>

Table 2 – The number of data of each class for the toy train dataset

<table>
<thead>
<tr>
<th></th>
<th>Normal</th>
<th>Fault</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>480</td>
<td>1440 (120 per each fault type)</td>
<td>1920</td>
</tr>
<tr>
<td>Validation</td>
<td>160</td>
<td>480 (40 per each fault type)</td>
<td>640</td>
</tr>
<tr>
<td>Test</td>
<td>160</td>
<td>480 (40 per each fault type)</td>
<td>640</td>
</tr>
</tbody>
</table>

For the experiment, we used the ToyADMOS2 dataset (5) consisting of sounds from two toy machines (a toy car and a toy train) and environmental noises. Among five different machine model types and speed levels in the dataset, we selected model type A and speed 1 for both toy car and toy train data. Conditions of each machine are labeled by four types of faults (a, b, c and d) and three damage levels (low, middle, and high), so there is a total of 13 conditions including the normal condition. Every data is 12 s long and the sampling rate is 16 kHz. Noise data of ToyADMOS2 include factory noise signals recorded in four different environments: N1, N2, N3, and N4. We denote these as the noise environment to indicate whether the model is trained and tested by the same or different noise data. To increase the diversity of noise data, the time shift (0–2 s) and volume perturbation (0.5–2 times) augmentation was applied to each factory noise signal during the training step.

To simulate noisy environments, we mixed machine sound and noise data with SNR from -10 dB to 0 dB. Considering the rarity of faulty machine data in real situations, we used four times as many normal data as fault data. Details of the train, validation, and test dataset are presented in Tables 1 and...
2. In this work, noise-only data are also required for simulating noise-only periods, so we separately prepared the noise dataset by collecting the noise data in ToyADMOs2.

### 4.2 Model architecture

<table>
<thead>
<tr>
<th>Operation</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mel spectrogram</td>
<td>$(B,1,192000)$</td>
<td>$(B,1,128,374)$</td>
</tr>
<tr>
<td>Batch normalization</td>
<td>$(B,1,128,374)$</td>
<td>$(B,1,128,374)$</td>
</tr>
<tr>
<td>Feature extractor</td>
<td>$(B,1,128,374)$</td>
<td>$(B,1024,1,1)$</td>
</tr>
<tr>
<td>Squeeze</td>
<td>$(B,1024,1,1)$</td>
<td>$(B,1024)$</td>
</tr>
<tr>
<td>Linear classifier</td>
<td>$(B,1024)$</td>
<td>$(B,13)$</td>
</tr>
<tr>
<td>Softmax activation</td>
<td>$(B,13)$</td>
<td>$(B,13)$</td>
</tr>
</tbody>
</table>

We modified ResNet for ASC (Acoustic Scene Classification) proposed by Koutini et al. (6). The model structure and sizes of input and output are presented in Table 3. The model first transforms a raw audio signal into a Mel spectrogram, which is then standardized by batch normalization and put into the feature extractor. The output of the feature extractor is squeezed and fed into the linear classifier layer. Lastly, softmax activation is applied to generate the predicted probability. Since there are 13 machine condition classes, the output of the model has a shape of $(B, 13)$ for batch size $B$.

### 4.3 Train and validation

We trained the model for 100 epochs. The batch size of machine sound and the noise was 8, respectively, so the total batch size was 16. Adam optimizer was used, and the learning rate was 1e-4 from epoch 1 to epoch 30. Until epoch 90, the learning rate was linearly decreased to 1e-5 and then maintained to epoch 100.

We tested and compared the model trained by ONE with the model only using the softmax score, free energy score (FE), energy-bounded learning (EB), and classifying noise-only periods as an additional class (AC). Details of all techniques are described in Table 4.

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Loss</th>
<th>Noise Score</th>
<th>Noise exposure data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax score</td>
<td>CCE</td>
<td>Negative MSP</td>
<td>Not exist</td>
</tr>
<tr>
<td>Noise exposure (NE)</td>
<td>CCE, noise exposure loss</td>
<td>Negative MSP</td>
<td>Exist</td>
</tr>
<tr>
<td>Energy score (FE)</td>
<td>CCE</td>
<td>Negative energy score</td>
<td>Not exist</td>
</tr>
<tr>
<td>Energy-bounded learning (EB)</td>
<td>CCE, regularization loss</td>
<td>Negative energy score</td>
<td>Exist</td>
</tr>
<tr>
<td>Additional class with noise data (AC)</td>
<td>CCE</td>
<td>-</td>
<td>Exist</td>
</tr>
</tbody>
</table>

To construct loss functions, the balancing weight was set to $\alpha = 0.5$ for noise exposure, and $\beta = 0.1$ for energy-bounded learning. The margins of energy-bounded learning, $m_M$ and $m_N$, were set to $-25$ and $-7$, respectively. The temperature parameter $T$ was equal to 1. The model was validated after the end of each epoch. The noise score threshold $\eta$ was set to optimize the model performance and recorded for each epoch. After 100 epochs, the best model parameter and threshold were used for the test.

### 4.4 Test

The evaluation metric used for all experiments is the macro F1 score, which is the average of all F1 scores across 14 different classes (one normal class, one noise class, and 12 fault classes). Since the noise class is included in the classification labels, both the fault-type classification and noise detection performance can be evaluated by a single measure. In detail, the macro F1 score is defined as
Each experiment was repeated three times with different random seeds, and the distribution of scores was presented with its mean value.

5. RESULTS

We conducted three different experiments to verify the noise exposure. In section 5.1, we compare ONE with other techniques by training and testing the models using the noises from the same environment. In section 5.2, we examine the performance change when the noises used for the train are different from those for the test. Lastly, in section 5.3, we compare the performance of noise exposure for on-site noise and other-site noise datasets.

5.1 Performance under the same noise environment

In the first experiment, we compared five different methods using different on-site noises. The noises from the same environment were used for the train and test datasets but with no overlapping samples between them. Figure 2 shows the macro F1 scores of the comparison sets. We can see that noise exposure (NE) performs best for the toy car. For the toy train, noise exposure outperforms the others for noise environment N1, but energy-bounded learning is better than the others for noise environments N2, N3, and N4. Nevertheless, noise exposure still shows the second-best performance among techniques. A similar trend can be found for the toy train data (Figure 3), where noise exposure and energy-bounded learning show the best performances.

![Figure 2 – Performance comparison of different models when the noises used for the training and test are from the same environment (toy car data). (N1–N4: noises from different environments)](image)

![Figure 3 – Performance comparison of different models when the noises used for the training and test are from the same environment (toy train data). (N1–N4: noises from different environments)](image)
5.2 Performance under the unseen noise environment

Test results when noises from different environments were used for the train and test are presented in Tables 5 and 6. The first cell of each row indicates the noise environment used for train, validation, and test, respectively. Each column represents the applied technique. The bold text highlights the best case for each row. Tables 5 and 6 reveal that noise exposure performs best for most cases of the toy car and toy train. Even when the noise exposure does not score the best, it still shows the second-best performance.

Table 5 – Performance comparison of different models when the noises used for the training (validation) and test are from different environments (toy car data).

<table>
<thead>
<tr>
<th>Train Noise / Validation Noise / Test Noise</th>
<th>Softmax score</th>
<th>Noise exposure</th>
<th>Energy score</th>
<th>Energy-bounded learning</th>
<th>Additional class with noise data</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1 / N1 / N2</td>
<td>0.564</td>
<td>0.777</td>
<td>0.509</td>
<td>0.640</td>
<td>0.692</td>
</tr>
<tr>
<td>N1 / N1 / N3</td>
<td>0.490</td>
<td><strong>0.676</strong></td>
<td>0.463</td>
<td>0.594</td>
<td>0.617</td>
</tr>
<tr>
<td>N1 / N1 / N4</td>
<td>0.386</td>
<td>0.518</td>
<td>0.387</td>
<td>0.189</td>
<td><strong>0.521</strong></td>
</tr>
<tr>
<td>N2 / N2 / N1</td>
<td>0.536</td>
<td><strong>0.703</strong></td>
<td>0.543</td>
<td>0.487</td>
<td>0.427</td>
</tr>
<tr>
<td>N2 / N2 / N3</td>
<td>0.657</td>
<td><strong>0.799</strong></td>
<td>0.644</td>
<td>0.690</td>
<td>0.668</td>
</tr>
<tr>
<td>N2 / N2 / N4</td>
<td>0.343</td>
<td><strong>0.655</strong></td>
<td>0.322</td>
<td>0.218</td>
<td>0.377</td>
</tr>
<tr>
<td>N3 / N3 / N1</td>
<td>0.528</td>
<td><strong>0.601</strong></td>
<td>0.575</td>
<td>0.293</td>
<td>0.489</td>
</tr>
<tr>
<td>N3 / N3 / N2</td>
<td>0.534</td>
<td><strong>0.610</strong></td>
<td>0.553</td>
<td>0.336</td>
<td>0.611</td>
</tr>
<tr>
<td>N3 / N3 / N4</td>
<td>0.380</td>
<td><strong>0.397</strong></td>
<td>0.392</td>
<td>0.240</td>
<td>0.324</td>
</tr>
<tr>
<td>N4 / N4 / N1</td>
<td>0.483</td>
<td><strong>0.539</strong></td>
<td>0.512</td>
<td>0.387</td>
<td>0.348</td>
</tr>
<tr>
<td>N4 / N4 / N2</td>
<td>0.576</td>
<td><strong>0.590</strong></td>
<td>0.587</td>
<td>0.361</td>
<td>0.476</td>
</tr>
<tr>
<td>N4 / N4 / N3</td>
<td><strong>0.579</strong></td>
<td>0.577</td>
<td>0.551</td>
<td>0.377</td>
<td>0.522</td>
</tr>
</tbody>
</table>

Table 6 – Performance comparison of different models when the noises used for the training (validation) and test are from different environments (toy train data).

<table>
<thead>
<tr>
<th>Train Noise / Validation Noise / Test Noise</th>
<th>Softmax score</th>
<th>Noise exposure</th>
<th>Energy score</th>
<th>Energy-bounded learning</th>
<th>Additional class with noise data</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1 / N1 / N2</td>
<td>0.572</td>
<td><strong>0.779</strong></td>
<td>0.534</td>
<td>0.628</td>
<td>0.683</td>
</tr>
<tr>
<td>N1 / N1 / N3</td>
<td>0.589</td>
<td><strong>0.769</strong></td>
<td>0.591</td>
<td>0.587</td>
<td>0.687</td>
</tr>
<tr>
<td>N1 / N1 / N4</td>
<td>0.475</td>
<td><strong>0.734</strong></td>
<td>0.435</td>
<td>0.586</td>
<td>0.609</td>
</tr>
<tr>
<td>N2 / N2 / N1</td>
<td>0.493</td>
<td><strong>0.583</strong></td>
<td>0.483</td>
<td>0.415</td>
<td>0.443</td>
</tr>
<tr>
<td>N2 / N2 / N3</td>
<td>0.575</td>
<td><strong>0.694</strong></td>
<td>0.566</td>
<td>0.551</td>
<td>0.574</td>
</tr>
<tr>
<td>N2 / N2 / N4</td>
<td>0.441</td>
<td><strong>0.683</strong></td>
<td>0.442</td>
<td>0.541</td>
<td>0.551</td>
</tr>
<tr>
<td>N3 / N3 / N1</td>
<td>0.445</td>
<td>0.497</td>
<td>0.412</td>
<td><strong>0.521</strong></td>
<td>0.368</td>
</tr>
<tr>
<td>N3 / N3 / N2</td>
<td>0.418</td>
<td><strong>0.520</strong></td>
<td>0.407</td>
<td>0.519</td>
<td>0.414</td>
</tr>
<tr>
<td>N3 / N3 / N4</td>
<td>0.528</td>
<td><strong>0.720</strong></td>
<td>0.523</td>
<td>0.541</td>
<td>0.557</td>
</tr>
<tr>
<td>N4 / N4 / N1</td>
<td>0.367</td>
<td><strong>0.433</strong></td>
<td>0.379</td>
<td>0.311</td>
<td>0.262</td>
</tr>
<tr>
<td>N4 / N4 / N2</td>
<td>0.429</td>
<td><strong>0.537</strong></td>
<td>0.424</td>
<td>0.470</td>
<td>0.393</td>
</tr>
<tr>
<td>N4 / N4 / N3</td>
<td>0.420</td>
<td><strong>0.519</strong></td>
<td>0.429</td>
<td>0.367</td>
<td>0.389</td>
</tr>
</tbody>
</table>

5.3 Performance of noise exposure using on-site noise and other-site noise

In this experiment, we used different noise datasets for the train, validation, and test. The first columns of Tables 7 and 8 indicate the noise used in the train and validation step, while the other columns represent the macro F1 scores when tested by machine sounds mixed with noises indicated on the first cell of the corresponding column. The noise environment of noise-only data for the test was the same as that of machine sound data for the test. The diagonals of Tables 7 and 8 (grey color) hence indicate the results of on-site noise exposure, and off-diagonals show the results of other-site noise exposure. The test with the toy car and toy train dataset demonstrates that the on-site noise is more beneficial than the other-site noises for the noise exposure. One exceptional case (test with N3) exists in the toy car test, but in general, the model exposed to similar types of noises is more robust to the noisy data. This may seem to be an obvious conclusion but also stresses that we can robustly train a model by exposing it to in-situ noises measured in the factory without the need of collecting various noise data.
Table 7 – Performance comparison of different models trained by noise exposure with on-site (ONE) and other-site noises (toy car data).

<table>
<thead>
<tr>
<th>Noise data used for noise exposure</th>
<th>Machine sound with N1</th>
<th>Machine sound with N2</th>
<th>Machine sound with N3</th>
<th>Machine sound with N4</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>0.841</td>
<td>0.865</td>
<td>0.985</td>
<td>0.925</td>
</tr>
<tr>
<td>N2</td>
<td>0.773</td>
<td>0.875</td>
<td>0.988</td>
<td>0.919</td>
</tr>
<tr>
<td>N3</td>
<td>0.701</td>
<td>0.778</td>
<td>0.980</td>
<td>0.889</td>
</tr>
<tr>
<td>N4</td>
<td>0.704</td>
<td>0.792</td>
<td>0.982</td>
<td>0.930</td>
</tr>
</tbody>
</table>

Table 8 – Performance comparison of different models trained by noise exposure with on-site (ONE) and other-site noises (toy train data).

<table>
<thead>
<tr>
<th>Noise data used for noise exposure</th>
<th>Machine sound under N1</th>
<th>Machine sound under N2</th>
<th>Machine sound under N3</th>
<th>Machine sound under N4</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>0.755</td>
<td>0.761</td>
<td>0.801</td>
<td>0.752</td>
</tr>
<tr>
<td>N2</td>
<td>0.732</td>
<td>0.777</td>
<td>0.776</td>
<td>0.778</td>
</tr>
<tr>
<td>N3</td>
<td>0.668</td>
<td>0.705</td>
<td>0.845</td>
<td>0.743</td>
</tr>
<tr>
<td>N4</td>
<td>0.625</td>
<td>0.714</td>
<td>0.743</td>
<td>0.796</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS

In this paper, we exposed a DNN model to various noises to build a noise-robust classifier and detect noise-only data. We compared five different outlier exposure methods for this objective. With noise exposure, the classifier was trained to classify machine conditions while producing a uniform predicted probability for noise-only data. The comparison of outlier exposure methods shows that noise exposure has the best or second-best performances irrespective of the types of machine sounds and noises. We also tested noise exposure using different on-site and other-site noises. For the on-site noise test, the same noise dataset split for train and test was employed, whereas different datasets were used for the other-site noise test. Results show that noise exposure with on-site noise outperforms the other-site noise test. These results demonstrate that the noises measured in the same environment where the machine operates can be used as a noise exposure dataset and can improve the robustness of fault classifiers.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (Ministry of Science and ICT) (No. NRF-2020M2C9A1062710) and supported by the BK21 Four program through the National Research Foundation (NRF) funded by the Ministry of Education of Korea.

REFERENCES

Impact of temporal Context on Sound Event Detection in Complex Head Related Impulse Response Acoustic Scenes.

Adeoluwasale ADEWUSI¹; Klaus OBERMAYER²

¹,²Department of Electrical Engineering and computer Science Technische Universität, Berlin, Germany

ABSTRACT

Temporal context is an inherent property of audio event that describes the energy activity within an audio sample over entire duration of the sample. The perception of sound sources in polyphonic acoustic scenes is hinged on the temporal and spatial signatures of foreground and background sources in these scenes. Presently, the exact temporal size of audio signal that is required to predict an audio event in acoustic scene is not known. Thus, this work attempts to model the sufficient length of sound event data that sound classifiers can access towards robust audio event detection in complex acoustic scenes. Complex binaural acoustic scenes are simulated by convolving audio from sound events classes with head-related room impulse responses under diverse acoustic conditions such as source azimuths and signal to noise ratio. Furthermore, deep learning-based sound classifiers are trained with varying temporal window lengths of scene data using multi-conditional training approach. Moreover, the influence of acoustic conditions on sound classification models' strength with different temporal context lengths are evaluated. The results reveal how temporal context length and these acoustic scene conditions affect the generalization performance of sound event classifiers in complex audio scenes.

Keywords: Temporal context, sound event detection, complex acoustic scenes

1. INTRODUCTION

Acoustic scene often contains multiple sources emitting sounds in an arbitrary temporal sequence, such that the listener ability to localize and detect specific sound of interest is largely influenced by the temporal distribution of the stimuli. From the studies in auditory scene analysis, the perception and separation of sounds (1, 2) the acoustic scene information that enters human ears is a stream of all available environmental sources that are emitting sounds.

Acoustic scenes differ greatly in their characteristics, particularly the foreground sound events and the background noise, thus the total length of time required for human auditory system for perception and identification of sound events in these scenes vary from one scene to another. For example, the time span that is required to comprehend a moving aircraft in an airplane acoustic environment will be less as a result of its loud and discernible background noise. In contrast, longer time of listening is required to aggregate enough
acoustic information to be able to clearly differentiate a cafe from a busy street if both have similar babble noise signature (3).

Major efforts in sound event detection domain have been majorly concentrated on the general modeling of sound event detection tasks. Some of these tasks include audio-tagging (4,5), rare audio detection (6) and polyphonic sound event detection (7, 8).

However, little or no effort has been directed towards the determination of sufficient temporal context length of acoustic scene data that sound detection models can access for optimal performance in a complex polyphonic acoustic scene condition. The question of ‘what is the sufficient size of context information for sound event recognition’ is left unanswered. Furthermore, the influence of acoustic conditions in non-stationary scenes on sound events classifier performance is yet to be clearly understood.

In this work we, investigate actual length of acoustic scene sample that a sound event classifier can access towards optimal sound event recognition in the presence of dynamic acoustic conditions such as multiple energy ratios and increasing number of co-occurring sources. Besides that, we examine the influence of these conditions on the generalization strength of the classifiers under multi conditional training scheme. Since the goal of sound event detection task is to build a machine listening system for environmental sounds that exhibits comparable performance with human ears. Thus, we explore the use of binaural feature representations from acoustic scene as input features. The binaural signals are simulated using a binaural robotic system.

In recent times various deep network architectures have been developed and applied to sound event detection tasks such as Convolution Neural Networks (9, 10) which have strong feature extraction capability, due to its translation invariant property. The Recurrent Neural Networks (RNNs) (11,12) can implicitly model temporal contexts by aggregating information over several time windows, thus, it can theoretically process infinite context length of information. The advantages of these two deep architectures have been explored and this has resulted in Convolution Recurrent Neural Networks (CRNNs) in sound events detection tasks (13,14).

2. BACKGROUND

2.1 Problem Formulation

The sound event detection in polyphonic scenes is the prediction of temporal stamps of each sound event in the acoustic scene sample and their associated semantic label. Frame level acoustic features are first extracted for each time frame \( t \) from the acoustic data spectrogram to yield a set of feature vector \( x_t \in \mathbb{R}^F \) such that \( F \in \mathbb{N} \) denotes the sum of features in each frame. Then, the main classification task is to evaluate the likelihoods of \( P(y_t(k)|x_t, \theta) \) for sound event classes \( k = 1,2,\ldots,K \) that is present in each frame \( t \), where \( \theta \) denotes the classifier parameters which are learned with deep networks. Then, the probabilities of the event activity in successive frames are binarized over a constant value by thresholding in order to successfully compute the predictions activity \( \hat{y}_t \in \mathbb{R}^K \).
2.2 Temporal Context and Spatio-Temporal Learning with 3D Convolution

Temporal context is an inherent attribute of acoustic sequence. It has been shown in speech recognition tasks that long temporal context improves the accuracy of automatic speech recognition systems (15) and phoneme recognition algorithms (16). Moreover, it is established in (17) that audio features evolve over different time scales in auditory scene and these features cannot be processed in isolation, rather perception is achieved by integrating information over different time scale. Thus, models can access longer temporal context are required for acoustic event processing.

The 3D-Convolution has been used in computer vision tasks (18, 19), particularly in action recognition (20) where it is applied to jointly aggregate information that are related to objects, scenes and actions in video clips. Furthermore, spatio-temporal convolution has attracted attention in sound event detection tasks in which 3D Convolution has been adopted in Acoustic scene classification (24), and multichannel sound event detection (22,23). In these tasks, 3D convolution jointly learns features from both spatial and temporal data dimensions by capturing the inter-dependence among the three-dimensional signals.

3. METHODOLOGY

3.1 Model Description

Illustrated in Figure 1 is the proposed model architecture for investigation of the effect of temporal context length in sound event detection systems. The model is a variant of Convolution Recurrent Neural Network (CRNN), and it consists of two separate Convolution architectures (3D and 2D convolutions), with a layer of recurrent networks. The recurrent architecture as used here is a stack of Long Short-Term Memory Networks (LSTM) (34).

![Figure 1 – Proposed model architecture to investigate the effect of temporal context length in sound event detection task](image-url)
The concatenated input representation of the acoustic scene data first sliced into the respective low-level features, which are rate maps and Amplitude Modulated Spectrogram (AMS). Thereafter, 3D convolution is applied on the AMS features to learn of inter-channel feature (spectro-temporal features) while spatial convolution operation is implemented on the rate maps features. The learned features are concatenated along the temporal dimension per time step into the LSTM networks for temporal context modeling. The output from the last LSTM layer is connected to layers of feed-forward networks and output layer prediction nodes. We named our proposed model CLDNN3D.

3.2 Data and Scene Creation

The sound data is sourced from the NIGENs database (25). This audio repository consists of thirteen strongly labeled audio event files and a general class from fourteen sound classes. The general class consists of random environmental sounds other than those in the other thirteen sound types. This class is included to serve as a negative example to the classifiers during training because it exhibits a wide variability.

3.3 Binaural Scene Synthesis

Point sound sources are convolved with Head Related Impulse Response (HRIR) measured with the KEMAR head binaural simulator in Two Ears (26) to generate binaural auditory scenes data. The binaural simulator synthesizes ear signals through the convolution of mono audio signals with HRIR signals. The resulting ear signals are combined at specific Signal to Noise Ratio (SNR) as in (27). The SNR is varied in each scene instance to reflect scene complexity defined by noise level.

The average length of each simulated auditory scene-instance data is 30-seconds long. Additionally, the number of sound sources in each scene sample is varied from one to four sound sources. For instance, Figure 2 shows an acoustic scene instance where three sources are simultaneously present (three-source scene). This data creation method allows us to modulate each acoustic condition (number of sources and signal to noise ratio) such that their individual effect on the classifier performance can be studied.

Figure 2 – Acoustic scene simulation with the head: three sources with the target source at $45^0$ while the second and the third sources are displaced at azimuths of $90^0$ and $135^0$, respectively to the head.

The target sound event in each scene instance is emitted from sound files of a particular sound category from the thirteen classes while the ‘distractor’ sources are emitted from other sound classes other than the target class (including the general class). The scene parameters are varied in order to create a complex auditory scenario and difficult listening situation that is close to what is available to human ears in a real acoustic environment. These variations favor multi-conditional training approach as used in this work.
towards increasing the model’s robustness in uncertain acoustic conditions.

The training scene data parameters are highlighted in Table 1. Here the number of sources is varied for each scene instance while the co-occurring sound events in each scene are combined at different Signal to Noise Ratio (SNR).

<table>
<thead>
<tr>
<th>Define Scene parameter</th>
<th>Values used</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR (dB)</td>
<td>-20, -10, 0, 10, 20</td>
</tr>
<tr>
<td>Number of sources</td>
<td>Varied between one to four</td>
</tr>
</tbody>
</table>

A total of 80 scenes are defined for each of the six folds of the training data while 168 scenes are specified for each of the two folds of the test data. The layout of the test scene data is shown in Figure 3. The minimum number of the sources used in the acoustic scene is two.

![Figure 3](image)

Figure 3 – Arrangement of the test scenes with the head located at the middle: (a) two-source scene (b) three-source scene (c) four-source scene

The layout in Figure 3 is such that the head is located at the middle and the scenes are confined to a minimum of two sources with spread angles that are larger than 0°. The sources are arranged in the order of increasing number of co-occurrences. Each arc represents an auditory scene sample, with a black dot representing a distractor source while a target source is indicated by an orange dot. The neighboring scenes have the same number of sources.

3.4 Feature Representations and scene Data Labeling

AMS (28) and Ratemaps (29) are the two feature representations processed from the resulting binaural scene signals. The AMS is three dimensional biologically inspired feature closely related to inner ear signals in humans. The representation is computed by spatial-temporal and the bark scale decomposition operations. The output signal is averaged into an overlapping frame length of 20 ms at 10 ms shift to yield 128 features vectors per frame. Ratemaps is similar in derivation to AMS but two-dimensional feature representation. The output is 32 rate maps features per frame. The final output feature is the concatenation of both AMS and rate maps along the feature dimension such that a total of 160 feature vectors per frame is produced.

Two labeling approaches are adopted as ground truth annotations in this task. These are the segment-label(block) and the frame-labels. In segment-based label, a target sound event is predicted to be present if its activity spans 75% of 500 ms block length of the scene data. In the frame-based label, the presence of a
sound is determined at the time of a frame (instantaneously). Given that the frame length is 20 ms the sound event is said to be active during the entire length of the frame and the duration of the signal.

3.5 Model Training and Optimization

The number of convolution units per layer in this model is \{8, 16, 32\} while the kernel size of \((3, 3, 3)\) and \((3, 3)\) is applied across all the layers for both 3D and 2D convolutions, respectively. The number of LSTM layers per model is randomly sampled from \{3, 4, 5\} and the number of fully connected (FC) layers equally sampled from \{1, 2\}. Moreover, the total number of neurons is selected from \((500, 1200)\) with sampling factor of \((0.75, 0.25, 0.5)\) as a function of recurrent layer depth and this is distributed between LSTM and FC. A data batch size is 128 and an initial learning rate of 0.001 is chosen which is later adjusted during hyper-parameter optimization. Several models are trained with varying temporal window lengths of the input data. For instance, in the segment-based labeling, the input temporal context length into the sound classifier is varied between \((500 \text{ms})\) to \(19\text{seconds}\) \((\text{s})\) while in instantaneous labeling, models are trained with context length from \(50\text{ms}\) to \(19\text{s}\).

A six-fold cross partial cross validation strategy is implemented with data sub-sampling for each training epoch. The sub-sampling technique ensures that all scene-instances selected from each training fold truly reflects even distribution of all the sound event classes at every epoch. Variational recurrent dropout \((30)\) is applied to the recurrent layers, binary cross-entropy is adopted as the objective function, and Adam optimizer \((31)\) is the learning algorithm. Moreover, early stopping criterion is invoked during model training such that the training terminates if the training performance does not improve after five successive epochs. Balanced Accuracy (BAC) is used as the objective measure of choice during cross validation. This is expressed in \((32)\) as:

\[
BAC = \frac{1}{2} \left( \frac{TP}{TP+FP} + \frac{FN}{TN+FN} \right)
\]

\[
= \frac{1}{2} (\text{Sensitivity} + \text{specificity})
\]

where \(TP, TN, FP\) and \(FN\) are True Positive, True Negative, False Positive and False Negative, respectively.

The BAC penalizes the sensitivity and the specificity values in order to mitigate the effect of data imbalance on the prediction results.

For benchmarking, two classifiers are trained as baselines in this work. First, a LSTM model with feed-forward networks called \((\text{LDNN})\) and second, a variant of Convolution Recurrent Neural Network \((\text{CLDNN2D})\) in which only spatial features are extracted from both feature representations \((\text{AMS} \text{ and ratemaps})\) of the scene data.

4. RESULTS AND DISCUSSION

The results are pooled across the test scenes and sound event classes independently in order to observe the relationship between the model performance and the scene parameters. It should be noted that segment- and frame-labels are indicated by \((\text{SL})\) and \((\text{FL})\), respectively, in the results. Furthermore, the results are averaged over the test scenes, test files and sound event classes in order to isolate the best and the worst model performances. Obtained results are presented in what follows.
4.1 Overall Balanced Accuracy and Temporal Context Length

The overall models' performances with respect to the temporal context length while isolating the scene parameters are shown in Figure 4.

The results are aggregated across all the test scenes, test files. The lines depict the temporal context lengths computed at 95% confidence interval.

The best model in this task achieved 89.5% BAC utilizing a temporal context length of 19 s in the segment-based models depicted in Figure 4(a). This is explainable because the sound event presence is predicted over the entire block length of scene sample. On the other hand, the frame-based models shown Figure 4(b) are not able to access more than a temporal context length of 15 s. This is because, the predictions are not done over a smooth block of features unlike its block-based counterparts, rather on frame-by-frame basis. The frame-based models are thus difficult to train and therefore unable access as much temporal information as the segment label models.

It is obvious that the CLDNN3D models outperform all other models by accessing the longest temporal context of 19 s. This suggests that spatio-temporal features that are jointly learned by the model had captured important inter-channel and intra-channel details that are pertinent to detection in the audio scene sample. The LDNN model is not able to accumulate context information beyond 10 s perhaps this is attributed to the fact that LSTM networks are difficult to train over a long temporal window and that spatial features in the acoustic scene data are not exploited. Interestingly, it displays a higher performance over CLDNN2D at temporal context length of 10 s.

For the frame-based models, a performance trend that is similar the segment based-models are observed with the CLDNN3D achieving the highest prediction performance. Remarkably, the LDNN model exhibited a higher performance over CLDNN2D at lower temporal context lengths up to 100 ms. The reason for this performance behavior is not clear, perhaps the CLDNN2D is not able to extract sufficient feature at this temporal window length for the classification task.

4.2 Effect of Signal-to-Noise Ratio on Classifier Performance

Figure 5 illustrates the variation of BAC performance of the various sound event classifiers with the scene signal-to-noise ratio.
Segment-based models

Frame-based models

Figure 5 – Variation of BAC performance of the various sound event classifiers with the scene signal-to-noise ratio. Results are aggregated over all the test scenes and test files. The lines plotted at 95% confidence interval

A close look at Figure 5(a) reveals that the CLDNN3D (19 s) performs best as the noise level in the acoustic scene progressively increases. This means that given a difficult acoustic scene that is characterized with high noise distribution, the model can accumulate temporal information towards a potential gain in detection performance. This result describes how human beings attend to sounds of interest even in a noisy scenario by listening more to gain more temporal information about the sounds of interest.

From the same Figure 5(a), it is worth noting that the LDNN slightly outperformed CLDNN3D and CLDNN2D at low noise level (−20 dB), this is rather strange, but it may be that the features that are learned by CLDNN models are not useful for sound detection at very low noise energy. Additionally, the best LDNN_10s model seems to show a clear performance over CLDNN2D_15s up till 0 dB, while CLDNN2D_15s performance increases marginally over the LDNN models at higher noise level in the scene.

For the models with frame-based label in Figure 5(b) an improved performance in CLDNN3D is observed with 10 dB gain from the easy scenes (low noise level) to very difficult scene. Strangely, the LDNN_10s model exhibits a superior performance to CLDNN2D_15s up till 10dB. This result is unexpected because CLDNN2D should have utilized invariant features to improve the model’s performance. The worst performance is exhibited by CLDNN2D_0.05s.

4.3 Effect of Increasing number of sources (Scene Polyphony) on the Classifier’s Performance

Figure 6(a) displays the performance trend for segment-based classifiers as the number of emitting sources are increasing in the acoustic scene so that there are more impediments to the detection of target sound events. The CLDNN3D models show clear performance over the LDNN and CLDNN2D models as the level of scene polyphony (distractor sources) Though the performance degrades as the scene becomes more difficult due to the increasing level of acoustic scene polyphony (from left to right), nevertheless CLDNN3D exhibited a marked performance in this range. The results here further confirm the ability of humans to give attention to sounds of interest in the presence of multiple competing sources by listening more. Furthermore,
CLDNN2D_15s shows a clear performance over its LDNN_10s counterpart in an acoustic scene where two sound events co-exist, while the model performance became marginal as the level of scene polyphony increases. This is quite different from the SNR situation where the LDNN model exhibit a distinct performance at lower SNR. This means that the number of sources in the acoustic scene and the scene SNRs can be considered as mutually exclusive scene parameters.

The performance margin between the best and the worst models particularly for the frame-based models Figure 6(b) are not significant, which may imply that temporal contexts does not help much in this regard. However, CLDNN2D_SL_0.5s shows a clear deviation in performance (worse) compared with other models.

Figure 6 – BAC performance of the various sound event classifiers over increasing number of sources in the scene. Results are aggregated over all the test scenes and test files.

4.4 Sound Event Category Detection Performance

We analyze the classifiers performances based on the class-wise detection of all the sound event categories in dataset.

<table>
<thead>
<tr>
<th>Sound event categories</th>
<th>CLDNN_3D_1 19s</th>
<th>CLDNN_3D_1 0.5s</th>
<th>CLDNN_2D_15s</th>
<th>CLDNN_2D_0.5s</th>
<th>LDNN_10s</th>
<th>LDNN_0.5s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm</td>
<td>0.73</td>
<td>0.73</td>
<td>0.76</td>
<td>0.67</td>
<td>0.66</td>
<td>0.68</td>
</tr>
<tr>
<td>Phone</td>
<td>0.78</td>
<td>0.74</td>
<td>0.76</td>
<td>0.74</td>
<td>0.73</td>
<td>0.7</td>
</tr>
<tr>
<td>crash</td>
<td>0.8</td>
<td>0.75</td>
<td>0.65</td>
<td>0.65</td>
<td>0.79</td>
<td>0.73</td>
</tr>
<tr>
<td>Piano</td>
<td>0.79</td>
<td>0.84</td>
<td>0.84</td>
<td>0.73</td>
<td>0.77</td>
<td>0.77</td>
</tr>
<tr>
<td>Engine</td>
<td>0.83</td>
<td>0.81</td>
<td>0.75</td>
<td>0.74</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>Fire</td>
<td>0.87</td>
<td>0.83</td>
<td>0.79</td>
<td>0.68</td>
<td>0.86</td>
<td>0.84</td>
</tr>
<tr>
<td>Baby cry</td>
<td>0.84</td>
<td>0.83</td>
<td>0.84</td>
<td>0.8</td>
<td>0.86</td>
<td>0.82</td>
</tr>
<tr>
<td>Footsteps</td>
<td>0.87</td>
<td>0.85</td>
<td>0.85</td>
<td>0.81</td>
<td>0.86</td>
<td>0.82</td>
</tr>
<tr>
<td>Knock</td>
<td>0.9</td>
<td>0.86</td>
<td>0.9</td>
<td>0.84</td>
<td>0.77</td>
<td>0.8</td>
</tr>
<tr>
<td>Dog</td>
<td>0.89</td>
<td>0.88</td>
<td>0.89</td>
<td>0.86</td>
<td>0.92</td>
<td>0.85</td>
</tr>
<tr>
<td>Scream</td>
<td>0.83</td>
<td>0.84</td>
<td>0.83</td>
<td>0.79</td>
<td>0.82</td>
<td>0.8</td>
</tr>
<tr>
<td>Male speech</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.88</td>
<td>0.95</td>
<td>0.89</td>
</tr>
<tr>
<td>Female speech</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.88</td>
<td>0.94</td>
<td>0.88</td>
</tr>
</tbody>
</table>

A cursory look at the Table 2 shows that certain sound event categories profited from the long temporal context length of the input signal. For instance, the classes of ‘phone’, ‘crash’ and ‘engine’ use a context length of 19s with CLDNN_3D exhibit an average of 0.08, 0.15 and 0.02 performance gain respectively with respect to the worst performing model in the tabulated results. Furthermore, the sound class ‘footsteps’ has a performance gain of 0.1 with same temporal context length (19s). It is noteworthy that the performance gain
in the footsteps class over the increasing context length is significant, this is because this sound category is an impulse sound which requires accumulation of cues over a long period of time for its detection. It is obvious that the ‘female and the ‘male speech,’ ‘dog bark’ and ‘baby cry’ utilized the best LDNN_10s to achieve the best detection performance results.

This is probably because these sound categories inherently have sufficient long temporal distributions and distinctive features so that the models do not necessarily need to aggregate many features or ‘listen’ longer to detect these sounds. Therefore, it is not clear how the performance gain over temporal context size impacts the detection among different sound categories in a dynamic acoustic scene. It will be safe to infer that, profiting from long context is dependent on the characteristics of specific sound event in consideration.

Table 3: BAC performances with respect to specific sound event category detection (Frame-label)

<table>
<thead>
<tr>
<th>Sound event categories</th>
<th>CLDNN_3D_15s</th>
<th>CLDNN_3D_0.05s</th>
<th>CLDNN_2D_15s</th>
<th>CLDNN_2D_0.05s</th>
<th>LDNN_10s</th>
<th>LDNN_0.05s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm</td>
<td>0.72</td>
<td>0.76</td>
<td>0.67</td>
<td>0.76</td>
<td>0.68</td>
<td>0.72</td>
</tr>
<tr>
<td>Phone</td>
<td>0.74</td>
<td>0.69</td>
<td>0.71</td>
<td>0.65</td>
<td>0.72</td>
<td>0.7</td>
</tr>
<tr>
<td>crash</td>
<td>0.76</td>
<td>0.74</td>
<td>0.73</td>
<td>0.7</td>
<td>0.77</td>
<td>0.69</td>
</tr>
<tr>
<td>Piano</td>
<td>0.82</td>
<td>0.8</td>
<td>0.75</td>
<td>0.77</td>
<td>0.8</td>
<td>0.77</td>
</tr>
<tr>
<td>Engine</td>
<td>0.83</td>
<td>0.78</td>
<td>0.81</td>
<td>0.77</td>
<td>0.8</td>
<td>0.76</td>
</tr>
<tr>
<td>Fire</td>
<td>0.88</td>
<td>0.83</td>
<td>0.85</td>
<td>0.77</td>
<td>0.89</td>
<td>0.79</td>
</tr>
<tr>
<td>Baby cry</td>
<td>0.83</td>
<td>0.79</td>
<td>0.83</td>
<td>0.87</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>Footsteps</td>
<td>0.81</td>
<td>0.8</td>
<td>0.79</td>
<td>0.77</td>
<td>0.9</td>
<td>0.8</td>
</tr>
<tr>
<td>Knock</td>
<td>0.88</td>
<td>0.82</td>
<td>0.8</td>
<td>0.86</td>
<td>0.88</td>
<td>0.82</td>
</tr>
<tr>
<td>Dog</td>
<td>0.85</td>
<td>0.87</td>
<td>0.84</td>
<td>0.86</td>
<td>0.83</td>
<td>0.75</td>
</tr>
<tr>
<td>Scream</td>
<td>0.83</td>
<td>0.8</td>
<td>0.78</td>
<td>0.75</td>
<td>0.82</td>
<td>0.79</td>
</tr>
<tr>
<td>Male speech</td>
<td>0.92</td>
<td>0.88</td>
<td>0.8</td>
<td>0.87</td>
<td>0.94</td>
<td>0.85</td>
</tr>
<tr>
<td>Female speech</td>
<td>0.89</td>
<td>0.87</td>
<td>0.85</td>
<td>0.85</td>
<td>0.92</td>
<td>0.84</td>
</tr>
</tbody>
</table>

The trend in Table 3 is like what is observed in the block-based performances except that the ‘alarm’ utilized the lowest of 0.05s with CLDNN_2D and CLDNN_3D models to achieve the best performance while there is a tie in the performance of the CLDNN_3D_19s and LDNN_10s in the detection of ‘scream class’. The probable cause of this strange behavior in the performances of these two different models is hard to explain, it may not be unconnected to the model training.

5. CONCLUSION

The effect of temporal context size on sound event model robustness in head related impulse response acoustic scene using multi conditional training approach has been investigated. The study reveals that models trained with longer temporal context information exhibit higher detection performance gain. The is because classifier can aggregate more features about the target sound events over time.

In addition to that, robustness of sound classifiers in difficult acoustic scenes is enhanced by jointly learning spatio-temporal features through 3D convolution.

Furthermore, multi-conditional training of sound detection models under complex acoustic scene conditions has proven to appreciably improve the classifier’s generalization strength particularly in a difficult acoustic scene condition under intense noise condition and the presence of multiple co-occurring sources. It will be interesting to investigate the influence of sound segmentation on the model’s detection performance in the future.
REFERENCES
22 Adavanne S, Politis A, Virtanen T. Multichannel Sound Event Detection Using 3D Convolutional Neural Networks for Learning Inter-channel Features.
23 Scenes A. 3D CONVOLUTIONAL RECURRENT NEURAL NETWORKS FOR BIRD SOUND DETECTION Technical Report Science and Engineering Faculty, Queensland University of Technology. 2018;3–6.
29 May T, Kohlrausch A. Simultaneous Localization and Identification of Speakers in Noisy and Reverberant Environments. (c):2121–6.
Sound source localization and detection based on parameter transfer learning

Xinghao Sun\textsuperscript{1,2}, Mengzhen Ma\textsuperscript{1,2}, Ying Hu\textsuperscript{1,2}

\textsuperscript{1}Department of Information Science and Engineering, Xinjiang University, Urumqi 830000, China
\textsuperscript{2}Key Laboratory of Signal Detection and Processing in Xinjiang, China

ABSTRACT

Sound source localization and detection is a joint task of identifying the presence of individual sound events and locating the sound sources in space. In order to promote the combination of two different tasks, we propose a sound source localization and detection method based on parameter transfer learning. Firstly, to solve the problem of uneven distribution of time-frequency dimension features, we propose a time-frequency dimension feature extractor, which extracts the features of time dimension and frequency dimension respectively and superimposes them together. Secondly, in order to promote feature interaction between different resolutions, we propose a refined feature learning module that can fuse features with different resolutions. Then, in order to reduce the training and reasoning time, we explore a temporal context representation module for learning temporal context information. The time context indicates that the module has better global feature capture capability and better parallelism capability compared with the recurrent neural network and gated recurrent unit. Finally, the rationality of our proposed model was verified by ablation experiments, and the effectiveness of our proposed model was verified by comparison with the best methods.

Keywords: Sound source localization, Sound event detection

1 INTRODUCTION

Sound event localization and detection (SELD) is a rapidly developing research field, which aims to analyze and locate acoustic events in urban and natural environments. The sound event detection in this task is different from the audio tagged in that the sound event detection also needs to estimate the start and end time of the sound. Sound event location and detection has a profound impact on many applications\cite{1}. For example, man-machine interaction\cite{2}, bioacoustic monitoring\cite{3}, smart cities\cite{4}, and timely warning of dangerous acoustic signals\cite{5}.

The research methods of sound event detection (SED) can be divided into two categories, which are based on parameter-based method and deep learning-based method. In recent years, neural network-based methods have been proved to be particularly effective for acoustic event detection tasks. SED requires recognizing not only the presence of each sound event but also the boundaries in the timeline of each sound event. Most of the initial neural network-based methods adopt the full connection layer as the neural network architecture. Recently, due to the great success of convolutional neural network (CNN) in the field of image recognition, CNN has become the mainstream framework in the field of speech recognition\cite{6}. Another mainstream neural network framework is recurrent neural network, which can learn information in long time series and has a good effect on context modeling, and is very suitable for sound event detection tasks\cite{7}. At present, the effective neural network architecture are the convolutional recurrent neural network and Transformer, and has a good improvement on sound event detection task.

The objective of sound source localization (SSL) is to determine the spatial position of sound source. It plays an important role in robot hearing, speech enhancement, sound source separation and acoustic visualization. Similar to the method based on sound event detection, sound source location mainly adopts two methods: parameter-based method and deep learning-based method. SSL based on parameter method include: signal parameter estimation based on arrival time difference (TDOA)\cite{8}, steering response power (SRP)\cite{9}, multi-signal
classification (MUSIC)[10] and rotation invariance technology (ESPRIT)[11]. These methods vary in algorithmic complexity, array geometry constraints, and model assumptions for acoustic scenarios. The sound source localization method based on deep learning has good generalization under different reverberation and noise levels. A series of studies on addressing direction of arrival estimation using deep neural networks. The results show that this method has a good application prospect and can be compared with parametric method.

In real application, not only the type of sound event, but also the specific location of the sound event. Considering this practical application, it is reasonable to combine sound event detection with estimating their respective relevant spatial positions, identifying the type and time information of the sound. In this paper, the log-linear spectrogram and normalized sound intensity vector are used as input features to realize the joint training of SELD. In this paper, we propose a SELD method based on parameter transfer learning. The SELD task is divided into two stages: the sound event detection stage and the sound source location stage, which correspond to the SED branch and SSL branch in the model respectively. In the training process, the SED branch is trained first, and then the learned feature weight parameters are transferred to the SSL branch. The SSL branch fine-tunes the feature weight parameters of the transfer and trains SSL using the output of the SED branch as a mask. Experimental results show that the performance of sound source location and sound event detection can be improved simultaneously by using this method. The performance of this method is significantly better than that of baseline method.

The organization of this paper is as follows: the second section will introduce input features of the proposed network and each specific part of the network. The third section will analyze the experimental results, and the last section is the summary of this paper.

2 THE PROPOSED METHOD

We propose a network based on parameter transfer learning as shown in Figure 1, which achieves great performance to deal with SELD in noisy and reverberant scenes. For the multi-channel audio, we adopt log-linear spectrum and normalized sound intensity vector as input features. These two features have accurate time-frequency mapping between signal power and source direction cues, which is the key to solve the problem of overlapping sound sources. The extracted features will be sent to the sound source location branch and the sound event detection branch for training. The network of the two branches is exactly the same in the way of composition. Because the more close to the characteristics of the input layer can save a lot of original fea-

Figure 1. The overall network architecture
tures, so we will sound event detection branch of the weight parameters of the two modules before migrating to beamformer branch to continue training. Event detection after SED branch output results will pass a threshold settings, choose the active sound, and as a masking to influence the output of the SSL branch. The SSL branch outputs only the spatial coordinates of active sound events.

2.1 Input features
In this paper, the samples are First-Order Ambisonics (FOA). \( M \) is the number of microphones and \( L \) is the number of sound sources. The short-time Fourier Transform (STFT) signal observed by arbitrary geometric \( M \)-channel microphone array in time-frequency domain is given as follows:

\[
X(t, f) = \sum_{i=1}^{L} S_i(t, f)H(f, \theta_i) + V(t, f) \in \mathbb{C}^M
\]  

(1)

Where \( t \) and \( f \) are time index and frequency index respectively, \( S_i \) is the \( i \)th sound source signal; \( H(f, \phi, \theta) \) for the \( i \)th source direction of arrival \((\phi_i, \theta_i)\) corresponding to the frequency domain of the steering vector. Where \( \phi \) and \( \theta \) are azimuth and elevation angles respectively; \( V \) is a noise vector. For mobile source \( \phi = \phi(t) \) and \( \theta = \theta(t) \) is a function of time. For simplicity, in some equations \( \phi_i \) and \( \theta_i \) omitted variable time.

The log-linear spectra and normalized sound intensity vectors of FOA dataset are calculated and served as input features. The multichannel log-linear spectra are popular for SED task. Log-linear spectra can be calculated as

\[
\text{LINSPEC}(t, f) = \log(|X(t, f)|^2) \in \mathbb{R}^{M \times T \times F}
\]  

(2)

Where \( T \) is the number of time frames and \( F \) is the number of frequency.

The normalized sound intensity vectors are compressed to guarantee the dimension of normalized sound intensity vector is the same as that of the log-linear spectra. The FOA consists of four channels, and its short-time Fourier transform (STFT) outputs \( W_{f,j}, X_{f,j}, Y_{f,j}, Z_{f,j} \). Where \( W_{f,j} \) is 0-th order of spherical harmonics and \( X_{f,j}, Y_{f,j}, Z_{f,j} \) correspond to 1st order of spherical harmonics. Here, \( f \) and \( t \) are indexes of frequency bin and time frame, respectively. The normalized sound intensity vectors are calculated from the 4-channel spectra as:

\[
I_{f,j} \propto \Re(W_{f,j}^*h_{f,j}) = [I_{X_{f,j}}, I_{Y_{f,j}}, I_{Z_{f,j}}]^	op
\]  

(3)

\[
h_{f,j} = [X_{f,j}, Y_{f,j}, Z_{f,j}]^	op
\]  

(4)

where \( \Re(\cdot) \) denotes the real-part of complex numbers, \( * \) is the conjugate of complex numbers. In addition, the direction of normalized sound intensity vector is necessary for DOA estimation. Then the normalized sound intensity vectors are normalized as:

\[
\tilde{I} = \frac{I_{f,j}}{I_{f,j}}
\]  

(5)

2.2 Time-frequency dimension feature extraction module
The audio samples are very different in time dimension and frequency dimension, which make it difficult to obtain features of these events in a single scale. Inspired by this, this section will propose a time-frequency dimension feature extractor to obtain time-frequency dimension features respectively.

The time-frequency dimension feature extractor(The left side of Figure 2) is to fully obtain the features of time dimension and frequency dimension. \( d \) stands for convolution of dilations with different dilations rates, and then concatenate them. Temporal attention and frequency attention are proposed in order to enhance features from related temporal dimensions and frequency segments. Applying different weights on time frame and frequency band can lead the network to pay different attention to the time and frequency characteristics of ambient sound. The structure of these two attention mechanisms is shown in the right side of Figure 2. For the frequency attention unit, first of all, the splicing feature of the upper level operation will undergo a convolution to transform the feature dimension \((C, T, F)\) into the feature \((1, T, F)\) containing only time-frequency dimension with channel number 1. Then perform a global average pooling operation on the feature.
graph, and only the frequency dimension will remain. Then let the feature pass through the activation function, change the value of the frequency dimension into the weight value between [0, 1], and recalibrate the frequency dimension by multiplying the weight value with the driving feature.

In this case, each representation learning is concentrated on a specific differentiated local region, rather than evenly distributed across the entire feature graph, which results in interference when a single branch is selected for a fragment without sound events, making the network more robust.

2.3 Fine feature learning module
After the time-frequency dimension features are extracted, the features will be refined. In previous experiments, we often failed to make good use of the data after and before the downsampling. Features sampled below the network are then sent to the next level of the network, often losing some information. When we start with the high resolution subnetwork as the first phase, gradually increase the high resolution to low resolution subnets, form more stages, and connect the multi-resolution subnets in parallel. We perform multiple multi-scale fusions so that each high to low resolution feature repeatedly receives information from other parallel representations, resulting in rich low-resolution features. As a result, the prediction of acoustic event categories may be more accurate and spatially located.

The purpose of the fine feature learning module design is to make use of features at different resolutions. In Figure 3, each square represents the feature graph under the current resolution. The black arrow represents the convolution unit, and the convolution kernel is 3x3. The green arrow represents the down-sampling operation, and the red arrow represents the up-adoption operation. In this module, average pooling is used for lower sampling and sub-pixel sampling is used for upper sampling. In the last layer, the features of the first two layers will be downsampled and spliced together. The dimensions will be reduced to the same dimension with low resolution through convolution and then output. Such features will have both high resolution and low resolution features, which is equivalent to fine processing of features.

Figure 2. Time-frequency dimension feature extraction module

---

**Figure 2. Time-frequency dimension feature extraction module**

- Input
- Frequency attention unit
- Temporal attention unit
- Output

- Frequency attention unit
- Temporal attention unit
- $U
- U1$ $U2$
- $U1'$ $U2'$
- $U'$
- $U2'$

- $f_{conv}$
- $f_{gap}$
- $f_{scale}$
- $f_{conv}$
- $f_{gap}$
- $f_{scale}$
2.4 Temporal context representation

Recent years, Transformer based models have attracted more and more attention in the field of sound event location and detection for their high-precision and efficient training process. A core part of the Transformer based model is the so-called self-attention(SA) mechanism, which uses the dot product to calculate the weight of attention. Although content-based dot product does a good job of capturing global interactions, it increases the computational complexity of the self-attention layer to a quadratic increase in the length of the input feature. Therefore, it is necessary to reduce computational complexity without sacrificing performance. Tay et al.[13] proposed dense synthesizer attention (DSA), which uses two feedforward layers to predict attention weight. Compared with SA, DSA completely eliminates point product and explicit pairwise interaction. It achieves competitive results using DSA in multiple language-processing tasks.

As shown in Figure 4, the main difference between attention of dense synthesizer and self-attention lies in the calculation method of attention weight. Intensive comprehensive attention eliminates the concept of query key value in the attention module and directly synthesizes attention weight. In practical application, dense synthesizer attention adopts two feedforward layers with ReLU activation to predict the attention weight, and the formula is as follows:
\[ B = \text{softmax}(\sigma_R(XW_1)W_2) \]  

(6)

Where \( \sigma_R \) are ReLU activation functions, \( W_1 \in \mathbb{R}^{d \times d} \) and \( W_2 \in \mathbb{R}^{d \times T} \) are learnable weights. The output of attention of dense synthesizer is:

\[ DSA(X) = B(XW_3W'^o) \]  

(7)

Where, \( W_3 \in \mathbb{R}^{d \times d} \) and \( W'^o \in \mathbb{R}^{d \times T} \) are the weight matrix of the linear mapping layer.

Under the excitation of convolutional neural network, WE adopt LDSA to solve the shortcomings of DSA. LDSA restricts the current frame to interacting only with neighboring frames. As shown in Figure 4, it defines a hyperparameter C, called context width, to control the length of the predicted attention weight, and then assigns the synthesized attention weight to the current frame and its neighboring frames, where \( C = 3 \) in Figure 4. The weight of other frames outside the context width will be set to 0. The calculation method of B in LDSA is the same as that in DSA. However, due to the existence of LDSA, the computation time and complexity are greatly reduced. The output of LDSA is:

\[ V = XW_3 \]  

(8)

\[ Y_t = \sum_{t=1}^{c-1} B_{t,j}V_{t,j-\lfloor \frac{j}{2} \rfloor} \]  

(9)

\[ LDSA(X) = YW'^o \]  

(10)

Our final time context representation module is similar to the coding part of Conformer structure. We replace the self-attention structure with the locally dense synthetic attention structure, and the rest of the structure is the same as Conformer.

3 RESULT AND DISCUSSION

In this section, the results of the network will be analyzed and compared in the DCASE2021 development dataset. First of all, the ablation experiment is conducted for the proposed module to verify the feasibility of the proposed module. Then, the proposed model is compared with other models to verify the effectiveness of the proposed network model.

3.1 Ablation experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>( ER_{20%} )</th>
<th>( F_{20%}(%) )</th>
<th>( LE_{CD} )</th>
<th>( LR_{CD}(%) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0.60</td>
<td>52.7</td>
<td>18.6</td>
<td>58.7</td>
</tr>
<tr>
<td>+TFFE</td>
<td>0.56</td>
<td>57.7</td>
<td>16.3</td>
<td>62.6</td>
</tr>
<tr>
<td>+TFFE+FFL</td>
<td>0.49</td>
<td>63.4</td>
<td>15.5</td>
<td>67.0</td>
</tr>
<tr>
<td>+ALL</td>
<td>0.45</td>
<td>67.2</td>
<td>13.9</td>
<td>69.8</td>
</tr>
</tbody>
</table>

In order to verify the effectiveness of the time-frequency dimension feature extraction module, a series of ablation experiments were conducted on the module. First, the baseline is the feature extraction module without going through the time-frequency dimension, and the fine feature learning module is replaced by three convolution blocks, each of which contains two convolution layers. The weights of the three convolution blocks
of the SED branch will still migrate to the three convolution blocks of the SSL branch. The time context representation is replaced by Bi-GRU, which is the same as the final model.

The tired line is added the TFFE module to the baseline method. It can effectively enhance the performance of the network, especially on $F_{20}(\%)$ increased 5.0%. In the experiment of the fourth line, a fine feature learning module was added on the basis of the third line, all the four metrics have improved. The last line in the table is the final result of the network proposed in Figure 1.

3.2 Comparison with other methods

The proposed model was compared with other models on the TAU-NIGENS Spatial Sound Events 2021 dataset. To ensure fairness, all models adopt the same training set, validation set and test set.

Table 2. Comparison with other methods

<table>
<thead>
<tr>
<th>Method</th>
<th>$ER_{20}$</th>
<th>$F_{20}(%)$</th>
<th>$LE_{CD}$</th>
<th>$LR_{CD}(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCASE2021 baseline[14]</td>
<td>0.73</td>
<td>30.7</td>
<td>24.5</td>
<td>44.8</td>
</tr>
<tr>
<td>SELD-TCN[15]</td>
<td>0.50</td>
<td>53.2</td>
<td>19.8</td>
<td>57.4</td>
</tr>
<tr>
<td>SSLDnet[16]</td>
<td>0.51</td>
<td>63.2</td>
<td>14.4</td>
<td>68.7</td>
</tr>
<tr>
<td>Ours</td>
<td>0.45</td>
<td>67.2</td>
<td>13.9</td>
<td>69.8</td>
</tr>
<tr>
<td>Ours (w/o Transfer)</td>
<td>0.50</td>
<td>65.4</td>
<td>16.1</td>
<td>66.1</td>
</tr>
<tr>
<td>Ours (single)</td>
<td>0.57</td>
<td>61.0</td>
<td>19.6</td>
<td>60.7</td>
</tr>
</tbody>
</table>

The proposed network is compared with other networks in Table 2. Ours represents the performance of the model proposed in this section, Ours (w/o Transfer) represents that there is no process of parameter transfer in the proposed model, and the two branches are trained separately. The last line used the single branch training mode, and then let the network output the results of sound event detection and sound source location respectively, that is to say, the method of parameter transfer learning can not be applied in this model.

4 CONCLUSIONS

In this paper, we propose a method of sound event localization and detection based on parameter transfer learning, which can transfer the parameters of the sound event detection branch to the sound source localization branch, and promote the training of the network branch of the sound source localization. Experiments verify that this network framework is effective. In the proposed network, the time-frequency dimension feature extractor can extract features of time dimension and frequency. By operating the features of two different dimensions, the output features can have time-frequency dimension features at the same time, and the features can be sent to the next level network more clearly. The fine feature learning module adopted in this model makes each feature of high resolution to low resolution repeatedly receive information from other parallel representations through multiple multi-scale fusion, so as to obtain rich low-resolution features. Different from the previous time context representation module, the GRU is partially replaced by the Conformer encoder embedded with the attention unit of local dense synthesizer. On the basis of saving computing time, the performance of the network is greatly improved and the network is closer to the lightweight network.

REFERENCES


Machined Learning-Based Active Target Detection
Using Temporal Variation Features From Spectrogram

Youngmin Choo¹; Sungjin Shin¹; Youngsang Hwang¹; Geunhwan Kim¹; Juho Kim²
¹ Ocean Systems Engineering, Sejong University, Korea
² Maritime Technology Research Institute, Agency for Defense Development, Korea

ABSTRACT
A scheme for underwater target detection is developed using machine learning (ML) with a deficient dataset. Temporal variations in amplitude and frequency around a target signal are different from those around a nontarget signal. Features are defined to reflect the differences and extracted using preprocessed spectrograms of beam time-series. An ML-based target detector using the temporal variation features is trained with the small dataset consisting of target and non-target samples and applied to the acoustic sonar data for testing, which deviate from the training data samples; in the test phase, the time window used for defining the features slides over the spectrograms of the acoustic sonar data such that the target signal is detected, resulting in target classification as per the classify-before-detect strategy. Finally, the detection results of the ML-based target detector are compared to those from deep learning schemes using automatically extracted features.

Keywords: Active target detection, Feature analysis, Sonar signal processing

1. INTRODUCTION
One of purposes of operating sonar systems is to detect signals of interest such as targets passively or actively. Recently, passive target detection has become more difficult owing to the reduced level of underwater radiated noise from ships as well as louder ocean ambient noise. Alternatively, an active sonar system transmitting a source signal is applied to detect the target signals with a higher signal-to-noise ratio by exploiting the known transmitted signal. However, when operating an active sonar system, returns from clutters exist, including fish schools and outcrops of the sea bottom, which cannot be removed using a filter owing to the similar frequency band with the target signal. Threshold-based decision rules, such as the constant false alarm rate method, are applied to filter out returns from clutters (Abraham, 2019). However, this simple decision causes many false alarms when the clutter signals are strong compared with the noise around them, and sonar operators are required to discriminate the target signals from the clutter signals at the end. To relieve sonar operators from labor-intensive tasks (i.e., the manual classification of target and clutter signals), a more sophisticated decision rule is required.

In this study, data-driven decision rules for detecting active target signals were derived based on machine learning (ML). Multiple aspects of scattered signals were exploited from the target and clutter with features reflecting their difference in terms of statistics and timbre, whereas the classical method with threshold uses simple information of signal amplitude.

2. FEATURES FOR TARGET DETECTION
In ML, handcrafted features are one of the key parameters determining the performance of a target classifier. In this study, features were defined to reflect the physical and perceptual differences between the target and nontarget, and they were extracted from the spectrogram of the acoustic data.

The amplitude distribution of the target signal differed from that of the clutter signal owing to the distinct characteristics of the scatterers. Several studies have investigated the amplitude probabilities of the target, clutter, and reverberation (Abraham, 2019), which were used to identify the target signals. Correspondingly, in the current study, the temporal amplitude variation around the target signal was
used as the physical feature. However, it was difficult to obtain the temporal amplitude variation from a raw time-series owing to its complicated pattern. Thus, the features for temporal amplitude variation were defined using power spectral densities at frames in spectrogram.

Meanwhile, a human listener can be cognizant of the difference between similar sounds, such as scattered signals from a target and clutter, using timbre. Perceptual features are defined to reflect the timbre in multiple domains, including spectrograms, auditory models, and energy envelopes, resulting in high-dimensional feature vectors after extraction from acoustic signals. Although complicated multidimensional perceptual features displaying sound in various aspects may be beneficial for distinguishing targets from clutter (Young and Hines, 2007), it is inappropriate to exploit them to train the target classifier with the deficient dataset owing to the curse of dimensionality. Alternatively, a temporal frequency variation of the acoustic signal in the spectrogram, which is correlated with timbre, was used as the other feature vector in this study.

3. ML-BASED TARGET DETECTION

A simple automatic target detector was contrived using concatenated physical and perceptual feature vectors. Principle component analysis (PCA) was applied to reduce the feature vector dimension, and two main principal components from PCA were used as input for the SVM. In the SVM, the hyperplane separating the samples belonging to different classes was obtained by maximizing the margin of the two classes.

Two main components that were sufficient for the classification were selected as input for validation (Figure 1). The process was repeated five times based on different validation datasets to estimate the accuracy (five-fold cross validation). The mean accuracy, precision, and recall of the simple target detector were approximately 94.2%, 92.7%, and 95.9%, respectively. Precision is lower than recall. Therefore, it is difficult for the classifier to identify clutter signals that are more scattered over the first principal axis; however, it still demonstrated satisfactory performance in the classification. Furthermore, under the small dataset, the ML approach using the hand-crafted features displays better generalization than deep learning approaches.

4. CONCLUSION

In this study, we proposed an automatic target classifier for an active sonar system based on an ML algorithm and data-driven decision rules. Useful features considering multiple aspects of scattered signals from the target and clutter were extracted from the preprocessed acoustic spectrogram of the beam time-series.

![Figure 1. Hyperplane in principle component domain based on (a) training dataset. Subsequently, it was applied to (b) the remaining (validation) dataset. The simple lines can separate two classes.](image)

REFERENCES

Comparative Study on Various Acoustic Features for Active Sonar Target Detection

Sungjin Shin¹; Youngsang Hwang¹; Geunhwan Kim¹; Juho Kim²; and Youngmin Choo³*
¹ Ocean Systems Engineering, Sejong University, Republic of Korea
² Maritime Technology Research Institute, Agency for Defense Development (ADD), Republic of Korea
³ Department of Defense Systems Engineering, Sejong University, Republic of Korea

ABSTRACT
Machine learning (ML) has been implemented to classify underwater targets. Sufficient data are required for the target classification using ML (particularly, deep learning). However, data are often insufficient and limited [1]. Therefore, to overcome this limitation, features of the data have been extracted and used as input. From numerous studies, various methods have been proposed for feature extraction. As well as active sonar cases, data are very limited and insufficient. Thus, it is necessary to extract suitable features for active target detection under the small dataset. Therefore, in this paper, we compare well-known feature extraction methods and find the most suitable feature. We extract the features such as 1) cepstrum [2], 2) Mel-frequency cepstral coefficients (MFCC) [3], 3) Gammatone-frequency cepstral coefficient (GFCC) [4], 4) power-normalized cepstral coefficients (PNCC) [5], 5) short-time Fourier transform (STFT) 6) wavelet packet decomposition (WPD) [6], 7) constant-Q transform (CQT) [7], and 8) timbre-based features [8]. To examine the utility of the feature for classification, data visualization is performed for each extracted feature using uniform manifold approximation and projection (UMAP) [9]. Lastly, the Gaussian mixture model (GMM) is implemented to analyze the data distribution of each extracted feature.

Keywords: Active Sonar, Feature extraction, Target detection

1. INTRODUCTION
For the active sonar target detection, data are often insufficient and limited [1]. In this paper, various hand-crafted acoustic features are researched and extracted to overcome this limit. We compare various acoustic features to find out appropriate and reliable features for the active sonar target detection. Following feature extraction methods are used. Section 2 simply explains feature extraction methods. Section 3 visualizes the distribution of extracted features using uniform manifold approximation and projection (UMAP) [9]. Probability distributions of the features are approximately expressed using Gaussian mixture model (GMM) for further analysis. Finally, Section 4 discusses the results of the comparison of extracted feature data distributions.

2. FEATURE EXTRACTION METHOD

2.1 Cepstral Coefficients features
Various cepstral coefficients analysis techniques are developed in the field of musical instruments, speech recognition and defense research purposed analysis. Following cepstral coefficients are used in this paper for active sonar data feature extraction.

Cepstrum becomes the basis for the cepstral coefficient as cepstral analysis [2]. Mel-frequency cepstral coefficient (MFCC) is based on the human auditory system which uses a filter bank called
Mel-filter bank [3]. Gammatone frequency cepstral coefficient (GFCC) uses the Gammatone filter bank which is an improved filter bank from the Mel-filter bank. Power-normalized cepstral coefficient (PNCC) also uses the Gammatone filter bank. However, it uses different time frames (short and medium) to analyze noise and speech separately [5]. Figure 1 represents the flow diagram describing the extraction of cepstral coefficients.

![Flow Diagram](image)

**Figure 1.** Flow diagram for the various Cepstral Coefficient extraction methods

### 2.2 Spectral features

For the spectral features, short-time Fourier transform (STFT), wavelet packet decomposition (WPD), and constant-Q transform (CQT) are implemented. STFT is the frequency analysis by windowing a short interval of signal. WPD decomposes the wavelet basis function of the signal and is improved from discrete wavelet transform by filtering deeper. CQT is one of the spectrogram analyses implemented for the musical instrument signal analysis.

### 2.3 Human Auditory System feature

Timbre-base features based on the human auditory system consist of various audio descriptors. Representations are based on the human auditory system and the audio descriptors are the feature vector of each representation. The timbre-base feature consists of 6 representations such as temporal energy envelope, audio signal, STFT magnitude, STFT power, equivalent rectangular bandwidth (ERB) FFT, and ERB gammatone. Each representation contains its audio descriptors, a total of 68 audio descriptors [8].

### 3. DATA DISTRIBUTION AND COMPARISON

#### 3.1 Data distribution using UMAP

The feature distributions are visualized with UMAP (Figure 2). Red points represent the target data distribution of each feature and blue points are nontarget data points. We compared it with other feature distribution analyses such as PCA or t-SNE. However, we choose UMAP as our prior method for feature distribution analysis for this paper because more reliable feature distribution is observed.

![Feature Data Distribution](image)

**Figure 2.** Extracted Feature data distribution using UMAP
3.2 Comparison of extracted feature datasets

To compare each feature distribution, GMM is implemented. In Figure 3, the probability density functions of each target and nontarget are plotted according to MFCC and timbre-based features after applying UMAP.

![Figure 3. Contour plotted according to the probability density function of the MFCC and timbre-based feature](image)

KL-divergence is estimated according to the probability of the target and nontarget distributions obtained from GMM (Table 1). Higher KL-divergence represents that target and nontarget feature data are distributed distantly. Therefore, the timbre-based feature is observed with the best separation ability.

<table>
<thead>
<tr>
<th>Feature</th>
<th>KL-Divergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cepstrum</td>
<td>1.0756</td>
</tr>
<tr>
<td>MFCC</td>
<td>4.5722</td>
</tr>
<tr>
<td>GFCC</td>
<td>4.548</td>
</tr>
<tr>
<td>PNCC</td>
<td>4.7707</td>
</tr>
<tr>
<td>STFT</td>
<td>5.258</td>
</tr>
<tr>
<td>WPD</td>
<td>5.8204</td>
</tr>
<tr>
<td>CQT</td>
<td>7.5258</td>
</tr>
<tr>
<td>Timbre-based feature</td>
<td>29.331</td>
</tr>
</tbody>
</table>

4. DISCUSSION AND FUTURE WORK

4.1 Discussion

We observed the timbre-based feature has the best separation ability for the dataset we used. However, with another dataset, the timbre-based feature is not performing greatly. We concluded that it is because timbre-based feature consists of numerous parameters to adjust which can increase the sensitivity of the feature extracting technique.

4.2 Future work

Finding similarities between the feature extraction methods will increase the efficiency of the feature fusion and feature selection for machine learning and deep learning. To find out the similarities between feature extraction methods, a deeper understanding of data distribution will be required. Data analysis schemes will be implemented and adopted on active sonar data.
ACKNOWLEDGEMENTS

This study is supported by Agency for Defense Development (ADD) (UD210005DD).

REFERENCES

Effects of resizing the image data to the underwater acoustic classification from ship radiated noise using convolutional neural network

Hye sun LEE\(^1\); Woo-young HONG\(^2\); Youngmin CHOO\(^3\); Keunhwa LEE\(^4\)

\(^1\) Sejong University, Republic of Korea
\(^2\) Sejong University, Republic of Korea
\(^3\) Sejong University, Republic of Korea
\(^4\) Sejong University, Republic of Korea

ABSTRACT

Underwater acoustic target classification based on ship radiated noise has emerged as an impressive topic since the deep learning-based method has dramatically improved the state-of-the-art performance. Recently, some deep learning studies using a large dataset such as ShipsEar, DeepShip, and ONC(Ocean Networks Canada) have been presented and shown higher accuracy for the underwater target classification. Most of these deep learning models are based on the convolutional neural network(CNN) with input data of image type, whose initial size depends on the choice of a pre-processing method, mainly a time-frequency representation. In this proceeding, we evaluate the effects of resizing input data on the classification accuracy using the DeepShip datasets and VGG16. It is shown that the proper resizing of the data can improve the performance of the classification.

Keywords: Underwater acoustic noise, Deep learning, DeepShip, ShipsEar

1. INTRODUCTION

The ship classification using underwater radiated noise measured by a passive sonar system is useful for application to military technology. Several researches have been actively conducted to achieve good performance using deep learning based on various feature of the acoustic signal radiated from ship\([1]\). Public datasets such as ShipsEar, ONC are the most famous, and the state-of-the-art result is achieved using DNN(Deep Neural Networks) models\([1,2,3]\). Traditionally, the conventional techniques combine classifiers using DNNs with feature extraction such as MFCC(Mel-Frequency Cepstral Coefficients) and GFCC(Gammatone Frequency Cepstral Coefficients) \([4,5]\).

The feature extraction technique, which is mainly used in deep learning, is a method of transforming raw time-series data into t-f presentation consisting of two dimensions with time and frequency axes. This technique has been developed to efficiently extract features using various scale filters, based on the spectrum, and are recently considered an important role in preprocessing as input features in DNN models.

The CNN model is one of the DNN models, which began to be developed in the field of image
recognition [6] and is also widely used in the field of speech recognition using t-f presentation [7]. In the underwater acoustic environment, deep learning has also achieved good performance on target detection and recognition [1]. However, the t-f presentation for low frequency range has a low resolution, unlike the general size of real image data (128 × 128, 256 × 256, etc). In the field of image recognition, it has been already confirmed that image resolution affects the performance of deep learning, which is rapidly lowered with extremely low resolution [8, 9, 10].

In this proceeding, we extracted four features (MFCC, GFCC, NL (1/3 octave noise level), CQT (Constant Q transform)) [11, 12] using the recently released DeepShip dataset [13], and the features are applied to the VGG16 model [14] by increasing the resolution using nearest neighbor interpolation and B-spline interpolation from the initial size. In this simulation, we can be confirmed the small size of the t-f presentation feature affects the deep learning performance similar to the image data in CNN models. In addition, it can be seen that each feature has a performance difference depending on the interpolation method. It can verify how robust the features are to the image resolution, and overall, we can obtain improved performance through resizing (increasing the resolution).

2. TECHNICAL DETAIL OF RESIZING FEATURES

2.1 Nearest neighbor interpolation

In image processing, nearest neighbor interpolation is one of the simple methods to interpolation in one or more dimensions. The generated values of new pixel are determined by the nearest neighboring pixel value. An Example of the method is depicted in Figure 1. A blockiness problem may occur when the destination image is quite larger than the initial image because the generated pixel is not assigned to a value other than the source pixel value [15].

![Image of Nearest neighbor interpolation](image.png)

Figure 1. Example of Nearest neighbor interpolation [16]

2.2 B-spline interpolation

As the more advanced technique of image interpolation, B-spline interpolation has been proposed. The newly generated pixel is the sum of the weighted sixteen nearest pixels. Each weight is determined by the n-order B-spline function and Figure. 2 shows how the new pixel value is generated by the normalized zero-order B-spline function [17].

![Image of B-spline interpolation](image.png)
2.3 Resizing two-dimensional feature

In this proceeding, we can obtain feature images of higher resolution using resizing based on interpolation. The process of resizing features with the nearest neighbor interpolation and B-spline interpolation described above was conducted. The initial dimensions and resized dimensions of each feature are shown in Table 1. The sample images of resizing features are shown in Figure 3.

![Figure 2. Example of B-spline interpolation(zero-order) [16]](image)

### Table 1. The initial dimension and resized dimension

<table>
<thead>
<tr>
<th>feature</th>
<th>Class</th>
<th>Initial</th>
<th>Almost Initial</th>
<th>Nearest</th>
<th>B-Spline</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>46 × 13</td>
<td>46 × 13</td>
<td>46 × 32</td>
<td>138 × 96</td>
<td>138 × 96</td>
</tr>
<tr>
<td>GFCC</td>
<td>46 × 13</td>
<td>46 × 13</td>
<td>46 × 32</td>
<td>138 × 96</td>
<td>138 × 96</td>
</tr>
<tr>
<td>NL</td>
<td>43 × 35</td>
<td>43 × 35</td>
<td>43 × 35</td>
<td>138 × 105</td>
<td>138 × 105</td>
</tr>
<tr>
<td>CQT</td>
<td>46 × 64</td>
<td>46 × 64</td>
<td>46 × 64</td>
<td>138 × 192</td>
<td>138 × 192</td>
</tr>
</tbody>
</table>

3. EXPERIMENT AND DISCUSSION

3.1 Experimental setup

In the experiments using VGG16, we performed on the DeepShip dataset for train, test. DeepShip, dataset has four classes of cargo, passengership, tanker, and tug with two years of data collected in the Strait of Georgia, and has a duration of approximately 47 hours with a sampling frequency of 32 kHz.[13] We downsampled DeepShip datasets at a sampling frequency of 16 kHz and the length of a segment is 250ms with hopped 64ms. Each duration of the sample is approximately 3s. As previously mentioned we extracted 4 features, MFCC, GFCC, NL(1/3 octave noise level spectrum), CQT(constant Q transform). After randomly selecting 70% training DeepShip dataset once, a total of
three simulations was conducted in almost initial size and larger size by applying nearest neighbor, B-spline interpolation to the same data. The learning rate is 1e-6, the batch size is 128, and the epochs are about 1000, and the VGG16 model of pretrained weight provided by PyTorch was used.

3.2 Result and discussions

The DeepShip dataset results of all features are shown in Table. Among the four features, the resizing CQT feature achieved up to 97.2% accuracy and has little difference in performance over the almost initial size feature. Following this, MFCC shows an 8.2% performance improvement when using B-spline interpolation, and seems to be most affected by image size. NL showed a performance improvement of 3.7% and GFCC showed a performance improvement of 2.8%.

Table 2. The accuracy result of Deep dataset

<table>
<thead>
<tr>
<th>Feature</th>
<th>Almost Initial(%)</th>
<th>Nearest(%)</th>
<th>B-Spline(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>50.5</td>
<td>58.4</td>
<td>58.7</td>
</tr>
<tr>
<td>GFCC</td>
<td>84.0</td>
<td>86.8</td>
<td>86.8</td>
</tr>
<tr>
<td>NL</td>
<td>88.5</td>
<td>92.2</td>
<td>91.1</td>
</tr>
<tr>
<td>CQT</td>
<td>97.0</td>
<td>97.0</td>
<td>97.2</td>
</tr>
</tbody>
</table>

4. CONCLUSIONS

In this paper, we have conducted Underwater acoustic target classification based on ship radiated noise using the interpolation method and VGG16 model. As the result of DeepShip dataset [13], it shows the most performance improvement in MFCC and little change in CQT. Based on this simulation, it can be seen that the robustness of the image size is different for each feature, and CQT is the most robust among the above features.

ACKNOWLEDGEMENTS

We thank M. Irfan et al.[13] and D. Santos-Domínguez et al.[2] who have allowed the use of the datasets. This work has been funded by the Ministry of Oceans and Fisheries of Korea under contract No. 20210067.

REFERENCES
ABSTRACT
Nowadays, continuous and unattended noise monitoring systems are being installed in the vicinity of airports, railways and roads, with the aim of more precisely controlling this kind of noise sources. A key part of these solutions is the detection and classification of the captured acoustic events. International standards permit the use of different methods to identify if the detected events are of interest or should be discarded, including: those based on a human operator (carefully trained to discern the nature of an event by evaluating the time profile of an unique parameter, mainly the $L_{Aeq}$); and those based on automatic processing. However, most of the automatic acoustic event detection solutions proposed by the machine learning community are based on complex features, different to those normally measured by sound analyzers, or even employ raw audio as input. Hence, it is difficult to embed these solutions into standard real-time measuring systems. In this paper, we propose an automatic event detector for train pass-by based on machine learning techniques, trying to reproduce the expertise of a human operator. The resulting event detector is robust, with low computational cost, and employs features directly obtained by the measuring system, which allows its implementation into the sound analyzer software.

Keywords: Machine Learning, Event Detection, Data Augmentation

1 INTRODUCTION
The growth and development of cities usually lead to high concentrations of population. This fact can have a significant impact on noise pollution, being transport infrastructures one of the most relevant noise sources related with it.

Environmental noise is defined by the European Parliament as the unwanted or harmful outdoor sound created by human activities, including noise emitted by means of transport, road traffic, rail traffic, air traffic, and from sites of industrial activity [1] Hence, the vicinity of airports, railways and industry complexes are areas with a high risk of noise pollution, in which the measurement and control of the environmental noise become essential to comply with the international regulations [1].

When countries develop laws related to noise pollution [2], two main approaches are usually followed:

- The first one is focused on the receiver. It assesses the overall amount of noise in a specific acoustic area, independently of which particular source is producing the pollution.
- The emitter is the target of the second approach. In this case, laws related to noise usually limit the amount of noise that a source can emit. The emitter perspective requires an accurate identification of the noise emitted by each source, isolating it with respect to that generated by other sources present in the measurement scenario.
Noise monitoring systems are being deployed in areas of high risk of noise pollution, continuously checking the noise level throughout the whole year. For instance, such systems are being installed in the vicinity of airports to quantify the noise related with aircraft operations. A key part of these systems is the correct and accurate identification of each independent aircraft, in order to only compute its associated noise. In this sense, the ISO 20906:2009 standard [3] allows the use of various methods for the detection of aeronautical events, ranging from those based on the expertise of human operators to others related with automatic event identification. Nonetheless, the existing noise monitoring systems often rely on a manual operator to perform the discrimination between the events of interest and those coming from other noise sources. Sometimes, the expertise of this human operator is the unique identification tool, but, in others, the human operator only evaluates challenging cases marked by a previous and simple automatic detection process.

The criteria followed by these human experts is usually based on a rather simple visual analysis of the sound level profiles. While this task might look easy, it is the consequence of a long training process. For instance, it is common to evaluate the temporal evolution of the $L_{A\text{eq}}$ in order to conclude the relevance of a certain event. In addition, some systems allow the listening to the audio of the event in case of doubt [4].

Recently, several approaches based on Machine Learning have been designed for acoustic event identification. These are not only focused on noise pollution applications: we can find acoustic event classifiers for a wide range of applications, such as acoustic surveillance, communication with robots and personal assistance, among others [5, 6]. In fact, the machine learning research community usually participates in competitions related to acoustic identification, being DCASE (Detection and Classification of Acoustic Scenes and Events) [7] one of the most important challenges.

With regard to noise pollution, various systems have been proposed, but these tend to focus on more complex acoustical features than the $L_{A\text{eq}}$ profile. For instance, in [8], the proposed approach trusts in biologically-inspired Gammatone Cepstral Coefficients (GTCC) to model the spectro-temporal evolution of the noise events. GTCC coefficients are also employed in [9], where the designed solution has to record the audio signal to perform the identification. Raw data are also the input to obtain the features in the system proposed in [10]. Short-time Fourier Transform features are the basis of the automatic environmental sound classifier proposed in [11]. Other complex spectral features are selected in [12], apart from temporal characteristics obtained directly from the raw data. It seems clear that employing this kind of features in automatic event classifiers makes difficult the implementation of the classifier in standard sound level meters. Therefore, it would be attractive to design a solution, based on machine learning techniques, that mimics the operation of human experts, only utilizing a simple evaluation of the $L_{A\text{eq}}$ profiles.

In this paper, using railway pass-by events as case study, we propose an automatic classifier that tries to replicate the expert knowledge of a human operator. In this sense, we have designed some specific features over the $L_{A\text{eq}}$ temporal evolution that summarize the acoustic profile of detected events. Furthermore, through a simple machine learning classifier, trained with novel data augmentation techniques, our proposal can be adapted to distinct environments with background noises of different natures and levels.

The resulting system presents a two-fold advantage. On one hand, $L_{A\text{eq}}$ profiling is based on statistics that can be computed directly by the measuring system, which allows its efficient implementation into the sound analyzer software, reducing the computational and economic costs of the solutions, including those of memory burden. On the other, it is robust to adapt to new scenarios under different background noise conditions without the need for further training.

The rest of the paper is organized as follows: Section 2 provides a detailed description of our method. Section 3 enumerates and discusses the experimental results that support our proposal and, finally, Section 4 summarizes our conclusions and outlines future lines of research.

## 2 METHODOLOGY

In this section we first give a general description of our acoustic railway pass-by detector; then, a detailed explanation of its constituent processing blocks will be provided in subsequent subsections.

The full pipeline of our system, depicted in Figure 1, can be decomposed into several steps:

- The $L_{A\text{eq}}$ profiles (every one second) provided by the noise monitoring system $L = \{l_1, l_2, \ldots, l_T\}$ are the
input of the Acoustic Event Detector (AED), being \( T \) the total number of profiles. This module outputs the collection of detected acoustic events \( E = \{e_1, e_2, \ldots, e_N\} \), being \( N \) the total number of detected events, each one with its corresponding label \( y = \{y_1, y_2, \ldots, y_N\} \) (when available), which indicates if the event is of interest or not. These labels are used for the training process of the classifier.

- The Feature Extraction Module (FEM) is in charge of profiling each event \( e_i \) by characterizing its dynamics along time. This module extracts eight statistical features to describe each event, \( f_i \).
- The Event Classifier (EC) is trained through the feature description of the acoustic events, \( F = \{f_1, f_2, \ldots, f_N\} \), and their corresponding labels \( y \). During test, for each feature description \( f_i \), outputs the probability of the event to be an event of interest, \( p_i \).

2.1 Acoustic Event Detector (AED)

The Acoustic Event Detector analyzes each one of the \( L_{A_{eq}} \) profiles obtained by the noise monitoring system \( l_i \) and outputs the detected events \( E \) and their labels \( y \). It imposes two requirements for a sequence of \( L_{A_{eq}} \) values to be a event of interest: 1) the \( L_{A_{eq}} \) in each one of its points must be above a threshold \( S_{th} \); and 2) the minimum length of an event to be of interest must be above the threshold \( L_{th} \). These thresholds will be adapted to the particular task in Section 3. The labels (when available) can obtained through an automatic temporal alignment of the \( L_{A_{eq}} \) profiles with a ground-truth (a file that contains the time instants of each event), through a manual labeling operation, etc.

2.2 Feature Extraction Module (FEM)

Once the events have been detected, we model the \( L_{A_{eq}} \) profiles of each event through a series of features which meaningful characterize the \( L_{A_{eq}} \) dynamics. The set of features that have been designed are inspired by the operation of human experts, who pick out the events of interest at a glance, by only analyzing the dynamics of the \( L_{A_{eq}} \) profile: magnitude, shape and variations. Based on this expert knowledge, we have designed the following features:

1. \( L_{A_{eq}} \) mean. The \( L_{A_{eq}} \) mean levels of the acoustic events have importance to distinguish the events of interest from other events, especially in scenarios with low environmental noise. The acoustic events of interest have usually higher \( L_{A_{eq}} \) levels than those of other events.

2. \( L_{A_{eq}} \) standard deviation. Other events usually have larger standard deviations than those of the events of interest. The \( L_{A_{eq}} \) profiles of the events of interest often have a softer evolution along time.
3. $L_{A_{eq}}$ skewness. The skewness value of a distribution measures its asymmetry, which will be larger for other events.

4. $L_{A_{eq}}$ kurtosis. The kurtosis value of a distribution indicates if the data are distributed near the mean of if they are located at the tails of the distribution, i.e., it is a measurement of the ‘peakness’ of the distribution. Acoustic events of interest usually present a peak which concentrates most of its energy.

5. $L_{A_{eq}}$ max. The maximum value of the $L_{A_{eq}}$ curve can be useful for classifying acoustic events depending on the noise level in the scenario under test.

6. $L_{A_{eq}}$ Crossing Rate. This feature refers to the number of times that the $L_{A_{eq}}$ curve crosses with its mean. Acoustic events of interest often concentrate the energy in a peak, so they cross with the mean only twice. Other events usually cross with the mean several times.

7. $L_{A_{eq}}$ Crossing Rate mean. It is the mean value of the largest peak of the $L_{A_{eq}}$ curve. Its value should be significantly above for the events of interest.

8. $L_{A_{eq}}$ Crossing Rate width. This last feature is the width of the largest peak of the $L_{A_{eq}}$ curve, where higher values should belong to the events of interest.

Figure 2 shows a typical acoustic event (a railway pass-by) in comparison with two events of no interest: a typical one, and a difficult event. As observed, the features that we have designed are correlated with the differences between the $L_{A_{eq}}$ profiles and should be useful to distinguish between them.

2.3 Event Classifier (EC)

The Event Classifier receives the complete set of features and the labels of the acoustic events and outputs the probability of each event to be of interest. It relies on a specific Data Augmentation technique and a Logistic Regression classifier.

Data Augmentation is defined as the process of applying some realistic transformations to a reduced dataset, employed specially for images [13], and more recently in Natural Language Processing [14]. The use of data augmentation usually follows two main objectives: 1) increase the size of the dataset in case of scenarios with reduced data; and 2) increase the variability of the dataset to avoid overfitting (especially when dealing with deep-learning techniques or performing domain adaptation, applying the detector in unseen scenarios, as it is our case) [15, 16]. In this work, we have proposed a data augmentation algorithm with the aim of reducing the signal-to-noise ratio (SNR) of data measured in favorable conditions of background noise. Our implementation of the algorithm can be explained in two steps:

- Apart from our $L_{A_{eq}}$ profile dataset $L$, we propose to collect representative examples of the types of noise that can be present in the typical scenarios, the set $Z = [z_1, z_2, ..., z_M]$, being $M$ the total number of samples. In our specific case, $Z$ contains other events, such as cars and motorcycles passes-by, people speaking, birds, etc.
• With a SNR of reference \(SNR_{\text{ref}}\) between the dataset samples and the noise, we combine the \(L_{\text{Aeq}}\) levels of the original sample \(i\) and the sample \(z_i\) (increasing or attenuating its level as a function of our design parameter), obtaining the noise compensated dataset \(S = [s_1, s_2, ..., s_M]\).

The augmented samples are the input to a Logistic Regression classifier [17]. A Logistic Regression classifier obtains the probability of each sample \(p_i\) to be of interest from the features \(f_i\) as:

\[
p_i = \frac{1}{1 + \exp(-\beta_0 + \beta_1 f_{i1} + ... + \beta_G f_{iG})}
\]  

(1)

where the weights \(\beta\) are inferred from the data and \(G\) is the total number of features. The classifier is trained by minimizing a binary cross entropy function with L2 regularization as follows:

\[
\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} y_i \log(p_i) + (1 - y_i) \log(1 - p_i) + \lambda \sum_{j=1}^{G} \| \beta_j \|
\]  

(2)

being \(\lambda\) the optimum weighting parameter between the regularization and the binary cross-entropy loss, obtained by cross-validation.

We have selected this type of classifier for three main reasons: 1) its computational cost at test stage is very reduced, since it can be implemented as a weighted sum of the features and a sigmoid function. Hence, it is completely suitable for its embedding into the software of a standard noise monitoring system; 2) it outputs a soft probability for each event to be of interest, allowing us to assess the degree of confidence of this classification, and to decide if additional methods should be applied for a specific event; and 3) the operation of the classifier is explainable through the weights, which indicate the particular relevance of each feature for this task.

3 EXPERIMENTS

3.1 Dataset Description

To assess our classifier, environmental noise measurements were carried out at two outdoor locations near the railway tracks to build a railway pass-by (RPB) database, containing more than 930 hours of data. The two selected scenarios were:

- **Scenario A**: the soundmeter was very close (i.e., 15 m. in straight line) to the railway track and facing it, at a height of 1.5 meters, in an area with little noise pollution from other noise sources. Therefore, the railway events at this point can be observed very clearly. It can be considered a clean scenario with a high SNR, where we have measured approximately 20 hours of data.

- **Scenario B**: the microphone was at a slightly greater distance from the tracks (i.e., 28 m.) than it was in scenario A. Also, it was placed at a height of about five meters, as the microphone was fixed outside the façade of the first floor of a building, two meters away from the façade. In scenario B, railway events were highly contaminated by events from other types of sound sources, such as voices, cars, motorcycles, and buses, that were even closer to the microphone than the tracks were. More than 900 hours have been measured in this challenging scenario, with degraded SNR.

It is worth noticing that the database includes four types of train: commuter trains, two types of interurban trains and freight trains. However, the most common ones are commuter trains.

The measurements in both scenarios were obtained with the same sound analyzer, Brüel and Kjaer Type 2250, by using the enhanced logging module, BZ-7225. All measurements were verified, prior to the start, by using a Brüel and Kjaer Type 4231 calibrator. The equipment was set up to taken periodically every second various broadband, including \(L_{\text{Aeq}}\), and spectral parameters. Therefore, it should be noted that, when the \(L_{\text{Aeq}}\) profile is mentioned in this paper, we are referring to a set of values of the \(L_{\text{Aeq}}\) taken every second.

In total, our AED provides 1472 samples (when operating with \(S_{th} = 66dB\) and \(L_{th} = 4s\), the optimum thresholds for our scenario since no RPB is missed), 289 coming from scenario A and 1183 coming from
scenario B. From these 1472 samples, 775 are labeled as RPB events and 697 are other events. Figure 3 shows the $L_{Aeq}$ profiles for the dataset scenarios A and B. As observed, samples are both variable in their duration and $L_{Aeq}$ levels, and scenario B is significantly more challenging in terms of RPB detection than scenario A.

![Figure 3. $L_{Aeq}$ profiles for the clean scenario (left) and the noisy scenario (right).](image)

To perform data augmentation, we have employed a noise dataset containing more than six hours of urban noise, taken in four different scenarios: roads, narrow streets, wide streets and normal streets. If we apply our AED to these set, 165 events can be obtained to be added as other events to the experiments.

The process of data augmentation is performed as described in Section 2, over the samples of scenario A. Depending on the objective SNR, we obtain both the RPB events contaminated with noise, and a varying number of other events because some parts of the $L_{Aeq}$ profiles not belonging to the RPB events trigger the AED $S_{th}$ threshold (this number will increase along with the decrease of the objective SNR). For our experiments, we have selected different objective SNRs: 0 dB, 3 dB, 5 dB and 10 dB.

### 3.2 Results

The goal of this section is to experimentally assess the proposed RPB detector in terms of its ability to adapt to new scenarios. For that purpose, our proposal has been trained and tested with different sets and the results have been compared in terms of their performance measurements.

The acronyms for the sets used for training and testing the detectors are A for scenario A, B for scenario B, UN for the urban noise samples, and SNRX for the noisy compensated scenario A, being X the objective SNR. The Area Under the Receiver Operator Characteristic (ROC) curve (AUC) is chosen as performance measurement. The ROC curve evaluates how well our classifier ranks the samples, i.e., its ability to distinguish between RPB and other event samples, as its discrimination threshold is varied. The AUC summarizes this curve in a single value to ease the comparison of different systems. Table 1 shows the results in terms of the AUC for the different configurations.

<table>
<thead>
<tr>
<th>Train sets</th>
<th>B</th>
<th>A</th>
<th>A + UN</th>
<th>A + UN + SNR10</th>
<th>A + UN + SNR5</th>
<th>A + UN + SNR3</th>
<th>A + UN + SNR0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test sets</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>AUC (%)</td>
<td>94.19</td>
<td>81.82</td>
<td>86.23</td>
<td>86.81</td>
<td><strong>88.01</strong></td>
<td>87.92</td>
<td>87.37</td>
</tr>
</tbody>
</table>

Some insights can be extracted from the results:

- First, if we train with scenario B (noise contaminated) and test with scenario A, the classification is very accurate (94% of AUC). Even in this case, it is challenging to correctly classify the most difficult events. This value should be considered as an upper bound to the results that we can obtain.

- The case of use proposed in this paper is the following one: we train the detector with samples of scenario A (a clean scenario), and we evaluate it in other scenarios (scenario B), where it has to work...
with different background noise levels. An optimum training of the detector allows it to adapt to other scenarios without re-training. In this sense, the direct approach (baseline) is to use the detector trained exclusively with the samples obtained from scenario A. However, the classifier provides moderate results (with a value of AUC of 81.82%).

- In that regard, it is more effective to add some samples of other events to the train set. Following this approach, the AUC is improved by 4%.

- Finally, with our data augmentation approach, the results depend on the objective SNR. The best results are obtained with a $SNR_{ref} = 3$ dB or $SNR_{ref} = 5$ dB (with minimal differences), reaching the 87.93% and 88.01% of AUC, respectively. With that values of SNR, we obtain a training set maximally aligned with our test scenario. Lower SNRs obtain worse results because the level of noise is above the one in the test scenario, and higher SNRs do not achieve to change the training samples enough to adapt to scenario B. Figure 4 shows the effect of the data augmentation policy over the $L_{A_{eq}}$ profiles of scenario A, bridging the gap between this one and scenario B.

![Figure 4. $L_{A_{eq}}$ profiles. From left to right: scenario A, SNRs of 10, 5 and 3 dB, and scenario B.](image)

Finally, we can take advantage of the descriptive nature of this classifier and analyze the weights of the Logistic Regression classifier for the baseline approach (trained with A and tested in B) and the optimum case (trained with A+UN+SNR5 and tested in B) to obtain some information about the most relevant features for the classification. Table 2 shows the weights for each considered feature. As observed in the table, in the baseline case the most relevant features for the classification are the mean value of the $L_{A_{eq}}$, the $L_{A_{eq}}$ CR mean and the skewness of the distribution. This is a trivial version of the classifier, able to distinguish between the RPB events and the other ones just because they have high $L_{A_{eq}}$ values and their values are centered around the mean. However, this is not the case of the optimum classifier (see Figure 3). Hence, when we adopt our data augmentation policy, the trivial solution is not valid, and the classifier puts more effort in classifying the samples according to more descriptive features. In the optimum case, the most relevant features are the standard deviation of the $L_{A_{eq}}$ profile, its kurtosis (the ‘peakness’ of the $L_{A_{eq}}$ profile) and the $L_{A_{eq}}$ CR mean.

<table>
<thead>
<tr>
<th>Feature</th>
<th>$L_{A_{eq}}$ mean</th>
<th>$L_{A_{eq}}$ std</th>
<th>$L_{A_{eq}}$ skew</th>
<th>$L_{A_{eq}}$ kurt</th>
<th>$L_{A_{eq}}$ max</th>
<th>$L_{A_{eq}}$ CR</th>
<th>$L_{A_{eq}}$ CR mean</th>
<th>$L_{A_{eq}}$ CR width</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/B ($\beta$)</td>
<td>0.7292</td>
<td>0.2666</td>
<td>0.9208</td>
<td>-0.4544</td>
<td>-1.0370</td>
<td>-0.077</td>
<td>2.7938</td>
<td>-0.3250</td>
</tr>
<tr>
<td>A+UN+SNR5/B ($\beta$)</td>
<td>-0.4719</td>
<td>2.2991</td>
<td>0.0498</td>
<td>-1.1805</td>
<td>0.0939</td>
<td>0.3199</td>
<td>0.5998</td>
<td>0.1808</td>
</tr>
</tbody>
</table>

4 CONCLUSIONS

In this paper, we have proposed an automatic event detector for railway pass-by based on machine learning techniques, trying to reproduce the expertise of a thoroughly trained human operator. It relies on ad-hoc features describing the $L_{A_{eq}}$ profile to discriminate between railway pass-by events and other ones. By means of a novel data augmentation policy, our proposal is versatile and robust enough to adapt to new scenarios without further training. The resulting RPB detector, which employs features directly obtained by the measuring system and is based on Logistic Regression (with low computational cost), can be implemented into any noise monitoring
system software and perform in real time. Furthermore, it is completely explainable, giving information about the most relevant features.

Our experiments have shown that the RPB detector is able to adapt to a challenging scenario when trained in a high SNR one, just with a proper data augmentation strategy, therefore bridging the gap between different noise conditions in different scenarios. Likewise, the resulting detector puts its effort in relevant features for the classification avoiding non interesting ones.

The main lines of further research involve the use of non-linear classifiers (neural networks), able to model more complex relationships among the features. In addition to this, more scenarios must be considered to further demonstrate the versatility and robustness of the detector.

ACKNOWLEDGEMENTS
This work has been partially supported by the Regional Grant 2020/NMT-6660 of the Autonomous Region of Madrid. Miguel Molina-Moreno is supported by the Spanish Ministry of Education, Culture and Sports FPU Grant FPU18/02825.

REFERENCES
Model-Agnostic Meta-Learning based active target classification for small objects near sea bottom

Heewon You\textsuperscript{1}; Sung-Hoon Byun\textsuperscript{2}; Youngmin Choo*\textsuperscript{3}

\textsuperscript{1}Ocean Systems Engineering, Sejong University, Republic of Korea
\textsuperscript{2}Ocean System Engineering Research Division, Korea Research Institute of Ships and Ocean Engineering (KRISO), Republic of Korea
\textsuperscript{3}Department of Defense Systems Engineering, Sejong University, Republic of Korea

ABSTRACT

Recently, machine learning-based methods have been widely used to discriminate targets from clutters in the ocean, but abundant training data is required. Furthermore, it takes intensive time and labor to obtain acoustic data in various ocean environments. To overcome these limitations, we propose a model-agnostic meta-learning (MAML) based active target classifier, which is adapted quickly to a new environment with less training data by learning how to learn. The MAML classifier is trained with two phases (inner and outer loops) using synthetic target and clutter data samples from various environments. The methodology for target classification is learned according to the various environments (tasks) in the inner loop. Then, the outer loop facilitates the learning using the knowledge accumulated across tasks. Lastly, performance of MAML based classifier is compared to those of conventional schemes in terms of recall, false alarm, and accuracy.

Keywords: Model-agnostic meta-learning, active target detection, target scattering

1. INTRODUCTION

Research for automation of target and clutter signal classification in sonar systems has been steadily conducted. Recently, many schemes using machine learning have been proposed [1-3]. Conventional machine learning achieves good performance when abundant training data are available. However, intensive time and labor are required to collect the abundant training data.

Meta-learning learns ability to adapt quickly to new tasks with less training data and competently adapt to new tasks even when there is no training data for new tasks. Meta-learning has three main approaches: 1) metric- 2) model- 3) optimization-based technique [4]. The key idea of metric-based meta-learning is to learn good kernel functions. Model-based meta-learning can be achieved by an internal architecture or controlled by other meta-learner models. Optimization-based meta-learning uses a few sample examples to extract meta-knowledge that can enhance optimization performance.

In order to classify targets and clutters in various ocean environments using machine learning, abundant amounts of training data are required for each ocean environment. Therefore, in this study, we propose a method to distinguish targets and clusters in a new environment without learning data, using a model learned with model-agnostic meta-learning (MAML) [5], an optimization-based approach among the three approaches.

2. MAML

MAML is a meta-learning using optimization-based approaches, which obtain model parameters that reflect the general characteristics of various tasks and use them as initial model parameter values in the new task. Here, distinguish between synthetic targets and clusters in various environments, MAML classifiers are trained in two stages: inner loop and outer loop. First, in the inner loop, an
optimization strategy such as gradient descent is used to update model parameters for identifying the target in each of the various environments, which is indicated by the dashed line in Figure 1. Then, the outer loop facilitates the learning using the knowledge accumulated across the tasks, which is shown as a solid line in Figure 1. The parameters $\theta$ have proper generalization performance because they have learned various tasks, so they easily adapt to new tasks.

![Figure 1](image)

**Figure 1.** In the inner loop update, the MAML classifier obtains the parameters of each of the various tasks, and in the outer loop update, the accumulated knowledge throughout the task can be used to quickly adaptation for new tasks.

3. SIMULATION

To analyze the MAML-based classifier, we simulated a situation to distinguish the clutter from the target on the seabed. The target and the clutter were assumed to be in the form of a sphere with the same material, but they have different shell thickness the target thickness is much thinner. The inner diameters of the target and the cluster are 0.3 to 0.499m and 0.001 to 0.2, respectively, and the outer diameters are the same as 0.5m. The receiver and transmitter are located the same position, and target and clutter data are collected according to various incident angles and bottom (sandy, silt, clay). The source frequencies range from approximately 10 kHz to 30 kHz.

In the training step, the MAML-based classifier and the conventional machine learning learn the characteristics of the target and the clutter in silt and clay bottom shown in Figure 2(a). In the test phase, we compare the performance of the MAML classifier and the conventional machine learning in the new environment (sandy bottom) in figure 2(b) with respect to a zero-shot learning.

![Figure 2](image)

**Figure 2.** (a) Features of silt and clay bottom frequency and thickness of (b) Features of sandy bottom frequency and thickness from subcritical at low SNR.
4. CONCLUSION

We proposed a MAML-based active target classifier that learns a learning method and quickly adapts to a new environment with less training data or without training data. The MAML-based classifier trained general features across the task to define appropriate initialization. As a result of comparing the performance of existing machine learning with the learned MAML-based classifier for tasks that have never been seen before, the MAML-based classifier performs better than existing machine learning.

ACKNOWLEDGEMENTS

This study was supported by the research project funded by the Korea Research Institute of Ships and Ocean Engineering (KRISO, NTIS 1525012176).

REFERENCES

An on-set detection algorithm for sound recognition system using random forest

Ju-man Song(1), Changmin Kim(1), Minook Kim(1), Yongjin Park(1), Seoyoung Lee(1), and Jungkwan Son(1)

(1) Advanced Research Robotic Lab, LG Electronics, Republic of Korea, juman.song@lge.com, changmin.kim@lge.com, minook83.kim@lge.com, yongjinn.park@lge.com, seoyoung28.lee@lge.com, jungkwan.son@lge.com

ABSTRACT
This paper introduces an on-set detection algorithm for real-time sound recognition system that detects various sound events. Based on random forest algorithm, proposed algorithm is designed to generate event on-set signals for real-time applications. Using a probability vector from a VGGish model with some convolutional neural network (CNN) layers as an input signal, the proposed algorithm can determine the on-set of events required by applications. The proposed algorithm is designed as a causal system by considering real-time applications on device with a microphone. This algorithm can be used for certain classes of acoustic events with better on-set detection rates than thresholding probability vectors from network model. Through several experimental results, the performance of the proposed on-set system is shown on a real-time device.

Keywords: Sound, Insulation, Transmission

1 INTRODUCTION
Sound event detection has been researched in various ways with microphones to copy the ability of ears, which hear sounds and analyze those signals to get meaningful information. To get valuable information from microphones, some researchers use deep learning techniques that can detect various sound events. A research group of name Visual Geometry Group proposes a VGG model and Parks, Jeong and Lee suggest other CNN variants to classify acoustic scenes of 10 seconds audio files in [2]. Seo, H., Park, and J. Park, Y use multiple VGGish models which have similar structure to VGG models with ensemble method to classify sound scenes. To detect sound events on off-line system, the random forest (RF) [4] is used as deep RF in [5]. Abnormal events are detected in [6], and urban sounds are recognized in [7]. Such papers use fully connected layers, average pooling layers or just thresholding values as a decision logic.

They show sound event detection performance at specific environments, but the real world has more variety of sound environments. At first, there are so many background sounds, thus signal to noise ratio (SNR) can be changed into various values. To overcome noisy signals in real world, many efforts have been done to ensure robustness of sound recognition system [8, 9]. Secondly, the reverberation information has to be considered for real-time implementation of sound recognition system. A sound event appears in a place will propagates in various ways and if the sound source moves, then the reverberation properties will be changed into more sophisticated properties. This complex reverberate environment effects are shown in [11] that the performance of the sound recognition network model degrades as the reverberate environment differs from the training environments.

Furthermore, the sound recognition system should have worked in real-time system like robots. Many applications that react to detected sound events require low response times to use event detection results. Various applications are using deep learning techniques in real-time on device like Nvidia Jetson AGX Xavier [12]. In [13], NVIDIA’s Jetson AGX Xavier development kit was used, and model inference performance was optimized using TensorRT. With this device, in this paper, a sound recognition on-set detection algorithm using RF with VGG model is proposed. To get robustness against noisy inputs, data augmentation with various reverberation...
and SNR environments has done for a VGG model. To reflect needs of applications in real-time system, a
decision logic of RF is applied to send detection messages.

2 DESIGNING A SOUND RECOGNITION MODEL

In this section, we will develop a network model which can be worked on devices in real-time. At first, we
choose various classes as possible which are exclusive and define. After that, a VGG model is designed to be
used in real-time process. To overcome complex sound environments, it is trained with data mixed with various
impulse responses and two kinds of SNR.

2.1 Selecting various classes

To determine the number of classes, we select as various as possible acoustic events in home environments, but
each classes are exclusive and definite. If some classes are inclusive, developed network would be confused to
choose in those classes. For example, if we select children playing event and speak event simultaneously, trained
model can detect female voices as children playing, because children playing event definition includes chattering
sound in playgrounds which has similar pitches to female voices. As exclusion of events is an important factor,
clarity of events is also important for sound recognition system. If the definition of certain event is ambiguous,
collected wav data can affect adversely to trained network model. Thus, with exclusive and definite classes, we
choose 29 classes to be recognized in home environments.

In addition to 29 selected classes, there is one more supplement class with the name of noise. This class
is an essential class for when the captured audio buffer has no class to choose in classes of model. Further-
more, classes which would be not used in applications will be determined as noise class in the decision logic.
Data for noise class were collected by recording acoustic wav in a living room where there is nobody and no
particular events. Data for other classes are collected in various ways. Some sound data are recorded in home
environments or experimental places. We also use crowd workers to get big data. The list of selected events is
described in Table I.

2.2 Data augmentation

For the robust operation of the proposed sound event detection system, we increase the size of data in various
ways. The first method is data augmentation to various impulse responses. Impulse response is a important
factor in real-time sound recognition, because the performance is restricted to the environments where training
data is obtained is used for training as shown in [11]. To get various impulse responses, we measure impulse

<table>
<thead>
<tr>
<th>Class Number</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class name</td>
<td>applause</td>
<td>baby cry</td>
<td>bike bell</td>
<td>blender</td>
<td>cat</td>
<td>dog</td>
</tr>
<tr>
<td>Class Number</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>Class name</td>
<td>doorbell</td>
<td>doorlock</td>
<td>drilling</td>
<td>electric shaver</td>
<td>engine idling</td>
<td>fan motor</td>
</tr>
<tr>
<td>Class Number</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Class name</td>
<td>glass break</td>
<td>gun shot</td>
<td>jackhammer</td>
<td>keyboard</td>
<td>kitchen hood</td>
<td>klaxon</td>
</tr>
<tr>
<td>Class Number</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>Class name</td>
<td>knock</td>
<td>laugh</td>
<td>motorcycle</td>
<td>music</td>
<td>noise</td>
<td>scream</td>
</tr>
<tr>
<td>Class Number</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>Class name</td>
<td>siren</td>
<td>sneeze</td>
<td>snore</td>
<td>speak</td>
<td>toilet</td>
<td>vibration</td>
</tr>
</tbody>
</table>
responses of our laboratory and get impulse responses of $T_{60}$ range from 0.21 to 1.10.

With twelve impulse responses, we mix the original wav set with two kinds of SNRs, 10dB and 20dB. Belongs to SNR, performance of sound event detection is remarkably changed, thus we add cafe noise with SNR 10dB and 20dB. The cafe noise is recorded in a cafe place for an hour in Seoul, and we use random start points to add the randomness of signals. Therefore, there are 25 datasets of 12 impulse responses, 2 SNRs and an original set.

2.3 Feature Extraction
The sample rate of mixed DB is 16kHz, in other words, there are 16000 samples in one seconds. In our network model, a frame has 320 audio samples, and we transformed frames into power spectrogram by shifting every 320 samples with 640 length Hann window. For the Fourier transform, 1024 points transform is used. Then, a spectrum of 96 frames was yielded from 1.92 seconds audio file, and each spectrum was compressed into 128 bins of Mel frequency scale. The final shape of the input feature becomes $[128 \times 96 \times 1]$.

![Figure 1. The block diagram of sound recognition model VGG11.](image)

2.4 Model Design
To prove the performance of the proposed on-set detection algorithm, we use the VGG11 model as a base deep learning model, which is usually used in many sound detection researches. However, to enhance the real-time operation, we do not use padding in CNN layers. If paddings are not used in CNN layers, output layers of CNN block have lower dimensions than input layers of CNN block. Additionally, if there are too many CNN layers in the model, the output of last CNN layer has smaller dimension then the number of sound event classes. Thus, among VGG models in [1], we can only use VGG11 in this paper.

Each convolutional block has one or two convolutional layer of $[3 \times 3]$ size kernel, a batch normalization layer, an activation layer of ReLU, a max pooling layer and a dropout layer as depicted in Figure.1. Conv block 1 has one CNN layer, and conv block 2 has two CNN layers. Kernels of convolutional layers in our model are initialized with He normal distribution [14]. Each conv block has 64, 128, 256, and 512 filters. After filtering conv blocks, we use a flatten layer, three dense blocks and additional dense layer. Finally, we put a dense layer and a softmax layer to get probability vectors as the output of our model. Dense block has a dense layer and
a ReLU activation layer. The overall structure of VGGish model is shown in Figure 1.

3 AN ON-SET LOGIC WITH RANDOM FOREST

Through the above process, it is possible to obtain sound recognition probability vectors. However, it is still a problem for sound recognition system to figure out when to send event messages for real-time applications. Thus, in this section, a decision logic with RF algorithm will be introduced.

3.1 On-set detection logic

From a sound recognition probability vector, the easiest way to determine event detection and when to send messages to applications is taking the event of highest probability value, and send messages if the probability is higher than a threshold value. However, it is difficult to determine the threshold value with the best performance. The highest probability value can be high in high SNR while it can be low in low SNR environment. Furthermore, if a sound event not included in target classes in Table I is generated, for example a laundry sound, the probability of similar sound like the kitchen hood class can get highest value. Above situation could make false detection, because the induced class of proposed strategy is the noise class. Additionally, the output of the VGG11 model changes as the sound source moves around. Thus, it is necessary to have a decision logic to determine on-set of sound detection from varying probability vectors. To resolve such problems, we use RF as our on-set decision logic.

To use a RF as a decision logic, a probability output vector from the VGG11 model is transferred to the RF as an input vector. Then, the RF determines whether there are valuable sound events in coming sound signals, and output comes out as an integer of 0 for event on and 1 for event off. This RF logic is trained by outputs of the VGG11 model and labels of true on-set. If there is no valuable event, the output of RF is 0, and if there is an event which is required for applications, the output of RF is 1. Absent environment is also used to learn noise classes for RF. The RF consists of 15 estimators with maximum 100 depth as depicted on Fig 2.

![Figure 2. The block diagram of random forest logic.](image)

RF can also be used for specific target events. As described in section 2.1, there are already 30 classes in Table I, which are determined as many as possible. However, a device with some applications will not use them all. Just few classes would be used in real-time applications. As the target environment of our research is home environment, the target output of the RF logic is reduced to 9 classes in Table II. Then, we have to match other classes not included in Table II to noise class.

The RF decision logic can be applied to a real-time device, by using a python toolkit of name sklearn-porter [16]. It generates a c++ code from a RF model in python, and this automatically generated c++ code can be easily applied to real-time devices.
4 EXPERIMENTS

4.1 Training VGG model

The dataset has maximum 10 seconds length audios. All audio data were resampled to 16kHz and subjected to maximum normalization. Total data length of each class is about an hour in most cases, while for some classes total length of data is less than an hour. Features of the VGG11 model is extracted from the dataset of 30 classes in Table I. To evaluate training model, 81% of data is used for training and 9% of data is used for validation. Remaining 10% of dataset is used for test.

The proposed network builds an optimal model using Adam optimizer[17] with default parameters. We use the categorical focal loss function with $\alpha = 0.25$ and $\gamma = 2$. For training, we use a batch size of 180, and the maximum number of epochs is set to 150. With two NVIDIA 2080Ti GPUs, training VGG11 model of given dataset took about 4 weeks. That takes too long time for changing classes, features or other parameters in short times. If more environments are added, it would tack more than 4 weeks. However, the RF decision logic takes short time to train with probability vectors from the pre-trained VGG11 model, as it is described in section below.

4.2 Training setup of random forest decision logic

To train the RF decision logic, a stream wav with labeled on-set is used. At initial state, a stream wav has only zero value samples stream, and all test event audio dataset are added to the stream wav with 2.56 seconds with no wav duration between each event. There is also a silent duration at the front of stream wav with 10 seconds. Then, the stream wav is mixed with three impulse responses which are generated from a rir generator of [18] with 48kHz sample rate, 96,000 samples, and room dimensions of [5 x 8 x 2.5]. Each generated impulse responses has RT60s of 0.3, 0.5, and 0.7. Three stream wavs are also mixed with four kinds of SNR -10, 0, 10, and 20 dB, using brown noise.

Mixed wavs are transferred to the proposed sound recognition system with a buffer handle scheme. The buffer handle scheme sends audio data with 256 samples to the sound recognition system, and a internal buffer sends a frame of 320 samples to feature extraction system. Then VGG11 model generates sound event probability vectors. Considering independencies of events, 80% of these generated probability vectors are used for training and remaining 20% are used for test.

4.3 Time consumption in devices

To test time consumption of the proposed system, the proposed VGG11 model is trained with keras model, and converted to ONNX model which can be used in TensorRT[13]. The converted ONNX model is implemented to NVIDIA Xavier [12] with c++ codes and test using gpus of Xavier and a connected microphone. With 1,000 inferences after 10 test buffers, the VGG11 takes average 6.575ms for each inferences and real-time feature extraction. The maximum time of an inference is 10.646ms. In other words, the proposed sound recognition system can detect sound events in maximum 10.646ms. If there are other programs to use CPUs and GPUs, then, the maximum inference time could increase. The RF decision logic is also tested in same environments, and it shows about 0.2ms, thus the maximum consumed time of the proposed strategy is 10.846ms.

4.4 Experimental results from stream data

Using mixed stream wavs, on-set f1-scores are calculated for each inferences from experimental results. Stream wavs are used to test the performance of the proposed sound recognition system in controlled environments from complex properties of sound fields for exact comparative results. Table III shows experimental results comparing with and without the decision logic. Figure 3 shows that the proposed decision logic improves the performance.
Table 3. F1-scores of experiments in various environments

<table>
<thead>
<tr>
<th>SNR(dB)</th>
<th>20</th>
<th>10</th>
<th>0</th>
<th>-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without RF (RT60=0.3)</td>
<td>0.68</td>
<td>0.71</td>
<td>0.74</td>
<td>0.63</td>
</tr>
<tr>
<td>Without RF (RT60=0.5)</td>
<td>0.67</td>
<td>0.7</td>
<td>0.73</td>
<td>0.61</td>
</tr>
<tr>
<td>Without RF (RT60=0.7)</td>
<td>0.65</td>
<td>0.68</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>With RF (RT60=0.3)</td>
<td>0.84</td>
<td>0.83</td>
<td>0.83</td>
<td>0.85</td>
</tr>
<tr>
<td>With RF (RT60=0.5)</td>
<td>0.83</td>
<td>0.84</td>
<td>0.85</td>
<td>0.86</td>
</tr>
<tr>
<td>With RF (RT60=0.7)</td>
<td>0.83</td>
<td>0.83</td>
<td>0.85</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Figure 3. Experimental results with or without random forest.

of on-set with VGG11 model about 30% for each results. Especially, for -10dB SNR, the performance of on-set is increased about 35% than before using RF as the on-set decision logic. Furthermore, figure 4 shows the proposed decision logic result comparing to true label and baseline thresholding result. At each inferences, if there is an event to be detected, then the result of on-set should be 1. If there are no events to be detected, then, the result of on-set should be 0. Red dots show true label, and the blue line shows the result of baseline method. The baseline method shows wrong results with 40th to 80th inference as fault detection, and it could send false detection messages to applications. However, the result of RF on-set logic with orange line is quiet similar to true label.

5 CONCLUSIONS

In this paper, we propose an on-set decision logic for real-time sound recognition system with RF. The sound recognition model generates probability vectors of sound events and these vectors are used as inputs to the RF decision logic. The RF algorithm determines on-set to send event detection messages for applications. The proposed decision logic can also be used in real-time devices with fast response time and achieves better performance than the system without decision logic even if the environment of sound changes.
ACKNOWLEDGEMENTS
This project was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (No. 2020-0-00857)

REFERENCES


Far-field Source Localization in Spherical Harmonics Domain using Acoustic Intensity Vector

Priyadarshini Dwivedi, Gyanajyoti Routray, Rajesh M. Hegde

Indian Institute of Technology Kanpur, India
E-mail: {priyadw, groutray, rhegde}@iitk.ac.in

ABSTRACT
Source localization in the presence of reverberation and a noisy environment is still a challenging research problem and has various signal processing applications. This paper proposes a novel far-field source localization method using the acoustic intensity vector in the spherical harmonics domain. The mathematical model for the sound pressure captured by the spherical microphone array (SMA) is first developed in the spherical harmonics domain. Subsequently, the acoustic intensity vector is derived from the spherical harmonics decomposition of the pressure and acoustic velocity. As the acoustic velocity efficiently preserves the directional information, the intensity vector also contains directional and energy information. The dependency of location on the intensity vector is further explored. Since the intensity vector in the azimuth and elevation plane varies with the location, a unified convolutional neural network (CNN) model is selected to map the intensity features to the locations in reverberation and noisy conditions. Extensive simulations and experiments are conducted both on simulated and real speech data for evaluating the performance of the proposed localization method. The results show a significant improvement in localization accuracy and mean square error (MSE) compared with the state-of-art methods.

Keywords: Direction of arrival, spherical harmonic domain, learning approach, convolutional neural network

1 INTRODUCTION
Acoustic source localization is still a challenging research problem in signal processing. In this context, numerous array signal processing techniques have been developed with various microphone array configurations [1]. Among the others, circular microphone array geometry is widely accepted for source localization [2]. Since circular microphone array exhibits directional ambiguity, the performance is constrained. On the contrary, the spherical microphone array (SMA) [3] provides equal resolution in all directions, avoiding directional ambiguity and adopted for accurate source localization [4]. The direction of arrival (DOA) estimation algorithms developed are broadly categorized as time-delay of arrival (TDOA) based techniques such as generalized cross-correlation (GCC) [5], steered response power with phase transform (SRP-PHAT) [5, 6, 7]. The other category utilizes the noise-subspace for DOA estimation, including multiple signal classification (MUSIC), MUSIC group delay [8, 9], and adaptive eigenvalue decomposition [10] among the others. These algorithms based on the planar wavefront assumption and performs well for far-field. In near-field source localization, the planar wavefront assumption is irrational and leads to spurious source localization [11, 12, 13]. The spherical wavefront better approximates the near-field sources and advantageous for source localization [11]. In these techniques, the source mode strength derived from the spherical harmonics decomposition (SHD) of the SMA recordings. The spherical harmonics representation facilities the separation of the time, space, and direction-dependent parameters. Further, these separated parameters are used for localizing the source. Since the neural network models efficiently decorrelate the noise and signal components, it is investigated for source DOA estimation. In [14, 15], signal independent spherical harmonics magnitude and phase features are extracted and utilized for the DOA estimation.
This work proposes a unified convolutional neural network (u-CNN) that estimates the DOA classes from the sound intensity as the input features. The significant contributions of the proposed work are as follows. First, the sound intensity over the rigid SMA is formulated in the modal domain from the received sound pressure and the acoustic velocity. Although the sound intensity in SH has been investigated by Zuo et al. [16] and applied to spatial sound field reproduction [17], it applies to the free space. However, the proposed work aims to obtain the sound intensity over the rigid sphere. Here the sound pressure observed on the rigid sphere comprises both incident and scattered pressure, and the acoustic velocity depends only on incident pressure. Subsequently, sound intensity is decomposed into the spherical harmonics domain through spherical harmonics decomposition and features depending on azimuth and elevation are separated. Since the acoustic source is sensitive to noise and reverberation, a unified convolutional neural network (u-CNN) model is investigated that maps the azimuth and elevation classes with the derived sound intensity features. Further, exhaustive simulations and experiments are carried out to validate the performance of the proposed DOA estimation methods and compared with the state-of-art spherical harmonics multiple signal classification (SH-MUSIC) [18] and spherical harmonics phase and magnitude feature based CNN (SH-P-PM-CNN) [14] methods.

The rest of the paper is organised as follows. The system model is given in section 2. Section 3 describes the sound intensity feature extraction and the proposed learning framework for DOA estimation. The performance of the proposed model is evaluated in section 4. Finally, section 5 concludes the paper.

2 SYSTEM MODEL IN SPHERICAL HARMONICS DOMAIN

The proposed representation of sound intensity on the rigid sphere in the spherical harmonics domain is detailed in this section. Prior to it, an overview of the sound pressure and acoustic velocity is given herein. Further, the spherical harmonics intensity (SH-INT) coefficients in the azimuth and elevation directions are described.

2.1 Acoustic Scenario and Pressure in SH Domain

Consider a far-field source at \( r_s = (r_s, \theta_s, \phi_s) \), with \( r_s \) is radial distance, \((\theta_s, \phi_s)\) is the elevation and azimuth direction of the source. The source is captured by rigid SMA of radius \( r_a \) with \( L \) flush mounted microphones and placed such that the center of the microphone coincides with the origin. The complex acoustic intensity at the observation point \( r = (r, \theta, \phi) \) on the SMA in modal domain is represented by [16]

\[
I(r, k) = p^*(r, k)V(r, k)
\]

where \( I(r, k) = [I^r(r, k), I^\theta(r, k), I^\phi(r, k)]^T \) and \( V(r, k) = [V^r(r, k), V^\theta(r, k), V^\phi(r, k)]^T \) represent the acoustic intensity and acoustic velocity in the spherical coordinate system. \( p(r, k) \) represents the acoustic pressure at the observation point and \((\cdot)^*\) for the complex conjugate. \( k \) denotes the wave number related to the frequency \( f \) as \( k = 2\pi f / c \) being the velocity of sound. The real part of \( I(r, k) \) represents the active acoustic intensity which exhibits the propagation of acoustic energy and direction of propagation. The intensity vector is defined on the SMA and can be decomposed into the spherical harmonics functions as

\[
I^\rho(r, k) = \sum_{n=0}^{N} \sum_{m=-n}^{n} I^\rho_{nm}(k, r)Y_{nm}(\theta, \phi) ; \quad \rho = \{r, \theta, \phi\}
\]  

where \( I^\rho_{nm}(k, r) \) is the acoustic intensity coefficient in the \( \rho \) direction in the spherical harmonics domain. The signal recorded by the microphones of the SMA is characterized by the spherical harmonics coefficient of the acoustic pressure, which can be extracted using the spherical harmonics decomposition. The sound pressure at \( r = (r, \theta, \phi) \) in the modal domain is represented as [19]

\[
p(r, k) = \sum_{n=0}^{N} \sum_{m=-n}^{n} b_{nm}(kr)[Y_{nm}(\theta, \phi)]^*s(k)Y_{nm}(\theta, \phi)
\]

where \( n \) and \( m \) denote the order and degree of the spherical harmonics basis function. During the source acquisition, the microphone also experienced the sensor and acoustic noise and added to the sound pressure.
For ease of representation, the noise component is dropped from the expression (3). $b_n(kr)$ is the far-field mode strength and composed of both the incident and scattered sound pressure, represented as

$$b_n(kr) = 4\pi i^n \left[ j_n(kr) - \frac{j'_n(kr)}{h_n(kr)} h_n(kr) \right]$$

where $j_n(\cdot)$, $h_n(\cdot)$ represent the spherical Bessel function of first kind and Hankel function of second kind respectively. $(\cdot)'$ is the derivative of the function. $Y_{nm}(\theta, \phi)$ is spherical harmonics basis function defined as

$$Y_{nm}(\theta, \phi) = \sqrt{\frac{2n+1}{4\pi} \frac{(n-m)!}{(n+m)!}} P_{nm}(\cos \theta) e^{im\phi}$$

where $P_{nm}$ is the associated Legendre function. The total sound pressure experienced by all the microphone of the SMA is expressed in matrix form as

$$p(k) = Y(\theta, \phi) B(kr) Y^H(\theta, \phi) s(k)$$

The received pressure by the SMA is represented in the spherical harmonics domain by multiplying $Y^H(\theta, \phi)$ on both sides of the above expression and expressed as

$$p_{nm}(k) = B(kr) Y^H(\theta, \phi) s(k) \quad \forall k. \tag{7}$$

where $p_{nm}(k)$ represents the spherical harmonics pressure coefficient. Further, it can be observed that for $n = 0$ and $m = 0$, the SH pressure coefficient $p_{00}(k)$ directly guided by source signal $s(k)$. Therefore, dividing (7) by $p_{00}(k)$ removes the dependency on $s(k)$ and normalizes (7) and expressed as

$$\tilde{p}_{nm}(k) = \frac{p_{nm}(k)}{p_{00}(k)} = \begin{bmatrix} p_{1-1}(k) & p_{10}(k) & \cdots & p_{nm}(k) \end{bmatrix}$$

2.2 Acoustic Velocity Around the SMA

The SMA is characterized as a rigid sphere on which the microphones are mounted. Further, the sound pressure experienced by the microphones is composed of both incident and the scattered component of the pressure. The total acoustic velocity is the sum of the radial and incident velocities. However, for the rigid SMA, the radial velocity is zero. Therefore the acoustic velocity is determined by the incident sound pressure. The incident sound pressure for a far-field source is expressed as

$$p_i(r, k) = \sum_{n=0}^{N} \sum_{m=-n}^{n} c_{nm}(k) j_n(kr) Y_{nm}(\theta, \phi)$$

where, $c_{nm}(k) = 4\pi (-i) h_n(kr) \left[ Y_{nm}(\theta, \phi) \right]^* s(k)$ are the sound pressure coefficients. The acoustic velocity $V(r, k)$ and the pressure are related through the momentum conservation or Euler equation and expressed as

$$V(r, k) = \frac{i}{k c \delta_0} \nabla p_i(r, k)$$

where $\delta_0$ is the air density and $\nabla$ represents the gradient in the spherical domain given as

$$\nabla(\cdot) = \frac{\partial(\cdot)}{\partial r} \hat{r} + \frac{1}{r} \frac{\partial(\cdot)}{\partial \theta} \hat{\theta} + \frac{1}{r \sin \theta} \frac{\partial(\cdot)}{\partial \phi} \hat{\phi}$$

where $\{\hat{r}, \hat{\theta}, \hat{\phi}\}$ represents the unit vector. Since the presented work focuses on DOA estimation, the azimuth and elevation acoustic velocity parameters are considered and are expressed as

$$V^\theta(r, k) = \frac{i}{k c \delta_0} \frac{1}{r} \frac{\partial p_i(r, k)}{\partial \theta} \tag{12}$$

$$V^\phi(r, k) = \frac{i}{k c \delta_0} \frac{1}{r sin \theta} \frac{\partial p_i(r, k)}{\partial \phi} \tag{13}$$
3 ACOUSTIC INTENSITY AROUND THE SMA

The acoustic intensity around the SMA is obtained from the total pressure and the acoustic velocity using (1). Further, the spherical harmonics intensity (SH-INT) coefficient corresponding to the azimuth and elevation are discussed herein.

3.0.1 SH-INT Coefficients in the φ Direction

The acoustic velocity corresponding to azimuth is calculated by substituting (9) in (13) and expressed as

$$V^\phi(r, k) = \frac{1}{kc} \sum_{n=0}^{N} \sum_{m=-n}^{n} im\alpha_{nm}(k) \frac{j_{n}(kr)}{r \sin \theta} A_{nm} P_{nm}(\cos \theta) e^{im\phi}$$  

(14)

where $A_{nm} = \sqrt{\frac{2n+1}{4\pi} \frac{(n-m)!}{(n+m)!}}$. The modal domain representation for the intensity vector in $\phi$ direction is obtained by substituting (3) and (14) in (1) and expressed as

$$I^\phi(r, k) = \sum_{n=0}^{N} \sum_{m=-n}^{n} \sum_{\hat{m}=-\hat{n}}^{\hat{n}} \frac{i \epsilon_{nm} A_{\hat{n} \hat{m}} A_{nm} \hat{P}_{nm}(\cos \theta)}{r \sin \theta} \frac{\mathcal{H}_{nm}(kr)}{\mathcal{F}_{nm}(kr)} \frac{Y_{nm}(\theta, \phi) Y_{\hat{n} \hat{m}}(\theta, \phi)}{r \sin \theta} \sin \theta d\theta d\phi$$

(15)

where $\hat{P}_{nm}(\cos \theta) = \frac{P_{nm}(\cos \theta)}{\sin \theta}$. The integral of product of three Legendre polynomial function is solved using [20], and the SH-INT coefficient in the $\phi$-direction is expressed as

$$I^\phi_{nm}(r, k) = \sum_{n=0}^{N} \sum_{m=-n}^{n} \sum_{\hat{m}=-\hat{n}}^{\hat{n}} \frac{i \epsilon_{nm} A_{\hat{n} \hat{m}} A_{nm}}{r} \mathcal{H}_{nm}(kr) \mathcal{F}_{nm}(kr)$$

(17)

where $\epsilon_{nm} = 2\pi$, if $\hat{m} - m - \hat{n} = 0$ and $\epsilon_{nm} = 0$, elsewhere. $\mathcal{F}_{nm}(kr)$ is given as [16, 21]

$$\mathcal{H}_{nm}(kr) = \mathcal{H}_{nm}(kr) \left( \frac{m + \hat{n} + \hat{m} + 1}{2}, \frac{4 - d_{m+n} - d_{d_{m+n}}}{2}, \frac{1 + m - n - d_{m+n}}{2}, \frac{1 + \hat{m} - \hat{n} - d_{\hat{m} + \hat{n}}}{2}, \frac{1 + \hat{m} - \hat{n} - d_{\hat{m} + \hat{n}}}{2}, \frac{1 + \hat{m} - \hat{n} - d_{\hat{m} + \hat{n}}}{2}, \frac{2 + \hat{m} + \hat{n} - d_{\hat{m} + \hat{n}}}{2}, \frac{2 + \hat{m} + \hat{n} - d_{\hat{m} + \hat{n}}}{2}, \frac{m + 1}{2}, \frac{\hat{m} + 1}{2}, \frac{\hat{n} + 1}{2} \right)$$

(18)

where with $\mathcal{H}_{nm}(kr) = H(n, m) H(\hat{n}, \hat{m}) H(\hat{n}, \hat{m})$, $H(n, m) = (-1)^{m(n + m)!}/[2^{m} n! (n - m)!]$ and

$$\mathcal{H}(\alpha, \beta; -1, -n_{2}, -n_{3}; a_{1}, a_{2}, a_{3}; c_{1}, c_{2}, c_{3}) = \sum_{j_{1}=0}^{n_{1}} \sum_{j_{2}=0}^{n_{2}} \sum_{j_{3}=0}^{n_{3}} \left( \begin{array}{c} n_{1} \end{array} \right) \frac{(-1)^{j_{1}} (a_{1})_{j_{1}}}{(c_{1})_{j_{1}!}} \left( \begin{array}{c} n_{2} \end{array} \right) \frac{(-1)^{j_{2}} (a_{2})_{j_{2}}}{(c_{2})_{j_{2}!}} \left( \begin{array}{c} n_{3} \end{array} \right) \frac{(-1)^{j_{3}} (a_{3})_{j_{3}}}{(c_{3})_{j_{3}!}} \right)$$

where $B(j) = \beta_{j}$ is the beta function, $(a)_{j} = \left\{ \begin{array}{ll} 1, & \text{if } j = 0 \\ a(a+1) \ldots (a+j-1), & \text{if } j = 1, 2, \ldots \end{array} \right.$ and $d_{M} = \left\{ \begin{array}{ll} 1, & \text{if } M \text{ is even} \\ 0, & \text{if } M \text{ is odd} \end{array} \right.$
3.0.2 SH-INT coefficients in the \( \theta \) direction

The SH-INT coefficients in the \( \theta \) direction is obtained similar to the SH-INT coefficient in the \( \phi \) direction using (9), (12), (3), and (1). The intensity vector in \( \theta \) direction is derived as

\[
I^\theta(r,k) = \sum_{n=0}^{N} \sum_{m=-n}^{n} i^{n} \tilde{P}_{nm}(kr) \psi_{nm}(\theta, \phi) A_{nm} P_{nm}(\cos \theta) e^{im\phi}
\]

(19)

where \( \tilde{P}_{nm}(r,k) = \frac{1}{\sqrt{4\pi}} \beta_{nm}(k) \times \alpha_{nm}(k) j_{n}(kr) \). The SH-INT coefficient in \( \theta \) direction is obtained similar to \( \phi \) direction and expressed as

\[
I_{rn}^\theta(r,k) = \sum_{n=0}^{N} \sum_{m=-n}^{n} \sum_{\hat{n}=0}^{\hat{m}} \sum_{\hat{m}=-\hat{n}}^{\hat{m}} A_{nm} A_{\hat{n}\hat{m}} \mathcal{P}_{nmn\hat{n}\hat{m}} \mathcal{T}_{nmn\hat{n}\hat{m}}(r,k)
\]

(20)

where \( \mathcal{P}_{nmn\hat{n}\hat{m}} = (\hat{n} - \hat{m} + 1)P_{1} - (n + 1)P_{2} \) with \( \mathcal{T}_{nmn\hat{n}\hat{m}} = H(n,m)H(\hat{n}, \hat{m})H(\hat{\theta}, \hat{\phi}) \) and

\[
P_{1} = \mathcal{H}_{nmn\hat{n}\hat{m}} \left( \frac{m + \hat{m} + \hat{n} + 1}{2}, \frac{4 - d_{m+n} - d_{\hat{m}+\hat{n}+1} - d_{\hat{m}+\hat{n}}}{2}, \frac{1 + m - n - d_{m+n}}{2}, \frac{\hat{m} - \hat{n} - d_{\hat{m}+\hat{n}} + 1}{2} \right)
\]

\[
P_{2} = \mathcal{H}_{nmn\hat{n}\hat{m}} \left( \frac{m + \hat{m} + \hat{n} + 1}{2}, \frac{5 - d_{m+n} - d_{\hat{m}+\hat{n}} - d_{\hat{m}+\hat{n}+1}}{2}, \frac{1 + m - n - d_{m+n}}{2}, \frac{\hat{m} - \hat{n} - d_{\hat{m}+\hat{n}}}{2} \right)
\]

4 PERFORMANCE EVALUATION

This section discusses the extraction of the spherical harmonics intensity coefficients from the SMA recordings in the \( \theta \) and \( \phi \) direction. Subsequently, the u-CNN learning model is investigated to map these features with the DOA classes. The training and testing data are used to evaluate the performance of the proposed system. Further, the DOA is estimated for the real-time recording using Eigenmike using different methods.
4.1 Feature Extraction and Dataset Generation

The SH-INT coefficients in the elevation ($\theta$) and azimuth ($\phi$) direction are obtained from the SMA recordings. First, the SHD is applied to the recordings, followed by removing the mode strength to obtain the normalized spherical harmonics pressure coefficients. Subsequently, the acoustic velocity is obtained for the $\theta$ and $\phi$ directions. Using the pressure and the velocity component, the acoustic intensity is calculated. The complete process is illustrated in Figure 1. In this work, Eigenmike with $r_a = 4.2$ cm and 32 microphones is considered for recording the sound source. The SH-INT coefficients $I_{\theta m}$ and $I_{\phi m}$ are obtained using (20) and (17). These features are considered the desired features for training the CNN. The model learns these features corresponding to the labels of DOA classes. For given features set $\Xi$ containing the DOA information, the CNN computes the posteriori probability $Pr(\{\theta_s, \phi_s\}|\Xi)$, which is trained with these SH-INT features and labels.

$$\{\hat{\theta}, \hat{\phi}\}_s = \arg\max_{\theta_s, \phi_s} Pr(\{\theta_s, \phi_s\}|\Xi)$$  \hspace{1cm} (21)

The probability $Pr(\theta_s, \phi_s|\Xi)$ is obtained from the learning approach. The CNN classifies the input SH-INT features to classes $\theta_s \in [0^\circ, 10^\circ, \ldots, 350^\circ]$ and $\phi_s \in [10^\circ, 20^\circ, \ldots, 170^\circ]$.

Exhaustive synthetic data is generated using Spherical microphone array room impulse response (SMIR) [22] for the training and testing of the CNN model. The data is generated for various room dimensions with variation in room reverberations ($RT_{60}$) and signal to noise (SNR). The dimension for the room is taken as $5 \times 6 \times 7$ m with random variation in the range $\pm 2$ m. $RT_{60}$ and SNR are considered in the range $[0.2 \text{–} 0.9]$ s and $[5 \text{–} 15]$ dB. The center of the SMA coincides with the center of the room and the source is placed at a distance of 1 m from the center of the SMA. The azimuth and elevation angles for the are taken from $[0^\circ \text{–} 360^\circ]$ and $[0^\circ \text{–} 180^\circ]$ respectively in steps of $5^\circ$. The speech signals are taken from LIBRISPEECH [23] dataset. The received signals are analyzed in the frequency domain using STFT, with 512 length Hanning window and 50% overlap.

4.2 Unified CNN Framework

A CNN architecture, as illustrated in Figure 1 is adopted that maps the input SH-INT features to DOA classes. The model considers the spectrogram of the speech intensity as the input. The framework contains two convolutional layers having kernel size $(3 \times 3)$ with a stride of 1, for feature learning followed by batch normalization and max-pooling of size $(1 \times 2)$. The learning layer is followed by two dense layers and the final layer. Each layer contains rectified linear unit (ReLu) activation function, except the final layer, which has a sigmoid activation function. A dropout of 0.2 is included before the dense layer to avoid over-fitting. During compilation, binary cross-entropy is used as the loss function, and the optimizer was adam with a learning rate of 0.01. The
model provides the posterior probability for classification as output.

4.3 Result Analysis

The performance of the proposed far-field DOA estimation is evaluated using root mean square error (RMSE) expressed as $RMSE[\theta] = \sqrt{\frac{1}{Q} \sum_{q=1}^{Q} (|\Omega_q - \hat{\Omega}_q|^2)}$. Where $Q$ is the total number of samples, $\Omega_q$ is the $q^{th}$ actual DOA of the source, and $\hat{\Omega}_q$ is the DOA estimate. Figure 2 illustrates the performance of various approaches for DOA estimation in far-field against signal-to-noise ratio (SNR) and reverberation $RT_{60}$. For test conditions, the dataset is generated considering random room conditions. The RMSE is evaluated for $SNR \in [5–30]$ dB in steps of 5 dB and $RT_{60}$ in the range $[0.25–1]$ s in steps of 0.25 s, for the entire range of $(\theta_s, \phi_s)$. These RMSE were averaged and plotted as illustrated in Figure 2. From Figure 2, it is observed that the proposed method shows a significant improvement in the RMSE compared to SH-MUSIC [18] and SH-P-PM-CNN [14] for variation in SNR and $RT_{60}$.

Further, the performance is also evaluated for the real-time recordings. The recordings are taken in a lab environment. The error for SH-MUSIC is $12^\circ$, and for SH-P-PM-CNN is $10^\circ$, where the proposed SH-INT feature-based DOA estimation error is $8^\circ$. Therefore, it can be concluded that the acoustic intensity features provided better DOA classification in contrast to the spherical harmonics phase and magnitude features.

5 CONCLUSION

The proposed work discusses a new method for far-field joint DOA (azimuth and elevation) estimation using spherical harmonic intensity coefficients. Further, a learning-based model is developed for training a unified CNN model for joint DOA estimation of the sound source. Performance comparison of the proposed method is made with various existing approaches. Experimental comparisons are also performed in the lab environment for all the approaches. The proposed approach for far-field joint DOA estimation performs reasonably better than existing ones in noisy and reverberant environments. In future, this work will be extended for near-field source localization for various acoustic applications.

ACKNOWLEDGEMENT

This work is supported by the SERB-DST under project no. SERB/EE/2021410.

REFERENCES


On the impact of input scaling strategies for deep learning based DOA estimation from Ambisonics signals

Nils POSCHADEL\textsuperscript{(1)}; Roman KIYAN\textsuperscript{(1)}; Stephan PREIHS\textsuperscript{(1)}; Jürgen PEISSIG\textsuperscript{(1)}

\textsuperscript{(1)}Institute of Communications Technology, Leibniz University Hannover, Germany
{poschadel, kiyan, preihs, peissig}@ikt.uni-hannover.de

ABSTRACT
When training deep neural networks, input scaling plays an important role and can contribute decisively to the performance of the resulting models. Here, we focus on the application of estimating the direction of arrival (DOA) from noisy Ambisonics speech signals with convolutional recurrent neural networks. The input features used for training the models are either amplitude and phase spectrograms or spectrograms of features derived from the intensity vector. In this work we systematically evaluate different input scaling strategies at the level of both audio data and spectrograms, as well as combined scaling. Our investigations give insights in the dependence of DOA estimation accuracy on various combinations of scaling across different dimensions of the input data. We evaluate both regression and classification models as well as single- and multi-speaker scenarios. Our results might serve as a guidance for design choices of preprocessing methods for similar applications.

Keywords: DOA, FOA, Input Scaling, Feature Scaling

1 INTRODUCTION
Direction of arrival (DOA) estimation is a well-known task in audio signal processing and an important component of many applications such as speech separation or speech enhancement. In addition, Ambisonics-based audio signal processing is becoming increasingly popular due to its flexibility and generalizability. Therefore, DOA estimation based on first-order Ambisonics (FOA) and higher-order Ambisonics (HOA) signals has been the subject of much attention \cite{11, 12}. Also in competitions such as the DCASE Challenge, DOA estimation with FOA signals is a highly regarded topic \cite{10}. Usually, spectrogram-based input features are used. Spectrogram features derived from the intensity vector (Int) have proven to be particularly suitable \cite{9, 11, 12}. However, decent performances can also be achieved with magnitude and phase (MP) spectrograms \cite{2, 12}.

An important component of such deep learning-based approaches is data preprocessing which usually involves input/feature normalization or scaling, with these terms often used interchangeably. There are different possible normalization strategies. On the one hand, each element of a dataset can be normalized individually; on the other hand, the totality of all elements in the dataset can be normalized as a whole, whereby these strategies serve different purposes. By normalizing each element individually, one can attempt to remove information/bias/variance from the data that is useless to the task, thus simplifying training and convergence. In addition, it has been seen that it is usually beneficial if the input features of a neural network are all in a similar order of magnitude \cite{3}. Different normalization strategies have been used in deep learning based DOA estimation, but to our knowledge without systematically investigating the differences and suitability of the approaches.

In this work, we will therefore test different input normalization strategies and analyze their influence on the DOA prediction accuracy. However, due to the large number of methods and parameters, a fully comprehensive and exhaustive investigation is not possible here either. We therefore mainly focused on investigating the influence of spectrogram scaling along the channel and/or frequency dimension, uniformly across the entire dataset. The question here is, for example, whether useful information about inter-channel dependencies is lost in a channel-by-channel normalization for DOA estimation, as it is and has been done, for example, in the baseline of the Sound Event Localization and Detection (SELD) DCASE challenge \cite{2, 1}. In addition, we inves-
tigate whether individual normalization of the audio data (in this case peak normalization) has a positive effect when using MP spectrograms. Individual normalization is already applied when calculating Int spectrograms, so normalization of the audio data would be compensated there and would have no further effect.

There are numerous input/feature scaling methods in the context of machine or deep learning like max-abs normalization, min-max normalization (also known as scaling to a range), logarithmic scaling, energy normalization, or standardization (also known as z-score normalization) as well as methods like peak normalization, rms normalization, or loudness normalization of audio signals. In this study, we have limited ourselves to standardization as a feature scaling method, since this is generally quite versatile and seemed to be the most suitable approach for the respective features. The basic principle of standardization is to scale the features in the (training) dataset so that they have zero-mean and unit-variance over the entire dataset. Details are described in Sec. 2.3. However, there are certainly many other methods that are at least as useful in some cases. Min-max normalization, for example, could be suitable for phase spectrograms due to the limited range of values, but is probably rather unsuitable for magnitude spectrograms. Logarithmic scaling, on the other hand, could be useful for magnitude spectrograms, but probably not so much for Int and phase spectrograms. Therefore, in order to keep the scope of these investigations manageable, we have restricted ourselves to the proven method of standardization. This is also used, for example, in the baseline of the SELD DCASE challenge [2, 1], where each combination of frequency and channel in the spectrogram is standardized independently.

Besides the different normalization strategies and MP/Int spectrograms, regression and classification models as well as single-source and multi-source models are considered. Despite the relatively large number of different approaches, parameters and models (84 different models in total), the investigations described here are not fully comprehensive but may serve as a guidance for preprocessing methods in similar applications. Since the principle and the results are presumably transferable, we have also restricted ourselves to FOA instead of general HOA and only consider up to two sound sources/speakers.

We present the details of the datasets and input features used in this work in Sec. 2. The configuration of the models and the metrics are described in Sec. 3. Finally, the results based on simulated and measured data are compared and discussed in Sec. 4 and summarized in Sec. 5.

2 DATA

2.1 Dataset

The training, validation and testing data was synthesized from a set of spatial room impulse responses (SRIRs) simulated with the MCRoomSim toolbox [14] as Ambisonics signals with N3D normalization. Further details are described in [11]. Altogether the dataset contained 8000, 500, and 500 rooms with random dimensions in the range of \( [3, 20] \times [3, 20] \times [3, 5] \) m for the training, validation, and testing set, respectively. The SRIRs were convolved with a randomly chosen sentence from the TIMIT database [4]. In the one-source case, ambient babble noise was added to the speech signal at a signal-to-noise ratio (SNR) between 0 and 20 dB. Finally, these sentences were cut to one-second-sequences.

For the two-source case, another SRIR was selected belonging to the same room but to a different source and having an angular distance of at least 15° from the first source. This SRIR was then convolved with a different speech sample from the TIMIT database. The second Ambisonics speech signal was added to the first one at a random signal-to-interference-ratio (SIR) between 0 and 10 dB. The signals were cut to the minimum length of the respective individual speech signals, such that the respective target number of speakers is active the entire duration of the signal. Again, ambient noise was added to the speech signal at a SNR of 20 dB and the resulting signals were cut to one-second-sequences.

In the evaluation, we used both a test set based on simulated SRIRs and an additional set consisting of Ambisonics speech signals synthesized from SRIRs measured in a real room at the Institute of Communications Technology of the Leibniz University Hannover [5]. We measured the SRIRs from each of our 36 KH120 loudspeakers to an em32 EigenMike\textsuperscript{®} [8] microphone at nine different positions, each with two different heights and eight different orientations of the microphone. In total, the described procedure led to 5184 measured SRIRs, which were afterwards encoded to an Ambisonics signal using the EigenUnits-em32-encoder\textsuperscript{1}. These measured

\textsuperscript{1}https://mhacoustics.com/products
SRIRs were handled according to the same procedure as for the simulated SRIRs to generate Ambisonics single- and multi-speaker signals.

### 2.2 Input features

As reported in the introduction in Sec. 1, we use both MP and Int spectrograms of FOA speech signals as input features for our models. Let \( F(t, f) \in \mathbb{C}^{50 \times 512 \times 4} \) be the Short-time Fourier transform (STFT) of the four-channel FOA speech signal with \( F(t, f) = [W(t, f) X(t, f) Y(t, f) Z(t, f)]^T \). Then the MP spectrograms are obtained by simply concatenating all the magnitude spectrograms with all the phase spectrograms along the channel axis.

On the other hand, Perotin et al. [9] proposed using the STFT of six-channel features derived from the FOA sound intensity vector according to (1) as input to the model. By using these Int spectrograms, they were able to improve the localization performance compared to using MP spectrograms.

\[
\begin{align*}
\frac{1}{C(t,f)} \begin{bmatrix}
I_a(t,f) \\
I_r(t,f)
\end{bmatrix} &= \begin{bmatrix}
\text{Re} \{W(t,f)X^*(t,f)\} \\
\text{Re} \{W(t,f)Y^*(t,f)\} \\
\text{Re} \{W(t,f)Z^*(t,f)\}
\end{bmatrix} - \begin{bmatrix}
\text{Im} \{W(t,f)X^*(t,f)\} \\
\text{Im} \{W(t,f)Y^*(t,f)\} \\
\text{Im} \{W(t,f)Z^*(t,f)\}
\end{bmatrix} \\
C(t,f) &= |W(t,f)|^2 + \frac{1}{2} \left(|X(t,f)|^2 + |Y(t,f)|^2 + |Z(t,f)|^2\right). \\
\end{align*}
\]

\( I_a(t,f) \) and \( I_r(t,f) \) describe the active and reactive intensity vector as a STFT expression of the FOA channels and \( C(t,f) \) is a normalization term according to (2). The normalization performed here with \( C(t,f) \) is already an individual time- and frequency-specific normalization (with the sound field energy). An additional peak normalization of the speech signal comparable to the one in the case of MP spectrograms described above would have no effect, as this would be compensated by \( C \). Since this normalization is already well established and proven to work, we did not examine any models without this normalization in order to keep the scope of these investigations smaller. For further details on these features and acoustic intensity in particular, see [9, 13].

### 2.3 Scaling methods

As discussed in Sec. 1, we have limited ourselves to standardization/z-score normalization as input normalization method. When standardizing the input, the data are shifted and scaled to have zero-mean and unit-variance for all features and across the whole (training) dataset. Therefore, the mean and standard deviation must first be calculated for each feature in the (training) dataset. Subsequently, the mean value is subtracted from each feature and the result is divided by the standard deviation of the feature. However, it must first be determined which data should be considered a separate feature and which data make up a common feature and should be scaled together.

Here, we have varied joint and separated scaling along the frequency and/or the channel dimension. The different cases and corresponding abbreviations are listed in Table 1. For the frequency dimension, there are only two cases: Separate scaling along the frequency dimension (Yes) and joint scaling of all frequencies (No). For the channel dimension, on the other hand, we distinguished 3 cases: Again, we investigated joint scaling (All) and separate scaling (Individual) of the channels. In the third case (Act./React. or M/P), all channels corresponding to the active intensity (or magnitude) are scaled together and all channels corresponding to the reactive intensity (or phase) are scaled together. This allows for both a compensation for the different magnitudes of the features (especially in the case of MP) and the preservation of the information about the inter-channel dependencies. In the following evaluations, the abbreviation None is used, when no standardization at all is applied.

<table>
<thead>
<tr>
<th>Channel</th>
<th>All Act./React. M/P Individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>Yes No</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To compensate e. g. for different distances of the sound sources or speech signal levels, we additionally investigated whether peak normalization of the individual speech signals could lead to more comparable data.
(especially for the non-simulated SRIRs) and thus better results. As mentioned above, such a normalization is already obtained for Int spectrograms by the normalization according to (1) and (2) and therefore only considered for MP spectrograms.

3 NETWORKS AND METRICS

The deep learning models used in this study are convolutional recurrent neural networks (CRNNs) that follow the same structure as the ones in [11, 12] and were implemented within the TensorFlow platform. A detailed overview of the network’s architecture is given in Table 2. The input shape of all the different networks is \((50, 512, \text{dim}_{\text{in}})\), where 50 is the number of frames, 512 the number of frequency bins, and \(\text{dim}_{\text{in}}\) the number of input channels with \(\text{dim}_{\text{in}} = 8\) for the MP CRNNs and \(\text{dim}_{\text{in}} = 6\) for the Int CRNNs. The output dimensions are \(\text{dim}_{\text{out}} = 3 \cdot N_s\) for the regression and \(\text{dim}_{\text{out}} = 425\) for the classification models, with \(N_s\) being the number of active sound sources. The STFT for the creation of the spectrograms was performed using the \texttt{stft} function of the python package librosa [7] on 640 samples, zero-padded to 1024 samples with a hop-size of 320 samples and using a Hann window. For all trainings, we used the Nadam optimizer. The activation function and dimension of the output layer depend on the model being a regression or classification model.

### Table 2. Architecture of the CRNNs for DOA estimation.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Details</th>
<th>Output Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Spectrograms</td>
<td>((50, 512, \text{dim}_{\text{in}}))</td>
<td></td>
</tr>
<tr>
<td>Conv2D</td>
<td>(3 \times 3)</td>
<td>((50, 512, 64))</td>
</tr>
<tr>
<td>BatchNorm</td>
<td></td>
<td>((50, 512, 64))</td>
</tr>
<tr>
<td>Activation</td>
<td>\text{elu}</td>
<td>((50, 512, 64))</td>
</tr>
<tr>
<td>MaxPooling</td>
<td>(1 \times 8)</td>
<td>((50, 64, 64))</td>
</tr>
<tr>
<td>Dropout</td>
<td>(0.2)</td>
<td>((50, 64, 64))</td>
</tr>
<tr>
<td>Conv2D</td>
<td>(3 \times 3)</td>
<td>((50, 64, 64))</td>
</tr>
<tr>
<td>BatchNorm</td>
<td></td>
<td>((50, 64, 64))</td>
</tr>
<tr>
<td>Activation</td>
<td>\text{elu}</td>
<td>((50, 64, 64))</td>
</tr>
<tr>
<td>MaxPooling</td>
<td>(1 \times 8)</td>
<td>((50, 8, 64))</td>
</tr>
<tr>
<td>Dropout</td>
<td>(0.2)</td>
<td>((50, 8, 64))</td>
</tr>
<tr>
<td>Conv2D</td>
<td>(3 \times 3)</td>
<td>((50, 8, 64))</td>
</tr>
<tr>
<td>BatchNorm</td>
<td></td>
<td>((50, 8, 64))</td>
</tr>
<tr>
<td>Activation</td>
<td>\text{elu}</td>
<td>((50, 8, 64))</td>
</tr>
<tr>
<td>MaxPooling</td>
<td>(1 \times 4)</td>
<td>((50, 2, 64))</td>
</tr>
<tr>
<td>Dropout</td>
<td>(0.2)</td>
<td>((50, 2, 64))</td>
</tr>
<tr>
<td>Reshape</td>
<td></td>
<td>((50, 128))</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>((50, 128))</td>
<td></td>
</tr>
<tr>
<td>BiLSTM</td>
<td>((50, 128))</td>
<td></td>
</tr>
<tr>
<td>Time-Dist. Dense</td>
<td>\text{elu}</td>
<td>((50, 128))</td>
</tr>
<tr>
<td>Dropout</td>
<td>(0.2)</td>
<td>((50, 128))</td>
</tr>
<tr>
<td>Time-Dist. Dense</td>
<td>\text{act}_{\text{out}}</td>
<td>((50, \text{dim}_{\text{out}}))</td>
</tr>
</tbody>
</table>

3.1 Classification

The classification models estimate whether or not each point on a predefined grid corresponds to the direction of an active source or not, assuming that the number of active sound sources is known. Here, we use the following quasi-uniform grid on the unit 2-sphere [9]:

\[
\theta_i = -90 + \frac{i}{I} \cdot 180, \quad \text{with } i \in \{0, \ldots, I\},
\]

\[
\phi_j = -180 + \frac{j}{J_i + 1} \cdot 360, \quad \text{with } j \in \{0, \ldots, J_i\},
\]

with \(I = \left\lfloor \frac{180}{\alpha} \right\rfloor\), \(J_i = \left\lfloor \frac{360 \cos(\theta_i)}{\alpha} \right\rfloor\), and a grid resolution parameter \(\alpha\) resulting in a grid of \(n_{\text{grid}} = \sum_{i=0}^{I} (J_i + 1)\) points. In this paper we set \(\alpha = 10\) which leads to a grid of \(n_{\text{grid}} = 425\) points. According to this classification setting, the target of our CRNNs is a multi-hot-encoded vector of size \(\text{dim}_{\text{out}} = n_{\text{grid}} = 425\), where each index...
corresponds to a DOA according (3). For each active speaker in the scene, the entry in the target vector corresponding to the direction closest to the respective DOA is set to one. Since we use a time-distributed dense layer as output layer, we get a $n_{\text{grid}}$-dimensional output vector for each time frame. In the one-source case, we used the softmax function as the activation function $\text{act}_\text{out}$ of the output layer together with the categorical-crossentropy loss function. In the multi-source case, we used the sigmoid function as $\text{act}_\text{out}$ and the binary-crossentropy loss function.

### 3.2 Regression
For a signal with $N_s$ active sound sources, the regression model directly outputs the Cartesian DOA predictions $\hat{\mathbf{X}}_t = [\hat{x}_1(t), \ldots, \hat{x}_{N_s}(t), \ldots, \hat{x}_{N_s}(t)]$ with $\hat{x}_i(t) = (\hat{x}_i(t), \hat{y}_i(t), \hat{z}_i(t)), i = 1, \ldots, N_s, t$ at the $t$-th time frame. The ground truth Cartesian DOA vector is denoted by $\mathbf{X}_t = [x_1(t), \ldots, x_{N_s}(t), \ldots, x_{N_s}(t)]$ with $x_i(t) = (x_i(t), y_i(t), z_i(t))$. Since we are only interested in the direction (and not the distance) of the sound source, both the ground truth DOA vectors and the estimates are normalized to unit length. For the estimates, this is achieved by adding a final (time-distributed) normalization layer to the model described in Table 2 which scales every Cartesian DOA vector to unit length. The final activation function $\text{act}_\text{out}$ is a linear function.

For the single-source regression models, the mean-squared error

$$\text{MSE}(\hat{x}_i(t), x_j(t)) := \frac{1}{3} \sqrt{ (\hat{x}_i(t) - x_j(t))^2 + (\hat{y}_i(t) - y_j(t))^2 + (\hat{z}_i(t) - z_j(t))^2 },$$

$i, j = 1, \ldots, N_s$ is used as loss function. As mentioned above, the permutation of the sources has to be taken into account for the multi-source regression models. Therefore, we used a permutation-invariant version $\text{MSE}_\pi$ of the MSE. Let $\mathcal{S}_{N_s}$ be the set of all permutations $\pi: \{1, \ldots, N_s\} \rightarrow \{1, \ldots, N_s\}$, then

$$\text{MSE}_\pi(\hat{\mathbf{X}}_t, \mathbf{X}_t) := \min_{\pi \in \mathcal{S}_{N_s}} \left( \sum_{i,j=1}^{N_s} \text{MSE}(\hat{x}_{\pi(i)}(t), x_j(t)) \right).$$

The final value of the loss function is calculated by averaging the loss values over time.

### 3.3 Metrics
The evaluations of our models are based on the angular distance $\delta_{ij} := \delta(\hat{x}_i, x_j) := \arccos(\hat{x}_i \cdot x_j / (\|\hat{x}_i\| \|x_j\|))$. Although the sources in our investigations are static, the models predict the DOA for every of the 50 time frames, so that our models can also be used for moving sources without further modification. Therefore, we also evaluate the models with a metric that is suitable for moving sources, i.e., we evaluate the predictions frame-by-frame and then average across frames.

For the regression models, the permutation of the sound sources or DOA predictions that leads to the smallest angular distance summed up over all sources is determined for each time frame. This is realized using the Hungarian algorithm [6]. These angular distances are then averaged over time for each sound source and called localization error (LE), which is the final metric considered in the following evaluations.

For the classification models, the largest $N_s$ values are determined in the output vector for each time frame, which have a minimum distance of $10^\circ$ between each other. This is to prevent neighbouring direction bins belonging to the same sound source being interpreted as two different sources. Subsequently, the DOAs belonging to the direction bins is determined. From here on, the procedure is identical to that of the regression models (Hungarian algorithm followed by time-average of angular distances to get the LE). This procedure slightly differs from that in [12]. However, the procedure here is more comparable to that in the multi-source regression and thus fairer in a direct comparison of the approaches.

### 4 RESULTS
The localization performances of the different models are shown in Figure 1 and Figure 2 for the one-source and two-source cases, respectively. The mean localization errors on the two testing sets are displayed in a scatterplot. When analyzing and interpreting the influence of the different input normalization methods, it must be taken into account that the differences between the individual models are sometimes very small and even almost lie in the
order of magnitude of random fluctuations in the training of the models. To keep these fluctuations as small as possible, we usually trained several models for each setting and took the best model on the validation dataset. However, due to the many different settings and models to be examined in this study (overall 84 different conditions), this was not possible for all the models. Therefore, only trainings that were obviously not fully converged were repeated. The individual, absolute results should therefore be treated with a little caution. In order to gain more significance, the performances are therefore additionally evaluated in Figure 3 averaged over the different standardization approaches.

It can be seen in Figure 1a that overall the Int models perform better than the MP models on both data synthesized using simulated and real SRIRs. However, peak normalization has a comparatively large influence on the MP models and leads to a considerable improvement with some of the MP models reaching similar performance as some of the Int models.

The performance of the one-source classification models shown in Figure 1b is overall worse than the performance of the regression models. Due to the principle of discretization of directions in the classification approach, these differences may partly be explained by discretization errors. However, the results of the different preprocessing methods differ less among themselves overall, with peak normalization again having a positive impact on the accuracy of the DOA estimation.

The results for 2-source models in Figure 2 show similar effects compared to those of the 1-source models: peak normalization is important regardless of the standardization approach and the Int models again perform considerably better than the MP models, especially in the regression formulation. However, it should be noted that appropriate preprocessing of the MP spectrograms in the classification setting leads to comparable accuracy of the MP models on the test set based on simulated SRIRs.

Figure 3 shows the mean LE for the different standardization approaches averaged over all the different models that meet the labeled criterion, with the idea of eliminating the influences of the random fluctuations in the training process as much as possible. For example, the scatter “F-yes” includes all one-source, two-source, regression and classification models, where (among other things) each frequency was standardized individually.\(^2\) In the case of the Int models in Figure 3a, there is a trend that it is better to scale across fewer dimension. It seems to be best to scale all channels and all frequencies together\(^3\) In the case of the MP models, peak normalization leads to considerable improvement in the estimation accuracy for all standardization variations. In addition,\(^2\)These are e.g. all models with the labels “F, C”, “F, C\(_1\)”, and “F, C\(_{12}\)”.\(^3\)F-yes is better than F-no, C\(_1\) is better than both None and C\(_{12}\), which are again better than C.
standardizing across the frequency dimension seems to be beneficial (F-yes is better than F-no). Furthermore, it appears to be important that the differences between magnitude and phase channels is compensated for by the standardization (C and C_{12} are better than C_1, which in turn is better than None).

5 CONCLUSION

In this paper we investigated the influence of feature normalization methods on the accuracy of single- and multi-speaker DOA estimation of noisy FOA speech signals. It has been shown that an appropriate normalization method plays an important role in the data preprocessing, especially when using MP spectrograms.

For the models trained on Int spectrograms, the different standardization approaches only have minor influence on the accuracy of the DOA estimation. Nevertheless, it seems to be most suitable to scale all channels and frequencies together (F, C_1). None, channel- and/or frequency-individual standardizations achieve slightly worse results on average.

In the case of models trained on MP spectrograms, on the other hand, input scaling should be a central part of the preprocessing. Both the peak normalization of the audio data to compensate for individual signal level differences and the standardization of the spectrograms are important. In the case of standardization, it seems to be especially decisive to align the different scales of magnitude and phase spectrograms (C or C_{12}). In addition, a scaling along the frequency dimension seems to be advantageous.

Overall, the Int models achieve better results than the MP models in DOA estimation. However, one can considerably reduce the difference to the MP models by choosing an appropriate preprocessing method. Moreover, it can be noted that the regression models achieve higher accuracy than the classification models. It should be mentioned here, however, that again with appropriate preprocessing of the MP spectrograms, similar accuracy can be achieved as with the Int spectrograms, at least on the data based on simulated SRIRs.

REFERENCES


Figure 3. Scatter plot of the mean localization error for the different scaling and normalization strategies averaged over all models that meet the labeled criterion and trained on Int (a) and MP (b) spectrogram.


Steered-Response Power Methods for Joint DOA and Ambient Temperature Estimation

Daniele MIRABILII, Deeksha DESAI, Emanuël A.P. HABETS

International Audio Laboratories of Erlangen*, Am Wolfsmantel 33, 91058 Erlangen, Germany. danielemirabilii@audiolabs-erlangen.de.

ABSTRACT
Direction of arrival (DOA) estimation refers to the process of using a sensor array to retrieve the direction of an acoustic source. It represents a task of interest in, e.g., acoustic scene analysis, source separation, and multi-channel signal enhancement. The propagation speed of acoustic sources is typically considered a known constant under the assumption of dry air and a constant temperature. However, as the speed of sound depends linearly on the temperature, temperature variations can lead to significant deviations from the assumed speed value. This work aims to jointly estimate the DOA of an acoustic source and the ambient temperature using steered-response power (SRP) methods, namely, the delay and sum and the minimum power distortionless response beamformers and the MUltiple SIgnal Classification algorithm. These methods are extended to account for a variable propagation speed. Firstly, the DOA and speed of sound are estimated as the maximizers of the SRP. Secondly, the temperature is inferred from the estimated speed. The performance is assessed and compared to an existing time difference of arrival-based method with simulated and experimental results under different noise and reverberation conditions. The proposed methods could enable temperature measurements for smart-home assistants without external thermometer sensors.

Keywords: DOA, Speed of sound, Temperature, Estimation

1 INTRODUCTION
The Direction of arrival (DOA) of propagating waves is an essential parameter in applications like radar, sonar, radioastronomy, and wireless communications. An array of sensors can measure an incident source from a particular direction. Each sensor signal can be seen as an attenuated and delayed copy of the signal measured at a reference sensor. The delay between two sensors is commonly called time of flight or time difference of arrival (TDOA) and corresponds to a phase shift in the frequency domain. Assuming a constant speed of light or sound, the TDOAs can be directly mapped to the DOA. The TDOA and DOA estimation problems are well studied in the literature, and several methods have been designed and exhaustively discussed.

In specific applications, the propagation speed is not constant and represents an additional unknown variable. An example is given in seismology, where multiple distributed seismometers are used to locate the source of a seismic event. In fact, seismic waves can propagate at a variable speed depending on the wave type (p-wave, s-wave, etc.) and on the density and elasticity of the propagation medium. In this case, we are not only interested in estimating the incident angle but also the propagation speed [7, 12]. Some estimation methods assume rectilinear (or plane-wave) propagation where the wavefronts are constant in the plane normal to the propagation direction at any given time.

Concerning acoustic wave propagation in acoustic array processing, the speed of sound is typically considered known (343 ms$^{-1}$) by assuming dry air at a temperature of 20°C. However, acoustic waves propagate at a speed that depends on specific properties of the propagation medium. In the air, the sound propagation speed proportionally increases with the ambient temperature. Although air cannot be considered an ideal gas

* A joint institution of the Friedrich-Alexander-University Erlangen–Nürnberg (FAU) and Fraunhofer IIS.
and humidity might be present, these two properties contribute marginally to variations in the speed of sound. In contrast, temperature variations can lead to significant deviations from the assumed speed value that can also impact the DOA estimation [5, 6].

TDOA-based methods [15, 9, 4] have been developed to jointly localize the acoustic source and estimate the propagation speed. In contrast to DOA estimation, source localization estimates the source position and not only the direction from a geometrical calculation based on the TDOAs. Such methods are formulated on the assumption of near-field propagation, i.e., the source distance is comparable to the inter-microphone distance. In [9], the authors formulated the joint localization and propagation speed estimation on the assumption of far-field, i.e., plane-wave propagation, to mitigate the problem of an ill-conditioned system matrix in existing methods. The performance of TDOA-based methods depends on two important aspects, namely the sensors synchronization and the time resolution. The second aspect becomes crucial, especially when closely spaced microphones are employed, i.e., the sampling frequency must be sufficiently high to resolve small TDOAs correctly.

In this work, we propose three methods to jointly estimate the DOA of a point source and its propagation speed. In addition, we resolve the ambient temperature using a simplified linear model based on the propagation speed. We focus on systems with closely spaced microphones and relatively low sampling frequency (16 kHz) and assume a far-field scenario. The proposed methods are based on the pseudospectrum or steered-response power (SRP) of the microphone array and can be seen as a modified version of classical DOA-estimation methods. In particular, we employ spatial filtering methods like the Delay-and-Sum Beamformer (DSB) and the Minimum Power Distortionless Response (MPDR) beamformer, and a signal subspace-based method, namely the MUltiple SIgnal Classification (MUSIC) method. Typically, the source DOA is estimated by searching the maximum of the SRP computed against candidate DOAs. The maximizer of the SRP then gives the estimated DOA under the assumption of a known and constant propagation speed. It follows that the problem can be extended to a variable propagation speed obtaining a 2D search grid, i.e., the search of the maximum is performed across both angular direction and propagation speed. In such a case, we obtain the DOA and speed estimates as the pair of candidates that maximizes the SRP. Similarly to [4], we employ a simple linear relation between the propagation speed and the ambient temperature to estimate the latter. The methods could enable passive temperature measurements using compact microphone arrays (e.g., for smart home assistants) when external thermometers are unavailable.

The remainder of the paper is structured as follows. In Section 2, we introduce the signal model and the problem formulation. In Section 3, the TDOA-based method presented in [9] is briefly discussed as a baseline approach. In Section 4, three SRP-based methods for jointly estimating the DOA and the propagation speed are presented. In Section 5, we evaluate the performance of the proposed method and compare it to the baseline method with simulations consisting of noisy and reverberant conditions, and with measurements in a low-reverberant room.

2 SIGNAL MODEL AND PROBLEM FORMULATION

Let us assume N microphones that are spatially distributed on a plane and an acoustic source emitting from a given point in space at a sufficiently large distance from the microphones. For simplicity, we consider a 2D scenario in which the source does not exhibit any elevation with respect to the plane of the microphone array. However, the problem can be extended to a 3D scenario by including a third variable representing the elevation angle. The propagation of the source signal to the microphones is modeled as follows

$$y_i(t) = d_i(t) * s(t) + n_i(t),$$ (1)

where $y_i(t)$ denotes the $i$-th microphone signal, $s(t)$ denotes the emitting source signal, $d_i(t)$ denotes the acoustic transfer functions (TF) from the source to the $i$-th microphone, $n_i(t)$ denotes an additive stationary noise term, $t$ denotes the discrete time sample, and $*$ denotes the convolution operation. We can write Eq. (1) in the Short-time Fourier Transform (STFT) domain as

$$Y_i(l,k) = D_i(k)S(l,k) + N_i(l,k),$$ (2)
where \( l \) and \( k \) denote the time and frequency instances index, respectively. Note that we assume a time-invariant TF (e.g., static or slowly-varying source position and propagation speed). In addition, we assume mutual statistical independence between the noise terms, i.e., \( \mathbb{E}[N_i N_j] = 0 \), and between the source and the noise \( \mathbb{E}[SN_i] = 0 \) for all \( i, j \) with \( i \neq j \), where \( \mathbb{E} \) denotes mathematical expectation. Furthermore, we assume homogeneous noise contributions at the microphones, i.e., equal power \( \mathbb{E}[|N_i|^2] = \varphi_n \) for all \( i \).

The TF can model different acoustic conditions. In the free-field, the TF models the direct-path propagation from the source to the microphones that is fully characterized by the TDOA \( \tau \). In turn, \( \tau \) depends on the source DOA and the propagation speed. Given a blind scenario where the source location is unknown, the direct-path TDOA and the propagation speed. Given a blind scenario where the source location is unknown, the direct-path TDOA \( \tau \) can be computed relative to a reference point and referred to as relative direct-path TF (RDP-TF). Typically, a reference microphone can be chosen as the reference point. Therefore, we rewrite Eq. (2) in terms of RDP-TFs as

\[
Y_i(l, k) = G_i(k)S(l, k) + N_i(l, k),
\]

where \( G_i(k) = D_i(k)/D_r(k) \), \( r \) denotes the reference microphone index, and \( G_r(k) = 1 \). The RDP-TFs in free-field and far-field conditions are expressed as pure delays between microphones, so that

\[
g_i(t) = \delta(t + \tau_i) \iff G_i(k) = \exp\left(i \omega_k \tau_i \right),
\]

where \( \delta \) denotes a Dirac delta function, \( t = \sqrt{-1} \). \( \omega_k = 2\pi F_s k / K \) denotes the discrete angular frequency, \( F_s \) denotes the sampling frequency, \( K \) denotes the discrete Fourier Transform (DFT) length, and \( \tau_i \) is the TDOA between the \( i \)-th and the reference microphone. The latter can be expressed as

\[
\tau_i = \frac{p_i^T u_s}{c},
\]

where \( u_s = p_i / \| p_i \|_2 \) denotes the unit vector pointing towards the source embedding the DOA, \( p_s \) and \( p_r \) denote the vectors between the reference point and the source, and the reference point and the \( i \)-th microphone, respectively, \( T \) denotes the transpose operator, \( c \) denotes the propagation speed in ms\(^{-1} \). For a simple scenario with two microphones, Eq. (5) becomes \( \tau_i = d \cos(\theta) / c \), where \( d \) denotes the inter-microphone distance and \( \theta \) denotes the DOA with respect to the axis identified by the microphones.

We express the signal model in Eq. (3) for all microphone signals in a vector notation

\[
y(l, k) = g(k)S(l, k) + n(l, k)
\]

where

\[
y(l, k) = [Y_1(l, k), ..., Y_i(l, k), ..., Y_N(l, k)]^T
\]

\[
g(k) = [G_1(l, k), ..., G_i(l, k), ..., G_N(l, k)]^T
\]

\[
n(l, k) = [N_1(l, k), ..., N_i(l, k), ..., N_N(l, k)]^T.
\]

The objective of this work is to estimate the DOA \( \theta \) (computed with respect to a reference axis) and the propagation speed \( c \) from the noisy microphone observations \( y(l, k) \). In addition, the ambient temperature expressed in °C can be estimated from the propagation speed [4] as

\[
\hat{T} = \frac{\hat{c}(T) - 331}{0.6}.
\]

3 EXISTING TDOA-BASED METHOD

In this section, we briefly discuss the TDOA-based method presented in [9] to jointly estimate the source DOA and the propagation speed in a far-field scenario. We can notice that Eq. (5) identifies an overdetermined linear system of \( N-1 \) equations in two variables. Rearranging the terms of Eq. (5) in a matrix form for all the TDOAs yields the linear system

\[
Ax = b.
\]
where

\[ A = [p_1, \ldots, p_i, \ldots, p_N]^T, \]
\[ x = \frac{u_s}{c}, \]
\[ b = [\tau_1, \ldots, \tau_i, \ldots, \tau_N]^T. \]

The least-squares solution of \( x \) is given by

\[ \hat{x} = A^\dagger b = (A^T A)^{-1} A^T b. \] \hfill (9)

where the estimated source propagation vector is obtained as

\[ \hat{u}_s = \frac{\hat{x}}{\|\hat{x}\|_2}, \] \hfill (10)

and the estimated propagation speed as

\[ \hat{c} = \frac{1}{\|\hat{x}\|_2}. \] \hfill (11)

In practice, only \( A \) in Eq. (8) is known and \( b \) must be estimated from the microphone signals. The TDOA estimation can be performed via maximization of the cross-correlation function by computing the time lag corresponding to the cross-correlation peak between two microphone signals, i.e.,

\[ \hat{\tau}_i = \arg\max_\tau R_i(\tau), \] \hfill (12)

where \( R_i \) denotes the cross-correlation between the reference and the i-th microphone signal. Pre-processing the cross-correlation can be beneficial in reducing estimation errors caused by reverberation and noise. In this respect, the Generalized Cross-correlation (GCC) [10] is commonly employed to mitigate the effect of adverse acoustic conditions on the TDOA estimation. The GCC is the inverse Fourier transform of the weighted cross-power spectrum, where different weighting functions can be used. A popular weighting is the Phase Transform (GCC-PHAT), where the cross-power spectrum magnitude is discarded, and the inverse transform ideally yields an impulse centered at the time lag corresponding to the TDOA, i.e.,

\[ R_i(\tau) = \mathcal{F}^{-1}\{\Psi_i(l, k)\}, \] \hfill (13)

where \( \mathcal{F}^{-1}\{\cdot\} \) denotes the inverse Fourier transform and

\[ \Psi_i(l, k) = \frac{Y_i(l, k) Y^*_r(l, k)}{|Y_i(l, k) Y^*_r(l, k)|}, \] \hfill (14)

where \((\cdot)^*\) denotes the complex conjugate. The GCC-PHAT was shown to be robust against reverberation but sensitive to noise. In fact, the PHAT normalization can potentially amplify frequency bands corrupted by noise even at high signal-to-noise ratio (SNR) conditions.

4 PROPOSED STEERED-RESPONSE POWER METHODS

In this section, we propose three methods based on the SRP maximization to jointly estimate the source DOA and propagation speed in a far-field scenario. The ambient temperature is then inferred using Eq. (7). The main idea is to extend existing DOA estimation methods to account for a variable propagation speed. This is achieved by computing the SRP at candidate speed and direction pairs and subsequently selecting the pair that yields the maximum SRP, i.e.,

\[ [\hat{\theta}, \hat{c}] = \arg\max_{\theta, c} P(k; \theta, c), \] \hfill (15)
In the MPDR beamformer, we minimize the beamformer output power subject to the distortionless constraint.

### 4.2 Minimum Power Distortionless Response

The DSB output power can be computed as the expectation of the magnitude squared value of Eq. (18), i.e.,

\[ P_{DSB} = \frac{1}{L} \sum_{l=1}^{L} |y(l,k)\Phi_{y}^{H}(l,k)|^2. \]

where \( L \) is the total number of considered time frames. We first present two spatial filtering methods (beamforming), namely the DSB and the MPDR, modified to account for a variable propagation speed. In these methods, we linearly combine the microphone signals to enhance the source signal coming from a candidate direction at a candidate speed and attenuate the signals coming from undesired directions and propagating at a different speed. This is achieved by applying complex weights to the microphone signals, i.e., \( Z(l,k) = h^{H}(k)y(l,k) \). The power of the beamformer output corresponds to the SRP so that Eq. (15) can be solved to infer the source direction and propagation speed. We then present a modified version of a subspace-based method, namely the MUSIC, where we factorize the microphone signal covariance structure into a source subspace and a noise subspace and exploit the orthogonality of the RDP-TF and the noise subspace basis to compute the SRP against candidate directions and speed. We subsequently apply Eq. (15) to the MUSIC SRP.

### 4.1 Delay and Sum

The DSB aims at delaying the microphone signals so that they are in phase and their sum yields the maximum output power. This way, the directional source signal adds coherently while the mutually uncorrelated noise is attenuated. We compute the beamformer output by selecting an appropriate phase shift identified by a candidate pair \([\theta, c]\)

\[ Z_{DSB}(k, \theta, c) = \frac{1}{N} \sum_{i=1}^{N} \exp(-j\omega_{i} \tau_{i}(\theta, c)) Y_{i}(k) = h^{H}(k, \theta, c) g(k, \theta, c), \]

where the dependency on the time frame was omitted for brevity. Therefore, the complex weights of the DSB are given by

\[ h(k, \theta, c) = \frac{g(k, \theta, c)}{N}. \]

The DSB output power can be computed as the expectation of the magnitude squared value of Eq. (18), i.e.,

\[ E[|Z_{DSB}(k, \theta, c)|^2] = E[|h^{H}(k, \theta, c)g(k, \theta, c)|^2] = h^{H}(k, \theta, c) \Phi_{y}(k) h(k, \theta, c), \]

yielding the SRP

\[ P_{DSB}(k, \theta, c) = \frac{g^{H}(k, \theta, c) \Phi_{y}(k) g(k, \theta, c)}{N^2}. \]

### 4.2 Minimum Power Distortionless Response

In the MPDR beamformer, we minimize the beamformer output power subject to the distortionless constraint \( h^{H}(k, \theta, c) g(k, \theta, c) = 1 \). The idea is to minimize the energy of interfering sources and noise while keeping a unitary response at the candidate pair \([\theta, c]\). The optimum solution for the complex weights is given by

\[ h(k, \theta, c) = \frac{\Phi_{y}^{-1}(l,k) g(k, \theta, c)}{g^{H}(k, \theta, c) \Phi_{y}^{-1}(l,k) g(k, \theta, c)}. \]

from which the output power can be obtained as in Section 4.1, yielding

\[ P_{MPDR}(k, \theta, c) = \frac{1}{g^{H}(k, \theta, c) \Phi_{y}^{-1}(l,k) g(k, \theta, c)}. \]
The matrix inversion in Eq. (22) might not exist due to a singular covariance matrix or lead to numerical issues. For this reason, we discuss a covariance matrix pre-processing method to mitigate this problem in Section 4.4. The MPDR beamformer is equivalent to the Minimum-Variance-Distortionless Response (MVDR) beamformer if we assume RDP-TFs known a-priori. However, this is not the case in practice since the free- and far-field assumptions can be violated in specific acoustic conditions. The MVDR can be employed by replacing the signal covariance matrix in Eq. (16) with the noise covariance matrix \( \Phi_n(l, k) \). However, the latter must be estimated from the microphone signals, and, due to this additional estimation step, we do not consider the MVDR in this work.

### 4.3 Multiple Signal Classification

Given the assumptions made in Section 2, it is possible to define the microphone signal covariance matrix in Eq. (16) as a superposition of the source signal \( \Phi_s(l, k) \) and the noise covariance \( \Phi_n(l, k) \) matrix, i.e.,

\[
\Phi_s(l, k) = \Phi_s(l, k) + \Phi_n(l, k) = \phi_s(l, k)g(l, k)g^H(l, k) + \phi_n(l, k)I,
\]

where \( I \) denotes an \( N \times N \) identity matrix and \( \phi_s(l, k), \phi_n(l, k) \) denote the power spectral densities of the source and the noise, respectively. The signal covariance matrix can be factorized through an eigenvalue decomposition obtaining the diagonal matrix \( \Lambda \) containing the eigenvalues, and the matrix \( Q \) containing the (column-wise) eigenvectors. Since the source signal and the noise are orthogonal, the diagonal matrix \( \Lambda \) is the sum of two diagonal matrices, the first containing the largest eigenvalue associated with the superposition of the source and the noise, and the second containing the remaining \( N-1 \) eigenvalues associated with the noise. The eigenvectors \( U, V \) associated with such eigenvalues span the source and the noise subspace, respectively.

MUSIC exploits the orthogonality of the RDP-TFs identified by the source propagation characteristics (speed and direction) and the noise eigenvectors, i.e., \( g^H V = 0 \) where \( 0 \) is a \( 1 \times (N-1) \) matrix of zeros. This is equivalent to \( g^H V V^H g = 0 \) and advocates for the following SRP

\[
P_{\text{MUSIC}}(k; \theta, c) = \frac{1}{g^H(k; \theta, c)\hat{V}(k)\hat{V}^H(k)g(k; \theta, c)},
\]

where \( \hat{V}(k) \) denotes the eigenvectors matrix associated with the noise obtained via factorization of the estimated covariance matrix in Eq. (17).

### 4.4 Pre- and post-processing

Some considerations are given in this section regarding the pre-processing of the microphone signal covariance matrix and the post-processing of the SRP. Similar to the GGC-PHAT, it is possible to whiten the microphone signals via magnitude normalization, i.e.,

\[
\hat{y}(l, k) = \exp(\mu \hat{y}(l, k)),
\]

where \( \hat{\cdot} \) denotes the element-wise argument operator. Consequently, we obtain a whitened covariance matrix using Eq. (17). Applying the DSB to the latter yields the so-called SRP-PHAT, which ideally presents sharper peaks than the SRP obtained with Eq. (20). In contrast, MPDR and MUSIC do not benefit to the same extent from the magnitude normalization in DOA-estimation tasks [13] at low SNR. This was also confirmed for joint DOA and propagation speed estimation by pilot experiments, where the MPDR and MUSIC in combination with the PHAT weighting in Eq. (25) performed worse than simply employing Eq. (17). Therefore, we employ such pre-processing only for the DSB in the following.

As mentioned in Section 4.2, the inversion of the signal covariance matrix in Eq. (22) cannot always be guaranteed. Moreover, an ill-conditioned matrix can lead to severe inaccuracies. For this reason, we employ a signal-dependent diagonal loading [8] to guarantee the invertibility of Eq. (17) in the following way

\[
\Phi_y^{\text{DL}}(k) = \Phi_y(k) + \mu I.
\]

The cost function in Eq. (15) yields narrowband estimates. As discussed in [13], a Normalized Arithmetic Mean (NAM) can be performed on the SRPs to obtain a broadband estimate of the source speed and direction.
Figure 1. SRP obtained with (a) the DSB, (b) the MPDR, and (c) the MUSIC methods, against the candidate DOA and propagation speed. In (d), the contour plot of (a), (b) and (c) from left to right is shown, respectively. The oracle velocity (source direction and propagation speed pair) is depicted with a red dot, and the estimated velocity is depicted with a green cross. The latter was obtained as the maximum of each corresponding SRP. This way, each frequency band equally contributes to the SRP, and estimation errors caused by a variable SNR across the spectrum can be mitigated. The NAM is defined as follows

$$p_{NAM}(\theta, c) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{P(k; \theta, c)}{\max_{\theta,c} P(k, \theta, c)},$$  \hspace{1cm} (27)$$

where $P(k, \theta, c)$ is a vector containing the narrowband SRP for all candidate $[\theta, c]$. In the following, we employ Eq. (27) consistently with the SRP in Eqs. (20), (22), and (24) to obtain broadband estimates of the source direction and propagation speed as follows

$$[\hat{\theta}, \hat{c}] = \argmax_{\theta,c} p_{NAM}(\theta, c).$$  \hspace{1cm} (28)$$

In Fig. 1, we show a simulated example of the SRP obtained with the proposed methods. The microphone array consisted of a Uniform Circular Array (UCA) with $N = 4$ microphones and a radius of 1 cm. The directional source was a 3 s realization of Gaussian white noise at 1.5 m from the array’s center and at a DOA of 180° from the horizontal axis. The oracle temperature was 24°C, and the corresponding propagation speed was 345.84 ms$^{-1}$. The TFs were simulated in anechoic conditions, and mutually uncorrelated white noise was added to the microphone signals at 20 dB of SNR. While all methods yield perfect accuracy, the MUSIC pseudospectrum presents the narrowest lobe that can be beneficial in multi-source scenarios.
5 PERFORMANCE EVALUATION

In this section, we assess the performance of the proposed methods compared to the existing TDOA-based method [9], which we refer to as Plane Wave Approximation (PWA) in the following. The evaluation consists of two parts. In the first part, we simulated different acoustic conditions of an emitting source given by male and female speech samples (16 kHz) in a room with a variable temperature. Reverberation and spatially white noise were included to investigate the robustness of the proposed methods against non-ideal conditions. The methods were also tested across different DOAs. In the second part, we conducted measurements in a low-reverberant room using a Raspberry Pi3B+ single-board computer and an 8-microphone array MATRIX Voice [2]. The room temperature was measured with a DHT22 hygrometer sensor [1]. A human male and a female speaker generated the source signals at three different DOAs. Temperature variations were achieved by using the air conditioner or opening the window.

To solve Eq. (28), we apply a search grid of candidate speed within a range $325 \leq c \leq 365 \text{ ms}^{-1}$ with a resolution of $0.5 \text{ ms}^{-1}$, and candidate DOA within a range $0^\circ \leq \theta \leq 330^\circ$ with a resolution of $1^\circ$. The covariance matrix in Eq. (17) was estimated over the entire duration of the samples (around 2 s). In the following, we employ the diagonal loading of the covariance matrix in Eq. (26) with $\gamma = 1e^{-5}$, and the NAM in Eq. (27) for the MPDR, spectral weighting in Eq. (25) and the NAM for the DSB, and the NAM for the MUSIC. The estimation of the TDOAs in the baseline PWA was performed via GCC-PHAT in Eq. (13).

Temperature, speed, and DOA estimate errors were obtained by computing the Root Mean Square Error (RMSE) between the estimates and the oracle values, as follows

$$\text{RMSE}_c = \sqrt{\frac{1}{Q} \sum_{q=1}^{Q} (\hat{c}(q) - c(q))^2},$$  

expressed in $\text{ms}^{-1}$ for the propagation speed, and

$$\text{RMSE}_\theta = \frac{180}{\pi} \sqrt{\frac{1}{Q} \sum_{q=1}^{Q} \left( \angle \{ e^{i(\hat{\theta}(q) - \theta(q))} \} \right)^2},$$  

expressed in degrees $^\circ$ w.r.t. the x-axis of the selected coordinate system, where $Q$ denotes the total number of considered measurements and $q$ denotes the measurement index. The complex exponential in (30) was employed to avoid an erroneous DOA average across different quadrants. The temperature RMSE$_T$ expressed in $^\circ\text{C}$ was computed in the same way as in Eq. (29), where $\hat{c}$ was used to compute $\hat{T}$ using Eq. (7). The measurements considered all oracle temperatures, DOAs, SNR levels, reverberation times, and were compared against the baseline.

5.1 Simulated data

The simulations were implemented using Pyroomacoustics [14], a prototyping Python-based platform for beamforming algorithms in indoor scenarios. The library was patched to (a) accommodate a variable propagation speed and (b) include the baseline and the proposed methods. We simulated the room TFs between the source and the microphones using a variable room temperature $T = [3,15,25,30]^\circ\text{C}$ and hence a variable propagation speed according to Eq. (7). The simulations were restricted to 2D scenarios. The room’s size was $4 \times 5$ m, and we simulated signals measured with a uniform circular microphone array (UCA) of 8 microphones with an inter-microphone distance of 0.02 m at the center of the room. The source distance was 1.5 m from the center of the microphone array, and multiple source DOAs were selected $\theta = [0,30,45,60,90,120,180,250,270,330]^\circ$. Mutually uncorrelated Gaussian white noise was added to the microphone signals to simulate microphone self-noise. We individually simulated anechoic and reverberant room conditions. We varied the noise power in the anechoic scenario, whereas we increased the T60 in the reverberant scenario. The source signals were processed at $F_s = 16$ kHz with a DFT length of 2048 over the entire bandwidth.
Figure 2. Results from anechoic simulations using speech samples and oracle temperature $T = [3,15,25,30]$°C, across oracle DOA = $[0,30,45,60,90,120,180,250,270,330]$° and SNR = $[15,10,5]$dB: (a) Temperature RMSE, (b) Speed RMSE and (c) DOA RMSE for the proposed methods and the baseline PWA. In (d) the temperature RMSE is depicted against sex. The black error bars correspond to the estimate standard deviation.

5.1.1 Anechoic and noise simulation
The anechoic room simulations were performed by generating the RDP-TFs, i.e., without considering early reflections or reverberation. Gaussian white noise was added according to a variable SNR = $[5,10,15]$ dB. Figures 2a, 2b, 2c depict the RMSEs of temperature, speed and DOA obtained with the baseline PWA and the proposed methods. The estimates were obtained across all DOAs, temperature, SNR levels, and speaker sex. The black error bars depict the error standard deviation. MUSIC yielded overall the best performance, closely followed by MPDR. In Fig. 2d, we show the performance based on male or female speech. MUSIC and MPDR performed comparably for the female speech samples, whereas MUSIC outperformed the competing methods for the male speech samples. The baseline method performed marginally better for female speech compared to male speech.

In Fig. 3, the plots depict the temperature RMSE against the three variable parameters used in the anechoic simulation, i.e., oracle temperature, source DOAs and SNR. The proposed methods yielded consistent results across oracle temperature (Fig. 3a) and DOA (Fig. 3b), whereas the baseline PWA accuracy significantly depended on the source DOA (Fig. 3b). For the proposed methods, the temperature and speed RMSE decreased by increasing the SNR (Fig. 3c). MUSIC consistently yielded the lowest RMSE across different parameters, thereby outperforming the competing methods. Overall, the proposed methods outperformed the baseline PWA in terms of estimate error mean and variance.
Figure 3. Results from anechoic simulations: temperature RMSE against (a) oracle temperature $T = [3, 15, 25, 30]^\circ C$, (b) oracle DOA $= [0, 30, 45, 60, 90, 120, 180, 250, 270, 330]^\circ$ and (c) SNR $= [5, 10, 15] dB$. The error bars on the plots correspond to the standard deviation of the observations.

5.1.2 Reverberant simulation
We selected increasing values of T60 = [0.1, 0.12, 0.13, 0.15, 0.2 and 0.22] s, to simulate increasingly reverberant conditions. As the maximum reflection order based on the input T60 is computed by inverting the Sabine formula [3] in Pyroomacoustics, the simulated T60 slightly varied from the desired values. The minimum T60 was dependent on the room size. The SNR was consistently 20 dB in every simulation. Additionally, the Direct-to-Reverberant Ratio (DRR) [11] was computed from the simulated TFs as an additional parameter to quantify the amount of reverberation. The DDR at each simulated T60 is shown in Table 1.

In Fig. 4, the simulation results indicate increasing speed and temperature RMSE compared to the anechoic scenario for the proposed methods. This suggests that the proposed methods are more sensitive to reverberation than noise. In particular, MUSIC overall outperformed MPDR and DSB, which yielded a comparable accuracy. Similar to the anechoic scenario, Fig. 5 shows the temperature RMSE against the selected parameters. While the proposed methods yielded consistent results across oracle temperatures (Fig. 5a) and DOAs (Fig. 5b), it is evident that the temperature RMSE increased by increasing the T60 (Fig. 5c). The baseline PWA yielded relatively consistent results across increasing values of T60 and outperformed MUSIC for T60 $> 0.19$.

5.2 Measured data
To assess the performance of the proposed methods in real conditions, we recorded speech utterances produced by human speakers in a low-reverberant room. The experiment was conducted in a room with dimensions 4.55 m x 4.45m x 2.55 m and an average reverberation time of 0.23s between 0.4 and 2 kHz. The set-up
Figure 4. Results from reverberant simulations using speech samples and oracle temperature $T = [3, 15, 25, 30]{^\circ}C$, across oracle DOA $= [0, 30, 45, 60, 90, 120, 180, 250, 270, 330]{^\circ}$ and $T_{60} = [0.1, 0.12, 0.13, 0.15, 0.2, 0.22]$ s: (a) Temperature RMSE, (b) Speed RMSE, and (c) DOA RMSE for the proposed methods and the baseline PWA. In (d) the temperature RMSE is depicted against sex. The black error bars correspond to the estimate standard deviation.

<table>
<thead>
<tr>
<th>$T_{60}$ (Oracle) [s]</th>
<th>DRR [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.12</td>
<td>7.756</td>
</tr>
<tr>
<td>0.13</td>
<td>6.657</td>
</tr>
<tr>
<td>0.15</td>
<td>5.158</td>
</tr>
<tr>
<td>0.2</td>
<td>3.058</td>
</tr>
<tr>
<td>0.22</td>
<td>2.516</td>
</tr>
</tbody>
</table>

Table 1. Direct-to-Reverberant ratio (DRR) in dB corresponding to each reverberation time $T_{60}$ value in seconds.
Figure 5. Results from reverberant simulations: temperature RMSE against (a) oracle temperature \( T = [3, 15, 25, 30]\degree C \), (b) oracle DOA = \( [0, 30, 45, 60, 90, 120, 180, 250, 270, 330]\degree \), and (c) \( T60 = [0.1, 0.12, 0.13, 0.15, 0.2, 0.22] \) s. The error bars on the plots correspond to the estimated standard deviation.

Figure 6. Experimental setup: (left) MATRIX Voice microphone array configuration with numbered microphones and related coordinate system, and (right) picture of the employed system with a Raspberry Pi3B+, the MATRIX Voice, and the DHT22 hygrometer.

was subsequently obtained by averaging four temperature measurements at the beginning and the end of the recordings. The room temperature was first lowered to 24.5\degree C for the male speaker and 25.2\degree C for the female
Figure 7. Results from measured data using human speakers and a variable oracle temperature: (a) Temperature RMSE, (b) Speed RMSE, and (c) DOA RMSE across oracle DOA = [60, 90, 135]°. In (d) the temperature RMSE is depicted against sex. The error bars on the plots correspond to the estimate standard deviation.

The source signal were female and male voices at DOA = [60, 90, 135]° counterclockwise w.r.t. the x-axis of the coordinate system depicted in Fig. 6. The elevation of the speech signals w.r.t. the array plane was approximately zero, with possible inaccuracies given by different positions of the speakers. The rationale behind using human speakers was to substantiate the feasibility of the proposed method for, e.g., virtual-voice assistants.

The audio samples were then processed offline at $F_s = 16$ kHz with a DFT length of 2048 samples over the entire bandwidth. The evaluation framework was the same as in the simulated data, where the microphone coordinates were adjusted according to the microphone locations in the MATRIX Voice array, and the microphone signal simulation was excluded. The performance with measured data is shown in Figs. 7 and 8. In Figs. 7a and 7b the MPDR outperformed the competing methods in terms of temperature and speed estimation. The performance of MUSIC was possibly reduced due to the presence of reverberation, as shown from the simulations in reverberant conditions. The baseline PWA yielded the best DOA estimation accuracy in terms of error mean and standard deviation (Fig. 7c). We observed differences in the performance between male and female speakers (Fig. 7d) due to possible variations in voice loudness and subsequent SNR level. Compared to the baseline PWA, the proposed methods yielded a lower error variability across oracle temperature (Fig. 8a) and DOA (Fig. 8b), but higher compared to the results from simulated data.
REFERENCES


Performance Analysis Of Binaural Signal Matching (BSM) in the Time-Frequency Domain

Ami Berger(1), Vladimir Tourbabin(2), Jacob Donley(2), Zamir Ben-Hur(2) and Boaz Rafaely(1)

(1) School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
(2) Reality Labs, Meta, 1 Hacker Way, Menlo Park, CA 94025, USA

ABSTRACT
The capture and reproduction of spatial audio is becoming increasingly popular, with the mushrooming of applications in teleconferencing, entertainment and virtual reality. Many binaural reproduction methods have been developed and studied extensively for spherical and other specially designed arrays. However, the recent increased popularity of wearable and mobile arrays requires the development of binaural reproduction methods for these arrays. One such method is binaural signal matching (BSM). However, to date this method has only been investigated with fixed matched filters designed for long audio recordings. With the aim of making the BSM method more adaptive to dynamic environments, this paper analyzes BSM with a parameterized sound-field in the time-frequency domain. The paper presents results of implementing the BSM method on a sound-field that was decomposed into its direct and reverberant components, and compares this implementation with the BSM computed for the entire sound-field, to compare performance for binaural reproduction of reverberant speech in a simulated environment.

Keywords: Binaural reproduction, BSM, Dynamic-arrays, Head tracking

1 INTRODUCTION
Binaural reproduction of acoustic scenes that are captured by microphone-arrays is becoming increasingly popular [1],[2],[3], with many applications in teleconferencing and virtual and augmented reality. A popular method for binaural reproduction involves convolving high order Ambisonics (HOA) signals with the head related transfer function (HRTF) [4]. This method is fairly accurate for sufficiently high spherical harmonics (SH) orders, and the incorporation of head tracking enhances the immersion experience of the listener. However, the main drawback of this method lies in the large number of microphones required when the audio signal is recorded by a spherical microphone array, and in the need for a spherical array geometry, which restrict the practical use of this method [12].

In order to reproduce binaural signals using more flexible array geometries, the beamforming-based binaural reproduction (BFBR) [5],[6],[7] method was proposed. In this method the microphone signals are filtered using a set of beamformers, and the output signals are later filtered using the HRTFs and then summed to reproduce the binaural signals. A theoretical framework to set the design parameters of BFBR, such as the look direction and beam number for spherical and planar arrays, was presented in [8]. However, for more general array geometries only limited guidelines were suggested, with no guarantee of accurate binaural signal reproduction, and so a comprehensive design methodology is still unavailable.

With the aim of overcoming the limitations of current beamforming-based methods, and in order to accurately reproduce binaural signals recorded by arrays of arbitrary geometry, the binaural signal-matching (BSM) [9],[10],[11] method was developed. BSM refers to the estimation of the binaural signals directly from the array measurements using optimal filters calculated for each ear separately. Recently, the design of a BSM system was described and studied for a varying number of microphones in a semi-circular array [13] incorporating head tracking [20],[22]; it was shown that the accuracy of BSM is sensitive to the position of the microphones rather
than to their number. In particular, it was shown that the closer a microphone in the array is to an ear, the better the binaural reproduction in that ear. Therefore, the main drawback of the BSM method is that it performs poorly for high frequencies, especially in cases where one of the ears is relatively far from all the microphones of the array. In order to improve the perceptual performance it was proposed to use Magnitude Least-Squares (MagLS) [28] instead of Least-Squares (LS) for high frequencies. However, while the use of MagLS improved the overall perceptual experience of a reproduced binaural signal, BSM still performs perceptually badly when one of the ears is relatively far from all of the array microphones. Parametric spatial audio and binaural reproduction have also been studied as an alternative to the BSM and beamforming methods described above. In this approach the sound-field is decomposed into components, typically direct sources and reverberant parts, and each is estimated and reproduced separately [21], [23], [24], [25], [26].

While these approaches show promising performance, the latter may depend on estimation accuracy, which becomes challenging in complex environments. Nevertheless, the parametric approach also motivated the study in this paper.

With the limitations of previous method in mind, this paper aims to study and analyze the performance of BSM with a parameterized sound-field in the time-frequency domain. Specifically, an acoustic scene that is separated into direct and reverberant components is investigated. The paper investigates the potential of improving BSM by incorporating sound-field parameterization, especially in cases where BSM currently fails.

2 MATHEMATICAL BACKGROUND

This section provides the mathematical model of BSM. Throughout the paper, the standard spherical coordinate system is used, denoted by \((r, \theta, \phi)\), where \(r\) is the distance to the origin, \(\theta\) is the elevation angle measured from the Cartesian z axis downwards to the Cartesian xy plane, and \(\phi\) is the azimuth angle measured from the positive x axis towards the positive y axis. A sound-field can be described as a linear combination of plane-waves \(a(k, \theta, \phi)\), where \(k = \frac{2\pi}{\lambda} \) is the wave-number, \(\lambda\) is the wave-length and \((\theta, \phi)\) is the wave arrival direction. Further, assume that the sound-field is composed of \(L\) far-field sound sources arriving from directions \((\theta_l, \phi_l)_{l=1}^L\), with source signals \(s_l(k)\). The sound-field is captured by an array of \(M\) microphones, which are located at \(r_m, \theta_m, \phi_m\), centered at the origin. The noisy array measurements can be described by the following narrow-band model [14]:

\[
x(k) = V(k)s(k) + n(k)
\]

where \(x(k) = [x_1(k), x_2(k), ..., x_M(k)]^T\) is the microphone-signal vector (measurements), \(V(k) = [v(k, \theta_1, \phi_1), v(k, \theta_2, \phi_2), ..., v(k, \theta_L, \phi_L)]\) is an \(M \times L\) complex matrix with columns \(v(k, \theta_l, \phi_l)\) representing the array steering vector from the \(l\)-th source to the microphone positions for all \(l = 1, 2, ..., L\) sources, \(s(k) = [s_1(k), s_2(k), ..., s_L(k)]^T\) is the source-signal vector, and \(n(k)\) is an additive-noise vector. In the BSM model we further assume that a listener is positioned with the center of the head coinciding with the origin, where \(h^{\ell'}(k, \theta, \phi)\) denotes the HRTF of the left and right ears of the listener using the superscripts \(\ell'\) for the left ear and \(\ell'\) for the right ear. The signal at the left and right ears can now be written as [13]:

\[
p^{\ell'}(k) = [h^{\ell'}(k)]^T s(k)
\]

where \(h^{\ell'} = [h^{\ell'}(\theta_1, \phi_1), h^{\ell'}(\theta_2, \phi_2), ..., h^{\ell'}(\theta_L, \phi_L)]^T\) contains the HRTFs corresponding to the directions of the sources.

Next, assume that the configuration of the microphone array is known, such that the steering matrix \(V(k)\) can be calculated analytically or numerically. In the first step, the array measurements are filtered and combined, in a similar manner to beamforming:

\[
z^{\ell'}(k) = [c^{\ell'}(k)]^H x(k)
\]

where \(c\) is an \(M \times 1\) complex vector holding the filter coefficients. Next, \(c\) is chosen to minimize the following mean-squared error between \(z^{\ell'}(k)\) and \(p^{\ell'}(k)\), the binaural signals in (2), for each ear separately:

\[
err_{bin}^{\ell'}(k) = E[(p^{\ell'}(k) - z^{\ell'}(k))^2]
\]
where $\mathbb{E}[]$ is the expectation operator. Next, assume that the noise is uncorrelated to the sources. This leads to the following error formulation:

$$err_{\text{hi}}^H(k) = \| [s(k)]^H ( [V(k)]^H c_{lR}(k) - [h(l') (k)]^* ) \|^2_H + \| [n(k)]^H c_{lR}(k) \|^2_H$$

(5)

Minimizing the error in (5) leads to the following solution:

$$c_{\text{opt}}^H(k) = (VR_sV^H + R_n)^{-1}VR_s[h(l')]^*$$

(6)

where $R_s = E[ss^H]$, $R_n = E[nn^H]$. Next, assume that the sources are uncorrelated as in [20], leading to $R_s = \sigma_s^2 I_L$, and also that the noise is uncorrelated between microphones, such that $R_n = \sigma_n^2 I_M$. These assumptions lead to the following simplification of (6):

$$c_{\text{opt}}^H(k) = (VV^H + \frac{1}{SNR} I_M)^{-1}V[h(l')]^*$$

(7)

where $SNR = \frac{\sigma_s^2}{\sigma_n^2}$. Equation (7) has been used in [20]. The advantage of (7) is that it does not require the estimation of $R_s$, but this in turn may lead to reduced performance. The next section presents an approach to enhance performance by incorporating limited information on $R_s$.

3 BSM WITH PARAMETERIZED SOUND-FIELD

This section investigates the BSM method described in [20], but extended to incorporate a parameterization of the sound-field. The parameterization is based on the assumption that the measured sound-field can be decomposed into two components as follows:

$$x(n,k) = x_d(n,k) + x_r(n,k) + n(n,k)$$

(8)

where $n(k)$ is an additive-noise vector, and $x_d(n,k)$ represents the direct signal from the source in the time-frequency domain, modeled as a single far-field plane wave, written as:

$$x_d(n,k) = v(k, \theta_d, \phi_d)s_d(n,k)$$

(9)

where $(\theta_d, \phi_d)$ represents the DOA of the direct signal, and $s_d(n,k)$ represents the source signal. $x_r(n,k)$ represents the reverberant part of the measured signal in the time-frequency domain and is typically composed of a large, unknown number of sources that arrive from unknown directions. This model can represent a single source in a room where $x_d$ are the measurements of the direct sound from the source and $x_r$ are the measurements of the source reflections from room boundaries. There are methods available that estimate $x_d(n,k)$ and its direction-of-arrival (DOA) for each time-frequency bin [21],[26],[27], but in this paper $x_d(n,k)$ and its DOA are assumed to be known. Knowing $x_d(n,k)$, there are methods that can reproduce the direct component of the binaural signal, $p_d^{lR}(n,k)$ [20],[21],[8],[1]. Nevertheless, in this paper the BSM method was chosen as this method is the focus of this paper. Reproducing the binaural signal of the reverberant component, $p_r^{lR}(n,k)$, from the reverberant component of the measurement, $x_r(k)$, is a more challenging problem since it usually requires an estimation of $R_s$. The simplification of $R_s$ applied in (7) assumes that the sources are uncorrelated, e.g. a diffuse sound-field, and so it could be a reasonable approximation for the reverberant part.

In summary, computing the BSM filters separately for the direct and reverberant components is expected to produce better results compared to computing the BSM filters for the entire sound-field, because in the former the direct sound components are expected to be reproduced more accurately. These components are formulated as:

$$\hat{p}_d^{lR}(n,k) = p_d^{lR}(n,k) + \hat{p}_r^{lR}(n,k)$$

(10)

where $\hat{p}_d^{lR}(n,k)$ and $\hat{p}_r^{lR}(n,k)$ are the reproduced binaural signals of the direct and reverberant components using the BSM method, as described in (7). The performance of BSM with the proposed decomposition is evaluated in this paper numerically using binaural signal error and perceptually, using a listening test.
4 SIMULATION STUDY

This section presents a simulation-based analysis of the performance of the proposed BSM method with sound-field decomposition, compared to the BSM method computed from the entire measured signal without applying decomposition.

4.1 SET-UP

A point source was simulated inside a room of dimensions $8 \times 5 \times 3$ m, having a reverberation time of $T_{60} = 0.68$ s, using the image method [18]. The source location in the room was $(2.47, 2.27, 1.7)$ m. The source signal was a 5s long recording of female speech, taken from the TIMIT database [19], sampled at 48kHz. A semi-circular microphone array was centered at $(2, 2, 1.7)$ m, with a DRR value of 4.5 dB, compromising $M = 6$ omni-directional microphones arranged on the horizontal plane. Microphone positions were denoted using spherical coordinates by $(r_m, \theta_m, \phi_m)$ for $m = 1, \ldots, M$ relative to the array center, where $r_m = 10$ cm, $\theta_m = \frac{\pi}{2}$ rad and $\phi_m = \pi - \frac{2(m-\frac{1}{2})}{M-1}$ rad. Microphone measurement signals are denoted by $x(t) = [x_1(t), x_2(t), \ldots, x_M(t)]^T$. The component of $x(t)$ representing the direct contribution from the source, $x_d(t)$ as in (8), was calculated by assuming a free-field environment. The HRTF in the simulation was taken from the Cologne database [17] measurements of the Neumann KU100 manikin, sampled at a frequency of 48kHz. The HRTFs for the DOAs of the assumed sources were interpolated in the SH domain using a SH order of 30. The head was centered at $(2, 2, 1.7)$ m and was aligned with the positive $x$ axis. An illustration of the array position relative to the head position and alignment is presented in Figure 1. Assuming the semi-circular array represents an array on AR glasses, for example, the selected orientation of the array relative to the head was chosen as this was found to be the most challenging for the BSM algorithm [20].

4.2 METHODOLOGY

Having computed $x(t)$ and $x_d(t)$ as described above, $x(n,k)$ and $x_d(n,k)$ were computed using the Short-Time-Fourier-Transform (STFT), with a Hamming window of 32ms and a hop length of 16ms. $x(n,k)$ was then calculated using (8). Using (7), two BSM filters were computed, one for the direct component and one for the reverberant component of the signal, denoted $c_{r}^{1/2}(k)$ and $c_{l}^{1/2}(k)$, respectively. As described in section 2, the BSM method was developed assuming a sound-field composed of $L$ far-field sources. When computing the filter for the reverberant component, it was assumed, similarly, that the sound-field is composed of $L = 240$ sources with DOAs that correspond to a spirally-nearly-uniform distribution [15]. A single source with $L = 1$ and a DOA of $(\theta_d = \frac{\pi}{2}, \phi_d = \frac{\pi}{2})$, relative to the array’s center, was assumed when computing the filter of the direct component. The steering vectors for the reverberant and direct components were generated analytically in the SH domain according to section 4.2 in [16] using the corresponding number of $L$ sources, the array geometry and the DOAs of the assumed sources. Next, $SNR = 20$ dB was assumed for the reverberant components, while $SNR = \infty$ was assumed for the direct component, and both values were substituted in (7). The BSM weights of
the reverberant component were calculated using MagLS [28] for frequencies in the range of $[1500, 24000]$ Hz. Next, $\hat{p}_{l,r}^r(n,k)$ and $\hat{p}_{l,r}^d(n,k)$, representing the output of the BSM method for the reverberant and direct components, were calculated according to (3) using $c_l^r(k)x_r(n,k)$ and $c_d^r(k)x_d(n,k)$, respectively.

$\hat{p}_{l,r}^{BSM}(n,k)$, representing the solution of implementing the BSM method directly from the array measurements without sound-field decomposition, were calculated according to (3) using $c_l^r(k)$ and $x(n,k)$.

$p_{l,r}^{ref}(n,k)$, the reference signals at the ears, were calculated by convolving the HRTFs of the left and right ears with the HOA signals of order 14 that were calculated using the image method as described in section 4.1.

$p_{l,r}^{ref}(n,k)$, the direct component of the reference signals at the ears, were calculated by convolving the HRTFs of the left and right ears with the HOA signals of order 14 that were calculated by assuming a free-field environment as described in section 4.1.

$p_{l,r}^{ref}(n,k)$, the reverberant component of the reference signals at the ears, were calculated using (10).

The normalized mean-squared error (NMSE) of the binaural signals was calculated as:

$$NMSE(k) = \frac{\mathbb{E}_n[|\hat{p}_{l,r}^{r}(n,k) - p_{l,r}^{ref}(n,k)|^2]}{\mathbb{E}_n[|p_{l,r}^{ref}(n,k)|^2]}$$  \hspace{1cm} (11)$$

where $p_{l,r}^{ref}(n,k)$ is the reference binaural signal of a certain time-frequency bin, and $\hat{p}_{l,r}^{r}(n,k)$ is the reproduced binaural signal of a certain time-frequency bin.

4.3 SIMULATION RESULTS

In order to study the performance of the BSM method when applied to a decomposed sound-field, the NMSE of the direct and reverberant components were calculated and are presented in Figures 2 and 3. Figure 2 shows that the NMSE of the reproduced direct component of the binaural signal is relatively low, for both the left and the right ears, in particular at the low frequencies. This result indicates a fairly accurate reproduction of the direct component, which agrees with the expected performance of the BSM method designed for a sound-field compromised of $L < M$ sources as shown in [13].

The NMSE of the reverberant component is presented in Figure 3. As can be observed, the NMSE is higher at higher frequencies and lower for the ear closest to the microphone positions (in this simulation the left ear), see Figure 1. These results agree with the expected performance of the BSM method as described in [20].

The NMSE of the reproduced binaural signal using the BSM method without sound-field decomposition, $\hat{p}_{l,r}^{BSM}(n,k)$, and the reproduced binaural signal using the BSM method with sound-field decomposition, $\hat{p}_{l,r}^{BSM} + \hat{p}_{l,r}^{d}$, were also calculated and are presented in Figure 4. As can be observed there is a clear overall improvement in the NMSE of the reproduced binaural signals for the BSM method with sound-field decomposition, compared to the standard BSM method, especially for the left ear (which is closer to the array). This demonstrates that a more accurate reproduction of the binaural signal using the BSM method can be achieved by sound-field decomposition.

By comparing the NMSE of the direct and reverberant components, we can deduce that the NMSE is dominated by the reverberant component. This simulation study shows the potential for sound field-decomposition - if such decomposition can be implemented in practice, it can significantly improve the performance of the BSM algorithm.
Figure 2. The NMSE of the direct component of the binaural signal for the left and right ears ($p^{l,r}_d, p^{l,r}_a$), calculated using $p^{l,r}_{d-ref}$ as the reference binaural signals.

Figure 3. The NMSE of the reverberant component of the binaural signal for the left and right ears ($p^{l,r}_v, p^{l,r}_v$) calculated using $p^{l,r}_{v-ref}$ as the reference binaural signals.

Figure 4. The NMSE of the left (a), and right (b) ears calculated using $p^{l,r}_{ref}$ as the reference binaural signals.
5 LISTENING EXPERIMENT

A preliminary listening test was reproduced with the aim of evaluating the similarity between the reference binaural signal $p_{l,r}^{ref}$, and the reproduce signals $\hat{p}_{l,r}^{BSM}$, $\hat{p}_{l,r}^{d}$, and $\hat{p}_{l,r}^{d} + \hat{p}_{r,l}^{d}$. The experiment was based on the MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) test [29] and incorporated 9 normal hearing participants. The preliminary listening test scores are presented using a box-plot in Figure 5. As can be observed in Figure 5, the scores of the reproduced binaural signal for the BSM with sound-field decomposition are higher than the scores of the standard BSM method. These results agree with the results presented in Figure 4; in particular, the score of the BSM with sound-field decomposition was very high and close to the reference, suggesting that this approach could also provide improvement that is perceptually notable.

![Figure 5. Box-plot of the listening experiment score. The median is marked by a red line, the 25th and 75th percentiles are marked as the bottom and top blue edges respectively, and the maximal and minimal grades are marked by black lines. The scores are based on overall quality.](image)

6 CONCLUSIONS

In this paper, binaural reproduction with the BSM method was studied with a semi-circular array and a decomposed sound-field. It was shown that the perceptual and overall accuracy of the reproduced binaural signal using a decomposed sound-field is higher than in the case of using the standard BSM method, especially where the distance between the listener ear position and the microphones is large. A listening test demonstrated that the accurate reproduction of the direct component was indeed important for perception. Future work could include the incorporation of a spatial coding method to implement sound-field decomposition, the development of a design framework that better reproduces the reverberant component, and an extension of the listening test performed in this work. Future work could also include the study of the BSM method with other array configurations, and the development of a design framework that improves binaural reproduction.

REFERENCES


[8] Ifergan, I., Rafaely, B. On the selection of the number of beamformers in beamforming-based binaural reproduction. J AUDIO SPEECH MUSIC PROC. 2022, 6 (2022)


ABSTRACT

The spatial super-hearing technology originally proposed by the present research group brings ultrasonic signals into the audible range and auralises them using headphones, in such a manner that the listener is also able to localise the sources through spatial hearing. The signals are captured using a microphone array, and the direction-of-arrival and diffuseness parameters are analysed in the time-frequency domain. This work describes the methods and hardware employed in a fully portable implementation of the system, for a frequency range from 20 kHz to 90 kHz. Six ultrasonic digital MEMS microphones are mounted onto a 3D-printed cube with rounded edges resembling a standard dice, with face-to-face distance of 25 mm. The directivity caused by the dice was simulated with COMSOL, and the resulting pattern is plotted and discussed. In the device, the microphone outputs are multiplexed using a custom-designed converter, which also houses a 2-channel D/A-converter for sound output. The microphone signals are delivered using USB to a pocket-sized standard computer, where the processing is conducted, and the binaural signal is sent back to the converter for DA conversion. Changes made to the processing, compared to the earlier implementation, are also discussed.

Keywords: late reverberation model, sound energy decay, coupled rooms, multi-exponential decay

1 INTRODUCTION

Although the sense of hearing is sensitive to a wide range of frequencies, there is an upper frequency limit. This limit is approximately 20 kHz for young human subjects and is gradually lowered with increasing age. The frequencies above 20 kHz are commonly referred to as ultrasonic frequencies. Many animals, such as bats, rodents, insects, reptiles and amphibians produce strong vocalisations in the ultrasonic range[1], and man-made devices may also generate ultrasonic sounds in their normal or abnormal operation; such as gas leaks in pipes[2]. Ultrasonic signals can be brought to audible frequencies using signal processing techniques, for example, bats are often monitored using specific detectors [3], which can play back the down-shifted sound through a miniature loudspeaker. However, while the sounds they produce are audible to the listener, such devices do not permit the perception of the direction of ultrasonic sound sources.

The present group has recently developed a technology to render ultrasonic frequencies audible within the range of human hearing, while simultaneously allowing the directions of the ultrasonic sources to be perceived by the listener in a real acoustic environment. The first published device [4] utilizes a miniature head-mounted ultrasonic microphone array, accompanied by parametric spatial audio reproduction of the down-shifted sounds over headphones. This article discusses an improved version of the device, which has been designed for mobile use, and some changes in the microphone array design are also reported.

2 Ultrasonic super-hearing technology

The ultrasonic super-hearing technology is based on the application of time-frequency-domain parametric spatial audio techniques, which have been previously developed for the enhancement of sound-field reproduction and compression of spatial sound scenes [5]. The methods first employ spatial analysis techniques over time and frequency, in order to extract spatial parameters describing the input sound-field as captured by the employed...
Figure 1. Left: Ultrasonic super-hearing device, where the microphone array and audio interface are mounted to headphones and signal processing is performed in miniature computer inside a black case seen on floor with its power bank. Right: Close-up figures of headphone-mounted parts of the device.

Figure 2. Signal processing chain for ultrasonic super-hearing.
microphone array. These analysed parameters are then subsequently used to synthesise the audio signals for
the target reproduction setup in an adaptive and more informed manner. The processing stages of the proposed
technique are depicted in Fig. 2. Note that each signal from the microphone array is first transformed into the
short-time frequency domain, and the spatial analysis is conducted for each time-frequency tile independently.
The parameter representing the most prominent direction-of-arrival is defined as a unit vector pointing to the
direction of arriving sound \( \hat{\mathbf{r}}(t, f) \), where \( t \) and \( f \) are time and frequency parameters, respectively. The
second analyzed parameter corresponds to the diffuseness of the sound-field, defined as a real-valued number
\( \hat{\psi}(t, f) \in [0, 1] \), where ideally \( \hat{\psi} = 0 \) is obtained for sound-fields comprising a single plane wave, and \( \hat{\psi} = 1 \) is
obtained for a purely diffuse-field or for fields where several sources are active at the same time.
One of the sensor signals is then also modified via a pitch-shifting or down-modulating method, in order to
bring captured ultrasonic sounds down to the audible frequency range. This modified signal is first attenuated
depending on analysed diffuseness, and subsequently spatialised for playback over headphones, by using the
appropriate binaural filters corresponding to the analysed direction-of-arrival parameters. In practice, the system
reproduces sound when a single source is dominant, and de-emphasises all other portions of signal. Non-
individualised head-related transfer function (HRTF) digital filters [6] were employed for the spatialisation for
the developed device.
In summary, the proposed processing approach permits frequency-modified signals to be synthesised with
plausible binaural and monaural cues, which may subsequently be delivered to the listener to enable the locali-
calisation of ultrasonic sound sources. Furthermore, since the super-hearing device turns with the head of the
listener, and the processing latency of the device was constrained to 44 ms, much of the dynamic cues should
also be preserved. Note that the effect of processing latency has been previously studied in the context of
head-tracked binaural reproduction systems, where it has been found that a system latency above 50-100 ms
can impair the spatial perception [7, 8]. Therefore, it should be noted that a trade-off must be made between:
attaining high spatial image and audio quality (which are improved through longer temporal windows and a
higher level of overlapping) and having low processing latency (which relies on shorter windows and reduced
overlapping). The current processing latency has been engineered so that both the spatial image and audio
quality after pitch-shifting, as determined based on informal listening, remain reasonably high.
3 Design of the device
A mobile version of the super-hearing device built for the first tests [4] was targeted in the current project. The
planning was based around miniature computers available, and on the options how to bring multichannel audio
with high sampling rate for real-time processing and immediate playback.
3.1 Hardware and audio software
A miniature computer running windows operating system (Latte panda 4GB/64GB) was selected to run the super-
hearing application, primarily because it was able to import 6 channels of audio at a 192 kHz sampling rate.
Many of other miniature computers investigated by the present authors did not have sufficiently fast buses
for this task. A portable computer was also preferred over a dedicated microprocessor, since the software
already developed by the research group could be easily utilized in the device. In practice, the REAPER audio
production tool is hosted on the device, and the super-hearing audio processing is applied therein through use
of an open-source VST plugin1.
Digital MEMS microphones were chosen (Knowles SPH0641LU4H-1), and a 8 x 12 mm board housing
them was designed and manufactured. The microphones were organized into groups of two, and stereophonic
audio from each pair in pulse density modulation (PDM) format is conveyed to the custom converter board.
On the converter board, dedicated format conversion IC’s (Texas Instruments PCM3180) are used to convert
the PDM formatted signals to inter-IC sound (I^2S) format, which is supported by the USB multichannel audio
interface (MiniDSP MCHStreamer). The format conversion board and USB audio interface were installed into a
3D-printed box mounted on the top of the headphones used in the device. Six audio channels with a 192 kHz
sampling rate are then fed using a standard USB cord to the miniature computer.

1The VST audio plugin, and related MATLAB scripts, may be found here: https://github.com/leonccormack/Super-Hearing
After processing, a stereophonic audio channel (also in I\textsuperscript{2}S format with sampling frequency of 192 kHz) is routed back to the format conversion board, where a two-channel digital-to-analog converter (Texas Instruments PCM1795) with 96 kHz signal bandwidth and high current output driver (JRC NJM4556) is implemented. The analog signals are then played over standard stereophonic headphones. For demonstration purposes, a pair of headphones with active noise cancellation and inbuilt headphone amplifier is used.

3.2 Sensor array
The previous version of the device utilized analog microphones flush-mounted to a 3D-printed spherical housing. Although a sphere seems to be a natural choice for geometry, it causes some issues in the direction-of-arrival analysis method utilized in the system. In the analysis stage, the alias-free short-time Fourier transform (afSTFT) filterbank described in [9] was employed to first divide the input signals into 512 uniformly spaced frequency bands, which are then analysed independently. These time-frequency domain signals $x(t, f) = [x(t, f), ..., x_Q(t, f)] \in \mathbb{C}^{Q \times 1}$ are denoted with $t$ and $f$ to represent the down-sampled time and frequency indices, respectively. Given that the intended operating range of the system is above the spatial aliasing frequency of the array, the direction-of-arrival (DoA) unit vector, $\hat{r}_{\text{DoA}}(t, f)$, is estimated using the sensor-amplitude driven space-domain approach proposed in [10]. This relies on first determining the instantaneous DoA estimates as:

$$\hat{r}_{\text{DoA}}(t, f) = \sum_{q=1}^{Q} |x_q(t, f)| n(\Omega_q),$$

where $n(\Omega_q) \in \mathbb{R}^{3 \times 1}$ are Cartesian unit vectors describing the direction of each sensor, $q$. Note that the array causes prominent acoustical shadowing of sound waves and the amplitude $|x_q(t, f)|$ is highest on the side of arrival, and lowest on the opposite shadowed side. When the direction vectors of the sensors are weighted with the amplitude values and summed, the resulting vector points to the most prominent direction-of-arrival of sound. Note that since these DoA estimates do not rely on inter-sensor phase relations, they are unaffected by spatial-aliasing.

The method is based on analysis of the shadowing effect by the rigid body, and optimally the directional pattern of the microphone should be a unidirectional cardioid pattern. However, the physical size of the array exceeds the spatial aliasing limit, and instead of unidirectional pattern, strong sidelobes are obtained, depending on frequency.

The digital MEMS microphones used in the device are mounted on small electronic boards, which calls for cubical arrangement of them, differing from spherical arrangement used before. The cubical arrangement was thought of as an ideal geometry for the array, and it was postulated that a combination of spherical and cubical geometries would provide more even shadowing patterns compared to those obtained by a spherical geometry. This proposed geometry was therefore first studied, where a sphere with a radius of 15 mm was cut on six sides of a cube with edge length of 25 mm. The geometry is therefore reminiscent of a dice, as shown in right side of Fig. 3. The geometry is not regular, and diffraction effects should be less systematic than with sphere. This was thought to produce pattern that would have less salient sidelobes. The shadowing effect was verified using COMSOL simulation. The simulation was run by placing an ideal pressure source on the position of the microphone mounted on the rigid bodies, and by computing the sound pressure level at a distance of 100 mm from the center of the body. The results of simulation for both spherical and dice-shaped arrays are shown in Fig. 3, where it can be observed, that on the contralateral side of the source the sidelobes are much less prominent, and more irregular than with spherical rigid body.

The dice-shaped body is thus practical in usage with microphones mounted on small circuit boards, and the more-diverse shawdowing effects produced by the body can be used by the direction-of-arrival estimator employed by the system.

3.3 Signal processing
The signal processing methods adopted in the revised version were modified only slightly in the current article. The pitch shifting method in earlier version was based on the use of phase vocoder [11, 12]. Phase vocoders are used typically to make moderate shifts in pitch, and the up-to three-octave shift required for the present super-hearing application did not always produce pitch-shifted signals of high signal quality. In the current version,
Figure 3. Sound pressure level simulated on a spherical surface at a distance of 100 mm from a rigid body, given a point source emanating from one of the microphones attached to the surface of the rigid body. Left: Spherical rigid body with 15-mm radius. Right: "dice" rigid body composed as intersection between a 15-mm-radius sphere and a cube with 25-mm edge length. Due to reciprocity, the SPL surface also represents the directional pattern of the microphone.

A simpler resampling method operating based on the time-domain signals was employed. In practice, in each time window with length of 2 ms, each microphone signal is time-stretched by a factor of 8, and overlap-added with the tails of earlier time windows. The pitch-shifted signal is then reproduced using the same methods as in the first version of the technology. The benefit of the simple resampling method is that the shift of each frequency is deterministic, although in some cases some frequency components can be heavily attenuated, and time structure can be smeared, due to destructive interference in the overlap-add processing.

4 Summary
This article describes the development a mobile version of an ultrasonic super-hearing device, which was previously proposed by the present audio research group in [4]. The device features an ultrasonic 6-microphone array, which is mounted onto a pair of headphones. The most prominent direction-of-arrival of sound in the region between 20 kHz and 60 kHz is then estimated, based on the array signals. The sum of all 6 signals is then pitch shifted to the audible frequency range, and subsequently auralised in the analysed direction. The device and proposed processing therefore allows the wearer to both hear and localise ultrasonic sound sources, such as bats or leaks in pressurised gas pipes. The paper details the development of a mobile/portable version of this device, and discusses the modifications made to the algorithms compared to the first implementation of the technology. Namely, the most significant changes were: the usage of digital MEMS microphones, as apposed to analog microphones; the development of a processing chain featuring a miniature portable computer, multichannel USB card and a dedicated processing board for input audio format conversion and output D/A conversion; and the usage of a dice-shaped rigid array, which housed the MEMS microphones.

ACKNOWLEDGMENTS
This research has received funding from the Aalto University Doctoral School of Electrical Engineering and the Academy of Finland, project no. 317341.

REFERENCES


Improving spatial cues for hearables using a parameterized binaural CDR estimator

Reza Ghanavi(1), Craig Jin(2)

(1)University of Sydney, Australia, reza.ghanavi@sydney.edu.au
(2)University of Sydney, Australia, craig.jin@sydney.edu.au

ABSTRACT
We investigate a speech enhancement method based on the binaural coherence-to-diffuse power ratio (CDR), which preserves auditory spatial cues for maskers and a broadside target. Conventional CDR estimators typically rely on a mathematical coherence model of the desired signal and/or diffuse noise field in their formulation, which may influence their accuracy in natural environments. This work proposes a new robust and parameterized directional binaural CDR estimator. The estimator is calculated in the time-frequency domain and is based on a geometrical interpretation of the spatial coherence function between the binaural microphone signals. The binaural performance of the new CDR estimator is compared with three state-of-the-art CDR estimators in cocktail-party-like environments and has shown improvements in terms of several objective speech quality metrics such as PESQ and SRMR. We also discuss the benefits of the parameterizable CDR estimator for varying sound environments and briefly reflect on several informal subjective evaluations using a low-latency real-time framework.

Keywords: CDR estimation, binaural speech enhancement.

1 INTRODUCTION
Speech enhancement and listening comfort improvement in multi-talker, noisy environments remains an active research area in binaural hearing and binaural signal processing for both hearables and hearing aids [1, 2, 3, 4]. Research has shown that exploiting the short-time spatial coherence estimate between two adjacent microphones is an effective way of calculating the gains required for spectral enhancement [5, 6, 7, 8, 9]. Among the model-based dereverberation methods, a limited number of them are proposed for binaural applications [10]. An attractive feature of the binaural spatial coherence approach is that applying simple Wiener post-filtering on the short-time binaural spatial coherence preserves auditory spatial cues such as interaural time difference and interaural level difference for all sources when the target source is located directly ahead [11, 12, 7]. However, many previous coherence-based methods do not consider binaural processing per se but focus on speech enhancement.

In [5] a reverberation suppression method is introduced that estimates the coherent-to-diffuse energy ratio (CDR) for post-filtering gain calculation in a complex noise field. In particular, this work considers the geometry of the spatial coherence function in the complex plane for direct and diffuse sound components. Further research in [8, 9] shows that CDR-based dereverberation improves when the estimator accounts for the direction of arrival (DOA) of the direct signal and the phase of the complex-valued spatial coherence function. An issue with the aforementioned CDR estimators is that they rely on models of the coherence of the direct signal and/or diffuse noise field to estimate the CDR, and these models may not always match complex binaural noise fields in natural environments. A few numbers of the several heuristic CDR estimators proposed by Schwarz et. al. [7] have shown greater robustness when compared with the CDR estimators introduced in [5], [8] and [9]. As well, a more recent study by Löllmann et al. [13] estimates the CDR based on the effective rank of the covariance matrix of the input signals. Although this method does not require a coherence model for the signal and noise sound fields, it has the drawback of higher computational cost in real-time applications compared with the coherence-based method and produces less target to masking ratios in multi-talker environments due to its omnidirectionality.

In this work, we propose a new robust directional CDR estimator derived from the complex-valued short time-frequency domain spatial coherence function between the observed binaural signals. The new CDR estimator requires neither a coherence model of the noise field nor an estimation of the room reverberation time.
In this formulation, the target direction for the desired coherent signal is always broadside and straight-ahead. Hence, the symmetrical phase-magnitude response of the new estimator can be physically and psychoacoustically matched with any head shape and size to preserve natural binaural cues recorded by conchal microphones. In addition, the online adjustment of the new formula’s real-valued parameter \( S \) enables precise binaural dereverberation and denoising in a given acoustic environment.

This paper is organized as follows. First, in Section 2, a novel parameterized CDR estimator is formulated and described. Further, in this section, the application of the new CDR estimator in reverberation suppression is illustrated, and the real-time implementation of the mentioned algorithm is briefly described. In Section 3, the objective and perceived sound quality of the new binaural speech enhancement algorithm is compared with several state-of-the-art counterparts.

2 METHODS

2.1 New CDR estimator

In this study, we consider the recording of a reverberated and/or noisy speech signal by two identical omnidirectional conchal microphones. We assume that the auto-power spectra of the microphone signals recorded for the broadside target are equal. In this case, the spatial coherence function, \( \hat{\Gamma}_{l,r}(m,k) \), with the frame index \( m \) and frequency \( k \) for two binaural microphone signals, is expressed in the time-frequency domain as:

\[
\hat{\Gamma}_{l,r}(m,k) = \frac{\hat{\Phi}_{l,r}(m,k)}{\sqrt{\hat{\Phi}_{l,l}(m,k)\hat{\Phi}_{r,r}(m,k)}},
\]

where \( \hat{\Phi}_{x,y}(.) \) is the estimated cross-power spectrum for signals \( x \) and \( y \) and we use the short-hand index notation \( l \) and \( r \) to represent the left and right ear microphone signals \( X_l(m,k) \), respectively. We estimate \( \hat{\Phi}_{x,y}(.) \) recursively across time frames by \( \hat{\Phi}_{x,y}(m,k) = \lambda \hat{\Phi}_{x,y}(m-1,k) + (1-\lambda)|X_l(m,k)X_y^*(m,k)| \), where \( \lambda \) is a smoothing factor between 0 and 1 and \( \ast \) indicates the complex conjugate operation. We propose a new heuristic and parameterized formula for a DOA-dependent, binaural CDR estimator, \( \widehat{CDR}(m,k) \), that is derived solely from \( \hat{\Gamma}_{l,r}(m,k) \). For brevity and clarity in specifying the new CDR estimator, we omit the time and frequency indices. The new CDR estimator is given by:

\[
\widehat{CDR}(\hat{\Gamma}_{l,r}, S) = \Re \left\{ \exp(\hat{\Gamma}_{l,r} + \cos(\arg(\hat{\Gamma}_{l,r}) - (\pi/2)\arg(\hat{\Gamma}_{l,r})) + \ln(\hat{\Gamma}_{l,r} + (S)^2\ln(\hat{\Gamma}_{l,r} + \cos(\arg(\hat{\Gamma}_{l,r}) + \pi) \right\},
\]

where \( \arg(\cdot) \) and \( \Re\{\cdot\} \) refer to the phase and real part of a complex number, respectively, and \( S \) is an adjustable positive real-valued parameter. To clarify this formula, we consider Fig.1a, which depicts the new CDR estimator as a function of the complex spatial coherence vector and several values of \( S \). The complex spatial coherence vector is calculated from the cross-power spectra of the microphone signals recorded for the broadside target.
coherence vector can be represented by $\hat{\Gamma}_{1,r} = Ae^{i\theta}$, with amplitude, $0 < A \leq 1$, and phase, $0 \leq \theta \leq 2\pi$. From Fig.1a we observe that $\overline{CDR}(\hat{\Gamma}_{1,r}, S)$ is a U-shape (parabolic-like) function of the phase of the spatial coherence vector that is mirror-symmetric about $\theta = \pi$. We consider also Fig.1c which shows the new CDR estimator for several values of the amplitude $A$ for $S = 1$. In general terms, the CDR estimator increases as $\theta$ approaches 0 and as $A \to 1$; in other words, the geometrical slope of the $\overline{CDR}(\hat{\Gamma}_{1,r}, S)$ graph scales with the magnitude and phase of the spatial noise field coherence vectors. The variation in the geometrical pattern of $\overline{CDR}(\hat{\Gamma}_{1,r}, S)$ with changes in the $S$-value modifies the spatial directivity of the microphone system, i.e., higher $S$-values can reduce $\overline{CDR}(\hat{\Gamma}_{1,r}, S)$ as $\theta \to \pi$ which is equivalent to increasing the suppression of sound as it becomes incident from the side. The adjustability of the estimated CDR patterns may be useful for matching their values with the actual noise diffuseness in a specific frequency band. Depending upon the value of $A$, e.g. $A < 1$, one notices that the CDR estimator demonstrates a peak that progressively moves away from $\theta = 0$ as $A$ decreases. More specifically, the formula has been empirically designed so that for $A < 0.94$, the CDR estimator goes to 0 as $\theta \to 0$ with a faster rate compared to $0.94 < A < 1$. This behavior has been explicitly designed into the CDR estimator in order to suppress coherent noise that might arise at lower frequencies in a highly diffuse noise field, such as that related to the late reverberation of a room [5]. On the other hand, when $\theta \to \pi$, the observed geometrical pattern (see Fig.1c) reduces the non-broadside PSD of the noise field in the binaural signals, which may be useful for preserving early source reflections and assisting with source localization based on interaural intensity differences.

To examine the relationship between of the new CDR estimator ($S = 1$), $\overline{CDR}(\hat{\Gamma}_{1,r}, S)$, and the estimated, complex-valued spatial coherence function, $\hat{\Gamma}_{1,r}$, consider Fig. 2 that depicts the estimated CDR levels for a speech signal in a lecture room based on the position of spatial coherence vectors on the complex plane compared with the ’Propose 2’ (P2) CDR estimator by Schwartz et al. [7]. Observe that the coherence vectors are more dispersed for higher frequencies but more concentrated around the positive real axis for lower frequencies. This phenomenon shows that low-frequency signals have higher correlations because of their comparatively long wavelengths concerning head size and microphone spacing. The contrast between the two CDR estimators suggests that the new estimator may offer more reliable and precise estimation of CDR across frequency for a broadside signal located in front. For example, consider that the P2 CDR estimator shows a significant abrupt drop in the estimated CDR level as frequency decreases below 500 Hz, i.e., it will likely underestimate the low-frequency incident sound, while the new CDR estimator shows a more unbiased response across all frequencies. Significantly, one may also observe a spatial notch along the real axis for $\hat{\Gamma}_{1,r} < 0.94$ corresponding to the decreasing peak height shown in Fig.1c. This spatial notch is intended to de-emphasize the diffuse noise signals while preserving the coherent direct signal.

2.2 Binaural spectral enhancement
The application of the new CDR estimator for binaural noise and reverberation suppression is tested and investigated using methods like those proposed by [11], as indicated in the block diagram in Fig. 3. Observe that, the spatial CDR is first estimated as described in Section 2.1 and a gain function, $G(m,k)$, is then derived from the CDR estimate in the time-frequency domain as follows:

$$\hat{G}(m,k) = \max \left( G_{\min}, \left( 1 - \frac{\mu}{\overline{CDR}(m,k) + 1} \right)^2 \right).$$  (3)
The aforementioned coherence-based gain function is equivalent to the square of a Weiner filter where $G_{min}$ is the gain floor to reduce the musical artifacts, and $\mu$ is referred to as the over-subtraction factor [7] and is commonly set to one. The gain function is applied equally to the left and right channels, preserving the spatial auditory cues of interaural time and level differences. The effect of the gain function is shown in Figs. 1b and 1d and generally follows the functional form of the new CDR estimator. As shown in Fig. 1b, smaller values of $S$ result in less spatial noise/reverberation suppression and wider spatial directivity. In contrast, the larger values of $S$ increase the suppression of the noise/reverberation, provide narrower spatial filtering and may also increase audible artifacts. Furthermore, the square of the Wiener filter has been selected as the gain function since empirical testing has shown that it performs well with the new CDR estimator. i.e., it produces less audible artifacts and higher background noise suppression than a gain function based on the spectral magnitude subtraction as suggested by [7].

2.3 Binaural room simulations

Three state-of-the-art coherence-based dereverberation algorithms are compared with the proposed speech enhancement algorithm discussed in Section 2.2 in a binaural format. The counterpart CDR estimators used in this study are: (1) the DOA dependent CDR estimator ‘Propose 2’ (P2) in Schwartz et al. [7]; (2) the DOA independent CDR estimator ‘Propose 3’ (P3) in Schwartz et al. [7]; and (3) the effective rank-based DOA independent CDR estimator proposed by Löllmann et al. [13]. All signal processing algorithms use a common 16kHz sampling rate, an FFT size of 512, a window length of 1024, and a hop size of 128 in MATLAB implementation. The gain function for the new CDR estimator was computed as described in Section 2.2 and for the counterpart algorithms the applied gain function is the spectral magnitude subtraction as described in [7, 13], with $\mu = 1$ and $G_{min} = 0.1$ for all algorithms. For the CDR estimators P2 and P3, the spatial coherence model for the diffuse noise is given as: $\hat{\Gamma}_{xy}^{diff}(3D) = \sin(2\pi f d_{mic}/c)/(2\pi f d_{mic}/c)$, where $d_{mic}$ is the distance between two conchal microphones and $c$ is the speed of sound. For the CDR estimator P2, the spatial coherence for the broadside direct signal is taken as 1 (real valued), while the CDR estimator P3 does not require an estimate of the spatial coherence of the direct sound [7]. For the three counterparts, the smoothing factor $\lambda$ was set according to the relevant reference publication (P2 and P3: $\lambda = 0.68$; Löllmann: $\lambda = 0.8$). For the new CDR estimator, we chose $\lambda = 0.72$.

For the room simulation, we used a set of binaural HRIRs recorded by the conchal microphones of a generic in-the-ear hearable for a male subject with large pinna provided by the database described in [15]. The database HRIRs were then evenly interpolated for 642 directions on the surface of an imaginary sphere and used as input for the room simulator MCROOMSIM [16] in order to obtain a set of BRIRs corresponding to a shoebox large room (20m x 16m x 5m) with 4 different reverberation times (0.3, 0.5, 1 and 2)s. The simulations were conducted with the listener positioned in the center of the room (ear level at 1.6 m). A target talker directional source is positioned in front of the listener at 0.5 m distance, and the subject is surrounded by a combination of one near-field time-reversed directional female speech masker located on the right and four far-field evenly distributed female speech maskers. The 34 s female utterances were derived from HARVARD speech corpus [17]. In order to simulate a more realistic environment, a low-pass-filtered white noise (cutoff frequency 400 Hz) was mixed with the five masker signals. The relative signal levels used for the target, maskers, and low-pass filtered noise signals were varied and specified as a triplet of numbers (-6, -10, -10) dB, (0, 0, -10) dB and (-6, 0, 0) dB, respectively. The process above was repeated for four different reverberation times, set by changing the room acoustic absorption settings in MCROOMSIM.

2.4 Broadband low-latency real-time framework

Fig.4 shows the actual Raspberry Pi-based embedded system [18] prototype adapted for high-quality and low-latency online implementation of the described new algorithm in Python [19]. The recorded binaural time signals (32kHz sample rate) are buffered (window size 512) using 50% overlap. The selection of larger window sizes enables more accurate short-time signal power estimation at lower frequencies and has shown fewer arti-
facts in a time-variant system. Furthermore, the number of FFT points has been doubled (FFT size = 1024) to improve the quality of spectral enhancement processing in the frequency domain. Using the Hanning window, the signal is then reconstructed via the weighted overlap-add (WOLA) technique [20]. The output buffer is filled by the second half of the previous segment in time and the first half of the current segment to reduce real-time latency by half. In this case, the total acoustic latency in this system is measured to be about 9 ms, which is comparable with the average latency in a high-quality hearing aid system [21]. The user interface for this system enables online adjustment of the S-parameter as well as other parameters.

3 RESULTS
3.1 Objective Speech Enhancement Performance
The performance of the new algorithm for speech dereverberation and denoising is compared with other state of the art algorithms as mentioned earlier using two intrusive methods: perceptual evaluation of speech quality (PESQ) [22] and cepstrum distance (CD); and two non-intrusive methods: speech-to-reverberation modulation energy ratio (SRMR) and word error rate (WER) calculated for an automatic speech recognition (ASR) system. The narrow-band results for PESQ and CD are averaged across the two binaural enhanced signals, while the SRMR and WER data are derived based on a monaural mix-down of the two full-band binaural signals.

![Figure 4.](image)

**Figure 4.** Real-time low-latency prototype of the new CDR-based true binaural speech enhancement system. Raspberry Pi 4B (a), HiFiBerry DAC plus ADC Pro (b), hearable interface (C), online user interface (d) and binaural in-the-ear earpieces [15] (e).

In Fig.5, the calculated objective values are averaged over the results derived for several signal-to-masker/noise ratios (refer to Section 2.3). Observe first that the S-values obtaining the best results for the newly proposed estimator vary across the various speech quality measures (to a larger extent) and also across the different re-
verberation conditions (to a lesser extent), with the optimal $S$-value for a given acoustic environment as the performance measure changes from PESQ to CD to SRMR. The performance variations depending on the $S$-parameter demonstrate that the three measures examine various aspects of the direct sound and background noise quality. The new CDR estimator with the optimal $S$-value generally improves the PESQ values compared with the other CDR estimators, e.g., for $S = 10$, the sound quality of the new algorithm generally outperforms the counterpart algorithms for all of the reverberant conditions. Interestingly, a significant increase in the SRMR performance values is found for the new CDR estimator with optimal $S$-values compared with the other CDR estimators. However, the $S$-value must be increased significantly to obtain these results, i.e., higher $S$ values result in a more direct-to-reverberant ratio (DRR). However, it can be observed that in a given room, for $S > 100$, the spatial gains for the direct signal can be declined significantly due to change in the shape and slope of the new CDR function curves (see Fig.1a). In addition, the increase in $S$ values can make more modifications to the background noise spectrum that may explain the slight increments in the CD values. The varying $S$-values obtaining optimal performance across the three speech enhancement measures indicate that there are different and likely conflicting requirements for optimizing speech enhancement performance based on CDR, depending on the significance given to a particular speech enhancement measure.

<table>
<thead>
<tr>
<th>RT60(s)</th>
<th>Unprocessed</th>
<th>Schwarz et al. (P2)</th>
<th>Löllmann et al. (P3)</th>
<th>New estimator $S = 0.1$</th>
<th>$S = 1$</th>
<th>$S = 3$</th>
<th>$S = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>55.67</td>
<td>45.33</td>
<td>62.67</td>
<td>60.33</td>
<td>45</td>
<td>48.33</td>
<td>50.67</td>
</tr>
<tr>
<td>0.5</td>
<td>53.33</td>
<td>50.67</td>
<td>57.67</td>
<td>57.67</td>
<td>50</td>
<td>48</td>
<td>44</td>
</tr>
<tr>
<td>1</td>
<td>55</td>
<td>52.33</td>
<td>58.67</td>
<td>53.67</td>
<td>49.33</td>
<td>47.33</td>
<td>47</td>
</tr>
<tr>
<td>2</td>
<td>62.33</td>
<td>59.33</td>
<td>64.33</td>
<td>61.33</td>
<td>59.33</td>
<td>60.67</td>
<td>60</td>
</tr>
<tr>
<td>Average</td>
<td>56.58</td>
<td>51.92</td>
<td>60.83</td>
<td>58.25</td>
<td>50.92</td>
<td>51.08</td>
<td>50.42</td>
</tr>
</tbody>
</table>

The objective speech intelligibility was also estimated by calculating the word error rate (WER) of the automated speech recognition (ASR) algorithm for the processed and unprocessed speech. The ASR engine Deepspeech 0.9.3 was used [23]. In this work, we used clean speech containing 34 s of female speech (100 words) from the HARVARD speech corpus that was 100% recognizable by the pre-trained ASR engine. The average word error rates can thus be attributed to the acoustic condition and binaural sound processing systems. Table 1 shows the WER results for the multi-talker scenarios. On average, the word error rate for the new CDR estimator enhancement algorithm with $S = 3$ is lower compared with the other CDR-based algorithms. For the multi-talker scenario, the WER results indicate potential advantages to be found by tuning the $S$-value specifically for a given sound environment.

### 3.2 Preliminary subjective evaluation

The binaural psychoacoustic perception of the newly proposed algorithm compared with Schwartz et al., [7] (P2 and P3) is evaluated through several informal listening tests in regular rooms and a large reverberant/noisy cafeteria using the real-time platform described in section 2.4. We are preparing to conduct a proper psychoacoustic experiment in the future. Here we report some anecdotal results. For all algorithms, the optimized online parameters $\lambda$ and $G_{\text{min}}$ are set to 0.02 and 0.1, respectively. In general terms, the perceived sound quality is compatible with the objective results discussed in Section 3.1; however, the informal listening tests have revealed that the new algorithm seem to significantly improve the spatial quality of the sound in a given environment compared with the the counterpart algorithms. For example, the perceived frontal near-field and far-field binaural intelligibility in the presence of several random distributed noise/masker sources is highly improved for the new CDR formula, while Schwartz et al., P2 has shown satisfactory results for only the near-field target and P3 has shown less enhancement for the target-to-masker ratio. Furthermore, the enhanced multi-talker and noisy spatial atmosphere reproduced by the new binaural algorithm was reported as perceived as more natural, robust and quiet compared with the two other algorithms. i.e., accurate source localization and externalization are preserved for the new CDR estimator resulting in improved listening comfort. The online adjustment of parameter $S$ has revealed that small changes in $S < 20$ are perceivable and may be advantageous in adjusting the enhanced target speech quality and spatial perception in natural environments. The increment of the $S$ value for $S > 10$ can produce minor audible artifacts due to a higher level of background noise modification.
4 CONCLUSIONS

A new binaural, directional coherent-to-diffuse power ratio (CDR) estimator has been proposed for noise reduction and dereverberation in multi-talker reverberant and noisy environments. The CDR estimator relies only on the observed complex coherence between the binaural microphones and maximizes for a broadside target signal. The binaural application of the gain function and the compatibility of the new formula with the actual binaural noise field preserves spatial hearing cues. Furthermore, the new CDR estimator employs a variable $S$-parameter to provide adjustable coherence-based spatial filtering for different noise conditions.

The objective numerical evaluations show that varying the $S$-parameter enables the CDR estimate to improve speech quality and/or the direct-to-reverberant ratio. Adjusting the $S$-parameter enables a trade-off between signal quality, degree of dereverberation, and the spatial quality of the sound. The results suggest that the new CDR estimator may improve existing coherence-based methods for denoising and dereverberation. To this end, several informal listening tests have shown more advantages of the new method compared with two counterpart algorithms in terms of sound naturalness, accurate source localization and voice intelligibility.

ACKNOWLEDGEMENTS

The authors would like to thank Jorg Buchholz (Macquarie University) for his constructive comments and suggestions. This research is supported by an Australian Government Research Training Program (RTP) Scholarship.

REFERENCES


ABSTRACT
Binaural reproduction is today a widely used technique for immersive spatial audio. To provide a listener with a realistic sensation of spatial hearing over headphones, sound signals are typically convolved with Head-Related Impulse Responses (HRIRs), or filtered according to their spectral equivalents, i.e. the Head-Related Transfer Functions (HRTFs). These describe the transfer properties of sound waves as they travel from a given sound source location in space to the ear canal in free space. Since HRIRs are highly individual (they depend on a subject’s anthropometric features), deviations from the user’s own HRIRs can affect negatively the listening experience. Therefore, the identification of relevant localization cues and their preservation is a topic of continuous interest within the spatial audio research community. In this context, while numerous studies have been carried out in the past to identify salient localization cues, for example by applying principal component analysis (PCA) to HRIR datasets, some recent works are exploiting the feature learning capabilities of deep learning-based approaches. In this work, we explore the use of common explainable artificial intelligence (XAI) techniques, such as class activation mapping, on convolutional neural networks (CNN) trained for classifying HRIR datasets into different directional sectors, exploring further this issue.

Keywords: Spatial audio, HRTF, XAI, deep learning, convolutional neural networks

1 INTRODUCTION
With the advent of immersive acoustic scenarios for virtual reality, achieving accurate localization of sound sources has become a major challenge. One of the peculiarities that complicates the development of accurate reproduction systems is that human spatial hearing is closely related to the listener’s anatomy. Acoustic effects caused by the head, torso, shoulders and pinnae have a great impact on human localization ability. It is a well-known fact that such human characteristics can be statistically described to aid the auditory localization process [6]. Most acoustic models usually utilize time-domain Head-Related Impulse Responses (HRIRs) or, equivalently, Head-Related Transfer Functions (HRTFs) in the frequency domain. With regards to HRTF signals, several works have proposed the use of preprocessing and postprocessing techniques to capture the relative influence of relevant anthropometric features [21]. The localization of virtual sources can also be improved via scaling the directional transfer functions [12]. An interesting finding that matches with the objective of this paper is that prominences (either peaks or notches) in the curves of the transfer functions may reveal information associated to the elevation of a source [8, 5].

The use of deep neural networks for decision aid in acoustic environments has been growing steadily. In particular the use of convolutional neural networks (CNNs) is crucial in most of the artificial support tasks like acoustic scene classification [3], music tagging [11], and speech emotion recognition [14], among many others. In the same line, the development of binaural spatial datasets allows to train neural networks in order to model personalized HRTFs and enable a more realistic listening experience. Recent studies have shown the possibility
to capture spatial audio features in HRTF datasets using CNNs [17]. These studies may be understood as a
new manner to study human performance in source localization. Other studies have recently focused on the
front-back discrimination of binaural music recordings [20], revealing interesting information about the relevant
frequency bands assisting such discrimination task.

In the same spirit of [17], this paper focuses on the application of explainable artificial intelligence (XAI)
techniques for the analysis of HRTF datasets. More specifically, we consider the analysis of the HRTF CIPIC
dataset [2] using a conventional 1D-CNN model. In contrast to [17], HRTFs transformed to a mel-scale are
directly used as input to the network. Moreover, we apply two different XAI techniques for saliency analysis,
namely class activation mapping (CAM) and the more general gradient-based CAM method (Grad-CAM). The
results obtained using these techniques will be analyzed to discover frequency bands in the HRTFs that encode
location cues relevant to the determination of the elevation of a source.

2 CASE STUDY
2.1 Experimental dataset and data preprocessing
The CIPIC HRTF Database is a public-domain database of high-spatial-resolution HRTF measurements for 45
different subjects, including the KEMAR mannequin with both small and large pinnae [2]. It includes 2,500
measurements of HRIRs for 45 subjects at 25 different azimuths and 50 different elevations (1250 directions) at
approximately 5° angular increments. The standard measurements were recorded at 25 different interaural-polar
azimuths and 50 different interaural-polar elevations. The sample duration of each HRIR is 200 samples (4.5
ms at a sample rate of 44.1 kHz). Additional special measurements of the KEMAR manikin were made for the
frontal and horizontal planes.

As in [17], we focus this study on the analysis of elevation cues. To this end, we divide all the responses
of the CIPIC database into a set of 9 spherical regions according to their elevation, as depicted in Figure 1.
Note that the number of sampled directions within each of the elevation classes is not uniform, although not
severely unbalanced. The full dataset is composed of 56,250 HRIR samples (corresponding to the combination
of the 45 subjects, 25 azimuth angles, and 50 elevation angles) with their associated ipsilateral and contralateral
channels.

To obtain the corresponding HRTF signals, we compute the one-sided fast Fourier transform of each channel
with 512 points, resulting in 257 frequency bins. In order to provide the network with a perceptually-motivated
input, we warp the frequency axis by considering a mel-scale mapping. This is achieved by dividing the range
[0-22050] Hz into 257 uniformly spaced points in the mel-scale, and taking the frequency bins that are closer
to their equivalent frequencies in Hz. Finally, only the magnitude spectrum of each channel in logarithmic scale
is considered. The shape of each dataset example is therefore (257,2).

The dataset was divided into two partitions for training and validation. The data of 36 subjects (45,000
samples) was used for the training partition and the data of 9 different subjects (11,250 samples) for validation.
2.2 Model architecture
This work considers a fully convolutional model with a straightforward architecture, represented in Figure 2. The design of the network was carried out with the aim of achieving a significant classification accuracy while keeping the model simple enough to facilitate the application of common XAI techniques. It consists of three 1D convolutional blocks with ReLU activation and max-pooling in between blocks to downsample the frequency information. A last convolutional layer followed by Global Average Pooling (GAP) is used to summarize the filter responses before the final dense layer, configured with softmax activation.

![Figure 2. Topology of the convolutional architecture developed for this paper to classify HRTFs into nine elevation sectors.](image)

<table>
<thead>
<tr>
<th>Table 1. Configuration of the CNN layers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Dimension</td>
</tr>
<tr>
<td>Filters</td>
</tr>
<tr>
<td>Sizes</td>
</tr>
<tr>
<td>Parameters</td>
</tr>
</tbody>
</table>

The information related with the construction of each layer of the CNN is shown in Table 1. It shows detailed information about how the model was built, including the dimensions involved at different depths and the number of associated parameters.

The model was trained with Adam optimizer ($\eta = 10^{-\mathbf{3}}$) using categorical cross-entropy as loss function. For the fitting we set a batch of 16 elements and 100 epochs with early stopping, which actually was activated during the training processes.

2.3 Feature explainability
The use of XAI techniques may be understood as an additional phase in model evaluation [3]. As a manner to give explainability in the same way than the studies above mentioned, we have analyzed the outputs of our deep convolutional architecture. Once our model is fitted and yields an acceptable performance, we want to remove the black-box produced by the CNN architecture. With that purpose, we have applied CAM [18] and Grad-CAM [16] to analyze both saliency maps and class activation maps. Even though both XAI techniques were designed solely to cover the limitations of CNNs, their implementation is widely extended thanks to their ability to generate localization maps highlighting significant spatial regions [12]. CAM originally arose as a technique that yields class activation maps of CNNs applied over object detection tasks. It allowed the trained models to localize class-specific patterns in image detection. Owing to the main disadvantage of CAM is the requirement of a GAP layer to operate over the convolutional filters, Grad-CAM overcomes that limitation by
the use of gradients over the first dense layer in the architecture. Regardless of their differences, both techniques rely on the assumption that the decision vector \( Y^c \) for the class \( c \) is described by means of the feature maps \( A_k \) of the last convolutional layer. Then, the saliency map for the class \( c \) is a weighted aggregation of the spatial components of each feature map, written as:

\[
\text{Class-score: } Y^c = \sum_k w_c^k \sum_i \sum_j A_{ij}^k, \quad \text{Class-Saliency: } L_{ij}^c = \sum_k w_c^k A_{ij}^k.
\]

(1)

Where the \( i \) and \( j \) components denote the matrix indices of the spatial components and \( k \) index associates the number filters. Regarding the weighting scheme \((w_c^k)\), CAM utilizes a backward projection of the output weights on to the maps of the last convolutional layer. On the contrary, Grad-CAM estimates each class-weight for each feature map as a linear combination of the partial derivatives \( \frac{\partial Y^c}{\partial A_{ij}^k} \). Their implementation is highly related with image classification [18, 16, 12] varying from clinical test [6] to computer games simulation through reinforcement learning [10].

3 RESULTS

3.1 CNNs performance

After training, the model achieved global accuracy of 0.8090 on the validation set, which suggests that the model performs reasonably well in the task of determining the elevation class of a given HRTF, regardless its azimuth. To analyze the classification performance in more detail, Figure 3 shows the confusion matrix (hit percentage) for the different elevation classes. The proportion of class examples is shown in red-scale. Note that most wrong predictions are misclassified to adjacent spatial regions, showing robustness in terms of understanding of the underlying spatial cues.

3.2 Saliency Maps

This section shows the results obtained by the XAI techniques considered in this work, analyzing the saliency maps provided by the CAM and Grad-CAM methods. These saliency maps provide meaningful information on how the convolutional kernels process a given input example to determine its corresponding elevation class. Figure 4 shows representative HRTF responses selected from the validation set together with their corresponding saliency in the background. The selection was made according to a maximum-probability criterion, i.e. they correspond to the responses pertaining to each class that the model classified with the highest confidence. Dark red zones correspond to frequencies having a high saliency, while light zones correspond to frequencies that are less relevant for the classification task. For each class, the CAM saliency is shown on top plt, while the Grad-CAM result is shown at the bottom. It can be observed for each of the represented examples that both CAM and Grad-CAM provide similar saliency zones, although the intensity of such zones may vary from one method to the other. In general, there is a high level of agreement in the results provided by both approaches.
Figure 4. CAM (top) and Grad-CAM (bottom) saliency maps over the most representative sample of each class. The predicted class probability ˆy is given in parentheses. The color bar is red-scaled, then white shades indicate low relevance and red ones high relevance.

To gain more insight into the relevant spatial components processed by the CNN model, we analyzed the average saliency maps resulting from CAM and Grad-CAM across all subjects and azimuth angles. The aim is to get an overall picture of the relevant frequencies assisting the classification for the whole dataset. The lateral regions were not considered as they do not really affect elevation. The averaged frequency-elevation saliency maps are shown in Figure 5, where red indicates more importance and blue less importance for classification.

Finally, Figure 6 shows as an image the saliencies of the individual responses belonging to each class in the validation set, both for CAM and Grad-CAM. Note that there is high correlation between the saliency results obtained for the different responses within each class, suggested by prominent vertical bands located at narrower or wider frequency regions. Again, the results obtained by CAM and Grad-CAM are considerably similar.

3.3 Discussion
The saliency maps obtained in the previous section are considerably in accordance with some of the results derived from previous psychoacoustical experiments. Next, we describe how our results are linked to some known spectral cues and effects related to elevation and front-back discrimination. Note that, since the average CAM saliency map in Figure 5(left) seems to be more consistent than the one obtained from Grad-CAM, we will discuss our findings taking into account mostly the CAM results.

Many studies have shown that spectral distortions caused by pinnae in the high-frequency range approximately above 4 kHz act as cues for median plane localization [9]. Indeed, by looking at the resulting saliency
Figure 5. CAM (left) and Grad-CAM (right) saliency maps averaged across subjects and azimuth, showed per elevation class.

Figure 6. Saliency bands of CAM (top) and Grad-CAM (bottom) per class in validation partition for correct predictions. The color bar is scaled, then light shades indicate high relevance and dark ones low relevance.

... maps, the most intense saliencies are found, as expected, on the mid and high frequency range. However, there are some interesting low-frequency effects (below 500 Hz) in the classes “up”, “back-up”, “back-level” and “back-down”. These may suggest that HRTFs from back directions show some low-frequency features assisting the perception of elevation that may not be present for frontal directions. In this context, although a behind cue was reported by [8] to appear as a small peak around 12 kHz, we observe the average saliency to be high at this frequency for “back up”, but also contributions from higher frequencies at “back level” and “back down”, suggesting a shift of cues towards very high frequencies when a behind source moves from down to up.

Butler and Belendiuk [5] showed that the prominent notch moves toward the lower frequencies as the sound source moves from above to below the aural axis in the frontal half of the median plane. For frontal directions, moving from “front-up” to “front-down”, we can indeed see the saliency change towards lower frequencies...
(from around 8 kHz to 3 kHz).

Additionally, as observed in Figures 4 and 6, there appears to be complementary saliencies on samples of opposite spatial classes that may hint on more cues, as follows. Samples from the classes “lateral-up” and “lateral-down” present opposite saliencies on the 8-15 kHz band, while being similar on the rest of frequencies. Similarly, samples from the classes “front-level” and “back-level” present opposite saliencies on the 2-10 kHz band, while being quite similar out of this range.

4 CONCLUSIONS
This work presented a preliminary study on the use of explainable artificial intelligence techniques, namely CAM and Grad-CAM, for assisting the interpretation of HRTF elevation cues. To this end, we trained a convolutional neural network on mel-scale-warped HRTF responses extracted from the CIPIC database. The model was trained to classify responses into 9 different spatial classes related to different elevation sectors, and showed considerable generalization capabilities over a validation set with responses from subjects different from the ones in the training set. The explainability techniques were applied over the trained model to obtain saliency maps indicating the relevant frequency bands used by the network to classify a given input response into one of the elevation classes. Although both CAM and Grad-CAM provided similar saliency regions, the results of CAM appeared to be more consistent across both the training and validation sets. The saliency regions identified by the applied explainable techniques were also consistent with most findings obtained through psychoacoustic experiments, although additional unexpected low-frequency effects were obtained for behind directions.

ACKNOWLEDGMENTS
This work received funding from Grant RTI2018-097045-B-C21 funded by MCIN/AEI/10.13039/501100011033 and “ERDF A way of making Europe”. Thanks also to Grant FPU20/05384, funded by MCIN/AEI/10.13039/501100011033 and "ESF Investing in your future".

REFERENCES


Spatial post-filter estimation based on low-order beamformers

Stefan A. WIRLER(1), Ville PULKKI(2)

(1) Aalto University, Finland, stefan.wirler@aalto.fi
(2) Aalto University, Finland, ville.pulkki@aalto.fi

ABSTRACT
This paper investigates the Cross-Pattern Coherence (CroPaC) algorithm using low-order beamformers to estimate a parametric spatial post-filter. The algorithm utilizes a coherence-based measure between the output of microphone signals and beamformers. The obtained spatial post-filter assigns attenuation values in the time-frequency domain, which results in the modulation of a microphone signal or the output signal of a beamformer. In this work, four microphones are used to create beamformers with different look directions to estimate the spatial post-filter for different directions. The low number of microphones makes it feasible for usage in mobile devices such as smartphones or hearing aid devices. A conventional delay-and-sum beamformer and the microphone signals are used to apply the estimated spatial post-filter. The performance evaluation is conducted by creating multi-speaker scenarios with different background noise levels. The algorithm is tested with different microphone positions suitable for usage in smartphones. Compared to commonly used beamforming techniques, the method shows improvement in background noise reduction and suppression of interfering sound sources.

Keywords: Spatial Post-Filter, Interference Suppression, Noise Reduction, Differential Microphone Array

1 INTRODUCTION
The use of signal enhancement techniques in handheld devices such as smartphones has become more applicable due to the higher amount of used microphones and the increased processing power over the last years. Signal enhancement techniques can provide an improvement in terms of noise reduction and suppression of unwanted sources, e.g., an interfering speaker. The typical approach to reduce those disturbing signals is the use of beamforming techniques. Beamforming tries to combine signals from different microphones to enhance a target source and decrease the noise level. The most well-known technique is the delay-and-sum beamforming (DSB) which improves the signal by delaying and adding the input of multiple microphones [1]. Another technique is the use of differential microphone arrays (DMAs), which measures the difference of the sound field between adjacent microphones [3]. Their main advantage compared to other beamforming methods is their frequency independence if the spacing is sufficiently small. Nonetheless, the frequency independence is always a trade-off as with decreasing inter-microphone spacing, the white noise gain in low-frequency regions will increase [3]. Further, the performance of microphone array beamforming is often determined by the number of used microphones, which is still comparatively low in small-scale devices.

A method to further improve the performance of beamformers is the post-filter technique. This technique can add further noise suppression by calculating a multi-channel Wiener Filter and applying it to a beamformed signal. One post-filter based on the multi-channel wiener filter is the Zelinski post-filter. It assumes the noise captured by different microphones is uncorrelated [4]. A generalization of the Zelinski post-filter is the McCowan post-filter, which models the coherence of the noise between the microphones [5]. A significant drawback of those post-filter is the assumption of rather big spacing between microphones and their decreased performance at low frequencies [6].

A post-filter that circumvent these drawbacks, especially in low-frequency regions, is the Cross Pattern Coherence (CroPaC) parametric post-filter [7]. The post-filter can be used to modulate omnidirectional or directional signals, i.e., a beamformed signal. It uses directional microphone signals to estimate a post-filter based on
the normalized cross-spectral density. For the calculation of the post-filter, the algorithm uses directional signals with the maximum gain and equal phase in the same direction. The algorithm can improve noise reduction further and is suitable for real-time applications [8].

This paper is motivated by studying the feasibility of using the CroPaC algorithm with low-order beamformers in devices such as a smartphone. Due to the number of microphones and their positioning restriction, higher-order beamformers are often not feasible to use directly in such devices. Hence, a low-order beamforming method is incorporated for usage with the CroPaC post-filter algorithm. In implementations before, the postfilter is estimated and tested with first-, second-, or higher-order beamformer [9].

To obtain the low-order directional signals, first-order fixed DMAs are employed. They provide the properties to estimate the CroPaC post-filter and are realizable with a small amount of microphones. With the estimated post-filter, the directivity can be increased compared to only using differential microphone arrays by utilizing the phase information and further increasing noise suppression, especially in the low-frequency regions. In this study, the spatial post-filter is obtained for four different fixed directions. The obtained spatial post-filter is then used to modulate the signal of an omnidirectional and the output of a DSB signal.

To obtain the low-order directional signals, first-order fixed DMAs are employed. They provide the properties to estimate the CroPaC post-filter and are realizable with a small amount of microphones. With the estimated post-filter, the directivity can be increased compared to only using differential microphone arrays by utilizing the phase information and further increasing noise suppression, especially in the low-frequency regions. In this study, the spatial post-filter is obtained for four different fixed directions. The obtained spatial post-filter is then used to modulate the signal of an omnidirectional and the output of a DSB signal.

The organization of the paper is as follows. Section 2 introduces the method, including the signal model, differential beamforming, and the estimation of the spatial post-filter. In Section 3, details about the used implementation is given. Section 4 shows the evaluation of the low-order post-filter and Section 5 concludes the paper.

2 METHOD
This section introduces the used signal model, the derivation of the used DMA and their beamforming weights, and the calculation of the post-filter with the CroPaC algorithm.

2.1 Signal Model
The signal arriving at a microphone array with \( N \) Sensors can be described as

\[
x_n(t) = b_n(t) + v_n(t),
\]

where \( x_n(t) \) is the signal obtained by one single sensor, \( b_n(t) \) the clean signal of a source, \( v_n(t) \) the additive noise and \( n \) the sensor number.

Here, we consider the classic far-field model, with sound propagating in an anechoic environment. Let the distance between the microphone be \( \delta \), the speed of sound \( c \), and the direction to the source \( \theta \). The steering vector then reads as

\[
d(\omega, \theta) = \left[1, e^{-j\omega \delta \cos \theta / c}, \ldots, e^{-j(N-1)\omega \delta \cos \theta / c}\right]^T,
\]

with the angular frequency \( \omega = 2\pi f \), the temporal frequency \( f \), superscript \( T \) the transpose operation and the imaginary number \( j = \sqrt{-1} \). With the steering vector, the signal model defined in Equation 1, can be written in the frequency domain as

\[
X_n(\omega, \cos \theta) = e^{-j(n-1)\omega \delta \cos \theta / c} S(\omega) + V_n(\omega),
\]

where \( S(\omega) \) is the signal of a sound source.

In vector notation, Equation 3 is given as

\[
x(\omega, \cos \theta) = [X_1(\omega), \ldots, X_N(\omega)]^T
= d(\omega, \theta)S(\omega) + v(\omega),
\]

where \( v(\omega) \) is the additive noise vector defined in the same manner as \( x(\omega, \theta) \).

To obtain the directional signal, the input signal of each microphone is weighted by a complex weight denoted as

\[
h(\omega) = [H_1(\omega), \ldots, H_N(\omega)].
\]
The output of the beamformer then simply reads as

\[ Y(\omega) = \sum_{n=1}^{N} H_n^{*} X_n \]

\[ = h^H d(\omega, \theta) S(\omega) + h^H v(\omega), \]

where \( Y(\omega) \) is the beamformer output signal and superscript \( ^H \) denotes the conjugate-transpose operation.

### 2.2 Differential Beamformer

In order to estimate the CroPaC post-filter, directional signals are needed. Due to the restricted amount of microphones in hand-held devices, i.e. Smartphones, only first-order directional signals are considered. One of the requirements is the maximum of the magnitude and the equal phase in the desired look direction. To create those beampatterns, differential beamformers are employed, which are briefly introduced in the following.

From the signal model, we can define constraints to fulfill the requirements to obtain a certain beampattern using DMAs. In the case of a first-order beamformer, two constraints are defined. The first constraint to consider is the distortionless response in the look direction (\( \theta = 0 \)), which corresponds to the maximum magnitude requirement of CroPaC, i.e.,

\[ h^H(\omega) d(\omega, \cos 0) = 1. \]

The second constraint determines the direction \( \theta \) of the null in the response, giving different beampatterns depending on the direction. The constraint can be formulated as [3],

\[ h^H(\omega) d(\omega, \cos \theta) = 0. \]

Solving for those constraints leads to the beamformer weights, expressed as

\[ h(\omega) = \frac{1}{1 - e^{-j\omega \tau_0}} \begin{bmatrix} 1 \\ -e^{-j\omega \tau_0} \end{bmatrix} \]

where \( \tau_0 = \delta / c \). The resulting beam pattern is determined by choosing different values for \( \cos \theta \). This resembles an alteration of a delay element or a phase shift in the frequency domain. In the following, only the dipole beampattern is used. The dipole beampattern is chosen as it has the highest directivity when only considering the front lobe. Therefore, \( \cos \theta \) is set to 0, which simplifies Equation 11 further to

\[ h(\omega) = \frac{1}{1 - e^{-j\omega \tau_0}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \]

### 2.3 Spatial Filter Estimation

The CroPaC spatial filter is originally defined with only directional signals. Here, only two directional signals and the omnidirectional signal are used due to the limited amount of microphones and, therefore, the lack of directional signals with an order higher than one. In the following, the time-frequency formulation is used with \( k \) denoting the frequency bin and \( i \) the time frame. The directional signals obtained by the differential microphone array will be denoted as \( Y_l(k, i) \). \( l \) indicates a different direction of the maximum magnitude and the positive phase, with \( l \in 1, \ldots, L \) and \( L \) equals the number of possible look directions with a positive phase. The estimation of the post-filter is based on the normalized cross-spectral density and a half-wave rectification

\[ G(k, i) = \max \left( \lambda, \frac{2\Re[\Phi_{01}(k, i)]}{\Phi_{00}^{m} + \Phi_{11}^{m} + \Phi_{11}^{p}} \right), \]
with the cross-spectral density \( \Phi_{01}(k,i) = E\{X^*_1(k,i)Y_1(k,i)\} \), where \( E\{\} \) denotes the expectation operator, the complex-conjugated operator \(^*\), \( \Re \) the real operator and \( \lambda = [0,1] \) is the spectral floor level to limit the attenuation introduced by the post-filter. The denominator is used for the normalization of the post-filter. It is calculated with the auto-spectral density of the omnidirectional microphones \( \Phi_{00}^m = E\{|X_m(k,i)|^2\} \) and the auto-spectral density of the directional signals of first-order \( \Phi_{11}^q = E\{|Y_q(k,i)|^2\} \) and \( \Phi_{11}^p = E\{|Y_p(k,i)|^2\} \), where the \( Y_p(k,i) \) has its look direction turned by 90° compared to \( Y_q(k,i) \). The half-wave rectification leads to one of the main advantages of using the CroPaC post-filter compared to only using the signal obtained by the DMA. In the case of only using the DMA signal, the performance will decrease as sources obtained from the opposite look direction will not be suppressed. With the CroPaC processing, the back-lobe is suppressed due to the negative phase coherence of the signal.

The estimation of the attenuation patterns for the CroPaC post-filter can suffer from sudden changes such as fluctuation of the sound level or changes in the noise. As the calculation proceeds in every time frame, these fluctuations can introduce a high variance of the values of \( G \) in each frequency-time frame. These variances will introduce artifacts hearable as musical noise [10]. This effect can be reduced by smoothing the post-filter with a recursive one-pole filter after the standard recursive update formula [11] as

\[
G_r(k,i) = 1 - \alpha(k,i)G(k,i) + \alpha(k,i)G_r(k,i-1),
\]

where \( \alpha(k,i) \) is the adaptive smoothing coefficient and \( G_r(k,i) \) the smoothed post-filter. To further improve the smoothing, the smoothing coefficient is estimated adaptively. Using the instantaneous signal-to-noise ratio (ISNR), the smoothing coefficient can then be determined by

\[
\alpha(k,i) = \frac{1}{1 + \xi^2(k,i)},
\]

where \( \xi \) is the ISNR of the signals.

The output signal \( Z(k,i) \) is then obtained by modulating a signal with the estimated post-filter as

\[
Z(k,i) = G_r(k,i)U(k,i).
\]

The signal modulated by the post-filter \( U(k,i) \) should originate from a signal with a spectrally flat response, such as an omnidirectional or a low-order directional signal. The last step is the inverse time-frequency transformation of the output signal to restore a time domain signal.

3 IMPLEMENTATION

The implementation of the method is based on the assumption, that each pair lies on the axis corresponding to the look direction. Using different microphone pair combinations will lead to different look directions, as the direction of a fixed first-order DMA is determined by the relative position of each microphone pair. The pairs’ positions were chosen so that the look directions correspond to an angle in the azimuth plane of 0°, 90°, 180°, and 270°. Figure 1 shows an example of a possible microphone layout. The distances \( \delta \) between the microphone pairs is denoted as \( d_1 \) and \( d_2 \), respectively. With hindsight of the distance, the frequency independency can’t be assumed anymore. With the chosen layout the beampattern will collapse at a certain frequency which is lower with higher distance between the sensors [12]. In the shown example, the microphone pairs do not lie exactly on the reference axis. This can lead to an angular error with respect to the maximum of the main lobe [13]. Here, it is assumed that the distances of the sources are big enough to neglect this angular error. The inversion of the phase is achieved by simply multiplying the weights of the beamformer by \(-1\) (c.f. Equation 12). The phase inversion is needed to to switch the direction of the positive phase and hence to the change to the opposite direction of the main lobe. The omnidirectional microphone closest to the look direction is chosen to calculate the cross-spectral density between the beamformed and the omnidirectional signal.

In Figure 2, the block diagram for processing one microphone pair is shown. In the used implementation, the weights of the beamformer were pre-calculated according to equation 12 and applied in the time-frequency
domain. In each time step, the post-filter is estimated according to the steps in section 2.3. The estimated post-filter is then used to modulate either the omnidirectional or the DSB signal. The microphones used for the DSB, correspond to the same as those used for the differential beamforming. The last step to regain the time-domain signal is the inverse time-frequency transformation.

4 EVALUATION

To assess the performance of the estimation of the CroPaC filter with low-order directional signals, the image source method was used to generate the evaluation scenario [14]. In order to simulate a smartphone, four microphones are placed in a reverberant room with coordinates compatible with a smartphone layout. Around those microphones, four speakers were placed at 0°, 90°, 180°, and 270° with a distance of 1.5m. The background noise was created by placing 12 pink noise sources around the microphones. The used window size of the STFT is 1024 samples, with a hop-size of 128 samples at a sampling frequency of 48kHz.

The room was simulated with a dimension of 5 x 4 x 2.5 m³ and a reverberation time of $T_{60} = 300$ms. The smoothing parameter $\alpha$ was calculated after equation 15. The floor factor $\lambda$ was set to 0.1 for all experiments. The distance between the microphones is $d_1 = 14$cm and $d_2 = 8$cm (c.f. Figure 1), to resemble microphones mounted at the edges of a smartphone. For all evaluation cases, the performance was evaluated by comparing the output signal of the DMA without post-filter, the CroPaC post-filtered signal of an omnidirectional signal, and the CroPaC post-filter applied to a DSB signal.

To evaluate the noise reduction, the enhancement of the segmental SNR is used [15]. The segmental SNR enhancement (segSNRe) is defined as the difference between the segmental SNR of the noisy input and the enhanced signal. For the estimation of the segSNRe, only one source is active at a time. This is done for all four directions. The performance was evaluated with input SNRs of -5, 5, 10dB. In Table 1 the segSNRe for the CroPaC post-filter applied to an omnidirectional (Omni - CroPaC), the DSB signal (DSB - CroPaC) and the DMA without post-filtering is shown for three different SNR levels. The best overall performance is seen for the CroPaC post-filter applied to an omnidirectional signal. Only in the low SNR case of -5dB the CroPaC post-filter applied on the output signal of a DSB is better than the modulation of the omnidirectional signal for the 0° and the 180° direction. This difference is mainly due to the chosen microphone layout. For those directions, the microphone spacing is bigger, therefore, the beampattern will collapse at a lower frequency compared to a smaller distances. This also explains the differences in performance of the 0° and the 180° to the 90° and the 270° directions. In overall, the CroPaC post-filter applied to the omnidirectional signal shows the best performance in segSNRe.

To evaluate the source separation capabilities, four sources are active simultaneously. In Figure 3, the signal
Figure 2. Block diagram of the processing with two microphones and a differential beamformer

Table 1. segSNRe and SIRe results

<table>
<thead>
<tr>
<th>SNR</th>
<th>segSNRe (dB)</th>
<th>SIRe (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-5 5 10</td>
<td>5 10</td>
</tr>
<tr>
<td>Omni - CroPaC ($0^\circ$)</td>
<td>4.31 4.12 3.66</td>
<td>16.55 17.98</td>
</tr>
<tr>
<td>Omni - CroPaC ($90^\circ$)</td>
<td>6.46 5.10 4.32</td>
<td>17.2 18.62</td>
</tr>
<tr>
<td>Omni - CroPaC ($180^\circ$)</td>
<td>5.22 4.22 3.48</td>
<td>10.72 11.57</td>
</tr>
<tr>
<td>Omni - CroPaC ($270^\circ$)</td>
<td>6.86 5.59 4.84</td>
<td>20.03 20.85</td>
</tr>
<tr>
<td>DSB - CroPaC ($0^\circ$)</td>
<td>6.13 3.99 2.93</td>
<td>11.21 11.59</td>
</tr>
<tr>
<td>DSB - CroPaC ($90^\circ$)</td>
<td>6.23 4.08 3.02</td>
<td>11.44 11.96</td>
</tr>
<tr>
<td>DSB - CroPaC ($180^\circ$)</td>
<td>5.55 3.47 2.44</td>
<td>7.20 7.33</td>
</tr>
<tr>
<td>DSB - CroPaC ($270^\circ$)</td>
<td>6.10 3.99 2.95</td>
<td>16.34 15.67</td>
</tr>
<tr>
<td>DMA ($0^\circ$)</td>
<td>3.33 3.52 3.46</td>
<td>3.53 3.82</td>
</tr>
<tr>
<td>DMA ($90^\circ$)</td>
<td>2.63 3.27 3.54</td>
<td>3.70 4.48</td>
</tr>
<tr>
<td>DMA ($180^\circ$)</td>
<td>3.30 2.95 2.82</td>
<td>0.04 0.16</td>
</tr>
<tr>
<td>DMA ($270^\circ$)</td>
<td>2.13 2.77 3.01</td>
<td>8.03 8.99</td>
</tr>
</tbody>
</table>

of the desired speaker, the spectrogram of one omnidirectional input microphone with four speakers, the output of the low-order CroPaC post-filter modulated an omnidirectional signal, modulated a DSB signal, and the output of the DMA is shown. It is shown that the main advantage of the low-order CroPaC post-filter is the ability to suppress the back lobe when comparing it to the output signal of the DMA. In the time-domain
Figure 3. Spectrogram and time-domain signal of the desired speaker with noise (at 270°), overlapping speaker with noise, the output of the CroPaC post-filter applied to an omnidirectional signal, the output of the CroPaC post-filter applied to the DAB output and the output of only using the differential microphone array.
signal of the DMA, an interfering speaker is still active at around 2.5s - 3s. Whereas for the CroPaC filtered signals, this interfering speaker is suppressed. Noise reduction is observable for the CroPaC post-filtered and DMA output signals. In the case of the signals processed by the CroPaC post-filter, better noise suppression for low-frequencies is evident compared to the output of the DMA.

In order to assess an objective measure of the ability to separate sources, the enhancement of the signal-to-interference ratio is calculated [16]. The signal-to-interference enhancement (SIRe) is calculated as the difference of the SIR of the processed signal and the SIR of the unprocessed signal. The results are shown in Table 1. The SIRe was evaluated for SNR levels of 5dB and 10dB. The biggest improvement is achieved with the CroPaC filter applied to an omnidirectional signal. The CroPaC post-filter applied on a DSB output signal also improves, but significantly less compared to the CroPaC post-filter applied to the omnidirectional signal. In terms of performance, it is a trade-off between the performance of the DMA used for the post-filter calculation and the DSB. The trade-off is mostly due to the dimension of the used microphone array, as the cut-off frequencies of both techniques have an opposite behavior in terms of the cut-off frequency and the distance between the microphones. The DMA performance was the worst in SIRe, leading to a maximum SIRe of 8dB. The main reason for this behavior stems from the back lobe of the DMA. In such a test setup, an interfering speaker is always present when using a DMA.

5 CONCLUSION

This paper proposes using a low-order beamformer to estimate a spatial post-filter based on the Cross Pattern Coherence algorithm, which is suitable for hand-held devices, e.g., smartphones. It can be used for real-time applications and only needs a low amount of microphones. With the setup, four possible look directions were realized. With different combinations of microphones, more directions would be realizable. The combination of the post-filter showed an improvement over only using a differential microphone array without post-filtering in terms of noise reduction and interference suppression. Using a delay-and-sum beamformed signal instead of the omnidirectional signal only leads to a slight improvement in terms of segmental signal-to-noise ratio enhancement in the low signal-to-noise ratio case for a bigger microphone spacing. A point to consider is that only one beampattern realizable with the differential microphone array was used and implemented. Further, only two microphones out of the possible four microphones were used for the beamforming. Using all of the microphones could further improve the performance of the estimation of the Cross Pattern Coherence post-filter with the differential microphone array.

REFERENCES


Multizone sound field reproduction with direction-of-arrival-distribution-based regularization and its application to binaural-centered mode-matching

Ryo MATSUDA$^{1,2}$; Yuki KAWASAKI$^{1}$; Makoto OTANI$^{1}$

$^1$Graduate School of Engineering, Kyoto University, Japan
$^2$Research and Development Division, Yamaha Corporation, Japan

ABSTRACT

In higher-order Ambisonics, a framework for sound field reproduction, secondary-source driving signals are generally obtained by regularized mode matching. The authors have proposed a regularization technique based on direction-of-arrival (DoA) distribution of wavefronts in the primary sound field. Such DoA-distribution-based regularization enables a suppression of excessively large driving signal gains for secondary sources that are in the directions far from the primary source direction. This improves the reproduction accuracy at regions away from the reproduction center. First, this study applies the DoA-distribution-based regularization to a multizone sound field reproduction based on the addition theorem. Furthermore, the regularized multizone sound field reproduction is extended to a binaural-centered mode matching (BCMM), which produces two reproduction points, one at each ear, to avoid a degraded reproduction accuracy due to a shrinking sweet spot at higher frequencies. Free-field and binaural simulations were numerically performed to examine the effectiveness of the DoA-distribution-based regularization on the multizone sound field reproduction and the BCMM.

Keywords: Sound field reproduction, Direction-of-arrival, Mode-matching

1. INTRODUCTION

Sound field reproduction is a technique for physically reproducing a real or virtually simulated sound field (1, 2). The sound field is generally recorded using a microphone array (3, 4) and reproduced using a loudspeaker array (5, 6, 7) or headphones with head-related transfer functions (8, 9). Higher-Order Ambisonics (HOA) is a framework for recording, analyzing, and reproducing a sound field by expanding a sound field in terms of spherical harmonic functions. However, an infinite order of spherical harmonic expansion is truncated depending on the number of microphones and loudspeakers in practice. The size of the recorded and reproduced sound field, which is so called sweet spot, is limited to a spherical region at a certain distance from the origin of the spherical harmonic expansion due to the nature of the radial function when the infinite order is truncated. In general, sweet spot size is inversely proportional to frequency.

When reproducing a sound field using a loudspeaker array in HOA, mode matching is generally used, which is a method of determining the loudspeaker driving signals so that the expansion coefficients of the primary sound field and those of the reproduced sound field synthesized by the loudspeaker array are matched in the least-squares sense. In mode matching, a regularization technique based on direction-of-arrival (DoA) distribution of wavefronts in the primary sound field has proposed (10). It has been shown that using DoA-distribution-based regularization suppresses unwanted output of secondary sources, especially in the opposite direction to the primary source, and improves the accuracy of reproduction at positions away from the sweet spot.

This study applies DoA-distribution-based regularization to multizone sound field reproduction and evaluates its accuracy. Furthermore, the DoA-distribution-based regularization is also applied to binaural-centered mode matching (BCMM) (11), which generates two sweet spots at the ear positions.
2. Multizone reproduction with DoA-distribution-based regularization

2.1 Formulation

The loudspeaker driving signals for multizone reproduction is expressed as,

\[ \mathbf{d} = (\mathbf{C}^\dagger \mathbf{C} + \lambda \mathbf{I})^{-1} \mathbf{C}^\dagger \mathbf{b}, \]

where \( \mathbf{d} \in \mathbb{C}^{L \times 1} \) is the vector consists of loudspeaker driving signals; \( \mathbf{C} \in \mathbb{C}^{Q(N_q+1)^2 \times L} \) consists of the spherical harmonic coefficients of the transfer function for the loudspeakers corresponding to the expansion around each local origin \( \mathcal{O}(q) (q = 1, ..., Q) \); \( \mathbf{b} \in \mathbb{C}^{Q(N_q+1)^2 \times 1} \) is the vector containing the spherical harmonic coefficients of the primary sound field corresponding to \( \mathcal{O}(q) \); \( L \) is the number of the loudspeakers; \( N_q \) is the truncation order for each zone; \( Q \) is the number of multizone; \( \lambda \) is the regularization parameter; \( \mathbf{I} \in \mathbb{R}^{L \times L} \) is an identity matrix (5, 6).

DoA-distribution-based regularization, proposed by Kawasaki et al. (9), is applied to mode matching for multizone reproduction, expressed as,

\[ \mathbf{d} = (\mathbf{C}^\dagger \mathbf{C} + \lambda \mathbf{\Sigma})^{-1} \mathbf{C}^\dagger \mathbf{b}, \]

where \( \mathbf{\Sigma} \in \mathbb{R}^{L \times L} \) is a diagonal matrix given by \( [\mathbf{\Sigma}]_{i,i} = 1/\|\mu(r_i, k)\| \). Here, \( \mu(r_i, k) \) represents the weights of the spherical wave arriving from each direction on the sphere, known as the single-layer potential (1),

\[ \mu(r_i, k) = \sum_{n=0}^{N} \sum_{m=-n}^{n} \frac{j h_n^m(k)}{k R_i^2 h_n^{(2)}(k R_i)} Y_n^m(\theta_i, \phi_i), \]

where \( k \) is the wavenumber; \( h_n^{(2)}(\cdot) \) is the spherical Hankel function of the second kind; \( Y_n^m(\cdot) \) is the spherical harmonic function of n-th order and m-th degree; \( r_i, \theta_i, \phi_i \) is the position of point source. Since the weights are relative value, the norm \( \|\mu(r_i, k)\| \) is normalized by the maximum value over all the loudspeakers. This weighting suppresses the output of loudspeakers located on the opposite side of the primary source.

2.2 Numerical simulation

A numerical simulation was conducted to compare mode-matching methods with two regularizations; one is the DoA-distribution-based regularization (Proposed) in Eq. (2), and the other is normal regularization (Conventional) in Eq. (1).

The loudspeakers were assumed as point sources, and 121 loudspeakers were placed on a sphere with a radius of 1.5 m whose angular coordinates were determined by spherical t-design (12). The truncation order \( N_q \) in Eqs. (1) and (2) was set to 6, and \( N \) in Eq. (3) was also set to 6. Two reproduced zones were centered at \((0.5,0,0)\) and \((-0.5,0,0)\). The simulation was performed for a single frequency, with the primary source being a point source with a frequency of 1 kHz in a free field.

The normalized reproduction error (NRE) (7) expressed as,

\[ \text{NRE}(r,k) = 10 \log_{10} \frac{\int_{\Omega} \left| p_{\text{rep}}(r,k) - p_{\text{des}}(r,k) \right|^2 d\Omega}{\int_{\Omega} \left| p_{\text{des}}(r,k) \right|^2 d\Omega} \]

was calculated over spherical regions centered at each local origin of zone with a radius \( R = 0.8N/k \), \( N/k \), and \( 1.2N/k \) for each direction of primary source. The spherical regions are discretized as an orthogonal grid in 0.01 m intervals. The regularization parameter \( \lambda \) in Eqs. (1) and (2) was set to the maximum eigenvalue of \( \mathbf{C}^\dagger \mathbf{C} \times 10^{-3} \).
Figure 1 illustrates the results of NRE for the zone centered at (0.5,0,0). Results for only one zone are demonstrated because the centers of two zones are symmetric. The figure reveals that the proposed method has a smaller NRE when $R$ is greater than the radius determined by $N = kR$, which defines the sweet spot.

3. BCMM with DoA-distribution-based regularization

The DoA-distribution-based regularization was applied to BCMM (11). Reproduced binaural signals were simulated under the same conditions as in Sec. 2.2, except that the center of two zone were set to (0,0.0705,0) and (0,−0.0705,0). Reproduction performances were compared among the BCMM with DoA-distribution-based regularization (BCMM w/ DoA), the BCMM with conventional regularization (BCMM w/ conv) and the conventional mode-matching in global region (MM). The transfer functions from each loudspeaker or primary source to the ears of the dummy head were numerically calculated using the boundary element method (13).

Figure 2 demonstrates the normalized error (NE) of reproduced signals. Results only at the left ear are shown because the results are almost symmetrical between both the ears. BCMM w/ DoA has smaller NE than the other methods, especially in the high frequency range. As shown in the free-field simulation results in Sec. 2.2, DoA-distribution-based regularization suppressed the output of the loudspeaker located on the opposite side of the primary sound source and, therefore, reduces the error outside the sweet spot defined by $N = kr$, resulting in smaller errors in the binaural signals.

4. Conclusion

The DoA-distribution-based regularization was applied to multizone reproduction and BCMM. Numerical simulations showed that the DoA-distribution-based regularization achieves smaller reproduction error in the region slightly outside the sweet spot; it also leads to smaller error in the reproduced binaural signal when it is applied to BCMM.
ACKNOWLEDGEMENTS

This work was partly supported by grants-in-aid from JSPS, Japan (Grant Nos. 19H04153 and 19H04145).

REFERENCES

Binaural Source Localization in Median Plane using Learning based Method for Robot Audition

Priyadarshini Dwivedi, Gyanajyoti Routray, Rajesh M. Hegde

Indian Institute of Technology Kanpur, India
E-mail: {priyadw, groutray, rhegde}@iitk.ac.in

ABSTRACT
This article presents a learning-based binaural source localization technique in the median plane and its application to robot audition. Binaural recordings capture the audio signal and acoustic transfer function from the source to the ears, known as the head-related transfer function (HRTF), which parameterizes spatial cues such as interaural time difference (ITD) and interaural level difference (ILD). ITD and ILD cues are prominent for source localization in the horizontal plane. Since ITD and ILD are nearly equal to zero in the median plane (the ear canal of both the ears is colocated), the localization is complex. Therefore, monaural spectral cues such as spectral notches are investigated for median plane source localization. The spectral notch represents the delay between the direct and the reflected wave. As it varies with the elevation angle, a learning-based model is developed to map the spectral notch with the elevation angle. The spectral notch features are extracted from the binaural recording using linear prediction cepstral coefficients (LPCC) and linear prediction residual coefficients (LPRC). Simulations and experiments are carried out using high-spatial-resolution HRTF measurements from CIPIC dataset to evaluate the performance. The results show a significant improvement in localization accuracy compared with existing methods.

Keywords: Direction of arrival, spherical harmonic domain, learning approach, convolutional neural network

1 INTRODUCTION
Sound source localization has always been a challenging and interesting signal processing problem. A wide variety of research is going on to improve localization accuracy. Humans have great potential to estimate the source position with respect to them. Therefore, to mimic localization like humans and to further improve the location accuracy, various methods have been explored so far for binaural source localization. In this context, spatial audio has been broadly investigated throughout. To perfectly exhibit spatial audio for every person in the virtual auditory space, each individual’s personalized head-related impulse responses (HRIRs) are required. If personalized HRIRs are not considered, it may lead to front-back ambiguity, elevation angle misperception, and inside-the-head localization [1]. Thus the localization accuracy gets greatly affected; therefore, human anatomy is important for accurate binaural source localization. The features that can help estimate the azimuth location, i.e., for horizontal plane localization, are considered interaural. The prominent features in this regard are interaural time difference (ITD) and interaural level distance (ILD). ITD is the difference in time of arrival of a signal between two ears and ILD is the difference in signal level. These occur due to diffraction from the head. However, ITD and ILD are primary cues for azimuth estimation in horizontal plane [2]. But they do not perform well for elevation estimation in the median plane. Also, there is front-back ambiguity is estimating the elevation as the ITD and ILD have negligible variation with elevation direction [3]. Therefore, it motivated us to explore various features for elevation estimation in the median plane. Interaural cues are almost equal to zero for the median plane as both the ears are co-located for the entering signal at the ear canal. Thus, monaural cues are realised for the median plane localization [4, 5]. Also, in noisy and reverberant environments, the binaural source localization is discussed in [6, 7].
The proposed work investigates binaural sound source localization in the median plane. The significant contribution of this work is as follows. Since the physical cues such as ITD and ILD are consistent at both the ears in the median plane, the elevation estimation is challenging. In this context, the monaural cues are investigated. The monaural cues capture the reflections from the pinnae and vary with the elevation. The spectral notches of the HRTFs are the prominent features for the elevation estimation. Linear prediction cepstral coefficients (LPCC) and linear prediction residual coefficients (LPRC) methods are adopted to extract spectral notches. The LP residue considers the source filter model and approximates the HRTF by all-pole coefficients. Then a Hanning window is deployed to extract the spectral notches. In this work, inverse discrete cosine transform is used to modify the cepstrum of the binaural signal. This modification improves the spatial resolution. Subsequently, a convolutional neural network (CNN) is developed to map these LPRC coefficients along with the spectral notches to the corresponding elevation angles to build a learning framework for binaural localization in the median plane. Comparing the proposed method’s performance with the state-of-art methods shows a significant improvement. Further real-time experiments are conducted to validate the performance of the proposed method.

The rest of the paper is organized as follows. The signal model for binaural source localization is given in Section 2. Section 3 describes the extraction of spectral notches and CNN framework. The performance of the proposed model is evaluated and compared with the state-of-art methods in Section 4. Finally, Section 5 concludes the paper.

2 SIGNAL MODEL AND PROBLEM FORMULATION

This section describes the signal model for binaural source localization. Subsequently, the various source direction-dependent binaural cues are depicted. The linear prediction cepstral coefficient (LPCC) and linear prediction residual coefficients for the binaural signals are described at the end of this section.

2.1 Binaural Source Localization

The binaural audio is a two-channel audio signal obtained by convoluting the monophonic audio with the head-related impulse response (HRIR). The discrete time signal model for the binaural signal is expressed as

\[ x_{l/r}[n] = h_{l/r} * s[n] + \eta_{l/r}[n] \] (1)

where \( x_{l/r} \) is the binaural signal reaching at the ears, \( s[n] \) is the monophonic discrete-time source signal, \( l/r \) represents left or right. \( \eta_{l/r}[n] \) is the uncorrelated additive noise for left or right ear received signals. \( h_{l/r} \) is the head-related impulse response (HRIR), defining the channel response between the source and the ears. The HRIRs also capture the pinna effects, head and torso reflection, head diffraction, and knee reflection, along with the room reverberation. The HRIR is referred to as the head-related transfer function (HRTF) in the spectral domain. The prominent cues for sound source localization are the physical cues such as interaural time difference (ITD) and interaural level difference (ILD). The ITD and ILD are time-invariant and, for a particular source direction, defined as

\[ ILD(\omega, \theta, \phi) = 20 \log_{10} \left| \frac{X_{l}(\omega, \theta, \phi)}{X_{r}(\omega, \theta, \phi)} \right| = 20 \log_{10} \left| \frac{H_{l}(\omega, \theta, \phi)}{H_{r}(\omega, \theta, \phi)} \right| \] (2)

\[ ITD(\omega, \theta, \phi) = \frac{1}{\omega} \left( \frac{\hat{X}_{l}^\omega(\omega, \theta, \phi)}{\hat{X}_{r}^\omega(\omega, \theta, \phi)} + 2\pi p \right) = \frac{1}{\omega} \left( \frac{\hat{H}_{l}(\omega, \theta, \phi)}{\hat{H}_{r}(\omega, \theta, \phi)} + 2\pi p \right) \] (3)

where \((\theta, \phi)\) represents the source elevation and azimuth, \(\omega\) represents the frequency. \( X_{l/r}(\omega, \theta, \phi) \) represents the short time Fourier transform (STFT) of the signal received by the left and right ears and \( H_{l/r}(\omega, \theta, \phi) \) represents the HRTFs. To estimate the relationship between the difference in distance traveled by the signal to reach the two ears and the ITD, phase unwrapping factor \( p \) is taken into account and is in the modulo-\(2\pi\) measure of phase. The physical cues ILD and ITD measure the time delay or the phase difference between the signals received by the ears. Further, the cross-correlation function (CCF) is investigated in [8]. The CCF measures the relative displacement between the two binaural channel signals and is computed as
where $N$ is the signal length and $k$ is the delay, i.e., the CCF uses a sliding window to measure the similarity between the two channels, and the maximum value is the time lag. Since CCF captures the variation in the source location, it is considered for binaural source localization. ITD is calculated from the time lag and the CCF-ITD joint parameter is considered for the source localization.

The binaural source localization methods that use the physical and time lag cues are sufficient in azimuth estimation. However, the binaural source localization is challenging when the source is present in the median plane as both the left and right HRTFs are equal. Hence the ILD and ITD value doesn’t change with the elevation variation. In this context, the monaural cues are considered for the source localization [9]. Gammatone filter energies (GFEs) and Mel-Frequency cepstral coefficients (MFCCs) are the monaural frequency-dependent cues among the others adopted for the elevation estimation. Gammatone filter works analogous to the cochlea and separates the sound into overlapping frequency bands. Further, the effects due to anthropometric parameters are modeled using the energy in each sub-band. In [10], MFCC features are explored for elevation estimation. The Mel-Frequency cepstrum is the short-term power spectrum of the cosine transform of a log power spectrum on a non-linear mel scale of frequencies and MFCC collectively combined to produce the mel-frequency cepstrum. Linear prediction cepstral coefficients (LPCC) are also one of the cues explored for the median plane localization. This is based on the speech production considering the vocal tract and is modeled using an all-pole filter. LPCC is computed from the covariance or auto-correlation of windowed speech signals. However, these features are not well explored for elevation estimation. Further, in the proposed work, spectral notches are extracted from the linear prediction residual cepstrum (LPRC). The LPRC coefficients well define the pinna spectral notches, which are the function of the spectral notch frequency and the elevation angles. The extraction of the LPRC residual signal is discussed in the subsequent section.

2.2 Modelling Spectral Notched from Binaural Signal Using LPRC

The all-pole model with $\eta$–order linear predictor (LP) is investigated to measure the LP residual of the HRTF. The $n$–th point of minimum phase signal $x[n]$ is predicted as a linear sum of previous $\eta$ samples. The prediction error $e[n]$, also known as LP residue, is expressed as

$$
e[n] = x[n] - \sum_{i=1}^{\eta} a_i x[n-i]$$

The LP coefficients $\{a_i\}$ are obtained by minimizing the expectation of squared error. The LP coefficients are exploited to compute the inverse filter $A(z)$ expressed as

$$A(z) = 1 - \sum_{i=1}^{\eta} a_i z^{-1}$$

The inverse filter $A(z)$ is operated on $H_l(z)$ to evaluate $E(z)$. The LP residue is favorable as it considers a source filter model and evaluates (i) all-pole model coefficients, (ii) residual corresponds to excitation of the source, and (iii) gains representing the energy of the signal [11]. The residue is obtained by applying a Hanning window, which eliminates fast varying frequency components. In the presented work, inverse discrete cosine transform (IDCT) is applied while calculating the cepstrum of $x[n]$. It improves the resolution of the spectral notches and eliminates the convolutional components due to the pinna reflections. The modified cepstrum is expressed as

$$\mathcal{C}_x[n] = \Re(IDCT(\log_{10}(\mathcal{F}\{x[n]\})))$$

where $\Re$ is the real part of the signal and $\mathcal{F}$ is a discrete Fourier transform. A schematic representation for the steps involved in calculating the LP residual from the binaural signal is shown in Figure 1.
3 SPECTRAL NOTCH FEATURES FOR BINAURAL SOURCE LOCALIZATION

In this section, first, the signal model for spectral notch features extraction using LPRC is described. Subsequently, the CNN architecture that maps these spectral features with the corresponding elevation is detailed.

3.1 Spectral Notch Feature Extraction using LPRC

The LP residual spectral features yield the monaural cues that define the dependence of monaural cues on the elevation angles and independence of the physical cues (ITD and ILD). The frequency of the spectral notches is obtained with the help of linear prediction residual cepstrum to map them corresponding to the elevation angle in the median plane. The local minima are known as the spectral notches of the given HRTF, and a reflection model is applied to extract the spectral notches as in [12, 13]. Let the ears receive the sound source signal $x(t)$ as observed at the pinna. The signal received at the ear canal, $s(t)$ is the sum of the direct signal and the reflected signal and is mathematically expressed as

$$s(t) = x(t) + \alpha x(t - t_d(\theta))$$

(8)

where $\theta$ represents the elevation angle, $\alpha$ represents the reflection coefficient, and $t_d(\theta)$ is the time delay. The incident and reflected waves should have destructive interference, therefore

$$t_d(\theta)2\pi f_n(\theta) = (2n + 1)\pi \quad \forall n = 0, 1, 2, \ldots$$

$$\Rightarrow f_n(\theta) = \frac{2n + 1}{2t_d(\theta)} \quad \forall n = 0, 1, 2, \ldots$$

(9)

For $n = 0$

$$f_0(\theta) = \frac{1}{2t_d(\theta)} = \frac{c}{4d(\theta)}$$

(10)

The above expression is multiplied by a factor of 2 considering Satarzadeh’s argument [14] and expressed as

$$f_0(\theta) = \frac{c}{2d(\theta)}$$

(11)

where $c$ represents the sound velocity in air, $f_0(\theta)$ represents the frequency of the first spectral notch, and $d(\theta)$ represents the incident and reflected wave path difference. From (11), it is observed that the notch frequencies are a function of the elevation angles. These notch frequencies are obtained from the LPRC, which is also uniquely mapped to the elevation angles. Therefore, these LPRC coefficients, along with the spectral notches, are the desired features for training the CNN model for binaural localization in the median plane.
3.2 CNN Framework
This section describes a CNN architecture that maps the elevation classes corresponding to the input LPRC spectral notches. Figure 1 illustrates the proposed CNN architecture. The spectral notches from the LP residual analysis along the LPRC are given as input to the proposed CNN model. The proposed model contains a convolutional layer for feature learning with a kernel size of $(3 \times 3)$ having stride of 1 followed by batch normalization and max-pooling of size $(1 \times 2)$. The feature learning layer is followed by the classification layer having two dense layers and a final layer. The rectified linear unit (ReLu) is the activation function at the dense layer, whereas the sigmoidal activation function is used at the output layer. The over-fitting is avoided by considering a dropout of 0.2 before the dense and final layer. The loss function considered during compilation is the categorical cross-entropy, and the optimizer used is adam, with a 0.01 learning rate. The posterior probability for classification is computed by the model. These probabilities are trained for each feature corresponding to the suitable class labels. The CNN output provides the elevation classes corresponding to each source.

4 DATASET AND PERFORMANCE EVALUATION
In this section, the performance of the proposed method is evaluated and compared with the existing DOA methods. Further, the analysis of the simulated results is obtained. An experiment is also carried out to illustrate the performance of the proposed method.

4.1 Dataset
The training and testing dataset is generated considering the HRTF defined in the CIPIC database [15]. The CIPIC database is a public domain database of high-resolution HRIR measurements considering the anthropo-
metric parameters of 45 different subjects. In this database, the elevation varies from $-45^\circ$ to $231^\circ$ with a step size of $5.125^\circ$ and the azimuth varies from $-80^\circ$ to $80^\circ$ in the head-centered interaural polar coordinate system. Further, the mono speech signals are taken from the LIBRISPEECH dataset [16]. The HRIRs are considered for the left and right ears for a fixed azimuth angle and varying elevation angles. These HRIRs are then convolved with the audio sources to get a binaural signal. Thus, the simulated binaural recordings are obtained for further preprocessing to extract desired features and localize the source.

4.2 Performance Analysis

Accuracy is considered as the performance metric and evaluated for the various methods. Accuracy is defined as

$$\text{ACC}(\%) = \frac{\hat{\Delta}}{\Delta} \times 100 \quad (12)$$

where $\Delta$ is the total number of DOAs and $\hat{\Delta}$ are correctly estimated DOAs. Accuracy for all the features is obtained and plotted at various SNR and shown in Figure 2 (a). Also, the accuracy is computed by varying the epochs and illustrated in Figure 2 (b). From Figure 2 (a) and (b), it can be observed that the proposed features perform best with the highest accuracy as compared with other features at all the SNR and epoch values. Also, the estimated elevation angles (predicted outputs) are plotted against the true elevation angles, as shown in Figure 3 (a). The green line shows the true elevation values, and the blue dots represent the predicted CNN outputs for each angle. All the plots in are obtained for the simulated datasets, and the results motivate enough to use the LPRC cues for binaural source localization in the median plane.

4.3 Experimental Validation

The performance of the proposed method is validated through real-time experimental data. The binaural signals are recorded using the low-cost mannequin [17] for three different speech signals taken from the LIBRISPEECH dataset [16] and played using the loudspeaker. Six different positions of elevation angles ($-30^\circ$, $0^\circ$, $30^\circ$, $120^\circ$, $180^\circ$, $210^\circ$) for the source are considered, three in the front and three at the back side in the median plane, as illustrated in Figure 4. The spectral notches and the corresponding frequencies are obtained. The elevation angle is obtained using the proposed method. The accuracy and the mean square error (RMSE) are obtained by comparing with the actual elevation and averaged for the three speech signals. The averaged accuracy and RMSE value are illustrated in Table 1. Further, the performance is obtained using different cues such as ITD, ILD, CCF, GFE, MFCC, MFCC/CCF, and LPCC. The corresponding accuracy and RMSE are shown in Table 1. Table 1 shows that the proposed methods perform better than other methods for binaural localization in the median plane. Subsequently, a subjective evaluation is carried out considering 10 subjects. The subjects must predict the elevation angle after listening to the binaural recordings. The angles predicted by the subjects, along with the actual and estimated angles by the proposed method, are illustrated in Figure 3(b). From Figure
Table 1. Accuracy and RMSE values for various test cases performed on the dataset in the lab environment

<table>
<thead>
<tr>
<th>Features</th>
<th>Case I (−30°)</th>
<th>Case II (0°)</th>
<th>Case III (30°)</th>
<th>Case IV (120°)</th>
<th>Case V (180°)</th>
<th>Case VI (210°)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>RMSE</td>
<td>ACC</td>
<td>RMSE</td>
<td>ACC</td>
<td>RMSE</td>
</tr>
<tr>
<td>ITD</td>
<td>05</td>
<td>45</td>
<td>08</td>
<td>41</td>
<td>02</td>
<td>46</td>
</tr>
<tr>
<td>ILD</td>
<td>06</td>
<td>41</td>
<td>09</td>
<td>37</td>
<td>04</td>
<td>41</td>
</tr>
<tr>
<td>CCF</td>
<td>10</td>
<td>37</td>
<td>11</td>
<td>33</td>
<td>09</td>
<td>34</td>
</tr>
<tr>
<td>GFE</td>
<td>08</td>
<td>35</td>
<td>09</td>
<td>36</td>
<td>04</td>
<td>37</td>
</tr>
<tr>
<td>MFCC</td>
<td>57</td>
<td>28</td>
<td>58</td>
<td>25</td>
<td>53</td>
<td>28</td>
</tr>
<tr>
<td>MFCC/CCF</td>
<td>63</td>
<td>15</td>
<td>65</td>
<td>14</td>
<td>58</td>
<td>21</td>
</tr>
<tr>
<td>LPCC</td>
<td>71</td>
<td>10</td>
<td>72</td>
<td>10</td>
<td>69</td>
<td>14</td>
</tr>
<tr>
<td>LPRC</td>
<td>75</td>
<td>08</td>
<td>79</td>
<td>07</td>
<td>77</td>
<td>10</td>
</tr>
</tbody>
</table>

3(b), it is observed that the LPRC features are more convenient for binaural localization in the median plane.

5 CONCLUSION

The proposed work presents a learning-based binaural source localization technique in the median plane and its application to robot audition. Binaural recordings capture the audio signal and acoustic transfer function from the source to the ears, known as the head-related transfer function (HRTF), which parameterizes spatial cues such as ITD and ILD. These cues are prominent for source localization in the horizontal plane. Since ITD and ILD are nearly equal to zero in the median plane (the ear canal of both the ears is colocated), the localization is complex. Therefore, monaural spectral cues such as spectral notches are investigated for median plane source localization. The spectral notch represents the delay between the direct and the reflected wave. As it varies with the elevation angle, a learning-based model is developed to map the spectral notch with the elevation angle. The spectral notch features are extracted from the binaural recording using linear prediction cepstral coefficients (LPCC) and linear prediction residual coefficients (LPRC). Simulations and experiments are carried out using high-spatial-resolution HRTF measurements from the CIPIC dataset to evaluate the performance. The results show a significant improvement in localization accuracy compared with existing methods.

ACKNOWLEDGEMENT

This work is supported by the SERB-DST under project no. SERB/EE/2021410.

REFERENCES


ABSTRACT

We conducted a study to examine the perceptual benefits of spatial cue preservation in binaural signals produced using binaural beamforming with microphone array signals. The focus of the study was to understand perceptual benefits in terms of speech intelligibility. In addition to the gain in Signal-to-Noise Ratio (SNR) that traditional beamforming techniques offer, binaural beamforming techniques offer the added benefit of spatial cue preservation and the perceptual benefits that come with it. We evaluated speech intelligibility under unprocessed, monaural beamforming (beamforming with no spatial cue preservation) and binaural beamforming stimuli conditions in a group of participants with normal hearing using the Matrix sentence test, using is a speech-on-speech masking task. Our results indicated consistently higher speech intelligibility with spatial cue preservation achieved by binaural beamforming. We observed an approximately 3 dB decrease in Speech Reception Threshold (SRT) with preservation of spatial cues, consistent with previous findings.

Keywords: Speech Intelligibility, Binaural Beamforming

1. INTRODUCTION

A key challenge in communicating in difficult acoustic conditions is the task of segregating the source of interest in the presence of various competing sources. A listener's ability to perform this task depends on several factors such as listener's hearing ability, noise levels, type, and spatial location of the competing sources relative to the source of interest etc.

Recently, various wearable products have been introduced that help mitigate these issues of listening and effectively communicating in difficult environments. Typically, devices designed to help with communicating in difficult conditions come equipped with multi-sensor and in some cases, multi-modal processing that might include visual inputs as well as audio inputs from a number of microphones. The signals from an array of microphones can be used to leverage spatial information represented in the differences between the various signals in order to spatially filter out the signals emanating from the competing sources that are not desired. This is usually done with a beamformer. This can lead to very significant improvement in Signal-to-Noise Ratio (SNR) and in cases of spatially separated desired and undesired sources, it can lead to significant gains in terms of Speech Reception Threshold (SRT).

However, a major shortcoming of traditional beamforming techniques is that they process inputs from multiple microphones to produce a single channel output that does not preserve any spatial information. This is particularly important in realistic and complex acoustic scenarios that are characterized by multiple
directional and diffuse sound sources that closely affect the overall perception of the desired parts of the scene. The absence of preservation of spatial characteristics of the scene leads to a mismatch between the visual and auditory stimuli associated with the scene. This also makes correct localization impossible since the processed signal no longer has any spatial cues associated with it. Moreover, an important phenomenon that helps with understanding speech in the presence of multiple directional sources is the ability to segregate sound sources based on their spatial location and to enable paying attention to the source of interest (Spatial Release from Masking). The absence of spatial cues hampers source segregation leading to poorer intelligibility compared to when spatial cues are correctly preserved.

Binaural beamforming techniques offer the advantages of spatial filtering while also offering the ability to preserve the spatial cues of a given acoustic scene. This spatial cue preservation can be limited to those corresponding to the desired source only or it may also refer to spatial cue preservation of all the directional sources in each scene, including residual noise sources.

In this study, we compare speech intelligibility of the output of binaural beamforming techniques with those of monaural beamforming techniques using a proprietary wearable microphone array. Our main aim is to understand the benefits of the preservation of spatial cues in terms of speech intelligibility on complex acoustic conditions.

2. METHODS

2.1 Beamforming

Three types of speech signals were used in this study namely unprocessed, monaurally processed and binaurally processed speech signals. A Binaural Linearly Constrained Minimum Variance beamformer (BLCMV) was used for the binaural processing condition. A Linearly Constrained Minimum Variance beamformer (LCMV) beamformer was used for the monaural processing condition. The BLCMV beamformer has the unique ability of suppressing noise while also preserving the spatial cues of each of the desired and undesired sources in an acoustic scene. The choice to use the LCMV beamformer for the monaural case was made to keep the parameters and gains of the system comparable across the monaural and binaural beamformer.

An LCMV beamformer imposes multiple linear constraints on its cost function to produce the spatially filtered output signal. Typically, the Acoustic Transfer Function (ATF) and an estimate of the noise covariance matrix are needed to construct this cost function. A binaural beamformer in general consists of two beamformers that work in conjunction to preserve the spatial cues of a given acoustic scene in addition to suppressing the noise. In the case of a BLCMV beamformer, constraints are imposed to preserve the Relative Transfer Function (RTF) of all directional sources in a scene. The RTF is basically a normalized ATF that varies just in the choice of the reference microphone. Preserving the RTF for each of the two beamformers thus leads to a preservation of spatial cues.

For a wearable array with M microphones, with inputs $x_1, x_2 \ldots x_M$, two independent sets of beamformer coefficients $w_l$ and $w_R$ are applied to generate the binaural signal $y$ consisting of two channels $y_l$ and $y_R$. One reference microphone is selected for each of the two beamformers from the available microphones. Specifically in the case of the BLCMV beamformer, two sets of filters are designed that recreate a processed version of the desired signal given the reference microphones selected for each of the two channels while reducing the undesired signals and minimizing the output power. The selection of the reference microphone is very important part of the system. For the study discussed here, binaural mics at each of the two ears were used as reference mics. RTFs were estimated using subspace techniques from the original desired and interfering signals. The proprietary wearable microphone array used in this study had a total of 8 microphones.

2.2 Speech Intelligibility testing

The Matrix sentence test was used to evaluate speech intelligibility. The speech stimuli consisted of speech from a target speaker masked by speech from four spatially separated interfering talkers. Target and interfering sources spoke five-word sentences. Each sentence was of the form Name-Verb-Number-Adjective-Noun e.g. "Sue bought three blue bags". Target and interfering speech were presented simultaneously, and participants were asked to identify the words spoken by the target talker from a matrix of words on a Graphical User Interface (GUI). Speech Reception Thresholds (SRTs) corresponding to 50% correct were computed at each Target-to-Masker Ratio (TMR). The TMRs here were computed per interfering source and were set at -20dB, -16dB, -12dB, -8dB and -4dB. Moreover,
Response Time (RT) from the end of the stimuli to when the participant hit the "Submit" button to proceed to the next stimuli was also recorded.

The test application was run on participant’s windows computers, and they listened to the stimuli over headphones. Participants set their Most Comfortable Level (MCL) at the beginning of the test and all stimuli were played at that level. They went through a short training session in the beginning of the test to familiarize themselves with the test and response formats.

2.3 Stimuli

The Boston University Corpus was used for this study. The corpus consists of various words recorded separately. Speech from twelve female talkers was used for this study. Individual words were concatenated to produce sentences of the form Name-Verb-Number-Adjective-Noun. Reverberation corresponding to RT₆₀=150ms was simulated using the Image-Source method. The target talker speech was always simulated from 0° in azimuth. There were four other talkers who acted as interfering sources. They were presented from -60°, -30°, +30° and +60° in azimuth from the participant. All target stimuli began with the word 'Sue'.

2.4 Participants

A total of 28 participants aged between 18-65 participated in the study. All participants self-reported having normal hearing.

3. Results

3.1 Speech Intelligibility

Because the response data are binomial by nature, we used a beta-binomial model to draw inference regarding the effects of beamformer processing and TMR. The beta-binomial distribution is the binomial distribution in which the probability of success at each of the N trials is not fixed but randomly drawn from a beta distribution. A beta-binomial model is a generalization of the binomial model. Rather than assuming an equal probability of correct word identification in each trial, a beta-binomial model estimates the probability from the data.

We analyzed the data in two ways. First, using input TMR to assess differential benefit from each processing scheme. Second, using output TMR to facilitate benefit of spatial cues independent of TMR differences.

![Figure 1 Probability of correct word identification. Points represent median probability of correct word recognition; bars represent the 89% highest density credible interval surrounding the median values; and shaded regions indicate the posterior distribution estimates](image)

Each panel in Figure 1 corresponds to one of the input TMRs tested. This figure represents average
performance and excludes varying effects of participant and utterance. Both the monaural and binaural beamformer typically had a gain of roughly 10-12dB. As shown in the figure, the binaural beamformer output consistently leads to higher accuracy in terms of word identification compared to the output of the monaural beamformer. The differences are most significant at -12dB and -16dB when the scores are somewhat moderate. At the highest and lowest TMRs these differences diminish.

Figure 2 depicts the probability of correct word identification (score) as a function of output TMR. This figure can be interpreted as psychometric functions of processing condition by output TMR. The dashed line corresponds to a score of 50% since SRTs corresponding to 50% accuracy are being compared. As with Figure 1, we see that the SRT with the binaural beamformer processing is lower by around 3 dB compared to that of the monaural beamformer.

![Figure 2 Probability of correct word identification (score) as a function of output TMR (dB). The lines depict a smoothed fit across median estimates and the shaded regions represent the 89% highest density credible intervals](image)

While the above figures depict average results excluding varying effects of participants and utterances, our data demonstrate variable performance by participants in the three processing conditions and variable correct word identification by utterance.

The range spanned by the scores averaged over TMR for the unprocessed condition was 38.5% when participant and utterance variance was excluded and 85.2% when included. Similarly, for the monaural beamforming condition, the range spanned went from 97.6% when participant and utterance variance were excluded and 99.4% when included. For the binaural beamforming condition, this range was 92.7% with variance excluded and 98.2% when included.

### 3.2 Response Time

We constructed a hierarchical model under a Bayesian framework to estimate the reaction time of responding to the speech recognition task. The model included beamformer processing (Unprocessed, Monaural, and Binaural) and TMR as population-level effects and group-level (varying) effects for Subject and Utterance, as well as the correlations between Beamformer Processing conditions.
Figure 3 Reaction time as a function of TMR (dB). Points depict median estimated reaction time, error bars indicate the 89% highest density credible intervals surrounding the median estimates, and shapes indicate processing conditions.

To analyze the response time data, some outliers were removed from the data and the response times were mean normalized on a per-participant basis. However, as seen in Figure 3, the response time is very noisy, and it may be difficult to draw any generalizable conclusions from them. The aim of this analysis was to use response time as a proxy for the effort needed to correctly identify speech.

4. Conclusion
We showed that there is an approximately 3 dB benefit in using binaural processing and in particular binaural beamforming to speech intelligibility while comparing it to monaural methods. We also looked at an analysis of the reaction time as a function of TMR, but the results were inconclusive.

ACKNOWLEDGEMENTS
We would like to thank Thomas Lunner, Owen Brimijoin, Jacob Donley, James Morrow, Honson Ling, James Manley-Buser and Shawn Featherly for their support in designing and conducting this study.

REFERENCES
Binaural Reproduction of HOA Signal using Temporal Convolutional Networks

Gyanajyoti Routray, Priyadarshini Dwivedi, *Raj Prakash Gohil, Rajesh M. Hegde

Indian Institute of Technology Kanpur, India, *Fraunhofer IDMT, Germany
E-mail: {groutray, priyadw, rhegde}@iitk.ac.in, *raj.prakash.gohil@idmt.fraunhofer.de

ABSTRACT

In this work, a temporal convolutional network (TCN) based binaural reproduction of higher-order ambisonics (HOA) signals in the spherical harmonics (SH) domain is proposed. The binaural rendering is characterized by the head-related transfer function (HRTF). Since the HRTFs cannot be measured for all the directions, it limits error-free binaural reproduction. The proposed work presents a data-driven approach to learning binaural cues from the anthropometric parameter and source directions. The task is to estimate masking functions that transform the higher-order ambisonics (HOA) signals into binaural signals. The learning framework takes the HOA signals as the input along with the anthropometric parameters to generate the binaural signals. In the proposed method, the TCN implicitly learns the HRTFs parameter and produces the binaural signal. The performance of the method is evaluated based on the reproduction accuracy and mean square error (MSE). Further real-time experiments are carried out using the CIPIC HRTF dataset and the binaural recording using the autogenously developed bionic ears to validate the performance of the proposed method.

Keywords: Direction of arrival, spherical harmonic domain, learning approach, convolutional neural network

1 INTRODUCTION

Binaural reproduction of spatial audio is a fast-growing technology in many audio-visual applications. Binaural reproduction has many applications, including augmented and virtual reality (AR/VR), personalized audio devices and systems, and teleconferencing. Preserving the complete spatial knowledge of the scenario is one of the significant challenges of this technology. Ambisonics is a promising technology for spatial sound recording and rendering [1]. The spatial resolution improves as the order of ambisonics increases and yields higher order ambisonics (HOA). Further, it is convenient to obtain the binaural signals from the HOA recordings to reproduce the HOA signal using headphones. In this context, many researchers have developed enormous methods that have been explored to reproduce binaural signals from the encoded ambisonics signals. Mainly these methods are classified broadly into two categories. In the first category, which is the non-parametric or direct method, head-related transfer functions (HRTFs) cues are utilized to obtain binaural signals from the locations of the virtual loudspeaker [2, 3, 4, 5, 6, 7, 8]. These methods apply to the B-format signals, the order-1 ambisonics signals. The encoded loudspeaker signals are decoded to obtain the virtual loudspeaker locations. These methods’ performance varies with the input signal’s spatial resolution. Usually, the spatial quality gets degraded for the B-format or the lower order ambisonics because they suffer from lower spatial resolution. Further, this degradation produces localization ambiguity, colorization effects, and deteriorated envelopment scene in the reverberant environments. The second category of the method is known as the parametric method. The operations in this method are performed in either time or frequency domain. The microphone array recordings extract the source signals’ direction of arrival (DOA). These DOAs are then integrated with the HRTFs and the decoding matrix to obtain the binaural signals [9, 10, 11, 12, 13]. The DOAs are obtained from B-format recordings by applying the QR decomposition after separating the real and imaginary components to get the
In this proposed work, a learning-based method using a temporal convolutional network (TCN) is developed to reproduce binaural signals from higher order ambisonics (HOA) signals. The TCN model learns the head-related transfer (HRTF) parameters from the anthropometric features and directional information embedded in the HOA signals. The advantage of this method is that it doesn’t estimate the directions and HRTFs. It directly produces the binaural signal by integrating the anthropometric features with the HOA signals. In this work, the proposed system takes the ambisonics signal of order 3 as input and reproduces corresponding binaural signals at the output. The input to the TCN network is the HOA signal derived from the sound file and HRTFs obtained by varying the source locations and the anthropometric parameters. The variation in the anthropometric parameters and the source location provides a wide range of the dataset that helps to train the learning model. This gives the model the advantage of learning and estimating binaural signals for a wide variety of datasets. The performance is evaluated and compared with the available state-of-art methods.

The rest of the paper is organized as follows. Section 2 describes the signal model for the HOA and binaural signal. Then the TCN learning framework is detailed. The performances evaluation and comparison to the state-of-art methods is given in Section 3, and Section 4 draws the conclusion and future scope.

2 SYSTEM MODEL AND PROBLEM FORMULATION

In this section, the signal model for HOA recording is discussed first. Subsequently, the binaural representation for the source is described, followed by the TCN model. The TCN model is trained to produce the binaural signal corresponding to the HOA signal and anthropometric parameters as input.

2.1 HOA Encoding

Consider the sound source acquisition scene where the source is located at \((\theta_1, \phi_1), (\theta_2, \phi_2), \ldots, (\theta_Q, \phi_Q)\) for \(Q\) distinct spatial audio sources \(s = [s_1, s_2, \ldots, s_Q]^T\). \((\theta, \phi)\) represents the elevation and azimuth angles, respectively. The ambisonics encoded signal is given as [16]

\[
B = Ys
\]

(1)

where \(Y \in \mathbb{C}^{(N+1)^2 \times Q}\) represents the spherical harmonics (SH) basis matrix. Each column of \(Y\) represents the spherical harmonics basis function corresponding to the source location and is expressed as

\[
Y = [Y(\theta_1, \phi_1), Y(\theta_2, \phi_2), \ldots, Y(\theta_Q, \phi_Q)]
\]

(2)

where \(Y(\theta, \phi) = [Y_{00}(\theta, \phi), Y_{1-1}(\theta, \phi), \ldots, Y_{NN}(\theta, \phi)]^T\) represents the spherical harmonics basis function in the direction \((\theta, \phi)\) and the SH basis \(Y_{nm}(\theta, \phi)\) is expressed as [17]

\[
Y_{nm}(\theta, \phi) = \sqrt{\frac{(2n+1)(n-|m|)!}{4\pi(n+|m|)!}} P_{nm}(\cos \theta) e^{im\phi}
\]

(3)

where \(n \in \mathbb{N}\) and \(m = -n, \ldots, n\) denotes the order and degree of the SH and \(P_{nm}(\cdot)\) expresses the associated Legendre function. The order of the spherical harmonics and the spatial resolution has a direct relationship. As the SH order is increased, the spatial resolution improves. Since the objective is to reproduce the binaural signal with high spatial resolution, the proposed work considers the higher order ambisonics encoded signals.
2.2 Binaural Reproduction

The proposed work aims to reproduce binaural signals for any person in any direction. The binaural signal reproduction prominently depends on head-related impulse responses (HRIR). The convolution of the HRIRs and the monaural audio signal represents the binaural signal. In the discrete time domain, the binaural panning signals to the headphones are given as

\[ x_{l|r}[t_d] = s[t_d] * h_{l|r}[t_d] \] (4)

where \( s[t_d] \) is the monaural source signal, \( h_{l|r}[t_d] \) is the HRIR for the left or right ear, and \( x_{l|r}[t_d] \) is the binaural signal for the left or right ears. \( t_d \) represents the discrete-time and * is the convolution operator. The HRIR represents the impulse response between the ear pinna and the source. The corresponding frequency domain representation is the head-related transfer function (HRTF). Since HRIR captures the reflections due to the anthropometric parameters and the source location, the binaural reproduction needs the exact knowledge of HRIR and the source signal. The binaural signal is obtained from the HOA encoded signal in the presented work. The HOA signal has the information of source location embedded in the encoded signal. During the reproduction, the prior information about the source location is unknown to the decoder. Hence a temporal convolution network (TCN) is explored that takes the HOA signal along with the anthropometric parameter to produce the binaural signals. After learning, the TCN model is ready to generate the binaural signal for any input direction and person with different anthropometric parameters.

2.3 Temporal Convolution Network Model

The TCN model is designed to train the dataset for learning-based binaural reproduction. Encoder and decoder modules with kernel sizes 40 and 10 are designed in the model. To avoid the learning of scaling factors by the filter, L2 normalization is applied in the encoder-decoder filter coefficients before obtaining the convolution, which is taken care of by masking the TCN module. The over-completeness of basis signals in the encoder and decoder increases with increasing channels (or the number of basis signals), further enhancing the performance. In the proposed work, a small number of channels is considered a trade-off between the model size and performance. Three sequential blocks are considered in the TCN module masking. Every block contains a stack of ResBlock with eight layers. The model contains three convolutional layers, the first layer of Conv1D decreases the channel number to 128, which is constant throughout the subsequent blocks. Another hyper dilated convolution layer has kernel size set to 4, with double dilation size in each layer. The final Conv1D layer matches the number of channels to twice the encoder output. An adapter block with Conv1D layer is introduced with ReLU activation function to add non-linearity. A multi-scale short-time Fourier transform (STFT) [18] is minimized for training the model. This loss function mostly replaces the point-wise losses estimated on the raw waveforms. The overall loss is computed as the total of all the spectral loss for right and left channels. A single STFT spectrogram loss is defined with the sizes 256, 512, and 1024. The adjacent frames in the STFT have an overlap of 75%. The model is trained with adam optimizer and the batch size is 196. The learning rate
is set to be 0.0001. The frame size of 8K audio samples at 48 kHz is considered for training at 50% overlap. The processes involved in reproducing binaural signals from the HOA signals are represented in Figure 1.

3 PERFORMANCE EVALUATION
The performance of the proposed binaural rendering of SMA recordings is evaluated in this section. First, the dataset for training is generated and trained using the TCN model. The performance of the proposed method is compared using the root mean square error (RMSE) against the epoch. Further, objective and subjective evaluations are carried out on the binaural signals.

3.1 Dataset Generation
There is no available dataset directly for the higher order ambisonics signals that can be used for the training of the learning model. Therefore, synthetic data is simulated for this purpose. The HOA signals are obtained using the spherical microphone array room impulse response (SMIR) generator [19]. For generating the signals, the audio files are taken from the LIBRISPEECH dataset [20]. The data is generated for various room dimensions with variation in room reverberations \((RT_{60})\) and signal to noise (SNR). The dimension for the room is taken as \(5 \times 6 \times 7\) m with random variation in the range \(\pm 2\) m. \(RT_{60}\) and SNR are considered in the range \([0.2 - 0.9]\) s and \([5 - 15]\) dB. The center of the SMA coincides with the center of the room and the source is placed at a distance of 1 m from the center of the SMA. The HRTFs are taken for source angles and the anthropometric parameters according to the CIPIC database [21]. The dataset contains 43 subjects with varying anthropometric parameters, azimuth angles in the range \([-80, 80]\), \([-65, -55, -45\) \((\text{in 5}\) steps\), \(55, 65, 80\), and elevation angles varying from \([-45, 230.625]\) in steps of \(5.625\). This provides a broad range of simulated training datasets, which is sufficient to train the learning model and generate binaural signals for any direction and anthropometric parameters. The simulated dataset generated provides a very much realistic binaural signal.

3.2 Numerical Simulation
The accuracy of the proposed TCN model method is computed for different subjects and source locations. Figure 2 (a) illustrates the accuracy of each subject at various angles. Each blue dot represents the accuracy for each angle. Figure 2 (b) represents the root mean square error (RMSE) against epochs for the test dataset when it is trained for different epochs. Also the accuracy for each azimuth and elevation angle in the range \(\phi \in [-80, 80]\) and \(\theta \in [0, 180]\), is represented in Figure 2 (c). The plots are motivating enough to use the proposed TCN learning-based model for binaural reproduction.

3.3 Subjective Evaluation
This section conducts the perceptual evaluation of the binaural signals obtained from the SMA recordings. Eigenmike is used for recording purposes. The HOA signals from the obtained recording, along with the anthropometric parameters, are given as the feature to train TCN based learning model. The binaural signal is
Figure 3. Illustration of the subjective analysis of the proposed method, (a) shows the experimental setup in the lab and (b) represents the box plot showing the scores of various subjects against various binaural reproduction methods.

Table 1. Comparison of objective evaluation scores for different methods

<table>
<thead>
<tr>
<th>Approach</th>
<th>PEAQ</th>
<th>DI</th>
<th>PSM</th>
<th>PSMt</th>
</tr>
</thead>
<tbody>
<tr>
<td>VL</td>
<td>-3.9945</td>
<td>0.0004</td>
<td>0.5532</td>
<td>0.3834</td>
</tr>
<tr>
<td>magLS</td>
<td>-3.9124</td>
<td>0.2057</td>
<td>0.6578</td>
<td>0.4028</td>
</tr>
<tr>
<td>MMV-P</td>
<td>-3.8810</td>
<td>0.2764</td>
<td>0.7292</td>
<td>0.4214</td>
</tr>
<tr>
<td>Bi-TCN</td>
<td>-2.3250</td>
<td>0.5764</td>
<td>0.9292</td>
<td>0.6414</td>
</tr>
</tbody>
</table>

obtained from the binaural recording as shown in Figure 3(a). The binaural recordings using the mannequin is reference binaural signal for the subjective and objective evaluation. For comparison purpose the binaural signals are also obtained using the virtual loudspeaker (VL) method [8], magnitude least squares (magLS) [22] and Multiple Measurement Vector Projections (MMV-P) [23] methods. A perceptual test with fifteen listeners is carried out for subjective evaluation. The listeners are requested to provide ratings to the perceived test sounds in terms of the overall quality they perceived progressive scale of 0 to 100. The opinion scores are also obtained for the VL, mag-LS, MMV-P, and proposed TCN-based binaural reproduction methods. The scores are averaged for the four test cases and are shown in Figure 3 (b). From Figure 3 (b), it can be observed that the proposed binaural reproduction method performs better than the VL, mag-LS, and MMV-P based binaural reproduction methods.

3.4 Objective Evaluation

The objective evaluation is performed by employing [24, 25] on the reproduced binaural signals. Various methods have been studied for binaural sound reproduction, and their perceptual qualities are measured with respect to the perceptual evaluation of audio quality (PEAQ) and perceptual similarity measure (PSM), the fifth percentile of the sequence of instantaneous audio quality (PSMt), and distortion index (DI). Table 1 represents the performance of various approaches using these methods. PEAQ is an objective measure on a scale from 0 (for excellent) to -5 (for annoying). DI compares the observed binaural sound with the reference sound to find the distortion between the two. The absolute value closer to unity represents reduced distortion for the reproduced audio. The results in the table show that the proposed TCN learning-based method for binaural reproduction performs much close to the reference audio. For PSM and PSMt, the perfect matching between the observed audio and reference audio is represented by unity. Therefore it is observed that the performance of the proposed approach is excellent using these approaches. The remarkable enhancement in the audio signal quality is
because of applying an exhaustive learning algorithm for various subjects and directions.

4 CONCLUSION
This work proposes a learning framework based on the temporal convolutional network (TCN) method for the binaural reproduction of HOA signals. HOA encoded signal obtained from the direct recording using a spherical harmonics basis. The spatial resolution improved by training the TCN model for various directions and anthropometric parameters. The binaural sound signals are obtained from the plane wave decomposition. The obtained results are compared with various baseline approaches. This work can be extended for various subjects by finding the anthropometric parameters. Further, a more robust binaural reproduction system can be developed by considering more subjects and continuous HRIR. This work can be useful in tracking sound sources and head movements in AR-VR, virtual concerts, and teleconferencing applications.

ACKNOWLEDGEMENT
This work is supported by the SERB-DST under project no. SERB/EE/2021410.

REFERENCES


A spatial enhancement approach for binaural rendering of head-worn microphone arrays

Janani FERNANDEZ1; Leo MCCORMACK1; Petteri HYVÄRINEN1; Archontis POLITIS2; Ville PULKKI1

1Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland
2Department of Information Technology and Communication Sciences, Tampere University, Finland

ABSTRACT

This paper builds upon a recently proposed spatial enhancement approach, which has demonstrated improvements in the perceived spatial accuracy of binaurally rendered signals using head-worn microphone arrays. The foundation of the approach is a parametric sound-field model, which assumes the existence of a single source and an isotropic diffuse component for each time-frequency index. The enhancement approach involves the post-processing of an initial estimate of the binaural signals, in order to obtain a refined estimate of binaural signals which more closely represent the inter-aural cues corresponding to the sound-field model. In this contribution, the enhancement approach has been implemented as an open-source framework, written in both the MATLAB and C programming languages, and as a real-time audio plug-in. The framework was also extended to offer direction-dependent gain control of sound sources relative to the listener, and a frequency-dependent control of the direct-to-diffuse balance, which are modifications that may find application within future augmented reality headsets and assistive hearing devices.

Keywords: microphone array processing, spatial audio, augmented reality, binaural hearing aids

1. INTRODUCTION

Binaural rendering of head-worn microphone arrays is receiving increased attention, largely due to the emergence of commercially-available augmented reality headsets and binaural hearing aid devices. In such applications, it is desirable for the captured surrounding sound scene to be reproduced over headphones with high perceived spatial accuracy and transparency, and for the reproduction method to facilitate some degree of hearing augmentation. There have been a number of signal-independent methods proposed recently [1, 2], which are specifically intended for processing head-worn microphone array input, and operate based on a linear mapping of the array signals to the binaural channels. However, while such methods have low computational requirements and achieve high signal fidelity, their maximum attainable spatial resolution is inherently limited by the number of microphones and their placement on the array geometry.

There are binaural rendering approaches which have been proposed by the hearing-aid research community [3, 4, 5], and proposals involving spherical microphone arrays (SMAs) within the spatial audio research community [6, 7], which also extract additional spatial information through observations of the inter-channel relationships between the array signals themselves. Typically, such methods (often referred to as parametric methods within the spatial audio field) impose assumptions regarding the composition of the input sound scene, and estimate meaningful spatial parameters over time and frequency. These elements are then used to dictate the rendering; typically, by informing the look-direction of spatial filters (beamformers) with their signals convolved with the respective head-related transfer functions (HRTFs). Provided that the spatial analysis techniques are robust, such signal-dependent methods have been shown to outperform their signal-independent counterparts in formal perceptual studies [7]. However, due to the nature of time-frequency domain processing, audible artefacts can occur.

1 janani.fernandez@aalto.fi
Recently, a general spatial enhancement approach, based on spatial covariance matching, was proposed in [8]. In principle, the enhancement approach may be applied to the output of any existing binaural rendering approach; but the focus of the study however, revolved around head-worn microphone arrays for augmented reality and hearing aid applications.

Building on the recent work of [8], this paper details a MATLAB toolbox and a real-time VST audio plugin implementation of the algorithms detailed therein.

2. BINAURAL RENDERING FRAMEWORK

This section provides an overview of the binaural rendering framework.

2.1 Sound-field model

The employed signal model assumes that the \( M \) input microphone array signals \( \mathbf{x}(t,f) \in \mathbb{C}^{M \times 1} \) describe a single source signal \( s(t,f) \), a diffuse field \( d(t,f) \in \mathbb{C}^{M \times 1} \), or a combination of the two at each time-frequency index

\[
\mathbf{x}(t,f) = a(\gamma,f)s(t,f) + d(t,f),
\]

where \( a(\gamma,f) \in \mathbb{C}^{M \times 1} \) is the array transfer function for incident direction \( \gamma \). Note that it is henceforth assumed that these array transfer functions, \( \mathbf{A} \in \mathbb{C}^{M \times V} \), have been either measured or simulated for a dense grid of \( V \) incident directions \( \Gamma = [\gamma_1, ..., \gamma_V] \).

The array spatial covariance matrix (SCM) is given as

\[
\mathbf{C}_x(f) = \mathbb{E}[\mathbf{x}(t,f)\mathbf{x}^H(t,f)] = a(\gamma,f)a^H(\gamma,f)\mathbb{E}[|s(t,f)|^2] + \mathbb{E}[d(t,f)d^H(t,f)],
\]

where \( \mathbb{E}[\cdot] \) denotes the expectation operator.

2.2 Spatial parameter estimation

Many source number detectors and diffuseness estimation approaches operate based upon the analysis of the eigenvalues of the time-averaged array SCMs. Typically, such approaches rely on the SCMs becoming diagonal (with all eigenvalues being equal) when there are no sources active or when high levels of diffuse sound is present. However, especially at lower frequencies, the array SCMs can deviate from this diagonal structure when capturing diffuse-fields. Therefore, a spatial whitening operation is applied as

\[
\mathbf{C}_x^{(w)}(f) = \mathbf{T}(f)\mathbf{C}_x(f)\mathbf{T}^H(f),
\]

where \( \mathbf{C}_x^{(w)} \in \mathbb{C}^{M \times M} \) is the spatially whitened array SCM, and \( \mathbf{T} \in \mathbb{C}^{M \times M} \) is the whitening matrix as described in further detail in [9].

The spatially whitened array SCM is then decomposed using an eigenvalue decomposition as

\[
\mathbf{C}_x^{(w)}(f) = \mathbf{V} \Sigma \mathbf{V}^H = \sum_{m=1}^{M} \sigma_m \mathbf{v}_m \mathbf{v}_m^H,
\]

where \( \sigma \) are the eigenvalues sorted in descending order and \( \mathbf{v} \) are their respective eigenvectors.

A diffuseness estimate is then obtained, based on the method described in [10], as

\[
\psi(f) = 1 - \frac{\beta}{\beta_0},
\]

where \( \beta_0 = 2(M-1) \), \( \beta = \frac{1}{(\sigma)} \sum_{m=1}^{M} |\sigma_q - \langle \sigma \rangle| \), and \( \langle \sigma \rangle = \frac{1}{M} \sum_{m=1}^{M} \sigma_m \).

For estimating the direction-of-arrival of the most prominent sound source in the scene, the MUtiple-Signal Classification (MUSIC) approach [11] is employed as

\[
P_{\text{MUSIC}}(\gamma,f) = \frac{1}{||\mathbf{V}_n^H \mathbf{T}(f)a(\gamma,f)||^2} \quad \text{for} \quad \gamma \in \Gamma,
\]

\footnote{The open-source framework and audio plugin may be found here: https://github.com/jananifernandez/HADES}
with the frequency-dependent DoA estimates subsequently extracted from the resulting pseudo-spectrum, \( P_{\text{MUSIC}} \), by identifying the direction at which the function is minimised.

### 2.3 Baseline binaural rendering approach

The spatial enhancement approach described in the following subsection is a post-processing operation, which is applied onto binaural signals which are produced by an existing baseline method. The baseline signals are therefore first obtained as

\[
y_{\text{bl}}(t, f) = Q(t, f)x(t, f),
\]

where \( Q \in \mathbb{C}^{2 \times M} \) is the baseline binaural rendering mixing matrix.

For example, in [8], three different baseline approaches, which are found in hearing aid research literature, were explored. One of the approaches involved the selection of two reference signals, the nearest microphones to each ear canal, and then to simply route them to the respective ear canals. The other two approaches were based on establishing a balance between binaural beamformers [3, 12, 13] and the reference signals, which has previously been formulated in [14, 15] using a user-controllable parameter. In [8], however, the balance was dictated by the time-frequency-dependent diffuseness term.

Note that in the case of simply routing two reference signals to the binaural channels, this baseline mixing matrix is both frequency- and time-independent. Whereas, for other time-invariant methods, e.g. [1, 2], the baseline mixing matrix may be frequency-dependent. While other baseline methods may employ signal-dependent beamformers, e.g. [3], and thus the mixing matrix may be both frequency- and time-dependent.

### 2.4 Spatial covariance matching based enhancement

The spatial enhancement approach is based on adaptively mixing the baseline binaural signals, in order to obtain refined estimates of binaural signals, which should more closely match the employed sound-field model. The enhancement approach is based on first defining target narrow-band binaural SCMs as

\[
C_y(f) = (1 - \psi(f))P_{\text{total}}(f)h(\gamma, f)h(\gamma, f)^H + \psi(f)P_{\text{total}}(f)D_{\text{bin}}(f),
\]

where \( P_{\text{total}} = \text{tr}[C_x] \) is an estimate of the total signal energy, \( h \in \mathbb{C}^{2 \times 1} \) are HRTFs corresponding to the analysed DoA, and \( D_{\text{bin}} = HH^H \in \mathbb{C}^{2 \times 2} \) is a binaural coherence matrix derived using a dense grid of HRTF measurements \( H = [h_1, ..., h_V] \in \mathbb{C}^{2 \times V} \).

The spatial enhancement operation is then applied via the mixing matrix \( M \in \mathbb{C}^{2 \times 2} \) as

\[
y_{\text{enh}}(t, f) = M(f)y_{\text{bl}}(t, f) = M(f)Q(t, f)x(t, f),
\]

which has been derived, through an optimisation process, as the matrix required to best ensure the following holds true:

\[
E[y_{\text{enh}}(t, f)\psi_y^H(t, f)] = M(f)Q(t, f)C_x(f)Q^H(t, f)M^H(f) \approx C_y(f).
\]

Note that the solution to this problem is detailed in [8].

### 3. IMPLEMENTATION

The binaural rendering framework and the spatial enhancement approach described in [8] was implemented in both the MATLAB and C languages. Both implementations are divided into two main stages: analysis and synthesis. In the analysis stage, the DoA and diffuseness parameters are estimated over time and frequency, based on the input microphone array signals. The time-frequency signals are stored in a signal container, and the results of the analysis are placed into a parameter container. These containers may then be optionally modified, before being passed onto the synthesis stage, which renders the binaural signals.

The C code implementation was also integrated into a real-time VST audio plugin; the graphical user interface for which is depicted in Fig. 1. The plugin supports the loading of arbitrary sets of array transfer functions and HRTFs via the SOFA standard. All three baseline binaural rendering methods described
Due to the decoupling of the analysis and synthesis stages of the framework, it is possible to manipulate the signal and/or parameters containers prior to synthesising the scene. This may be carried out to augment the rendering, in order to realise certain spatial audio effects and other sound-field modifications. Note that an overview of such parametric effects may be found in [16] using an alternative rendering framework; but with many of these effects still applicable to the present framework. In the developed VST audio plugin, the frequency-dependent direct-to-diffuse balance control is implemented, as described in [16], this allows diffuse sounds to be attenuated (i.e. de-reverberation), or exaggerated. Furthermore, a direction-dependent gain control was incorporated within the plugin, which enables an addition gain factor to be applied only to the direct stream components.

4. CONCLUSIONS

This paper has presented an open-source MATLAB, C, and audio-plugin implementation of a spatial enhancement approach, and the associated binaural rendering algorithms. The implemented spatial enhancement approach, which is based on signal-dependent spatial covariance matching operations, is a post processing algorithm intended to be applied to the output of an existing binaural rendering method. The implementation of the present binaural rendering framework also decouples the spatial analysis and synthesis stages. This allows spatial audio effects and sound-field modifications to be applied through simple manipulations of the estimated parameters. The developed audio plugin supports the application of direction dependent gain factors, which only affect sound components which are characterised as having a clear directionality, and also allows frequency-dependent balancing modification of the direct-to-diffuse balance.

REFERENCES

4. As‘ ad H, Bouchard M, Kamkar-Parsi H. A robust target linearly constrained minimum variance beamformer with spatial cues preservation for binaural hearing aids. IEEE/ACM Transactions on

Figure 1: The graphical user interface of the developed VST audio-plugin.


Sound-source position tracking from direction-of-arrival measurements: Application to distributed first-order spherical microphone arrays

Christoph Hold(1), Archontis Politis(2), Simo Särkkä(3)

(1)Department of Signal Processing and Acoustics, Aalto University, Finland, Christoph.Hold@aalto.fi
(2)Department of Information Technology and Communication Sciences, Tampere University, Finland
(3)Department of Electrical Engineering and Automation, Aalto University, Finland

ABSTRACT

Rendering 6-degrees-of-freedom (6DoF) spatial audio requires sound-source position tracking. Without further assumptions, directional receivers, such as a spherical microphone array (SMA), can estimate the direction of arrival (DoA), but not reliably estimate sound-source distance. By utilizing multiple, distributed SMAs, further methods are available that directly infer the position in 3-D space. Typically used DoA intersection by triangulation delivers problematically noisy estimates, therefore, statistical filters are better suited. In this study, we compare the performance of different DoA to position tracking strategies. DoA angles suffer from the well-known angle wrapping problem, which is especially problematic in Gaussian filters. However, these filters are attractive due to their low computational complexity. Using circular and spherical statistics, the non-linear extensions of the Kalman filter can be formulated to explicitly treat the discontinuity of DoA angles. Furthermore, we introduce a time adaptive regularization of the filter update by the instantaneous sound-field diffuseness estimate. An experiment with three first-order SMAs in a reverberant room shows an improved distance error compared to the mean DoA intersection baseline. The results highlight the importance of treating the angle wrapping and the stabilization when incorporating the sound-field diffuseness estimate.

Keywords: Spherical Microphone Array, Position Tracking, Parametric Spatial Audio

1 INTRODUCTION

Many applications demand for tracking a target position, while the sensors only give azimuth (or bearing) and elevation measurements. Such measurements are often direction of arrival (DoA) estimates, for example emanating from wave propagation, specifically sound-waves in acoustic signal processing. An array of microphones, in particular a spherical microphone array (SMA), allows to capture directional sound and several methods have been proposed to extract the angle of incidence from such recordings. Without further assumptions, however, SMAs only allow to estimate the DoA and thus not sound-source distance. Therefore, further methods are needed to localize a source position from DoA estimates.

Utilizing multiple microphone arrays simultaneously can improve the tracking performance. Distributed SMAs allow to also estimate sound-source distance, and therefore position, for example by triangulating the respective DoA estimates. However, triangulation is known to be error prone, e.g., because of noisy estimates, or calibration problems with multiple sensors. Particularly in acoustics, room reflections can heavily influence triangulation, since the reflections interfere with the source DoA measurement, hence leading to incorrect estimates. A strategy to mitigate the common issues arising from such geometrical approaches is to instead utilize statistical inference. Statistical filters are a powerful method in source tracking and trajectory smoothing, which can be formulated to infer a sound-source position estimate by processing the DoA measurements of distributed SMAs. The Kalman filter [4] is arguably the most prominent example here, due to its robust design and low complexity implementation. It has been employed in numerous target tracking applications with great success. For non-linear models, as in the present application, several extensions have been proposed. These methods typically rely on
linearization, or on sigma-point sampling, and the differences for the presented application will be investigated in this study.

A particular challenge of DoA angles is their circularity and inherent discontinuity. For example, an azimuth angle of 0 and $2\pi$ correspond to the same direction, and this behavior imposes further problems in formulating a tracking filter on the unit sphere. Several approaches treating angular measurements have been proposed, for example, the wrapped Kalman filter [15], modified coordinate systems [6], a formulation in spherical harmonics [11], or utilizing spherical statistics [1, 16, 5, 2]. The present article will apply and compare some (low-complexity) methodologies to first-order spherical microphone arrays and discuss their implications in the context of parametric spatial audio.

2 METHODS

2.1 Problem Formulation

The problem on hand requires tracking an object in 3-D space, by only observing angular DoA measurements. We will consider a target in 3-D with its position and velocity $x = [x, y, z, \dot{z}, \dot{z}]^T$, observed by receivers $r$ at position $p_r \in \mathbb{R}^3$ in Cartesian coordinates $p = [p_x, p_y, p_z]^T$, each delivering a DoA measurement $y_r$ in azimuth and elevation angles $\Omega = [\phi, \theta]^T \in \mathbb{S}^2$, where we may write the latter as a unit vector, formalized as a vector on the unit sphere manifold $u \in \mathbb{S}^2 = \{u \in \mathbb{R}^3 : ||u|| = 1\}$.

The quantity of interest is the state vector at the current time step $x_k$, which causes the observed measurements $y_k$. Because the true target state is unknown and can only be observed through noisy measurements, the target state is modeled by a probability density function (PDF). In a Bayesian framework, where the target is considered to move as a Markov process, the posterior PDF contains all information given all past and current measurements. This framework allows to formulate an optimal estimator as a recursive filter that consists of a prediction, and an update/correction step [13], thereby determining the most likely $x_k$ by statistical inference.

The state space model at time step $k$ is expressed in form

$$x_k \sim p(x_k | x_{k-1}) ,$$
$$y_k \sim p(y_k | x_k) .$$

We model the state of a target with a prior density

$$p(x) = \mathcal{N}(x; \mu, P) ,$$

where $\mathcal{N}(x; \mu, P)$ expresses a Gaussian PDF with mean $\mu$ and covariance $P$ evaluated at $x$. The dynamics of the system are modeled by a constant velocity model. Because we use discrete time steps $k$ in time intervals $\Delta t$, the model is discretized as

$$x_k = Ax_{k-1} + q_{k-1} ,$$

with

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} ,$$

where $A \in \mathbb{R}^{6 \times 6}$ is the transition matrix and $q$ process noise. The measurements at receiver $r$ are modeled as

$$y'_r = h(x_k; p_r) + r_k ,$$

with the non-linear measurement function $h : \mathbb{R}^n \mapsto \mathbb{S}^2$, where $n_x$ is the dimension of $x$, and the measurement noise $r_k$. The measurement function converts the target state $x$ to the observed DoA angles $y'$ and is hence
dependent on the position of the receiver \( p_r \), which leads to the measurement model of

\[
\begin{bmatrix}
\phi^r_k \\
\theta^r_k
\end{bmatrix} =
\begin{bmatrix}
\arctan \left( \frac{y_k - p_{ry}}{x_k - p_{rx}} \right) \\
\arctan \left( \frac{z_k - p_{rz}}{\sqrt{(x_k - p_{rx})^2 + (y_k - p_{ry})^2}} \right)
\end{bmatrix}.
\]

(5)

The observations will only contain the DoA in terms of azimuth \( \phi \) and elevation \( \theta \), per microphone array \( r \), and \( h \) can be conveniently implemented as \( \text{cart2sph}(x_{xyz} - p_r) \), which converts from Cartesian coordinates to azimuth and elevation angles. All measurements are then stacked into a single vector \( y_k \).

Uncertainty is modeled as the zero-mean Gaussian process noise \( q_{k-1} \sim N(0, Q) \) and Gaussian measurement noise \( r_k \sim N(0, R) \), which is assumed to be independent of the state and measurements. The process noise covariance after discretization is assumed to be related to \( q \) as

\[
Q = q
\begin{bmatrix}
\Delta^3/3 & 0 & 0 & \Delta^2/2 & 0 & 0 \\
0 & \Delta^3/3 & 0 & 0 & \Delta^2/2 & 0 \\
0 & 0 & \Delta^3/3 & 0 & 0 & \Delta^2/2 \\
\Delta^2/2 & 0 & 0 & \Delta t & 0 & 0 \\
0 & \Delta^2/2 & 0 & 0 & \Delta t & 0 \\
0 & 0 & \Delta^2/2 & 0 & 0 & \Delta t
\end{bmatrix}.
\]

(6)

2.2 Intersection

When trying to find a potential source position the geometrical approach may set the baseline approach. In the geometrical approach, rays are casted from the receivers in the direction of their estimated DoAs. In practice, however, these rays might not all intersect in (3-D) space. As in [8], we may define the intersection as the point of minimal distance between rays instead as

\[
p_{\text{isc}} = \left( p_1 + \tau_1 u_1 + p_2 + \tau_2 u_2 \right) / 2,
\]

(7)

with

\[
\tau_1 = \frac{(p_2 - p_1)\top u_1 + (p_1 - p_2)\top u_2 (u_1\top u_2)}{1 - (u_1\top u_2)^2},
\]

(8)

\[
\tau_2 = \frac{(p_1 - p_2)\top u_2 + (p_2 - p_1)\top u_1 (u_1\top u_2)}{1 - (u_1\top u_2)^2},
\]

(9)

between two receivers at \( p_1 \) and \( p_2 \), with their respective unit vector DoA estimates \( u \). In practice, both \( \tau \) are required to be positive values in order to produce an intersection in the same half-plane. The mean of all intersections from SMA receiver pairs is used as the baseline approach in this study.

2.3 Gaussian Filters

We consider a Gaussian filtering distribution

\[
p(x_k | y_{1:k}) \simeq N(x_k | m_k, P_k).
\]

(10)

The filter prediction step can be described in matrix form due to the linear transition model, leading to predicted \( m_k^- \) and \( P_k^- \) by

\[
\begin{align*}
m_k^- &= Am_{k-1} , \\
P_k^- &= AP_{k-1}A\top + Q_{k-1}.
\end{align*}
\]

(11)
The update step involves the non-linear measurement model function $h$, hence requires extension strategies discussed in the following sections.

As highlighted before, angular measurements call for special care when calculating their mean and difference, hence we adapt the classical filter solutions in the following. Angular means and differences occur in multiple filtering equations, for example when calculating the difference between the predicted and the observed state, also referred to as filter innovation. We will expect problems near the wrapping boundaries, since e.g., the angle $\phi = 0$ and $\phi = 2\pi$ represent the same angle, therefore, the mean and angular difference needs to reflect this property. One simple mitigation of the wrapping problem is to calculate the difference between two angles from $\pi$, or similarly, average the components of unit vectors $u$ and transform back to $y$ (see also [1]).

### 2.3.1 Extended Kalman Filter

A natural choice for non-linear Gaussian filtering is the extended Kalman filter (EKF) that is based on a local linearization of the model by Taylor series expansion. For a linear approximation, the first two terms are sufficient, hence differentiation stops after the Jacobian. The matrix form allows for a very efficient implementation, however, the filter requires the analytical derivation of the Jacobian. For the current problem, the Jacobian (where it exists) was implemented with entries

\[
\begin{align*}
\frac{d\phi}{dx} &= \frac{-1}{1 + \left(\frac{y-p_y}{x-p_x}\right)^2} - \frac{y-p_y}{(x-p_x)^2}, \\
\frac{d\phi}{dy} &= \frac{1}{1 + \left(\frac{y-p_y}{x-p_x}\right)^2} - \frac{1}{(x-p_x)^2}, \\
\frac{d\phi}{dz} &= 0,
\end{align*}
\]

\[
\begin{align*}
\frac{d\theta}{dx} &= \frac{1}{1 + \left(\frac{z-p_z}{\sqrt{(x-p_x)^2 + (y-p_y)^2}}\right)^2} - \frac{z-p_z}{(x-p_x)^2 + (y-p_y)^2}^{3/2}, \\
\frac{d\theta}{dy} &= \frac{1}{1 + \left(\frac{z-p_z}{\sqrt{(x-p_x)^2 + (y-p_y)^2}}\right)^2} - \frac{z-p_z}{(x-p_x)^2 + (y-p_y)^2}^{3/2}, \\
\frac{d\theta}{dz} &= \frac{1}{1 + \left(\frac{z-p_z}{\sqrt{(x-p_x)^2 + (y-p_y)^2}}\right)^2} - \frac{1}{(x-p_x)^2 + (y-p_y)^2}^{1/2}. \\
\end{align*}
\]

The filtering equations include the prediction step according to Eq.(11). The update step leads to the posterior with $m_k$ and $P_k$, using the Jacobian $H(\cdot)$, according to [13, Alg.5.4] as

\[
\begin{align*}
S_k &= H(m_k^-)P_k^-H^T(m_k^-) + R_k, \\
K_k &= P_k^-H_k^T(m_k^-)S_k^{-1}, \\
v_k &= y_k - h(m_k^-), \text{ subject to Eq.}(12), \\
m_k &= m_k^- + K_k v_k, \\
P_k &= P_k^- - K_k S_k K_k^T. \\
\end{align*}
\]

Note the angular difference in calculating the innovation $v$, wherefore we call this modification EKF SPH in the following.
2.3.2 Unscented Kalman Filter

A common criticism about the extended Kalman filter is the insufficient approximation of the non-linearity and hence inferior performance in some circumstances [14]. Therefore, the slightly more flexible unscented Kalman filter (UKF) has been proposed [18], where the Gaussian is described by a set of sigma-points \( \mathcal{X} \). These sigma-points can then be propagated through any (non-linear) model function \( g(\mathcal{X}) \) as 

\[
\mathcal{Y} = g(\mathcal{X})
\]

reflecting the angular wrapping, similar to [1], labeled UKF SPH in the following. The sigma-points are found as [13, eq. 5.74]

\[
\mathcal{X}^0 = m, \\
\mathcal{X}^i = m \pm \sqrt{n + \lambda} \mathcal{P}_{ii}^{1/2},
\]

where the associated weights \( W_i \) are given in [13, eq. 5.77] and sum to unity, and with the square-root of the covariance matrix \( \sqrt{\mathcal{P}} \) (e.g., by Cholesky factorization). The parameter \( \lambda \) is chosen as in [13, eq. 5.75]

\[
\lambda = \alpha^2 (n + \kappa) - n,
\]

where \( \alpha = \kappa = 1 \) were set without further optimization. A deterministic sampling scheme on the unit hypersphere has been proposed in [5].

The filter prediction is again given by Eq. 11. The sigma-point sampling is only necessary for the non-linearity of the filter update, propagating the predicted state through the measurement function \( h \). The exact filtering equations are given in [13, Alg. 5.14], however, the angular difference and weighted angular means are here subject to Eq. 12 and Eq. 13, respectively. The posterior of the update is carried out again as

\[
m_k = m_{k-1} + K_k v_k, \\
P_k = P_{k-1} - K_k S_k K_k^T,
\]

A square-root form can be found in [17], which requires fewer computational operations, besides numerical stability benefits.

2.4 Spherical Distribution Filter Variant

The von Mises-Fisher (vMF) distribution properly defines a distribution on the sphere, with mean \( \mu \) and concentration parameter \( \kappa \). The concentration parameter is inversely proportional to the variance. The distribution is derived by conditioning a Gaussian PDF on the hypersphere, and can therefore be seen as an intrinsic approach to the specific characteristics of angular measurements. In contrast to a Gaussian distribution in azimuth and elevation, the vMF shows no angular stretching for increasing elevation angles (see [2]).

Let \( \mathbf{u} \) again be the unit vector on the sphere, i.e., \( n = 3 \), the probability density function of the vMF distribution is given in [7, Eq. 9.3.4]

\[
f_{\text{vMF}}(\mathbf{u}; \mu, \kappa) = C_3 \exp \left( \kappa \mu^T \mathbf{u} \right),
\]

where for the present case the normalising constant simplifies to \( C_3 = \frac{\kappa}{\sin \kappa} \). The distribution is uniform for \( \kappa = 0 \) and unimodal for \( \kappa > 0 \) with mean

\[
\mathbb{E}[\mathbf{u}] = A_n(\kappa) \mu,
\]

with

\[
\mathbb{E}[g(x)] \simeq \mu_{U} = \sum_{i=0}^{2n} W_{i}^{(m)} \mathcal{Y}_{i}^m, \quad \text{subject to Eq. (13)},
\]

\[
\mathbb{C}[g(x)] \simeq \mathbf{S}_{U} = \sum_{i=0}^{2n} W_{i}^{(c)} (\mathcal{Y}_{i} - \mu_{U}) (\mathcal{Y}_{i} - \mu_{U})^T, \quad \text{subject to Eq. (12)},
\]

(16)
and covariance
\[ C[u] = \frac{A_n(\kappa)}{\kappa} I_n + \left[ 1 - A_n(\kappa) - n \frac{A_n(\kappa)}{\kappa} \right] \mu \mu^\top. \] (22)

For the present case of \( n = 3 \) the above simplifies with [7, Eq. 9.3.9]
\[ A_3 = \coth \kappa - \frac{1}{\kappa}. \] (23)

Based on the vMF distribution, a Gaussian filter has been formulated for position tracking with DoA measurements, which lie in a \( S^{n-1} \) manifold [2]. While the prediction step follows the standard Gaussian filter solution, they have presented a solution for the update step using sigma points as in the UKF, but modeling the measurements using vMF distribution, which is abbreviated as UKF vMF in this study. We re-arranged their solution to have access to the Kalman gain matrix \( K \), such that the update is available as in the form of Eq. (19).

It should be noted that the authors mention improved performance for an iterative optimization of \( A \) and \( R \), which was not carried out in favor for simplicity and comparability to the other presented methods.

2.5 Parameter Extraction
Considering a basic sound-field model of a sound-source in free-field conditions, the emitted sound impinges at the microphone as plane-waves. The DoA of a sound-source is in opposite direction of its net acoustic energy flow (i.e. acoustic intensity, or dependent on definition in the same direction). Extracting this intensity vector \( \mathbf{i} \) is particularly convenient for spherical microphone arrays, as the spherical harmonic (SH) expansion of a sound-field up to first order is proportional to the pressure \( p \) (zeroth order), and to the pressure gradient \( \mathbf{v} \) (first order). The (pseudo-) intensity vector is proportional to the measured
\[ \mathbf{i} \propto \Re \{ p^H \mathbf{v} \}. \] (24)

The vector direction directly estimates the DoA \( \Omega_{\text{DoA}} \) of its predominant signal component
\[ \Omega_{\text{DoA}} = \angle \mathbf{i}. \] (25)

The measured azimuth and elevation were extracted per time sample and then averaged per processing block over \( \Delta t \), and all simultaneous measurements stacked into \( \mathbf{y}_k \). It has been shown that this simple technique leads to reliable DoA estimates from first-order SH components [3]. However, reflections can influence the extracted direction, because typically reflections are linearly correlated to the sound-source signal, but from competing DoAs.

The estimates \( p \) and \( \mathbf{v} \) also deliver an estimate of the sound-field sector energy \( E \) and diffuseness parameter \( \psi \)
\[ \psi = 1 - \frac{||\mathbf{i}||}{E} = 1 - \frac{||\mathbf{i}||}{|p|^2 + \mathbf{v}^H \mathbf{v}}. \] (26)

as an indicator of the degree of deviation from a purely propagating soundfield to fully reactive or isotropic sound fields [12]. The diffuseness \( \psi \) is related to the length of the active intensity vector and is defined in \( \psi \in [0, 1] \), which results in \( \psi = 0 \) for a single impinging plane-wave or far-field source, and \( \psi = 1 \) for no observed net flow, occurring for example in a fully diffuse sound-field, such as dense reverberation. It can be interpreted as the directionality of the sound-field intensity flow, which we will explore as a measure of the reliability of the DoA estimate. The intuition here is that in a dry environment, the DoA is dominated by the direct path net intensity flow of the sound-source of interest, whereas in a reverberant environment the superposition with multiple reflections will result in a less reliable estimate. A similar and more elaborate concept is the direct path dominance (DPD) test [10], applied to the pseudo-intensity vector in [9], which separates the covariance matrix into sub-spaces instead and is therefore computationally more demanding.

2.6 Modification of Filter Update
Under the assumption that a highly diffuse sound-field results in less reliable sound-source direction estimates, the tracking algorithm may be further optimized for spatial audio applications. The sound-field diffuseness estimate
measured by each spherical microphone array can function as an indicator for unfavorable DoA estimation conditions. The estimated sound-field diffuseness value seems to be a promising parameter in order to incorporate additional sound-field information into the filtering algorithm. The diffuseness value does furthermore not only indicate an unreliable measurement due to reverberation, it also goes to one if a DoA estimation is not possible due to a lack of input signal. It therefore constitutes a threshold independent measure to detect insufficient input signal, as it may occur in a speaker pause. A moving sound-source is likely to continue moving during a short break (i.e., in between words of a moving speaker), wherefore loosely continuing the dynamics seem sensible.

The filter update is regularized under unreliable conditions, which are indicated by high diffuseness values. This work introduces a simple thresholding approach on the measurement noise $R$, influencing entry $R_r$ of any effected receiver $r$ as

$$R_r^{\text{mod}} = \tau_r R_r, \text{ if } \psi_r > \psi_{TH},$$

and updates $S$ accordingly in the form of

$$S_k = S_k + R_k^{\text{mod}}.$$  

Because the Kalman gain $K$ is inversely proportional to $S$, the update

$$m_k = m_k + K_k v_k,$$

$$P_k = P_k - K_k S_k K_k^T,$$

heavily favors the predicted solution $m_k$ over the innovation $v_k$ in case of high diffuseness estimates. This strategy avoids manipulating the Kalman gain directly, which might result in a inconsistent filtering formulation.

We tested a threshold of $\psi_{TH} = 0.1$ and modulated with the corresponding diffuseness value as $\tau_r = 10\psi_r$. Note that these values are chosen heuristically and further strategies should be investigated.

3 EVALUATION AND DISCUSSION

The evaluation of the presented methodology was carried out on simulated recordings of a moving source in a reverberant room. An image-source reverberation model with a reverb time of $RT_{60} = 0.5s$ calculated the reflection pattern of the room outlined in Fig. 1 and Fig. 2. The impulse response at the microphones was updated with the moving source in 0.2m increments. The sound-source was moving along a trajectory of 16m with a velocity of 1m/s in a setup shown in Fig. 1. The sound-source first moves along a typical movement for speech along the horizontal plane, and then corners sharply into an ascending motion. The virtual sound-source emitted a white noise sequence, band-passed between 100Hz to 10kHz. The virtual microphone arrays delivered a set of first-order spherical harmonic audio signals, which were then split into blocks of 1024 samples, followed by the parameter extraction detailed in Sec. 2.5. Virtual first-order spherical harmonic receivers captured the scenario in two different typical arrangements. First, an arrangement of three microphone arrays that captured the room from various angles was simulated, depicted in Fig. 1. It is noted here, that the method does not require the receivers to enclose the trajectory of the target, which is demonstrated by the target trajectory leaving the visualized triangulation. Second, an arrangement consisting of three microphone arrays spaced along a common axis, similar to a linear array, which is particularly relevant in practical applications. This arrangement shown in Fig. 2 demonstrates the performance with all receivers located at the same height, besides exemplifying the azimuthal angle wrapping challenge.

These scenarios were chosen to demonstrate multiple effects of the filter designs. We expect critical performance differences around the circularity/ angle discontinuity, since the source passes the $\pm \pi$ relative azimuth angle for the first part of the trajectory, i.e., for microphone m1 in scene 1 and for all three microphones in scene 2. The sharp corner should then uncover problems in the adaption of the filters. Furthermore, the increasing elevation in the second part of the trajectory challenges the assumptions, as only the vMF filter models the measurement statistics correctly in this case. The tracker was initialized with $m_0$ offset from the true value by a realization of standard normal noise, in order to investigate the convergence of the algorithms. The measurement noise was set to a diagonal matrix where the entries correspond to an uncertainty of 5°, and the process noise to $P_0 = I$. 

\footnote{implementation using : https://github.com/polarch/shoebox-roomsim}
Figure 1. Simulated Scene 1 in two cross-sections, m marks the virtual spherical microphone arrays, s the simulated source moving on the dashed line. The simulated room geometry is indicated by the solid boundary.

Figure 2. Simulated Scene 2 in two cross-sections, m marks the virtual spherical microphone arrays, s the simulated source moving on the dashed line. The simulated room geometry is indicated by the solid boundary.
Figure 3. Scene 1 filter estimates in comparison to the true trajectory. The room geometry is depicted as the coordinate limits. The virtual microphone array locations are marked as circles.

Figure 4. Position error distance of the evaluated estimators in comparison to the true trajectory for scene 1.
Figure 5. Filter estimates in comparison to the true trajectory. The room geometry is depicted as the coordinate limits. The virtual microphone array locations are marked as circles.

Figure 6. Position error distance of the evaluated estimators in comparison to the true trajectory for scene 2.
Table 1. Position error distance (RMSE) of filter estimate to true trajectory.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>RMSE Scene 1</th>
<th>RMSE Scene 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Intersections</td>
<td>0.9322</td>
<td>0.4066</td>
</tr>
<tr>
<td>UKF naive</td>
<td>5.9020</td>
<td>12.4264</td>
</tr>
<tr>
<td>UKF SPH</td>
<td>0.3749</td>
<td>0.3247</td>
</tr>
<tr>
<td>EKF SPH</td>
<td>0.3694</td>
<td>0.3179</td>
</tr>
<tr>
<td>UKF vMF</td>
<td>0.3716</td>
<td>0.3240</td>
</tr>
</tbody>
</table>

Summarizing the performance, Tab. 1 shows the evaluated root-mean-square error (RMSE) Euclidean distance between the true trajectory and the filter estimates. The filter performance results are similar and consistent between both scenes. All solutions using angular filtering produce nearly identical results for the presented scenes. In comparison to the naive UKF implementation (i.e., without adapting the filtering equations), the spherical variants show a clear improvement. Filters UKF SPH and EKF SPH use the spherical statistics Eq. 12 and 13, whereas UKF vMF utilizes the vMF distribution. Furthermore, the spherical filters improve on the geometric intersection approach baseline, with also much smoother and more stable results.

Figure 3 shows the tracking filter estimation in comparison with the simulated true trajectory for scene 1. It shows that all algorithms, except the naive UKF, perform equally well and deliver accurate tracking of the sound-source. Figure 4 visualizes the distance RMSE over time, indicating that all filters converge quickly from the intentional initialization offset. The results show a clear indication that neglecting the circular nature of spherical angle measurements leads to poor performance, which is consistent with the literature, e.g., [1, 2] The naive UKF without angular measurement adjustments to the filtering algorithm, leads to a significant estimation error and trajectory divergence just at the point where SMA $m_1$ produces measurements around $\pm \pi$ azimuth, which also the information of two additional SMAs can not counterbalance. After the detour, the filter converges again towards the solution of the other estimators. Scene 2 seems to highlight these problems even more significantly, as demonstrated in Fig. 5 and 6. Again, the naive UKF solution produces unacceptable estimation errors, whenever the trajectory passes the $\pm \pi$ azimuth wrapping of each receiver. The estimation error generally increased for increasing elevation. With all receivers on one axis, the relative differences decrease, hence, statistical inference becomes harder. Additionally, the measured diffuseness increased here, which led to higher uncertainty in the DoA measurements according to Fig. 7. Interestingly, UKF vMF could only show a marginal improvement over UKF SPH, which came at a significant increase in algorithmic complexity.

The estimated sound-field diffuseness value seems to be a promising parameter in order to incorporate additional sound-field information into the filtering algorithm. This is particularly interesting, as these parameters are usually extracted in parametric spatial audio at a very low computational cost. The diffuseness value does furthermore not only indicate an unreliable measurement due to reverberation, it also reacts when DoA estimation is not possible due to a lack of input signal. It therefore constitutes a threshold independent measure to detect insufficient input, as apparent from Fig. 7 for the first few time blocks. Future work could extend to multi-source algorithms and exploring dedicated subspace-methods, e.g., for the parameter estimation.

4 CONCLUSIONS

For this study, multiple computationally efficient non-linear Kalman filters were explored for 3-D target tracking from DoA measurements. This concept was then applied to a moving source in a reverberant room, resembling a both practical and challenging scenario. Concepts of addressing the particularities of spherical measurements were demonstrated and evaluated. In order to mitigate problematic behavior in low SNR scenarios, additional filter regularization dependent on the estimated sound-field diffuseness was explored. The concept of augmenting the classical Kalman filter with information available in parametric spatial audio seems a simple and promising
strategy in order to optimize sound-source position tracking.  

REFERENCES


---

2Implementation available online [https://github.com/chris-hld/doa2pos](https://github.com/chris-hld/doa2pos).


ABSTRACT
Headphones have become the most preferred medium for audio consumption in recent years. With increasing immersive content in Augmented, Virtual, Mixed reality formats, and in music such as Dolby Atmos, it is critical that headphones do not color the immersive experience. Headphones are not acoustically transparent and thus it affects both the timbral as well as the spatial quality of the input sound source. Headphone response depends on the headphone-ear coupling and thus displays high spectral variation between individuals. Inaccurate headphone compensation is observed to be worse than no equalization at all, therefore it is critical to use a headphone equalization filter that is personalized. In this paper, we present a technique to predict personalized headphone equalization curves with machine learning algorithms. Using personalized head-related transfer functions and a generic headphone response as input, we demonstrate neural network models that learn thousands of abstract features representing the ear as a combination of resonant, dissonant, reflective, refractive and other abstract structures. The model accurately predicts the personalized headphone-ear coupling for a given headphone from which the personalized headphone equalization can be computed. The accuracy of the model is corroborated with objective analysis.

Keywords: HRTF, Headphone EQ, Machine Learning

1 INTRODUCTION
Over the past few years, spatial audio has become a critical part of delivering a truly immersive sound experience. With the increasing immersive content in Augmented/Virtual/Mixed reality formats, and in music such as Dolby Atmos, it is important to understand the limitations of Spatial audio and technologies to enhance the experience. Headphones have become the most popular medium for audio consumption, and Binaural audio technology is one of the most convenient means to deliver accurate spatial audio.

Headphones are not acoustically transparent and thus it affects both the timbral as well as the spatial quality of the input sound source. The effect of the headphones has to be compensated by calculating the inverse of the headphones transfer function (HPTF) and convolving it with the binaurally synthesized audio. HPTF also depends on the headphone-ear coupling and thus displays high spectral variation between individuals.

Accurate equalization of headphone responses is necessary in order to have a true immersive perception of the sound [1]. Typically, the headphone response comprises the headphones transducer response and the acoustic coupling between the headphones and the listener’s ears (Figure 1). In order to compensate for the headphone response, the HPTF is first measured at either the blocked ear-entrance or at the eardrum. It is not a necessary requirement that the “reproduction point” (i.e. the point where the binaural signals are played back) is the same as the recording point. However, the location mismatch between the reproduction and the recording point has to be compensated [2].

In general, the headphone equalization accounts for the compensation of (Figure 1):

1. The frequency response of the acoustic emitter of the headphones.
2. The acoustic coupling between the headphones and the listener’s ears.
Because of the fit of the headphone cup on the ear, the ear pinnae also distorts in shape when wearing headphones. This increases the non-linearity in coupling. Considering that personalized HRTFs are obtained with a non-deformed ear, they contain information about the ear shape, which is encapsulated in the HRTF.

In addition, the equalization may also compensate for the frequency response of the microphone and the mismatch of the radiation impedance looking out from the ear-entrance. To compensate for the HPTF response correctly for an individual, one needs to measure the headphone transfer functions at the eardrum or the blocked ear-entrance of the individual. Moller et.al [2] measured the headphone transfer functions (at the blocked ear canal) of 40 individuals for 14 different headphones and found that the inter-subject variations at the high frequencies are much higher compared to the low-frequency region. Blocked ear-canal measurements tend to have lesser inter-individual variations compared to the open ear-canal measurements.

Pralong and Carlile [3] obtained the HPTFs of the left and right ears of 10 subjects with a Sennheiser HD 250 circumaural headphone. It was found that considerable variability in the responses occurred at frequencies above 6 kHz, with an inter-individual standard deviation peaking up to 17 dB at frequencies around 9 kHz for the right ear. The inter-individual differences are also found to be similar to that in the HRTFs. Since, the high-frequency spectral cues are extremely important for localization perception, personalized equalization is necessary to ensure that headphones do not degrade the spatial perception. HPTF also varies considerably every time the headphone is removed and repositioned. Kulkarni et al. [4] obtained 20 transfer function measurements on the KEMAR manikin with the removal and repositioning of the headphone after each measurement. It is observed that the spectral features in the headphone transfer functions are comparable to those observed in HRTFs. For the same reason, we use personalized HRTFs as one of the inputs to the neural network model. Variations of the responses at the low frequencies are lower than that at higher frequencies, since the pressure at low frequencies within the earphone cavity can be approximated to be the same everywhere. Because of this variation in the measurement of the headphone responses, effective compensation is not possible even with personalized equalization and the spectrum at the eardrum becomes unpredictable.

The mean response (across the different measurements) is still not adequate because the mean inverse filter cannot equalize the filter functions completely. Interestingly, it has been pointed out that the variance produced by the spectral artefacts due to repositioning is lower than the variance of the spectral features useful for the SC [5].

In this paper, we present a technique to predict the personalized headphone equalization responses using neural networks. The Neural Network model designed predicts the headphone-ear coupling by taking personal-
ized HRTF and a generic headphone response as input to the models. In the following sections, we describe the Personalized Headphone-Transfer Function (p-HPTF) Neural Network (NN) in detail. We will also present the architecture, validation, and the limitations of this model.

2 WHY NEURAL NETWORKS?

The underlying assumption of this model is that the size and shapes of the ear and the headphone itself are effectively captured within the p-HRTF and generic HPTF; a sufficiently valid argument to make since these responses are both captured at the entrance of the ear canal. This means that just using the p-HRTF and generic HPTF could be sufficient to predict a p-HPTF instead of using topographical representations of the ear and headphone cup. In addition, representing the ear and headphone cups topographically can be challenging and could require a more complex mathematical model (or a NN) to learn the mechanical coupling.

We choose neural networks over modeling this problem numerically because of the versatility and the non-linear complexities that they offer. NNs are wonderful at understanding innate non-linear abstractions that are many a times difficult to model. With the appropriate model architecture and loss function combinations, abstract representations can be learnt by NNs that cannot be modeled parametrically.

3 NEURAL NETWORK ARCHITECTURE

The realization that headphones should non-linearly couple with the HRTF response in order to generate p-HPTFs forms the basis of this NN. An assumption is made here that the space for HPTF and p-HRTF can be transformed with the help of NNs, in which linear addition could represent non-linear coupling. By using non-linear transformations represented by multiple layers of a NN, a feature space is learnt that supports this addition. Latent space representations can be added in a transformed space to come up with embeddings that semantically make sense. For example, in Natural Language Processing (NLP), a latent space vector can be derived for the words ‘good’ and ‘king’ with their respective embeddings. Because of this, the words ‘good’ and ‘king’ can be semantically added in the latent space to construct the concept of a ‘good king’ as follows:

\[ \text{Embedding('good king')} = \text{Embedding('good')} + \text{Embedding('king')} \] (1)

where ‘good’ and ‘king’ are represented numerically in vector space [9]. The same argument can be applied here that the latent embeddings for p-HRTFs and generic HPTFs can be added linearly to semantically represent p-HPTFs at the output shows as follows:

\[ \text{Embedding('p - HPTF')} = \text{Embedding('p - HRTF')} + \text{Embedding('HPTF')} \] (2)

With this concept as the basis, a double input NN that takes in p-HRTF and HPTF vectors as inputs for a frequency range and predicts its corresponding p-HPTF is designed (Figure 2). Initial approaches were attempted with latent spaces using fully connected (FC) layers on the p-HRTF and HPTFs, but the results were...
not satisfactory. The key difference here is that unlike words in NLP that are singular in nature and whose embeddings may not be orthogonal, vector inputs that represent magnitude responses have orthogonal components on the frequency axis. This means that use of FC layers should be avoided while going from one layer of a NN to another until the latent space. This is because the FC layers due to their inherent nature would muddle frequency information from one bin into another. In this work, in order to isolate each frequency bin, parallel paths are created to learn an embedding and representation for each frequency bin separately. Advantage of adding embeddings in this way is that it can help keep the orthogonality of the p-HPTF embedding intact. In this paper, these parallel path embeddings are referred to as trickle embeddings because of its nature of trickling the information down from one frequency bin/band to the next bin/band.

Figure 3 highlights the inner workings of the proposed trickle layer. The input layer is divided into sequential overlapping chunks and the number of frequency bins to consider in a chunk can be adjusted. In this study, a window size of 5 is chosen, which means 5 sequential elements being presented as 1 input sub-vector for the next layer. The window size can be any size between 1 and n; 1 being using a single frequency bin for the next layer’s sub-vectors and n being using all elements from the input response. In order to avoid muddling, window sizes of 5 or less were chosen, which in terms of frequencies correspond to a band of $\frac{44100}{256} \times 5 = 861$ Hz.

The sub-vector from the input layer is transformed into another sub-vector for the 1st layer using a FC layer. A collection of these sub-vectors is called a trickle layer as information trickles down parallel paths into this layer. As seen from Figure 3, trickle layer 2 sub-vectors are obtained by passing their corresponding trickle layer 1 sub-vectors through their own FC layers. This avoids any muddling between frequency bands and keeps the embeddings isolated in frequency. Future trickle layers do not involve combining different sub-vectors.

$N$ such trickle layers can be constructed depending on the amount of complexity needed. The final embeddings to be used for addition can be siphoned from the last trickle layer. In an attempt to learn different representations for the p-HRTF and HPTF, the same set of trickle layers for the p-HRTF and HPTF were not used. Instead, each of the inputs were allocated their own set of trickle layers. The intuition behind this is that the coupling information that needs to be extracted from p-HRTF and HPTF is different since they both represent different things.

The last layer trickle embeddings can be added together to form the coupled embedding which can be then passed into a FC Layer to get the final p-HPTF response. It could be argued here that an approach similar to
trickle embedding might work better to isolate frequency responses to go from the coupled embedding to the p-HPTF response but in this study, using a FC layer on the entire coupled embedding gave the best results. As the coupling happens, contributions from all frequencies play a role in generating appropriate responses for all frequencies.

It is to be noted that this being a pilot study does not exhaustively cover all possibilities of NN architectures that could yield a p-HPTF with p-HRTF and HPTF as inputs. The NN in this paper explores mainly the idea represented by Equation 2, that could be expanded upon with future work.

4 LOSS FUNCTION

In order to effectively learn embeddings that semantically represent coupling, several loss functions are explored in this paper. Initial studies explored common loss functions like Mean Absolute Error (MAE), Mean Square Error (MSE), and Huber Loss but none of these were suited for this experiment as they performed poorly. Divergence loss functions such as Kullback-Leibler (KL) and Itakura-Saito (IS) [10] [8], were considered suitable for this study. Equation 3 represents IS divergence encapsulated in the absolute function to make it completely non-negative.

\[
\text{Loss} = \frac{1}{2n} \sum_{i=1}^{n} (\frac{Y_i}{Y_{\text{pred}_i}} - \log_{10}(\frac{Y_i}{Y_{\text{pred}_i}}) - 1)
\]

The magnitude response outputs range in between 0-1 and within this range, the IS distance curve is steeper than MAE, MSE and Huber. Figure 4 shows this IS distance along with \( x = y \) plane (optimal solution) along with the MSE curves. The inclusion of an absolute sign results in two different minimas for the IS distance, one at \( x = y \) plane and the other when the ground truth is minimal. Since the ground truth is normalized and does not go below \(-40dB\) in most cases, the only optimal solution is \( x = y \).
5 EXPERIMENTAL SETUP

In this paper, we use personalized HRTFs and HPTFs from the HUTUBS database from TU Berlin [11]. This database consists of HRTFs and HPTFs measured on 95 subjects with 10 repositions for Sennheiser HD 650 and Sennheiser HD 800. The dataset is split into 3 sets: Train set, Validation set, and Test set in the ratio 70% : 15% : 15%. The train and validation sets were used to train and fine-tune hyperparameters of the model. The test set was evaluated on the final, best performing model. The best performing model was chosen to be one, where both the training and validation losses are both minimal and stable over iterations before and after the point under consideration. This guaranteed a good model that generalized well over the training and validation set.

The p-HRTFs and p-HPTF responses were recorded at a sampling rate of 44.1 KHz, and a 256 pt FFT window is used yielding a 129 pt one-sided magnitude response. Ipsilateral p-HRTF responses at angles (±80, 0) were chosen as it captures all the idiosyncratic aspects of the pinnae. This angle is chosen as this comes closest to the flare angle of the ear when a user wears headphones [6]. To obtain the generic HPTF input (the second input to the NN apart from p-HRTF), the mean of all the p-HPTFs were computed for both left and right channels separately in the log magnitude space. This was carried out for all subjects in the dataset for HD650 and HD800 each. Ideally, NN model such as this would require a HPTF response that contains only the headphone-transducer characteristics decoupled from the ear.

The frequency range in the experiment for both the inputs (p-HRTFs, Generic HPTF), and the NN output generated (p-HPTF) is limited to 4 kHz to 16 kHz. The primary reason being this range encapsulates all the unique spectral features associated with the pinna (Pinna response transfer function). Responses below 4 kHz can be easily modelled with a generic response as it mainly consists of the ear-canal resonance, shoulder reflections, and other low-frequency non-idiosyncratic effects. In this experimental set up, 4 kHz to 16 kHz translates to 70 unique frequency bins in the 129 pt one-sided magnitude response.

As for the training scheme, the best performing NN had 6 trickle layers and 1 FC layer at the output. This data was then fed in batches of size 5 for both training and validation. Each trickle layer’s sub vector passed through a FC layer that has a batch normalization step before the activation function. A Leaky ReLU is used as the primary activation function. The last layer that generates the prediction utilized a sigmoidal activation function in order to predict responses between 0-1. Adam optimizer with a learning rate decay factor of 0.99 was used.

6 RESULTS

Figure 5 shows the p-HPTF magnitude response outputs generated by the NN model described above. IS distance measure is used to analyze the results generated in this paper. The origin of using IS divergence as a similarity score for audio applications was used by Cédric et al. [10] for Non-negative matrix factorization. A Mean score of IS score of 0.196 is obtained across all subjects and headphones. Figure 5 displays the p-HRTF generated outputs along with the true p-HRTF responses for both HD650 and HD800 taken from the test data set. Overall, it can be noted that the NN model captures the unique spectral features such as the notches and peaks accurately. It can also be seen that lower-the IS value, higher is the similarity between the predicted p-HRTF and the ground truth p-HRTF. In the next section, the limitations, caveats, as well as future work to improve the results further is explained.

7 DISCUSSION

This research study presents the importance of personalized HPTFs in spatial audio production. Here, a novel neural network based model is introduced that predicts accurate personalized HPTFs for a headphone model. A new scheme is proposed representing p-HRTFs and HPTFs in a space where non-linear coupling can be broken down into a linear sum. The proposed abstract space can be achieved by learning nonlinear complexities that make up this coupling, rather best with neural networks because of their versatility. A new kind of NN layer is described called the trickle layer that is inspired by transformer networks used in NLP and computer vision [7]. The trickle layer divides the input sequence into parts, sans the positional embeddings because the position
Figure 5. NN outputs for Personalized HPTF of both HD650 and HD800 headphones for different subjects. Blue : Predicted p-HPTF. Red : Ground truth p-HPTF

of the sub-vector itself represents positional information. The NN is trained on TU Berlin’s HUTUBS dataset [11] with the IS loss function and achieved a mean validation IS distance of 0.196.

Although, the NN model described here presents satisfactory results, there are several caveats and limitations to this study. To obtain a generic HPTF as an input to the model, average of all p-HRTFs from the database is taken. In an ideal scenario, the model should have an input that only encapsulates characteristics of the headphone transducer response and not ear features. Although taking an average eliminates most of the pinnae features, it does not eliminate all of them. In order to improve the results, a larger data set is required to train the NN that consists of p-HRTFs for multiple headphones. NN models with layer weight sharing for latent embedding calculations can further be explored to improve the mean IS metric on the validation set. Incorporating residual blocks into the training has in the past helped NN models with better results in a wide variety of fields. IS distance, although an objective score, is used as a measure that indicates perceptual distance in this study. The amount of tolerable IS distance or just noticeable difference for IS distance that is required for the predicted and true p-HPTF to be perceptually indistinguishable has to be investigated in future studies. In subsequent studies, these results also need to be validated by carrying out subjective experiments.
REFERENCES


ABSTRACT

We previously proposed a two-circular-loudspeaker-array (2CLA) model and investigated its focused source reproduction properties compared with those of a conventional circular loudspeaker array (CLA). The asymmetric array model exhibits direction-dependent reproduction performance with higher reproduction accuracy than that of CLA, for specific settings. In the present study, we investigated the properties of another asymmetric array model, the elliptical loudspeaker array (ELA), with respect to the focused source reproduction. For the focused source reproduction using this model, loudspeakers are mounted on an elliptical rigid baffle. We propose an exterior sound field reproduction method based on the Mathieu function expansion for a rigid ELA. Here, the transfer function of a rigid ELA was used instead of the free-field function. Numerical simulations were conducted, comparing the properties of the studied ELA to those of conventional arrays. The ELA performed similarly to the 2CLA for the same array dimension, suggesting that asymmetric array models have similar properties. Furthermore, detailed investigations were performed for various source directions and array dimensions.

Keywords: Sound Field Reproduction, Focused Source, Elliptical Loudspeaker Array

1 INTRODUCTION

Focused source reproduction was developed as a spatial audio technique [1, 2, 3]. This technique reproduces the sound field of a sound source between the loudspeakers and the listener; it yields the spatial perception of a virtual source.

Conventional studies of focused source reproduction have primarily considered wave field synthesis using linear loudspeaker arrays (LLAs). Such LLA-based reproduction yields a triangular listening area that limits the listener’s movement. In addition, exterior sound field reproduction methods have been proposed using circular loudspeaker arrays (CLAs), reproducing the entire radiating sound field with fewer limitations on the listening area. Furthermore, a two-circular loudspeaker array (2CLA) model has been proposed for the focused source reproduction. It was reported that the asymmetric array outperformed the conventional CLA under specific conditions [4, 5].

Another asymmetric array model is the elliptical loudspeaker array (ELA) model. Recently, we proposed an ELA model for interior sound field reproduction, together with its wave domain method, the Mathieu-function-expansion-based method [6, 7]. In this study, additional discussion is presented about using this ELA model for focused source reproduction (i.e., an application of exterior sound field reproduction). In addition, the ELA model is compared with the CLA and 2CLA models.

In Sec. 2, we modify the Mathieu-function-expansion-based method for exterior sound field reproduction using a rigid ELA. The transfer function of the rigid array is also introduced. In Sec. 3, we present the results of our numerical simulations. Reproduction examples at 1000 Hz are presented in Sec. 3.1. A comparison between the ELA, CLA and 2CLA models, for a frequency band, is shown in Sec. 3.2. The direction de-
dependence of ELA is discussed in Sec. 3.3. Finally, the effects of changing the axis length and flattening are discussed in Sec. 3.4. Note that this study only discusses the two-dimensional case.

2 METHOD

2.1 Elliptical Coordinate System and Sound Field Representation

First, we introduce the elliptical coordinate system and the orthogonal expansion of the sound field in the system. As shown in Fig. 1, the elliptical coordinate system is defined by the coordinates \( r = (u, v) \), such that

\[
\begin{align*}
  x &= a \cosh u \cos v, \\
  y &= a \sinh u \sin v,
\end{align*}
\]

(1)

where \((x, y)\) are the corresponding Cartesian coordinates and \( a \) represents the distance from the origin to one of the foci located at \( F(\pm a, 0) \) in the Cartesian coordinate system. In the elliptical coordinate system, an ellipse is defined by \( u = u_0 \).

With the Laplace operator in the elliptical coordinate system, the Helmholtz equation yields solutions, namely the Mathieu angular and radial functions \[8\]. In this study, the Mathieu angular and radial functions of the \( \zeta \)-th kind are denoted by \( \text{me}_n(q, u) \) and \( M_n^{(\xi)}(q, u) \), respectively. \( n \) denotes the order, and \( q \) is defined by \( q := \frac{\omega^2 a^2}{4 c^2} \), where \( \omega \) is the angular frequency and \( c \) is the speed of sound. The Mathieu functions considered in this study have only integer orders. These computational methods have been explained in the literature \[9\].

The Mathieu functions are mutually orthogonal, making them the orthogonal expansion bases of the sound field \[8, 9\]:

\[
p(u, v, \omega) = \sum_{n=-\infty}^{\infty} \left[ \alpha^i_n M_n^{(1)}(q, u) \text{me}_n(q, v) + \alpha^s_n M_n^{(0)}(q, u) \text{me}_n(q, v) \right],
\]

where \( \alpha^i_n \) and \( \alpha^s_n \) denote the expansion coefficients of the incident and scattered sound fields, respectively.

2.2 Rigid Elliptical Scatter

In this study, rigid ELAs (consisting of loudspeakers mounted on a rigid elliptical scatter) are discussed for reproducing focused sources. Rigid arrays have been reported \[5\] as more practical than open arrays, because they help to avoid forbidden-frequency issues \[10\] and are easier to implement. Here, we introduce the transfer function for a rigid ELA. With Eq. (2), the sound field in the elliptical coordinate system is the sum of the incident \( p_i(u, v) \) and scattering \( p_s(u, v) \) fields, where

\[
p_i(u, v) = \sum_{n=-\infty}^{\infty} \alpha^i_n M_n^{(1)}(q, u) \text{me}_n(q, v),
\]

(2)
\[ p_s(u, v) = \sum_{n=-\infty}^{\infty} \alpha_n^4 M_n^{(4)}(q, u) me_n(q, v). \]  

By applying the Neumann boundary condition to the rigid elliptical scatter, the particle velocity on its surface \((u = u_0)\) becomes zero in the normal direction, implying that the sum of the derivatives of the direct (the incident field) and scattered (the scattering field) sounds is 0. Then, we have

\[ \alpha_n^4 = -\frac{M_n^{(4)}(q, u_0)}{\pi M_n^{(4)}(q, u_0)} \alpha_n^4, \]

where \(M_n^{(4)}\) and \(M_n^{(4)}\) are the derivatives of the Mathieu radial functions. Substituting Eq. (4) into (3) and using the Wronskian of the Mathieu function

\[
\mathcal{W}\{M_n^{(1)}, M_n^{(4)}\} = -\frac{2j}{\pi},
\]

while the sound pressure on the elliptical surface is obtained as follows:

\[
p(u_0, v_0) = p_i(u_0, v_0) + p_s(u_0, v_0) = \sum_{n=-\infty}^{\infty} -\frac{2j}{\pi} \alpha_n^4 me_n(q, v_0).
\]

We set a point source at \((u, v)\) outside the elliptical scatter (i.e., \(u \geq u_0\)), its incident field is the free-field transfer function of a point source [8, 6], where

\[
\alpha_n^4 = -\frac{j}{4} M_n^{(4)}(q, u) me_n(q, -v).
\]

Finally, applying the reciprocity theorem (exchanging the source and observation point), the transfer function of a point source mounted on the rigid scatter, that is, the transfer function of the rigid ELA, becomes

\[
G(u, v|u_0, v_0) = \sum_{n=-\infty}^{\infty} -\frac{me_n(q, -v_0)}{2\pi M_n^{(4)}(q, u_0)} M_n^{(4)}(q, u) me_n(q, v).
\]

### 2.3 Focused Source Reproduction Using Rigid ELA

A Mathieu-function-expansion-based method was proposed for interior sound field reproduction using an ELA [5, 7]. Here, we modified it for the exterior sound field reproduction. Let the target source be located at \((u_t, v_t)\) outside the rigid ELA (where \(u_t > u_0\)). The primary (i.e., target) sound field outside the target source is the scattering field, written as

\[
p(u, v) = \sum_{n=-\infty}^{\infty} \gamma_n M_n^{(4)}(q, u) me_n(q, v),
\]

where \(u > u_t\). In particular, if the target source is a line source (i.e., a monopole source in the two-dimensional space), we have

\[
\gamma_n = -\frac{j}{4} M_n^{(1)}(q, u) me_n(q, -v_t).
\]

Next, we consider the secondary (i.e., reproduced) sound field. The sound field of the loudspeakers mounted continuously on a rigid elliptical scatter \(u_0\) is

\[
\hat{p}(u, v) = \int_0^{2\pi} G(u, v|u_0, v_1) D_t u_0 dv_1,
\]

where \((u_0, v_1)\) and \(D_t\) are the coordinates and driving functions of the \(l\)-th loudspeaker, respectively. Substituting Eq. (8) into (11), applying the Mathieu function expansion to the driving functions of the ELA, we obtain the following equation with the orthogonality of the Mathieu functions [9]:

\[
\hat{p}(u, v) = \sum_{n=-\infty}^{\infty} \hat{\beta}_n d_n M_n^{(4)}(q, u) me_n(q, v),
\]
\[ \beta_n = \frac{u_0}{M_n^{(4)}(q, u_0)} \]  
where \( d_n \) denotes the expansion coefficient of the driving function. For discretely placed loudspeakers, this becomes

\[ \beta_n = \frac{L}{2\pi M_n^{(4)}(q, u_0)} \]  
where \( L \) denotes the number of loudspeakers [7]. By matching the reproduced sound field in Eq. (12) to the target sound field in Eq. (9), we have

\[ d_n = \frac{\gamma_n}{\beta_n} \]  
where the orthogonality property is again used.

To suppress the filter gain (i.e., the output level of the loudspeakers), we use the Tikhonov regularization approach, as follows:

\[ d = (B^H B + \lambda I)^{-1}B^H \gamma, \]  
where \( \lambda \) is the regularization parameter. We truncate the infinite series at the maximum order of \( N \) such that

\[ d = [d_0, d_1, \ldots, d_N]^T, \]  
\[ \gamma = [\gamma_0, \gamma_1, \ldots, \gamma_N]^T, \]  
\[ B = \text{diag}(\beta_0, \beta_1, \ldots, \beta_N). \]  
The driving function in the frequency domain is

\[ D_l = \sum_{n=-N}^{N} d_n m e_n(q, v_l). \]  

3 NUMERICAL SIMULATIONS

In this section, we describe the results of our numerical simulations of the focused source reproduction. The results are compared with those of the CLA and 2CLA. Further characteristics of the ELA, including its direction dependence and array dimension, are also discussed.

3.1 Focused Source Reproduction Using ELA

We first conducted experiments to determine whether the proposed method can reproduce a focused source. Two examples of focused source reproduction were considered, at 1000 Hz. A monopole source located at \((x_s, y_s) = (0 \text{m}, 0.5 \text{m})\) was used as the target source. The ELA had 30 loudspeakers mounted equiangularly.
Figure 3. Focused source reproduction using an “on y” rigid ELA of 30 loudspeakers.

Figure 4. Reproduction error over the listening area, comparing the ELA, CLA, and 2CLA.

($\Delta v = \pi/15$) on a rigid elliptical surface. The lengths of the major and minor axes of the ELA were 0.8 m and 0.3 m, respectively. Two ELA settings were tested, namely “on x” (i.e., the major axis on the x axis) and “on y” (i.e., the major axis on the y axis). A elliptical coordinate system with $a = \sqrt{\frac{5}{20}}$ was set for the ELA, where $u_0 \approx 0.39$ and $q \approx 11.74$. The truncation order for all array models was $N = \left\lfloor \frac{L-1}{2} \right\rfloor = 14$, whereas the order for computing the transfer function was 20. Tikhonov regularization was applied, for suppressing the maximum driving function ($\max ||D_l||$) to less than 0 dB.

Figures 2 and 3 show examples of reproduction in a 2 m $\times$ 2 m sound field. The results show that the “on y” array configuration moderately reproduces the target sound field while the “on x” array configuration exhibits more errors, especially on both left and right sides of the target source. This can be explained by the fact that the array-source distance is smaller for the “on y” array configuration.

3.2 Comparison with CLA and 2CLA

To compare the performances of the ELA, CLA, and 2CLA with similar array dimensions, we conducted simulations with a fixed target source at (0 m, 0.5 m). The ELA was the same as that used in the previous experiment. The CLA had a radius of 0.15 m. The 2CLA model was set as two CLAs with centers 0.5 m away from each other. Note that the ELA and 2CLA had the same “length” (0.8 m) and “width” (0.3 m). Both the “on x” and “on y” ELAs and 2CLAs were tested for their asymmetry. The CLA and 2CLA were also rigid and had 30 loudspeakers mounted equiangularly on their surfaces. The CLA and 2CLA results were obtained using a circular-harmonic-expansion-based method [6]. The comparison was conducted for the 200-4000 Hz frequency band. The other conditions were the same as those in the previous experiments.

Figure 4 shows the reproduction error over the exterior listening area. The listening area was selected as a
Figure 5. Direction dependence of reproduction errors on focused source reproduction, using the ELA.

partial area of the entire exterior space: a ring area with a radius in the 0.4–4 m range. The reproduction error over the area was computed as

$$
\varepsilon_\Lambda(\omega) = 10\log_{10} \frac{\int_\Lambda |\hat{p}(x, \omega) - p(x, \omega)|^2 dx}{\int_\Lambda |p(x, \omega)|^2 dx},
$$

(21)

where \( \Lambda \) denotes the target area. The results show that the performance of the ELA was similar to that of the 2CLA with the same “length” and “width” of the array, while both the ELA and 2CLA outperformed the CLA.

3.3 Direction Dependence

Regarding the asymmetry of the ELA, we investigated its direction dependence by varying the target source directions. The same “on x” ELA was used throughout these studies. The target source had a fixed radius of 0.5 m. As the ELA is symmetrical with respect to the \( x \)-and \( y \)-axes, source directions between \( \nu_s = 0 \) and \( \nu_s = \pi/2 \) were tested. All other conditions were identical to those used in the previous simulations. The reproduction error over the listening area for the 200-4000 Hz range is shown in Fig. 5. The \( x \) and \( y \) axes in the figure represent the source direction and frequency, respectively. The results show the direction dependence, suggesting that the ELA performs better for lower angles of the source direction, which is the same for the 2CLA [5].

3.4 Array Dimension

To investigate how the array dimension influences the reproduction accuracy, simulations were conducted by changing the axis length or flattening of the array.

Flattening \( \mathcal{F} \) captures how flat an ellipse is: \( \mathcal{F} = 0 \) denotes a circle and \( \mathcal{F} \to 1 \) describes an ellipse that is almost a straight line. Flattening is defined as \( \mathcal{F} = \frac{a-b}{a} \), where \( a \) and \( b \) are the major and minor axes of an ellipse, respectively.

The same configurations as in the previous simulations were used, except that the focused source was at a single position, \((0 \text{m}, 0.5 \text{m})\). The simulations were conducted for the 200-4000 Hz frequency band.

The following four aspects of the ELA were investigated:

(i) how does the axis length perpendicular to the source direction affect performance (with another axis length fixed);

(ii) how does the axis length in the source direction affect performance (with another axis length fixed);

(iii) how does flattening affect performance with fixed perimeter of the array;

(iv) how does flattening affect performance with fixed area of the array.
Figure 6. Focused source reproduction errors, for changing the axis length (a) perpendicular to the source direction and (b) along the source direction.

(i) axis length perpendicular to source direction
The axis length in the source direction was fixed at 0.3 m in part (i). The axis length perpendicular to the source direction was varied from 0.01 m to 0.5 m. Note that the one with both axis lengths at 0.3 m was a CLA. Figure 6(a) shows the reproduction error for various axis lengths. The $x$-and $y$-axes in the figures represent the axis length and frequency, respectively. Color indicates the reproduction error. The results indicate that the axis length perpendicular to the source direction is roughly positively related to the reproduction accuracy.

(ii) axis length in source direction
The axis length perpendicular to the source direction was set to 0.3 m in part (ii). The axis length in the source direction was varied from 0.01 m to 0.5 m. Note that the one with both axis lengths at 0.3 m was a CLA. Figure 6(b) shows these results. The results indicate that the axis length in the source direction is roughly positively related to the reproduction accuracy.

Figure 7. Focused source reproduction errors for flattening variation (a) with a fixed perimeter and (b) with a fixed area.
(iii) flattening with fixed perimeter
In part (iii), the perimeter of the array was set to \(3\pi/5\) m, the same as that for a CLA with a radius of 0.3 m. Flattening was varied between 0 and 1. Note that the one with \(\mathcal{F} = 0\) denotes a CLA. The results of varying the flattening are shown in Fig. 7(a). The x axis in the figures represents the flattening. The results indicate that, for a fixed perimeter, the flattening is negatively related to the reproduction accuracy, indicating that the CLA exhibits the best performance.

(iv) flattening with fixed area
In part (iv), the area of the array was set to \(9\pi/100\) m\(^2\), the same as that for a CLA with a radius of 0.3 m. Flattening was varied between 0 and 1. Note that the one with \(\mathcal{F} = 0\) denotes a CLA. The results are shown in Fig. 7(b), indicating that for a fixed area, the flattening is negatively related to the reproduction accuracy; the dependence is weaker than that in part (iii).

4 CONCLUSIONS
In this study, we proposed a focused-source-reproduction method using an ELA. This method modifies and extends the previously proposed Mathieu-function-expansion-based method. Discussions were made based on the results of our numerical simulations. Overall, the results showed that the ELA and 2CLA with the same “length” and “width” have similar performances and direction dependence. Both asymmetric array models outperformed the conventional CLA. In addition, we found that for the ELA, the axis length was roughly positively related to its performance, whereas flattening had a slightly negative effect on the performance. A detailed discussion of the ELA model can be found in [11].

ACKNOWLEDGMENTS
This research was supported by JSPS KAKENHI Grant No. 19K12041.

REFERENCES
Near-field localization of sound sources using a spherical microphone array

Ning Xiang(1), Tommy Metzger(2)

(1) Graduate Program in Architectural Acoustics, Rensselaer Polytechnic Institute, Troy, New York USA, xiangn@rpi.edu
(2) Graduate Program in Architectural Acoustics, Rensselaer Polytechnic Institute, Troy, New York USA, tommy.r.metzger@gmail.com

ABSTRACT
Spherical microphone arrays have been widely used in various acoustical applications, such as detection and localization of sound sources. In this work, the spherical harmonics theory supports formulations of parametric models for a spherical microphone array to localize sound sources. In the source presence, the parametric models are critical to localize an unknown number of multiple sound sources. This work applies two levels of Bayesian inference for enumeration and localization of potentially multiple sources. This work focuses on near-field localization of sound sources. Upon a positive detection, Bayesian model selection and parameter estimation are carried out within a unified framework to localize near-field sound sources including source distances and direction-of-arrivals. This paper discusses model formulations, uncertainties of localization parameters using preliminary experimental data.

Keywords: Sound, Insulation, Transmission

1 INTRODUCTION
This work studies localization of sound sources in the near-field by a spherical microphone array for both the number and location(s) of the detected sound sources. Previous work on spherical microphone array for estimating direction of arrivals in far-field has been reported, among others, by Landschoot and Xiang [1]. The data are used to extract directional information via spherical harmonic beamforming, yet the extracted information is merely limited to the far-field directions of arrivals. In the far-field, source distance information is lacking, though. Bayesian analysis was applied first to enumeration of the number of concurrent sources, followed by their strengths and directions of arrivals (DoAs). The current work further extends the investigations using the spherical microphone array as done in [1], but processes the data obtained in the near-field. The spherical beamforming technique applies spherical Fourier transform to the sensed sound pressure on the surface of the spherical microphone array [1, 2, 3]. The processing yields a matrix/map of incident sound energy along with a distance filter [2]. This work investigates the capabilities of two-level Bayesian inference to carry out the source localization, particularly focus on formulation of source distances in terms of radial components of spherical wave solutions. Similar work using Bayesian inference has been reported in the literature to determine the number of sources and their positions. However, the data are collected using different types of microphone arrays [4]. This paper will present a brief description of the beamforming techniques and the model formulation applied in the two-levels of Bayesian inference followed by preliminary discussions of experimental results.

2 MODEL AND DATA FORMULATION
This paper intends to localize potentially multiple concurrent sound sources around the spherical microphone array based on parametric models. For this purpose, the prediction models need to incorporate an unknown number of sound sources along with localization parameters including source distances and directions of arrivals, while processing the sensed data from the spherical microphone array.
Figure 1. Beamforming data energy map of one and two sound sources using a rigid sphere microphone array of 32 channels. The source positions: $[r_1 = 5 \text{ cm}, \phi_1 = 50^\circ, \theta_1 = 88^\circ]$ for the single source; while $[r_1 = 5 \text{ cm}, \phi_1 = 50^\circ, \theta_1 = 88^\circ; r_2 = 5 \text{ cm}, \phi_2 = 300^\circ, \theta_2 = 50^\circ]$ for the two simultaneous sources.

2.1 Array beamforming

Microphone signals on the spherical array are processed to analyse incoming sound field and to estimate the sound energy around the spherical microphone array. Sound pressure signals $P_{\text{mic}}(k, a, \theta_i)$ of $M$ microphones flush-mounted on the rigid spherical surface of radius $a$ are Fourier transformed by

$$P_{nm} = \frac{4\pi}{M} \sum_{i=1}^{M} p_{\text{mic}}^2(k, a, \theta_i)[Y_m^m(\theta_i)]^*, \quad (1)$$

where $P_{nm}$ represents the transformed microphone signals in spherical harmonics domain, $\theta_i$ represents the microphone positions around the spherical microphone array of radius $a$. The spherical beamforming is accomplished in the spherical harmonics domain

$$D(\theta_l) = \sum_{n=0}^{N} \sum_{m=-n}^{n} Y_n^m(\theta_i) P_{nm}, \quad (2)$$

where $\theta_l$ represents (beamforming) looking directions. The microphone pressure signals are also squared in order to obtain signal 'energy' outputs. When sweeping the looking angles $\theta_l$, a data energy map can be obtained, in the following the data processed for further inferential analysis are denoted as $D$, they are all in two-dimensional matrix form. Figure 1 illustrates so achieved beamforming data for one or two concurrent sound sources.

2.2 Model formulation

This work specifically predicts the beamforming data of sound fields around the rigid sphere microphone array of radius $a$ as

$$g(\theta_s, r_s) = \sum_{n=0}^{N} \sum_{m=-n}^{n} B_n^m(k, r_s)[Y_n^m(\theta_k)]^* Y_n^m(\theta), \quad (3)$$
Figure 2. Near-field modal strengths (solid lines) of spherical order up to 5th in comparison with those of far-field ones (dot-lines)

where \( r_s \) represents the source distance, \( \theta_s = [\vartheta_s, \phi_s] \), represent the source direction, \( Y_{nm}(\theta) \) represents spherical harmonics with \( \theta = [\vartheta, \phi] \). \( k = \omega / c \) is the propagation coefficient with \( \omega \) and \( c \) being the angular frequency and sound speed, respectively. and \( B_n^s(k, r_s) \) is the near-field modal strength,

\[
B_n^s(k, r_s) = jk \left[ j_n(k a) - \frac{j'_n(k a)}{h'_n(k a)} h_n(k a) \right] h_n(k r_s),
\]

where \( j_n() \), \( h_n() \), and \( j'_n() \), \( h'_n() \) are spherical Bessel, Hankel functions and their derivatives, respectively.

Figure 2 illustrates the near-field modal strengths for the orders of spherical harmonics from 0th to 5th. Incorporating the near-field modal strength \( B_n^s(k, r_s) \) into the beamforming model side as in eq.(3) benefits a full parameterization of sound sources in the prediction models. Counting for potentially multiple sound sources, the prediction models can be established as

\[
M_S(\Theta) = \sum_{s=1}^{S} A_s g^2(\Theta, r_s),
\]

where \( \Theta = [\theta_s, \phi_s, r_s] \) collectively includes all the localization parameters for potentially \( S \) number of sound sources. Squared beamforming prediction \( g^2(\Theta, r_s) \) is consistent with the beamforming data energy as in Eq. (1). In the following \( M_S(\Theta) \) is denoted as the near-field prediction model for the beamforming data obtained from the spherical microphone array. The number of potential sound sources \( S \) is in the subscript of the model, while the parameter vector \( \Theta \) collectively includes all the source localization parameters.

The upper order \( N \) of the spherical harmonics in both Eqs. (2, 3) is dictated by the number of microphone channels \( M = 32 \). In this work, \( N = 4 \) is achieved. One of crucial localization issues is to process the data and the model for one-source against ‘no-source’ scenario. In this situation, the ‘no-source’ scenario is processed and modeled by the 0-order spherical harmonics. To be more precise, the background (no-source) model is to assign \( N = 0 \) in Eq. (3) with \( S = 1 \) in Eq. (5).

3 A UNIFIED BAYESIAN FRAMEWORK

This work applies Bayesian analysis for enumeration and localization of sound sources in near-field. The analysis is accomplished using Bayes’ theorem for the source localization tasks as described above which are included in the background information \( I \),

\[
p(\Theta|D, M_S, I) = \frac{p(D|\Theta, M_S, I)p(\Theta|M_S, I)}{p(D|M_S, I)}.
\]
Denoting $Z = p(D|M_S,I)$, $\mathcal{L}(\Theta) = p(D|\Theta,M_S,I)$, $\Pi(\Theta) = p(\Theta|M_S,I)$, a simplified form of Bayes’ theorem is used in the following as

$$\frac{\text{posterior}}{p(\Theta|D,M,I)} \times \frac{\text{evidence}}{Z} \times \frac{\text{likelihood}}{\mathcal{L}(\Theta)} \times \frac{\text{prior}}{\Pi(\Theta)},$$

(7)

where the subscript of model is omitted for simplicity from now on, but still bearing in mind that the sound source enumeration needs to select the number of concurrent sound sources $S$.

Equation (7) encompasses both levels of Bayesian inference (parameter estimation and model selection) in one unified framework. At the same time it explicitly relates two input quantities of Bayes’ theorem [5], the likelihood and the prior on its right-hand side, with two output ones, the posterior and the evidence on its left-hand side. In particular, the posterior is the key output for estimating the corresponding source localization parameters for a known number of sound sources, while the evidence represents the key quantity for selecting the correct number of sound sources. When pursuing the model selection, a logarithmic evaluation of Bayes’ factors [5]

$$\zeta_{ij} = 10\log_{10}(Z_i) - 10\log_{10}(Z_j), \text{ [decibans]}$$

(8)

needs to be estimated among the pre-selected model set, this estimation largely relies on the Bayesian evidence of each individual model $Z_i$. For computationally efficient estimations of Bayesian evidence, a nested sampling [6] has been successfully applied to this effort.

Within the unified Bayesian framework, the prior $\Pi(\Theta)$ is assigned to uniform distributions for each individual parameters, while the likelihood $\mathcal{L}(\Theta)$ is the quantity to incorporate the data and the model $M_S$. The model is part of prior knowledge given the data, and the likelihood is one of priors to be assigned as well. Applying the principle of maximum entropy leads to the likelihood assignment to be the Student’s $T$-distribution [5],

$$\mathcal{L}(\Theta) \propto \frac{\Gamma(Q/2)}{2} (\pi E)^{-Q/2},$$

(9)
Figure 4. Comparison between experimentally measured data maps with those predicted by the beamforming model. In one source scenario, the sound source finds itself at 5 cm distance from the origin of the sphere, while for two source scenario, the second source finds itself at 5.5 cm distance.

with

\[ E = \sum_{j=1}^{J} \sum_{k=1}^{K} \left[ D(\theta_j, \phi_k) - M_S(\theta_j, \phi_k) \right]^2, \]  

(10)

where \( \Gamma(\cdot) \) is Gamma function and \( Q = J \times K \), and \( 1 \leq j \leq J \) covers the entire elevation range of interest and \( 1 \leq k \leq K \) the entire azimuth range of interest, respectively. The data are processed by Eq. (2), while the model is determined by Eq. (5).

4 PRELIMINARY EXPERIMENTS

Figure 3 shows a photograph for the near-field experiments. The spherical microphone array holds 32 microphones embedded flush on the rigid sphere surface. The microphones are distributed nearly equidistantly. A sound source finds itself near the spherical microphone array. Impulse responses are measured between the sound source and the microphone array in a wide variety of directions and distances in the near-field.

Figure 4 illustrates the directional maps of two scenarios. One source and the two simultaneous sources near the microphone array with the first source located at 5 cm distance from the origin of the spherical microphone array. In the two-source scenario, the second source finds itself at 5.5 cm distance. The left column of Fig. 4 shows the model predictions using Eq. (5), while the right column are the data maps derived from the data processing based on the spherical Fourier transform via Eq. (2).

Figure 5 illustrates a group of results for the data of two-source scenario based on the higher level of Bayesian inference, which relies on estimation of Bayesian evidence in form of Bayes factors. The result indicate that two-source \( \zeta_{2,1} \) is strongly evidenced in the data over that of one source \( \zeta_{1,0} \), also over that of three-sources \( \zeta_{3,2} \), and four-sources \( \zeta_{4,3} \). Note that in order to estimate Bayes factor \( \zeta_{1,0} \), the '0-source' model has to be involved. As mentioned before, the '0-source' model is still the prediction model expressed in Eq. (5), yet with \( S = 1 \) along with Eq. (3) with \( N = 0 \).

Table 1 lists the estimated source distances for the two simultaneous near-field sources along with individual amplitude parameters \( A_2 \) relative to \( A_1 \) (in the two right-most columns). The probabilistic estimation also benefits quantification of uncertainties. They are also listed as standard deviations. Note that the left-most two
columns list the expected source distances physically set during the experiments, they are also subject to setting errors.

<table>
<thead>
<tr>
<th>Source 1 (cm)</th>
<th>Source 2 (cm)</th>
<th>Estimate 1 (cm)</th>
<th>Estimate 2 (cm)</th>
<th>Amplitude 1</th>
<th>Amplitude 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td>3.96±.0065</td>
<td>3.78±.0063</td>
<td>1</td>
<td>0.93±.0001</td>
</tr>
<tr>
<td>5</td>
<td>5.5</td>
<td>4.02±.0044</td>
<td>4.87±.012</td>
<td>1</td>
<td>1.1±.013</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>4.05±.0033</td>
<td>4.71±.019</td>
<td>1</td>
<td>0.27±.0047</td>
</tr>
<tr>
<td>5</td>
<td>7.5</td>
<td>4.06±.0032</td>
<td>4.52±.018</td>
<td>1</td>
<td>0.15±.0030</td>
</tr>
</tbody>
</table>

5 SUMMARY
This paper reports the preliminary investigations of the near-field sound source localization using a spherical microphone array. The current effort is to establish the distance-related prediction model using radial solution of Helmholtz wave equation in spherical coordinates along with angular solutions (spherical harmonics). The formulation of two levels of Bayesian inference heavily follows the previous work published most recently. Preliminary results indicate the formulated Bayesian estimation framework is capable of estimating the source distances with considerable uncertainties, particularly when competing sound sources are present in near-field. Further effort will orient research investigations toward systematic studies on source distance and amplitude dependence, estimation uncertainties examined by the quantitative near-field condition. Sound source detection using Bayesian model comparison should be another line of future research.

ACKNOWLEDGEMENTS
Authors would like to thank Dr. Efren Fernandez-Grande for his insightful comments on the early stage of this work.

REFERENCES


A spherical beamforming algorithm for acoustic centering and phase correction of source directivities

Samuel D. BELLOWS; Timothy W. LEISHMAN

1,2 Brigham Young University, USA

ABSTRACT
The directivities of acoustic sources have many applications in auralizations, room acoustical designs, and sound source modeling. Of practical importance in directivity measurements is the location of the source’s acoustic center relative to the array’s geometric center. The authors recently developed an acoustic source centering algorithm based on the equivalence of far-field magnitude directivity patterns of centered and uncentered directivities. This work explores how the far-field phase directivity patterns likewise lead to source-centering procedures. While the phase-based algorithm is less robust than the magnitude-based approach, the technique is notable for its significant improvements in computational efficiency. The paper provides algorithm validations with both theoretical sources and measured trumpet directivities.

Keywords:

1. INTRODUCTION
The directivity of a sound source characterizes the spatial dependence of its acoustic radiation over frequency. Understanding the directional characteristics of sources has broad applications, including auralizations, room acoustical design, microphone placement, and source modeling. Directivity measurements are typically performed by sampling at a constant-radius spherical surface with a specified sampling density, such as the 5 or 10-degree angular resolution suggested by the AES sampling standard (1). Of practical concern when measuring the directivity of a source is the location of the source relative to the geometric center of the microphone array. Source misalignment within the array can lead to several undesirable effects. First, the measured pressure on the array surface may not be representative of the desired far-field directivity pattern (2). Additionally, source translations increase the number of expansion terms required, which in turn increase the likelihood of spatial aliasing (3, 4). Thus, proper source positioning within the array is essential for practical directivity measurements.

More recently, the authors have shown that in the far-field, the magnitude pattern of centered and translated sources are equivalent (5). One can exploit this equivalence to determine the reference frame from which the near-field pattern converges most rapidly to the known far-field magnitude pattern. While this approach is robust at higher frequencies and useful for complex sources, its computational expense is limiting because the formulation requires extraction of magnitude patterns in the spatial domain rather than an efficient spherical-harmonic-based process. To improve upon these limitations, this work presents an acoustic centering algorithm based on a source’s far-field phase patterns. The relations between the far-field phases of centered and translated sources can cast the acoustic centering problem as the detection of arrival (DOA) of a local plane-wave with efficient solutions using spherical-harmonic-domain beamforming. Theoretical results help validate the robustness of the approach, and the technique successfully centers measured trumpet directivities.

1 samuel.bellows11@gmail.com
2 tim_leishman@byu.edu
2. ALGORITHM

2.1 Far-field Phase Relationship

Consider an acoustic source radiating into free space. If a closed spherical surface of radius \( r = a \) entirely encompasses the source, the exterior solution to the Helmholtz equation in spherical coordinates yields the pressure for \( r \geq a \) (6):

\[
p(r, \theta, \phi, k) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} c_{n}^{m}(k) h_{n}^{(2)}(kr) Y_{n}^{m}(\theta, \phi),
\]

where \( k \) is the wavenumber, \( h_{n}^{(2)}(kr) \) are the spherical Hankel functions of the second kind of order \( n \), \( Y_{n}^{m}(\theta, \phi) \) are the normalized spherical harmonics of degree \( n \) and order \( m \), and \( c_{n}^{m}(k) \) are the frequency-dependent expansion coefficients. Exploiting the orthogonality of the spherical harmonics over the sphere yields the expansion coefficients:

\[
c_{n}^{m}(k) = \frac{1}{h_{n}^{(2)}(ka)} \int_{0}^{2\pi} \int_{0}^{\pi} p(a, \theta, \phi, k) [Y_{n}^{m}(\theta, \phi)]^* \sin \theta d\theta d\phi,
\]

where \(*\) indicates complex conjugation. In the acoustic and geometric far-field of the source, where \( kr \gg 1 \) and \( r \gg d \), with \( d \) being the spatial extent of the source, the asymptotic form of the spherical Hankel functions simplifies Eq. (1) to the form

\[
p(r, \theta, \phi, k) \approx \frac{e^{-i kr}}{kr} \sum_{n=0}^{\infty} \sum_{m=-n}^{n} c_{n}^{m} i^{n+1} Y_{n}^{m}(\theta, \phi).
\]

The far-field simplification allows separation of the pressure field’s angular and radial dependence, yielding an unnormalized directivity function

\[
\tilde{D}(\theta, \phi) \approx \sum_{n=0}^{\infty} \sum_{m=-n}^{n} c_{n}^{m} i^{n+1} Y_{n}^{m}(\theta, \phi).
\]

Next, one may assume that the source has an acoustic center and let \( D_c(\theta, \phi, k) \) denote its far-field normalized directivity function with the origin of the coordinate system aligned with the acoustic center. One may further let \( D(\theta, \phi, k) \) denote the sources’ far-field directivity function measured with the acoustic center located at position \( r_c \). The first directivity product theorem then relates these two directivity functions as (5, 6)

\[
D(\theta, \phi, k) = e^{i \hat{r} \cdot r} D_c(\theta, \phi, k),
\]

where \( \hat{r} \) is the unit vector in the direction of \( r \). This key result shows that in the far-field, the magnitude directivity patterns of the centered and uncentered source are equivalent. However, the phase shift factor \( e^{i \hat{r} \cdot \hat{r}} \) between the centered and uncentered patterns remains. Figure 1 illustrates this shift through color-mapped far-field phase spheres for a monopole, dipole, and radially vibrating cap set on a rigid sphere. Each constant-radius plot describes the phase of the pressure using a cyclical color scheme. Figures 1(a), 1(d), and 1(g) show the far-field phase of each source when aligned with the array center. For the case of the vibrating cap on a sphere, the center of the sphere aligns with the array center. Next, Figs. 1(b), 1(e), and 1(h) show the far-field phase for each source after a translation of \( r_c = (0.0, 0.1, 0.3) \) m. Finally, Figs. 1(c), 1(f), and 1(i) show the difference in phase pattern between the translated and untranslated cases. The numerical results verify that regardless of how simple or complex the initial phase of an untranslated source may be, the phase difference depends only on the translation and wavenumber as described in Eq. (5).
Figure 1 – Far-field phase spheres for a monopole [(a)-(c)], dipole [(d)-(f)], and radially vibrating cap on a sphere [(g)-(i)]. These include cases for the sources located at the center of the measurement array [(a), (d), and (g)], translated to position (0.0, 0.1, 0.3) m [(b), (e), and (h)], and the phase differences between the untranslated and translated sources [(c), (f), and (i)].

2.2 Phase-based Centering Algorithm

Previous work exploited the equivalence of the far-field directivity magnitudes to develop a centering algorithm that is robust even at high frequencies and for complex sources (5). However, a disadvantage of the algorithm is that its use of magnitude patterns produces a computationally expensive approach in the spatial domain. This section leverages phase relationships between far-field directivities to yield a more computationally efficient algorithm with less robustness.

To begin with, for some sources one may assume that the phase of the centered pattern is roughly constant so that the measured far-field phase of the translated source is

$$\Psi(\theta, \phi, k) \approx k r_e \cdot \hat{r}.$$  \hspace{1cm} (6)

The pressure of a unit-amplitude plane wave is

$$p(r, \theta, \phi, k) = e^{-i k r} = e^{-i r k \hat{r}},$$ \hspace{1cm} (7)

where $k$ is the wavenumber vector with magnitude $k$, which points in the direction of propagation. By associating the term $k r_e$ of the far-field phase with the $-r \hat{k}$ term of the plane wave, one may cast the acoustic centering problem as determining the direction of arrival (DOA) of the plane wave. One key difference between the two problems is that for the plane-wave DOA, $r$ is typically known, so the steering need only be performed over the angular coordinates $\theta$ and $\phi$. However, for the source centering problem, $r_e$ is unknown; the varying radial positions require additional consideration.

A straightforward approach to the problem is to use a delay-and-sum beamformer. In this case one may let $y$ be the output of the beamformer so that (7)
\[
y = \mathbf{w}^H \mathbf{p}_{nm}.
\]  

Here, \( \mathbf{p}_{nm} \) is a vector containing the spherical harmonic expansion coefficients of the far-field phase \( e^{i\psi} \) and \( \mathbf{w}_{nm} \) are the spherical harmonic coefficients of the beamforming weights,

\[
w_{nm} = 4\pi n j_n(-kr) Y_n^m(\theta, \phi),
\]

where \( j_n \) are the spherical Bessel functions of order \( n \). The position \( (r, \theta, \phi) \) that maximizes the beamformer output \( y \) is then the acoustic source center. Figure 2 illustrates this approach for a monopole located at \( r_c = 0.3 \) m and \( (\theta_c, \phi_c) = (45^\circ, 90^\circ) \) with wavenumber \( k = 10 \) m\(^{-1} \). Six different spherical projections for varying radial steering positions \( r_c \) show the beamformer output. The color scheme of each projection is kept constant to highlight that the acoustic center must be determined not only from the angular portion but also from the radial component. The beamformer's maximum output coincides with the monopole's true location, indicated by a red dot.

Figure 2 – Delay-and-sum beamformer output for the acoustic source centering problem.

While the formulation based on Eq. (8) provides a satisfactory result, one can further improve the algorithm’s computational efficiency by noting the strong axial symmetry of the far-field phase as seen in Figs. 1(c), (f), and (i). Because the order \( m = 0 \) spherical harmonics are axially symmetric about \( \mathbf{z} \) for any degree \( n \), the rotation of the far-field phase function that maximizes the energy in the \( p^0_n \) coefficients can determine the direction in which \( \mathbf{r}_c \) points. The Wigner-D rotation matrices \( D(\theta, \phi, \psi) \) allow this rotation to be carried out in the spherical harmonic domain (7). By letting

\[
q_{nm}(\theta, \phi, \psi) = D(\theta, \phi, \psi) p_{nm}
\]

be the far-field phase spherical harmonic expansion coefficients after rotation, one finds that maximizing the objective function

\[
f(\theta', \phi') = \sum_{n=0}^{\infty} |q_{n0}(\theta', \phi', 0)|^2
\]

yields the rotation required to orient the phase so that it is axially symmetric about \( \mathbf{z} \). Once this rotation is known, the direction \( (\theta_c, \phi_c) \) can be determined, and the delay-and-sum equation weights can be used with fixed angular components and varying radial components. The ambiguity in the rotation angle between \( \mathbf{z} \) and \( -\mathbf{z} \) resolves by allowing \( r_c \) to vary over both positive and negative ranges. Furthermore, because the degree \( n = 1 \) expansion terms contain the relevant directional
information for a single plane wave, the expansion coefficients $p_{nm}$ may be truncated to a maximal $N=1$ expansion, with the associated Wigner-D rotation matrices being of size $4 \times 4$ for increased computational efficiency.

3. THEORETICAL RESULTS

The dodecahedron regular polyhedron loudspeaker (RPL) is an interesting source to study because, even though it behaves much like a simple source at low frequencies, its directivity becomes complex at high frequencies (8). If one assumes the source has a single acoustic center, it must fall at the RPL center due to geometrical arguments. Thus, the RPL provides an ideal case of a complex radiator with a known acoustic center. Figure 3(a) shows the centered far-field phase of a simulated dodecahedron RPL for wavenumber $k = 30 \text{ m}^{-1}$ and RPL radius $a = 0.2 \text{ m}$. Strong phase shifts are evident for each driver from the red patches in the general sphere of blue. Figure 3(b) shows the far-field phase after a source translation to $r_c = (0.0, 0.1, 0.2) \text{ m}$. Figure 3(c) then shows the simplified phase using the degree $N=1$ expansion, which correctly identifies the direction of translation.

Figure 3 – Far-field color-mapped phase spheres for the (a) centered, (b) translated, and (c) translated dodecahedron using a simplified degree $N=1$ expansion.

Figure 4 shows the centering results. The projected sphere plots the objective function $J(\theta', \phi')$ applied to the simplified phase shown in Fig. 3(c). The red dot indicates the true angular direction of the translation whereas the black $\times$ indicates the predicted direction. The line plot shows the delay-and-sum output using the identified direction and varying the radial parameter $r_c$. The vertical dashed red line indicates the true radial position.

Figure 4 – Centering results for dodecahedral RPL. Left: Objective function $J(\theta', \phi')$. Right: normalized delay-and-sum output using a fixed angular position and varying $r_c$.

4. EXPERIMENTAL RESULTS

A radius $a = 1.17 \text{ m}$ rotating semi-circular microphone array measured the directivity of a played trumpet with various mutes. The microphone array consisted of 36 12.7 mm (0.5") precision microphones that were relatively calibrated to a dedicated channel. The microphones were placed in $5^\circ$ polar angle increments. Subsequent arc rotations in $5^\circ$ azimuthal increments swept out a sphere with sampling density consistent with the AES standard on loudspeaker directivities (1), minus the
nadir (south pole) measurement position. A near-field reference microphone normalized varying excitation levels between the repeated measurements through frequency response functions (FRFs) as outlined in Ref. (9). A head restraint and laser mounted to the instrument restricted the musician’s movements for greater consistency between measurements. Figure 5 shows the trumpet player within the measurement system.

Figure 5 – Trumpet player within measurement arc while playing with a cup mute.

Because of the small arc radius, placement the trumpet’s bell close to the array’s center was not practical. Consequently, the authors anticipated source misalignment between the instrument's acoustic center and the array's geometric center. Figure 6 shows the raw FRF-based directivity balloons with 1 Hz narrowband resolution for the first three partials of the note E4: 329 Hz, 658 Hz, and 988 Hz, respectively). Color and radius both depict levels on the surface of constant $r = a$, with the $0^\circ$ marker indicating the direction in front of the musician. Figures 6(a)-(c) show the magnitude of the FRF-based balloon, whereas Figs. 6(d)-(f) show the phase. The magnitude balloons highlight the effectiveness of the FRF method, as the directivity functions show smoothly varying functions that reveal interesting directional characteristics, such as increasing diffraction lobes at higher frequencies. However, there appear to be reduced levels below the musician, likely due to the source placement within the array. The phase-based FRF-balloons validate this assertion, as similar patterns appear in Figs. 1 and 3(c).

Figures 6(g)-(i) show the far-field magnitude directivities after propagation via Eq. (3) using an $N = 34$ expansion, the maximal possible for the given sampling configuration. The directivity is much less directional, implying that many of the features evident in Figs. 6(a)-(c) are likely near-field effects. In addition, the principal axis of radiation lowers slightly.

The phase-based centering algorithm determined the acoustic center of each partial to be at $(0.25, 0.00, 0.43)$ m, $(0.25, 0.00, 0.42)$ m, and $(0.29, 0.00, 0.34)$ m, respectively. While the acoustic center is generally considered to be frequency dependent, the position remains relatively consistent for the three partials. Figures 6(j)-(l) show the corresponding centered directivities of Figs. 6(a)-(c) based on expanding the pressure about the estimated acoustic center at a $a = 1.7$ m radius. Importantly, they show strong similarities with the far-field pattern, even though these are not far-field directivities. This result shows the effectiveness of the centering algorithm, as the pressure measured about the acoustic center should quickly converge to the far-field pattern (5). The area-weighted RMS deviation values (8) show that the deviations between the centered patterns of 6(j)-(l) and the far-field patterns of 6(g)-(i) were only 1.3, 1.7, and 2.0 dB compared to 1.9, 2.1, and 2.5 dB for the measured patterns of 6(a)-(c). Thus, the centering algorithm effectively reduces deviations between far-field directivities and nearer-field measurements.
5. CONCLUSIONS

This work has presented an acoustic centering algorithm based on the far-field phases of sources. It allows the acoustic centering problem to be reformulated in terms of a spherical beamforming problem. Theoretical directivities and measured musical instrument directivities validate the method. Future work could include the application of the algorithm to other sound sources and exploring other approaches for sources with more complex phase patterns.

ACKNOWLEDGEMENTS

This work was funded by the William James and Charlene Fuhriman Strong Family Musical Acoustics Endowed Fellowship.

REFERENCES

5. Bellows SD, Leishman TW. Acoustic source centering of musical instrument directivities using acoustical
Study of speaker localization under dynamic and reverberant environments

Daniel A. Mitchell* and Boaz Rafaely†

School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Israel

ABSTRACT

Speaker localization in a reverberant environment is a fundamental problem in audio signal processing. Many solutions have been developed to tackle this problem. However, previous algorithms typically assume a stationary environment in which both the microphone array and the sound sources are not moving. With the emergence of wearable microphone arrays, acoustic scenes have become dynamic with moving sources and arrays. This calls for algorithms that perform well in dynamic environments. In this article, we study the performance of a speaker localization algorithm in such an environment. The study is based on the recently published EasyCom speech dataset recorded in reverberant and noisy environments using a wearable array on glasses. Although the localization algorithm performs well in static environments, its performance degraded substantially when used on the EasyCom dataset. The paper presents performance analysis and proposes methods for improvement.

Keywords: Direction of Arrival (DOA) estimation, Direct Path Dominance (DPD) test, wearable microphone arrays, EasyCom dataset

1 INTRODUCTION

Localizing multiple sound sources recorded with a microphone array in an enclosure is an important task used in a wide range of applications such as speech enhancement, source separation and video conferencing [2]. Therefore, many direction-of-arrival (DOA) estimation methods have been developed for this task. These include methods based on beam-forming [11], subspace methods such as multiple signal classification (MUSIC) [9], and time-delay of arrival estimation methods [3]. Many of the algorithms based on these methods were designed assuming a free-field environment, but when these algorithms are used in a more common reverberant environment their DOA performance degrades. The reason for this is that in a reverberant environment, room reflections mask the direct sound which carries the DOA information. Recently, however, several methods have been developed for DOA estimation of multiple speakers that are robust to reverberation. One such method processes the microphone signals in the time-frequency domain, and employs a direct-path dominance (DPD) test [7] to identify time-frequency bins that are dominated by the direct sound from the source. Algorithms which use this method have been widely studied assuming a static environment, where both the sound sources and the microphone array are stationary. On the other hand, these algorithms have been less intensely studied in a more realistic, dynamic environment, where the sound sources and/or the microphone array are moving.

In dynamic environments, the motion of the sound sources and/or microphone array may lead to the rapid change of DOAs in time. Thus, to accurately trace the DOA of speakers requires a short interval between successive DOA estimates. Additionally, the DOA estimates may be smoothed in time using a tracking algorithm. Although DOA estimation and tracking algorithms in dynamic environments have been the subject of several recent studies [5, 12], including the Acoustic Source Localization and Tracking (LOCATA) [6] challenge, none of these have included experiments with wearable microphone arrays - which may bring new challenges. Such scenes are becoming increasingly popular due to the increased interest in applications involving augmented reality.

In this article we address the problem of DOA estimation in a noisy dynamic environment involving a wearable microphone array. The experiments were performed using the Easy Communication (EasyCom) dataset

* mitchdan@post.bgu.ac.il † br@bgu.ac.il
[4], which was explicitly designed to represent a realistic cocktail-party environment. The DOA estimates were computed using a computationally efficient algorithm, which had been shown previously to have good source localization performance in a static reverberant environment [10]. The algorithm incorporates a DPD test and operates in the time-frequency domain. We study the performance and limitations of this algorithm on the EasyCom dataset under different operating parameters. We also introduce two modifications of the algorithm and study the consequent improvement in performance.

2 MATHEMATICAL MODEL

In this section we first briefly present the model assumed for the recorded signal as captured by the microphone array at each time-frequency bin \((t, f)\). Then, we describe the local space domain distance (LSDD) algorithm [10] for the DOA estimation at each bin \((t, f)\).

2.1 Signal model

Assume a microphone array with \(M\) microphones arranged according to a pre-defined geometry. Next, consider a sound field comprised of \(K\) far-field sources, arriving from directions \(\Psi_k, k \in \{1, 2, \ldots, K\}\). These sources represent the direct sound from the speakers in the scene, as well as reflections (reverberations) due to objects and room boundaries.

In the next step, the recorded microphone signals are transformed into the joint time-frequency domain by applying the short-time Fourier transform (STFT). This is done by first separating the speech signal into short time intervals of length \(\delta t\). A fast Fourier transform (FFT) is then applied to each time segment. Following these pre-processing steps, the signal received by the microphone array can then be described in the STFT domain as

\[
x(t, f) = \sum_{k=1}^{K} s_k(t, f)v(f, \Psi_k) + n(t, f),
\]

where \(x(t, f) = [x_1(t, f), x_2(t, f), \ldots, x_M(t, f)]^T\) is an \(M \times 1\) complex vector denoting the signal as measured by the microphones composing the array; \(v(f, \Psi_k) = [v_1(f, \Psi_k), v_2(f, \Psi_k), \ldots, v_M(f, \Psi_k)]^T\) is an \(M \times 1\) vector denoting the response of the microphone array to a unit-amplitude plane wave at frequency \(f\) arriving from the \(k\)th source in direction \(\Psi_k\); \(s_k(t, f)\) is a scalar which represents the amplitude of the \(k\)th sound source signal; and \(n(t, f) = [n_1(t, f), n_2(t, f), \ldots, n_M(t, f)]^T\) is an \(M \times 1\) vector denoting the noise in the signal \(x(t, f)\).

2.2 Local Space Domain Distance (LSDD) algorithm

The LSDD algorithm is a recently developed DOA estimation algorithm characterized by DOA performance that is robust to reverberation. The algorithm was first proposed in [10] and works as follows. The directional spectrum \(S(t, f) = [S_1(t, f), S_2(t, f), \ldots, S_L(t, f)]^T\) computed within this algorithm is an \(L \times 1\) vector defined over a grid of DOAs \(\Theta_l, l \in \{1, 2, \ldots, L\}\). The \(l\)th component is defined as

\[
S_l(t, f) = d(x(t, f), v(f, \Theta_l)) = d([x_1(t, f), \ldots, x_M(t, f)]^T, [v_1(f, \Theta_l), \ldots, v_M(f, \Theta_l)]^T),
\]

where \(d(a, b)\) is a function which measures the similarity between two vectors \(a\) and \(b\). In [10], \(d(a, b)\) was defined as

\[
d(a, b) = \frac{1}{\beta} \min_{\beta} \left( \frac{|a - \beta b|}{|a|} \right),
\]

where \(\| \cdot \|\) is the 2-norm. However, in this article we shall use the more conventional cosine similarity measure:

\[
d(a, b) = \frac{<a, b>}{\|a\| \|b\|},
\]

where \(<a, b>\) denotes the inner product between \(a\) and \(b\). Given the spectrum vector \(S(t, f)\), the estimated DOA for bin \((t, f)\) is computed by

\[
\hat{\theta}(t, f) = \arg \max_{\theta} \{S_l(t, f)\}.
\]
However, some of the bins \((t, f)\) do not contain a valid \(\hat{\theta}(t, f)\) value. These are bins in which the direct signal from the speaker is masked by noise and reverberations. We eliminate these bins by calculating a DPD measure value \(\chi(t, f)\) for each bin which we then test against a threshold \(\lambda\). Although there are different methods available for calculating \(\chi(t, f)\), we shall, for simplicity, use the following:

\[
\chi(t, f) = \max_{l} \{ S_l(t, f) \} .
\]  

(6)

Together, Eqns. (5) and (6) define a (joint) LSDD DOA/DPD algorithm.

### 2.3 Energy Weighted Local Space Domain Distance (LSDDe) algorithm

We describe an energy weighted modification for calculating the DPD test value \(\chi(t, f)\). In this equation, we weight the DPD test value in Eqn. (6) with its corresponding signal energy. The energy weighted DPD test value is then:

\[
\chi(t, f) = \max_{l} \{ S_l(t, f) \cdot MED[|x_1(t, f)|^2, |x_2(t, f)|^2, \ldots, |x_M(t, f)|^2] \} ,
\]

(7)

where \(MED\) is the median operator. Together, Eqns. (5) and (7) define a (joint) LSDDe DOA/DPD algorithm.

### 3 PROPOSED DOA ESTIMATION ALGORITHM

We propose a new DOA estimation algorithm which was the outcome of investigating the performance of the LSDD algorithm with the EasyCom dataset. It is clear from Eqns. (2), (4) and (5) that the LSDD algorithm does not use any information about the behavior of \(S(t, f)\) with respect to \(\Theta_l\). As such information may be useful, it is proposed to incorporate this information using a correlation process, as follows. For each frequency \(f\) we define an ideal two-dimensional spectrum represented by matrix \(W\), whose elements, \(W_{lh} \equiv W(\Theta_l, \Theta_h)\), represent the similarity between the \(lh\)th steering vector \(v(f, \Theta_l)\) and the \(lh\)th steering vector \(v(f, \Theta_h)\) defined as

\[
W(\Theta_l, \Theta_h) = d(v(f, \Theta_l), v(f, \Theta_h))
= d([v_1(f, \Theta_l), \ldots, v_M(f, \Theta_l)]^T, [v_1(f, \Theta_h), \ldots, v_M(f, \Theta_h)]^T), \forall l, h \in \{1, 2, \ldots, L\} ,
\]

(8)

where the measure \(d\) is defined in (4). Now, the similarity between each column vector in \(W\) and the spectrum \(S(t, f)\) is computed, which provides the following indication. Suppose that \(\Theta_h\) is a DOA of an actual source. Then, we expect the \(lh\)th column of \(W\) to be similar to \(S(t, f)\). This, in effect, defines the new directivity based Space Domain Distance (dSDD) DOA estimation algorithm. The corresponding DOA estimate for bin \((t, f)\) is now computed using

\[
\hat{\theta}(t, f) = \arg \max_h \{ d(S(t, f), W_h) \} ,
\]

(9)

where \(W_h\) denotes the \(lh\)th column in \(W\). We follow Eqn. (6) and define a corresponding DPD test measure as

\[
\chi(t, f) = \max_h \{ d(S(t, f), W_h) \} .
\]

(10)

Together, Eqns. (9) and (10) define a (joint) dSDD DOA/DPD algorithm. It should be noted that under ideal conditions where signal \(x\) in Eqn. (1) is composed of a single plane wave, the two algorithms, LSDD and dSDD, should provide the same estimate as they both rely on the same set of steering vectors. However, the motivation for proposing dSDD is the expected robustness against potential noise and reverberation due to the comparison of entire functions, or vectors. This is in contrast to the LSDD where DOA estimates are based on looking for a peak in a function.

As in the case of the LSDD algorithm, we describe an energy weighted dSDD algorithm (dSDDDe), in which, for each bin \((t, f)\), we weight the dSDD DPD test measure with the corresponding signal energy. The energy weighted DPD test value is therefore given by

\[
\chi(t, f) = \max_h \{ d(S(t, f), W_h) \} \cdot MED[|x_1(t, f)|^2, |x_2(t, f)|^2, \ldots, |x_M(t, f)|^2] .
\]

(11)
4 EXPERIMENTS
This section presents an experimental study that aimed to investigate the performance of the LSDD and the dSDD algorithms with the EasyComm dataset. First, the experimental setup and methodology are presented. This is then followed by the evaluation of the results.

4.1 Set-up
The experiments described in this article were performed on the EasyCom dataset [4]. This dataset was designed with the aim of analyzing the cocktail party effect with audio signals captured by augmented reality (AR) glasses equipped with an egocentric six-channel microphone array. Figure 1 shows a schematic drawing of the glasses with locations of the microphones [4].

The dataset contains recordings of natural conversations in a noisy restaurant environment. Participants were equipped with close-talk microphones, a camera and tracking markers. They were asked to engage in conversations during several tasks, including introductions, ordering food, solving puzzles, playing games and reading sentences. The recordings also contain an egocentric video viewpoint of the participants. The pose (position and rotation) of every participant was also recorded. The dataset was additionally labelled with annotators of voice activity.

Figure 1. Illustration of the AR glasses with locations of microphones [4]. Four of the microphones are fixed rigidly to the glasses and two of the microphones are placed in the user’s ears.

The signals recorded by the microphones were sampled at a rate of 48 kHz. The recorded data was transformed into the STFT domain using a 1024 samples ($\approx 20$ msec) Hann window with an overlap of 512 samples. The microphone signals in the STFT domain were employed as an input to the algorithms under study.

4.2 Methodology
Evaluation of the DOA/DPD algorithms incorporated a direction search with a resolution of $6^\circ$, which was limited to the horizontal plane. The ground truth azimuthal DOA ($\Psi_k$) was obtained from the EasyCom dataset as a function of time. Altogether, a series of three experiments was performed with the EasyCom dataset. The first experiment measured the effective frequency range of the array $[f_{\text{low}}, f_{\text{high}}]$, and the second and third experiments investigated the effect of frequency smoothing and the length of the time interval $\Delta T$ on the performance.

DOA estimation performance was evaluated as follows. For each $(t, f)$ bin, the absolute error,

$$\varepsilon(t, f) = |\Psi(t) - \hat{\theta}(t, f)|,$$

between the true DOA $\Psi(t)$ and the estimated DOA $\hat{\theta}(t, f)$ was computed. Note that Eqn. (12) assumes that both $\Psi(t)$ and $\hat{\theta}(t, f)$ are measured with respect to the same axis. In practice, in this dataset, $\Psi(t)$ is measured with respect to an axis defined relative to the room, while $\hat{\theta}(t, f)$ is measured with respect to the orientation of the glasses. Thus, before calculating $\varepsilon(t, f)$, $\hat{\theta}(t, f)$ is transformed to the fixed axis of the room by incorporating head tracking information. In addition, each bin was labeled a "hit" if $\varepsilon(t, f) \leq 10^\circ$; otherwise it was labeled a "miss".

The bins $(t, f)$ were divided into time blocks of length $\Delta T$. For each block, only bins which satisfied the following four conditions were regarded as valid and were used for DOA estimation:

1. The frequency $f$ lies within the effective operating frequency range: $f \in [f_{\text{low}}, f_{\text{high}}]$.

2. The time $t$ lies within the selected block. Denoting $T$ as the middle time of the block, then $t \in [T - \Delta T/2, T + \Delta T/2]$ belongs to block $T$. 
3. Voice activity was detected at time $t$.

4. The DPD test value $\chi(t,f)$ exceeds a threshold $\lambda$. For the purpose of this study, the threshold was determined from the percentage $p$ of bins which satisfy the first three conditions with the highest value $\chi$. Note that $\lambda$ is computed independently for each block $T$.

The mean value of $\varepsilon(t,f)$ was computed over valid $(t,f)$ bins, i.e. bins which satisfy the above four conditions. By definition, this is the mean absolute DOA estimation error $E(p,T,\Delta T)$ for the block $T$. Similarly, the mean hit ratio $H(p,T,\Delta T)$ was computed on valid $(t,f)$ bins by dividing the number of valid bins labelled "hit" by the total number of valid bins. Finally, the mean absolute error, $\bar{E}(p,\Delta T)$, and the mean hit ratio, $\bar{H}(p,\Delta T)$, for the entire experiment was computed by averaging $E(p,T,\Delta T)$ and $H(p,T,\Delta T)$ over all blocks $T$.

4.3 Effective operating frequency band

The EasyCom dataset involves speech sound which naturally limits the frequency range of interest [8]. This frequency band is reduced in practice by aliasing effects which arise from the microphone array. For a specific steering vector $v(f,\Theta_h)$ (corresponding to a frequency $f$ and direction $\Theta_h$), the similarity between $v(f,\Theta_h)$ and the set of steering vectors $v(f,\Theta_l), l \in \{1,2,\ldots,L\}$ was computed using Eqn. (4). This was repeated for all frequencies $f$ leading to the following measure:

$$\Lambda(f,\Theta_l) = d(v(f,\Theta_h),v(f,\Theta_l))$$

Figure 2 shows $\Lambda$ for $\Theta_h = 0^\circ$. Visual inspection shows that the preferred frequency band is about $1100-2000$ Hz, where at lower frequencies the directivity may be too wide, while at higher frequencies significant side lobes may degrade spatial processing. While this is a relatively narrow band of frequencies, in this work it led to the best performance. Extending the range of operation for both lower and higher frequencies is proposed for future work.

4.4 DOA error and frequency smoothing

The spectrum $S(t,f)$ plays a key role in the DOA/DPD algorithms. In particular, the authors in [1] have shown that it is beneficial to smooth $S(t,f)$ in frequency. We investigated the effect of smoothing $S(t,f)$ over frequency using a moving average filter of length $(2R+1)$. Let $\tilde{S}(t,f) = [\tilde{S}_1(t,f),\tilde{S}_2(t,f),\ldots,\tilde{S}_L(t,f)]^T$ denote the smoothed spectrum, which is computed by

$$\tilde{S}_l(t,f) = \sum_{r=-R}^{R} S_l(t,f + r\Delta f) / (2R+1) ,$$

Figure 2. The similarity matrix $\Lambda$ for $\Theta_h = 0^\circ$.  

4.3 Effective operating frequency band

The EasyCom dataset involves speech sound which naturally limits the frequency range of interest [8]. This frequency band is reduced in practice by aliasing effects which arise from the microphone array. For a specific steering vector $v(f,\Theta_h)$ (corresponding to a frequency $f$ and direction $\Theta_h$), the similarity between $v(f,\Theta_h)$ and the set of steering vectors $v(f,\Theta_l), l \in \{1,2,\ldots,L\}$ was computed using Eqn. (4). This was repeated for all frequencies $f$ leading to the following measure:

$$\Lambda(f,\Theta_l) = d(v(f,\Theta_h),v(f,\Theta_l))$$

Figure 2 shows $\Lambda$ for $\Theta_h = 0^\circ$. Visual inspection shows that the preferred frequency band is about $1100-2000$ Hz, where at lower frequencies the directivity may be too wide, while at higher frequencies significant side lobes may degrade spatial processing. While this is a relatively narrow band of frequencies, in this work it led to the best performance. Extending the range of operation for both lower and higher frequencies is proposed for future work.

4.4 DOA error and frequency smoothing

The spectrum $S(t,f)$ plays a key role in the DOA/DPD algorithms. In particular, the authors in [1] have shown that it is beneficial to smooth $S(t,f)$ in frequency. We investigated the effect of smoothing $S(t,f)$ over frequency using a moving average filter of length $(2R+1)$. Let $\tilde{S}(t,f) = [\tilde{S}_1(t,f),\tilde{S}_2(t,f),\ldots,\tilde{S}_L(t,f)]^T$ denote the smoothed spectrum, which is computed by

$$\tilde{S}_l(t,f) = \sum_{r=-R}^{R} S_l(t,f + r\Delta f) / (2R+1) ,$$

Figure 2. The similarity matrix $\Lambda$ for $\Theta_h = 0^\circ$.  

4.3 Effective operating frequency band

The EasyCom dataset involves speech sound which naturally limits the frequency range of interest [8]. This frequency band is reduced in practice by aliasing effects which arise from the microphone array. For a specific steering vector $v(f,\Theta_h)$ (corresponding to a frequency $f$ and direction $\Theta_h$), the similarity between $v(f,\Theta_h)$ and the set of steering vectors $v(f,\Theta_l), l \in \{1,2,\ldots,L\}$ was computed using Eqn. (4). This was repeated for all frequencies $f$ leading to the following measure:

$$\Lambda(f,\Theta_l) = d(v(f,\Theta_h),v(f,\Theta_l))$$

Figure 2 shows $\Lambda$ for $\Theta_h = 0^\circ$. Visual inspection shows that the preferred frequency band is about $1100-2000$ Hz, where at lower frequencies the directivity may be too wide, while at higher frequencies significant side lobes may degrade spatial processing. While this is a relatively narrow band of frequencies, in this work it led to the best performance. Extending the range of operation for both lower and higher frequencies is proposed for future work.

4.4 DOA error and frequency smoothing

The spectrum $S(t,f)$ plays a key role in the DOA/DPD algorithms. In particular, the authors in [1] have shown that it is beneficial to smooth $S(t,f)$ in frequency. We investigated the effect of smoothing $S(t,f)$ over frequency using a moving average filter of length $(2R+1)$. Let $\tilde{S}(t,f) = [\tilde{S}_1(t,f),\tilde{S}_2(t,f),\ldots,\tilde{S}_L(t,f)]^T$ denote the smoothed spectrum, which is computed by

$$\tilde{S}_l(t,f) = \sum_{r=-R}^{R} S_l(t,f + r\Delta f) / (2R+1) ,$$
where $\Delta f$ is the STFT frequency bin width and $S_l(t, f)$ is defined in Eqn. (2).

DOA estimation experiments were carried out on several 1-minute segments extracted from the EasyCom dataset. Altogether, $S_l(t, f)$ was frequency smoothed using (i) a 9-element filter, (ii) a 3-element filter and (iii) no smoothing. In this experiment, $\Delta T$ was fixed to 200 msec. The corresponding results are shown in Fig. 3.

Figure 3 shows that best overall performance, in terms of both $\bar{E}(p, \Delta T)$ and $\bar{H}(p, \Delta T)$, for the original LSDD algorithm as described in Sec. 2.2 is obtained with the 9-element filter. On the other hand, for both LSDDe as in Sect. 2.3 and dSDDe as in Sect. 3, the best overall performance is obtained when no smoothing is used. Moreover, for both algorithms the change in performance with respect to smoothing is relatively small.

Finally, we compare the results obtained with the variation of the LSDD algorithm (i.e with a 9-element smoothing filter) and the best variations of the LSDDe and dSDDe algorithms (i.e. no smoothing). At low percentages $p$ the mean absolute error $\bar{E}(p, \Delta T)$ obtained with the dSDDe algorithm is approximately $9^\circ$ lower than that obtained with the LSDD algorithm. Similarly, the mean hit ratio obtained with the dSDDe algorithm is approximately 5% higher than that obtained with the LSDD algorithm.

4.5 DOA error and time interval $\Delta T$

The choice of time interval $\Delta T$ may be directly related to the dynamic nature of the dataset. In general, we would like to use a value of $\Delta T$ which is small enough such that the environment can be considered spatially stationary within the interval. However, if the chosen value of $\Delta T$ is too small, DOA performance may degrade. In this experiment, three different values of $\Delta T$ were investigated: (a) $\Delta T = 200$ msec; (b) $\Delta T = 300$ msec; and (c) $\Delta T = 500$ msec. The DOA/DPD algorithms used the best frequency smoothing as presented in the previous section. Figure 4 shows $\bar{E}(p, \Delta T)$ averaged over ten 1-minute segments for three $\Delta T$ values. The figure shows a consistent and significant improvement in performance as $\Delta T$ increases. At $\Delta T = 500$ msec and $p = 1\%$, the dSDDe algorithm gave a mean absolute DOA error $\bar{E} \approx 20^\circ$. Similar to Fig. 3, the dSDDe algorithm achieved the best performance, and the LSDD algorithm the worst.
Figure 4. $E(p, \Delta T)$ as a function of percentage $p$ for the LSDD, LSDDe and dSDDe algorithms for three different $\Delta T$ values, averaged over ten 1-minute segments. (a) $\Delta T = 200$ msec; (b) $\Delta T = 300$ msec; (c) $\Delta T = 500$ msec. Best frequency smoothing was applied as in Sec. 4.4.

4.6 Summary

Figure 5 illustrates the performance of the dSDDe algorithm together with a timeline of several experimental data values, for an example taken from the EasyCom dataset. In this example there are two active speakers. The time interval $\Delta T$ used in the figure is 200 msec.

Figure 5. Ground-truth and estimated DOAs for a 1-minute long segment taken from the EasyCom dataset, including two speakers. Ground-truth directions are presented using dashed thick lines. The orientation of the microphone array is marked using a thick red line. The DOA estimates $\hat{\theta}(t, f)$ provided by the dSDDe algorithm with $p = 1\%$ and $\Delta T = 200$ msec are shown as colored circles.

The figure illustrates the dynamic nature of the environment. While many of the DOA estimates fall close to active speakers, some fall very far away from the true DOA. There is also a clear bias in the estimation toward a preferred direction. These findings raise the need for further research to better understand the dataset and the algorithms, and propose improved solutions.
5 CONCLUSIONS
This work presented three experiments for DOA estimation based on the EasyCom dataset. These preliminary experiments showed that:

1. The baseline performance using the original LSDD algorithm showed limited performance.
2. The dSDD algorithm improved performance by incorporating more detailed spatial information.
3. Energy weighting DPD test values was found to be useful.
4. Both the LSDDe and the dSDDe DOA algorithms do not require frequency smoothing.

ACKNOWLEDGEMENTS
This research was supported by Reality Labs at Meta.

REFERENCES
3D multi-source localization by joint ESPRIT with multiple spherical arrays

Jung-Woo CHOI(1) and Franz ZOTTER(2)

(1) School of Electrical Engineering, KAIST, South Korea, jwoo@kaist.ac.kr
(2) Institute of Electronic Music and Acoustics, University of Music and Performing Arts, Graz, Austria, zotter@iem.at

ABSTRACT
Simultaneous localization of multiple sound sources in 3D space by multiple compact microphone arrays has become an important problem, e.g., in the processing of 6 degrees-of-freedom audio recordings. Most techniques rely on detecting directions-of-arrival (DoA) independently per array to stay robust against unknown signal types, source directivity, or slight synchronization offsets between the arrays. However, the subsequent source localization requires to correctly assign every DoA observed in an array to every of the detected sources to avoid misdetections, so-called ghost sources. We present the Multi-Array EB-ESPRIT problem as a parametric approach with inherent DOA-to-source assignment. It is based on multiple compact spherical arrays and DoA detection but additionally considers both auto- and cross-covariances between arrays, which together yield a total signal subspace. This signal subspace links all the array-specific DoA detection problems to an unknown but common mixture of the independent sources, without introducing sensitivity to directivity- or synchronization-related gain differences between the arrays. We propose an algorithm using generalized joint Schur decomposition and geometric projection to solve the problem. Numerical experiments with randomly distributed sources indicate that the algorithm largely improves accuracy and robustness, even in cases when the assumed number of sources over-estimates the true number of sources.

Keywords: 3D source localization, Spherical microphone arrays, ESPRIT

1 INTRODUCTION
Reproduction of a real sound field in a virtual space is one of the key features of virtual reality (VR) and metaverse audio. A listener in a virtual space can navigate through different positions, and the corresponding real sound field at the changing listener position needs to be traced and reproduced, as stressed by 6 degree-of-freedom (6 DoF) audio encoding and rendering papers (1–5). In reality, sound fields can only be captured at a finite number of positions, and one has to interpolate or extrapolate sound fields from limited measurements. Such tasks are not easy especially when multiple sound sources are activated at the same time. For efficient encoding of a sound field, 3D positions of multiple sources are imperative prior information.

Currently, recording VR audio contents by a spherical microphone array (SMA) appears appealing, since SMAs allow us to seamlessly track sources by their directions-of-arrival (DoAs) via numerous existing beamforming techniques such as Eigenbeam ESPRIT (EB-ESPRIT) (6–12), EB-MUSIC (13, 14), and EB-MVDR (15). However, for 6 DoF applications, 3D positions of sound sources need to be traced rather than DoAs. For this reason, recent studies (1, 4, 16–20) focused on the 3D sound source localization (SSL) of multiple sources using multiple SMAs distributed in space.

Most SSL techniques employ a post-pairing strategy that finds directions of arrivals (DoAs) or time differences of arrivals (TDoAs) from individual microphone arrays and then pairs them using an association algorithm or DoA histogram mapped to grid points (1, 20). For simultaneous detection of multiple sources, however, post-pairing is ambiguous and entails an association problem. Whenever DoAs or TDoAs are associated but actually belong to different sources, it causes the erroneous display of ghost sources. Moreover, histogram-based approaches require a computationally expensive search process over pre-defined grid points.

This paper introduces a parametric 3D SSL technique called multi-array EB-ESPRIT (MEB-ESPRIT (21))
that can completely avoid association ambiguities. MEB-ESPRIT utilizes recordings from multiple SMAs and directly estimates 3D source positions based on the parametric DoA estimation technique: vector-based EB-ESPRIT (VEB-ESPRIT \(^{10, 12, 22, 23}\)). Unlike other DoA estimation techniques or SSL techniques based on DoA estimation, MEB-ESPRIT exhibits the following distinctive features:

• The proposed technique is a parametric approach that directly estimates 3D source positions without a grid search over a room.

• The proposed technique makes use of relations between different SMA recordings. Therefore, it can distinguish inline sources (Fig. 1(b, c)) positioned in the same direction from some SMAs.

• The technique is a subspace-based technique that performs subspace extraction before estimating source positions. This characteristic is in line with the conventional subspace-based technique such as ESPRIT and MUSIC. Still, the proposed technique overcomes the limitation of these techniques that the number of sources should be known or estimated first. We show that the proposed technique can accurately estimate source locations when the assumed number of sources is greater than the actual number of sources.

These advances in the proposed technique are accomplished by the generalized joint Schur decomposition (GJSD) algorithm combined with geometric projection (GP) in an iteration process. In the following sections, we define the 3D SSL problem and explain how we can detect multiple source positions by extending the concept of the conventional VEB-ESPRIT.

2 3D SOURCE LOCALIZATION PROBLEM

2.1 Problem statement

Consider \( Q \) sound sources positioned at \( r_q (q = 1, \cdots, Q) \) and \( L \) SMAs whose centers are at \( r_\ell (\ell = 1, \cdots, L) \), as illustrated in Fig. 1(a). When viewed from each array center, the position of each source in local coordinates is given by \( r_{q\ell} = r_q - r_\ell \). The goal of 3D source localization is to find positions of multiple sources \( r_q \) from the recording and known positions \( r_\ell \) of multiple SMAs.

SMAs recordings are usually processed by a spherical harmonics encoding system, which accomplishes the spherical Fourier transform \(^{24}\) of the microphone array signals and compensates the array’s acoustic surface impedance. This process yields spherical harmonic coefficients signals of a plane-wave decomposition up to a limited harmonic order \( N \). As a result, the sound field from the \( q \)th point source measured by the array \( \ell \) should transform to the spherical harmonic coefficients \( a_{n,m,\ell} \) \(^{21}\)

\[
a_{n,m,\ell}(\omega) = Y_{nm}(e_{q\ell}(\omega)) g_{q\ell}(\omega) s_q(\omega). \tag{1}
\]

where \( n \) and \( m \) are orders and degrees of the real-valued spherical harmonics \( Y_{nm} \), respectively. Here, \( e_{q\ell} = r_{q\ell}/\|r_{q\ell}\| = (e_{q\ell x}, e_{q\ell y}, e_{q\ell z}) \) is the unit-norm direction vector heading from the SMA center towards the source,
\( s_q(\omega) \) is the complex amplitude of the source signal at the frequency \( \omega \), and \( g_q(\omega) = b(-e_q)\exp(ik|r_q|)/|r_q| \) includes the source directivity \( b(-e_q) \), propagation delay and decay in proportion to the distance \( (k = \omega/c: \text{wavenumber}, c: \text{speed of sound}) \). The real-valued spherical harmonics are defined as

\[
Y_{nm}(\phi_q, \theta_q) = J_n^m |r_q|^m \sin^m |\phi_q|, \quad \text{for } m \geq 0, \quad J_n^m = (-1)^m \sqrt{\frac{(2n+1)(2-\delta_{nm})}{4\pi}} \frac{(n-|m|)!}{(|n+|m|)!},
\]

for source positions in the local azimuth \( \phi_q \) and zenith \( \theta_q \). The Kronecker delta is \( \delta_{nm} = 1 \) only for \( m = n \) otherwise zero, and the associated Legendre functions \( P_n^m \) follows the definition of (24). For multiple sources, Eq. (1) can be rewritten in a vector form as

\[
a_q = Y_q G_q s_q.
\]

Here, \( Y_q \in \mathbb{R}^{H \times Q} \) is the spherical harmonics (SH) matrix with \( H = (N+1)^2 \) coefficients for \( Q \) sources, the diagonal matrix \( G_q \in \mathbb{C}^{Q \times Q} \) contains source- and array-specific gains \( g_q \), and \( s_q \in \mathbb{C}^{Q \times 1} \) is a column vector of all source signals \( s_q \). In the definitions of \( s_q \) and \( G_q \) above, the dependency on the frequency \( \omega \) is implicit.

Equations (1) and (3) imply that the SH coefficients of a sound source are associated with the SHs evaluated at the direction \( e_q \). Therefore, we can extract the DoA \( e_q \) from the observed SH coefficients. In particular, the real-valued VEB-ESPRIT (22, 23) utilizes three recurrence relations of SHs to extract DoAs, which can be reduced to the following generalized eigenvalue decomposition (GEVD) problems:

\[
MY, \Phi_{\ell} = D_{\ell} Y_{\ell}, \quad MY, \Phi_{\ell} = D_{\ell} Y_{\ell}, \quad MY, \Phi_{\ell} = D_{\ell} Y_{\ell},
\]

for an order-reducing matrix \( M = [1 0] \in \mathbb{R}^{H \times N^2} \) and recurrence coefficient matrices \( D_{\ell} \) (for definitions of coefficients, see (23)). The subscript \( w \) represents three axis directions in Cartesian coordinates, so we get three GEVD matrix equations. The three DoA parameters \( e_q = (e_{qx}, e_{qy}, e_{qz}) \) are stored in the diagonals of the diagonal matrices \( \Phi_q, \Phi_{\ell}, \Phi_{\ell} \), respectively. Therefore, one can estimate DoA parameters by finding the generalized eigenvalues from Eq. (4). However, the SH matrix \( Y_{\ell} \) is not directly traceable from the measured SH coefficients \( a_q \) of Eq. (3), but it is structurally implied in the signal subspace of the observed SH coefficients. The signal subspace is estimated from the best rank-Q approximation of the covariance matrix

\[
R_{\ell} = E\{a_q a_q^H\} = Y_q G_q E\{s s^H\} G_q^H Y_q^T = Y_q G_q S S^H G_q^H Y_q^T 
\approx U_{\ell} \Lambda_{\ell} U_{\ell}^H.
\]

The approximation collects \( Q \) eigenvectors \( U_{\ell} \in \mathbb{C}^{H \times Q} \) belonging to the largest eigenvalues in a diagonal matrix \( \Lambda_{\ell} \in \mathbb{C}^{Q \times Q} \). Structurally, Eq. (5) reveals a relation of the SH matrix \( Y_{\ell} \) to \( U_{\ell} \) by a nonsingular transform \( X_{\ell} \),

\[
Y_{\ell} = U_{\ell} X_{\ell},
\]

whenever the matrices \( G_q S \) and \( \Lambda_{\ell} \) are nonsingular. This allows rewriting the VEB-ESPRIT equations as

\[
MU_{\ell} \Psi_{\ell} = D_{\ell} U_{\ell}, \quad \text{where } \Psi_{\ell} = X_{\ell} \Phi_{\ell} X_{\ell}^{-1},
\]

which in 3D space defines a system of 3 VEB-ESPRIT matrix equations for the SMA. The diagonal entries of \( \Phi_{\ell} \) can be found from the eigenvalues of \( \Psi_{\ell} \). In practice, \( \Psi_{\ell} \) is calculated by taking the left pseudoinverse of \( MU_{\ell} \). \( \Psi_{\ell} = (MU_{\ell})^+ D_{\ell} U_{\ell} \), and the rank of \( MU_{\ell} \) limits the number of identifiable sources as \( Q \leq N^2 \).

### 2.2 MEB-ESPRIT

For 3D SSL based on the DoA estimation technique, MEB-ESPRIT considers cross-covariances between different SMAs. For example, the total covariance between two arrays \( \ell, \kappa \) is and its rank-Q subspace approximation are written as

\[
R = E \left[ \begin{bmatrix} a_\ell & a_\kappa \end{bmatrix} \right] = \left[ \begin{array}{cc} R_{\ell\ell} & R_{\ell\kappa} \\ R_{\kappa\ell} & R_{\kappa\kappa} \end{array} \right] = \left[ \begin{array}{cc} Y_{\ell} G_{\ell} & Y_{\ell} G_{\kappa} \end{array} \right] S S^H \left[ \begin{array}{cc} G_{\ell}^H Y_{\ell}^H & G_{\kappa}^H Y_{\kappa}^H \end{array} \right] 
\approx U_{\ell} U_{\kappa} \Lambda_{\ell\kappa} U_{\ell}^H U_{\kappa}^H
\]

(8)
Here, \( U_\ell \) and \( U_\kappa \) are obtained by partitioning the total signal subspace eigenvectors \( U \). These partitioned matrices are different from \( U_\ell \) derived from a single SMA (Eq. (5)), e.g. their columns are not orthogonal. Nevertheless, comparing lines in Eq. (8) shows that the partitioned signal subspace eigenvectors are still related to the SH matrices multiplied by the respective array-specific gains \( G_\ell \) or \( G_\kappa \) via a transformation matrix \( X \), which is now common to the SMAs, \( Y_\ell G_\ell = U_\ell X \). \( Y_\kappa G_\kappa = U_\kappa X \), or
\[
Y_\ell = U_\ell X G_\ell^{-1}, \quad Y_\kappa = U_\kappa X G_\kappa^{-1}. \tag{9}
\]

Since the relative orders of sources in two SMAs are maintained by the common transform, and as after inserting Eq. (9) into Eq. (4), \( M U_\ell X G_\ell^{-1} \Phi_{\ell w} = D_\ell U_\ell X G_\ell^{-1} \), the diagonal gains \( G_\ell^{-1} \) commute with the diagonal eigenvalues \( \Phi_{\ell w} \) thus cancel, \( M U_\ell X \Phi_{\ell w} = D_\ell U_\ell X \), we avoid the association problem. The VEB-ESPRIT equations of Eq. (7) can accordingly be rewritten to the partition \( U_\ell \) with the common transform \( X \)
\[
M U_\ell \Psi_{\ell w} = D_\ell U_\ell, \quad \text{where } \Psi_{\ell w} = X \Phi_{\ell w} X^{-1}. \tag{10}
\]

For \( L \) SMAs in 3D space, this defines common eigenvectors \( X \) shared by a total of \( 3L \) MEB-ESPRIT matrix equations, \( L \) times as many as for VEB-ESPRIT above. With the left pseudo inverse for \( M U_\ell \), \( 3L \) matrices \( \Psi_{\ell w} = (M U_\ell)^{\dagger} D_\ell U_\ell \) can be calculated and jointly diagonalized by \( X \). The remaining task is to find the corresponding joint eigenvectors \( X \).

### 2.3 Joint diagonalization of multiple matrices

Simultaneous diagonalization of multiple matrices \( \Psi_{\ell w} \) using a common transform matrix has been tackled in various ways. Joint eigenvalue decomposition (JEVD) techniques \((25, 26)\) can be utilized for this purpose, but there are several issues that cannot be handled by JEVD.

- **Rank deficiencies in deriving \( \Psi_{\ell w} \)**: Most JEVD techniques assume that all matrices are non-singular. However, in Eq. (10), the pseudo-inversion of \( M U_\ell \) required to compute \( \Psi_{\ell w} \) can be singular in several situations. For sources lying on the same line from the array \( \ell \) (Fig. 1(b)), \( Y_\ell \) contains duplicate columns, so that the column rank of the related matrix \( U_\ell \) becomes deficient. Such inline sources are even more problematic when lying on a line between two arrays, affecting both of them (Fig. 1(c)). Rank deficiency also occurs in \( U_\ell \) if a source has a directional radiation pattern with a null directed towards the array \( \ell \).

- **Misdetection of the signal subspace dimension \( \tilde{Q} \)**: The number of sources should be known or estimated to determine the signal subspace \( U \), but the estimation can easily fail in complex acoustic scenes. For instance, the overestimation of \( Q \) (\( \tilde{Q} > Q \)) inevitably pollutes the partitioned eigenvectors \( U_\ell \) with different noise subspace eigenvectors. This pollution varies among SMAs and dissolves their coupling so that there is no common transform \( X \) that accurately diagonalizes the given ESPRIT matrices \( \Psi_{\ell w} \) jointly. If the assumed number \( \tilde{Q} \) is less than the actual number of sources (\( \tilde{Q} < Q \)), then the signal subspace contains incomplete and inconsistent information and it becomes impossible to extract accurate DoAs.

- **Room reflections**: When there are strong early reflections in a room, the identified signal subspace may include sound fields from several image sources. Therefore, there will be more sources that individually satisfy the recurrence relations than the assumed number of independent sources.

To resolve these issues, we propose an iterative approximate joint eigenvalue search that can be made robust. To this end, we first avoid pseudo inversion \((M U_\ell)^{\dagger}\) by casting the MEB-ESPRIT matrices of Eq. (10) into an equivalent problem for generalized joint eigenvalue decomposition. After right-multiplying Eq. (10) with the joint eigenvectors \( X \), we get a generalized joint eigenvalue decomposition problem with the matrix sets \( A_{\ell w} = D_\ell U_\ell \) and \( B_\ell = M U_\ell \) of the indices \( \ell, w \) with the diagonal eigenvalue matrices \( \Phi_{\ell w} \) for the joint right eigenvectors \( X \)
\[
A_{\ell w} X = B_\ell X \Phi_{\ell w}. \tag{11}
\]

Instead of diagonalizing both \( A_{\ell w} X \) and \( B_\ell X \) by left multiplication with an array-specific left eigenvalue matrix to find \( \Phi_{\ell w} \), the generalized joint Schur decomposition (GJSD) identifies the diagonals of \( \Phi_{\ell w} \) by using numerically robust unitary matrices \( Q_\ell \in \mathbb{C}^{H_\ell \times H_\ell} \) and \( Z \in \mathbb{C}^{Q \times Q} \) that contain the orthogonal, generalized left or right
Schur vectors \((25, 27)\), respectively, reducing the matrices to upper-triangular shaped ones, \(V_{\ell w}\) and \(W_{\ell}\),

\[
Q_{\ell}^H A_{\ell w} Z = V_{\ell w}, \quad Q_{\ell}^H B_{\ell} Z = W_{\ell}.
\]

(12)

The ratio of their diagonals \(\frac{\text{diag}(\{V_{\ell w}\})}{\text{diag}(\{V_{\ell}\})}\) yield the desired eigenvalues. The GJSD algorithm proposed here applies the algorithm \((27)\) for the given, nested sets of 3L matrices \(A_{\ell w}\) and \(B_{\ell}\). Whenever no accurate diagonalization exits, \(V_{\ell w}\) and \(W_{\ell}\) won’t be perfectly upper-triangular, but will be the best approximation through the successive reduction of the lower-triangular part of each column corresponding to each source.

A brief outline of the proposed GJSD-GP algorithm is illustrated in Fig. 2, whose details are as follows:

1. Initialization for the \(q\)th source \((q = [\hat{Q}, \cdots, 1])\)

A unit-length right eigenvector \(x\) is randomly initialized \((x = \text{randn}(q, 1), x = x/\|x\|)\)

2. Iteration cycle (i-iv) for the best \(y_{\ell w}\), and \(x\)

The right eigenvector \(x\) and eigenvalue \(e_{\ell w}\) are updated to robustly minimize the JEVD error

\[
\Delta_{\ell w} = (A_{\ell w} - B_{\ell} e_{\ell w}) x = A_{\ell w} x - y_{\ell w} e_{\ell w}
\]

(13)
in a least-square sense. To this end, left eigenvectors \(y_{\ell}\) and eigenvalues \(e_{\ell w}\) are updated by power iteration steps (i) and (ii), cf. \((21)\). The resulting DoAs/eigenvalues \(e_{\ell w}\) are then refined by the geometric projection algorithm step (iii) that forces the estimated DoAs at different SMA locations \(r_{\ell}\) to intersect at a single position \(\hat{r}\) in 3D space. This is accomplished by the geometric source localization

\[
\hat{r} = \left(\sum \alpha_i P_{ei}\right)^{-1} \left(\sum \alpha_i P_{ei} r_i\right), \quad \text{where } P_{ei} = I - e_i e_i^T.
\]

(14)

Here, the array dependent weight \(\alpha_i = \left(\|y_i\|/\sqrt{\sum ||y_i||^2}\right)^{1/3}\) is employed to suppress the contribution of SMAs recording inline sources or potential nulls of a directional source. The estimated position \(\hat{r}\) is used to get geometrically more consistent DoAs/eigenvalues \(e_{\ell w}\) estimates via projections applied to the error \(\Delta_{\ell w}\) before minimizing its squares via an inverse iteration step followed by normalization to unit norm (iv), see \((21)\). The steps i-iv keep repeating until the updated vector converges \((1 - \|x^H x_{\text{old}}\| < \text{tol})\) or a maximum number of iteration cycles is reached, using the constants tol, maxit.

3. Orthogonal reduction using Householder reflection

We utilize the Householder reflections to derive left and right reduction matrices \(\hat{Q}_{\ell}\) and \(\hat{Z}\) from the estimates \(y_{\ell}\), \(x\) whose task is to suppress the lower-triangular parts in the first column of \(A_{\ell w}\) and \(B_{\ell}\)

\[
\hat{Z} = \begin{bmatrix} \hat{Z}_1 \end{bmatrix}^H = I - 2z z^H/\|z\|^2,
\]

where \(z = \text{msgn}\left(i_1^T x\right) i_1 + x\) for \(i_1 = [1 \ 0 \ \cdots \ 0]^T\),

\[
\hat{Q}_\ell = I - 2q q^H/\|q\|^2,
\]

for \(q = \|y_{\ell}\| / \text{msgn}\left(i_1^T y_{\ell}\right) i_1 + y_{\ell},\)

(15)

for \(\text{msgn}(x) = 1\) for \(\text{Real}\{x\} > 0\), and \(-1\) otherwise. The orthogonal reduction to the top entry in the first column of \(A_{\ell w}\) and \(B_{\ell}\) becomes

\[
B_{\ell}' = \hat{Q}_{\ell}^H B_{\ell} \hat{Z} = \begin{bmatrix} B_{\ell} & b_{\ell}^T \\ e_{\ell} & \hat{B}_{\ell} \end{bmatrix}, \quad A_{\ell w}' = \hat{Q}_{\ell}^H A_{\ell w} \hat{Z} = \begin{bmatrix} A_{\ell w} & a_{\ell w}^T \\ e_{\ell w} & \hat{A}_{\ell w} \end{bmatrix}
\]

(16)

4. Completion of identification of the source \(q\) and proceeding to source \(q - 1\)

The reduction is accepted as a valid upper-triangular reduction set if it satisfies the following condition:

\[
\sqrt{\sum_{\ell w} \|e_{\ell w}\|^2} + \sqrt{\sum_{\ell} \|e_{\ell}\|^2} < \text{tolre} \times \left(\sqrt{\sum_{\ell w} \|A_{\ell w}\|^2} + \sqrt{\sum_{\ell} |B_{\ell}|^2}\right)
\]

(17)

In this case, the eigenvalue for the \(q\)th source is found as \(e_{q,w} = A_{q,w}/B_{q}\). For the next source, reduced matrices \(\hat{A}_{q,w}\), \(\hat{B}_{q}\) are used as \(A_{q,w}, B_{q}\), and new iterations begin from step 1 with \(q \rightarrow q - 1\). Otherwise, the current estimate \(x\) is inaccurate, we discard the attempted reduction and permit random re-initialization of \(x\) by re-entering at step 1 for the \(q\)th source; re-initializations are limited to maxre times per source.
3 EXPERIMENT

For simulations, four SMAs were positioned at vertices of a tetrahedron with edge length $2\sqrt{6}$ m (Fig. 3(a)). Each SMA had 32 omnidirectional microphones positioned at the sensor positions of a commercial device (Eigenmike EM-32™). The radius of the spherical surface was $\rho = 42$ mm, and the surface was assumed to be rigid. The rotational orientations of SMAs were configured in the same direction. The microphone signals were simulated up to order 30 and encoded up to the third order. To analyze signals by the short-time Fourier transform (STFT), the rectangular window of 1024 samples, a hop size of 512 samples, and an FFT size of 2048 samples were used at a sampling rate of 16 kHz. The total covariance matrices $\mathbf{R}$ were then constructed by taking an average over 300 temporal snapshots. The frequency of interest was $k\rho = 2$. For all the simulations listed below, the hyperparameters for the GJSD-GP algorithm were set to maxit $= 100$, tol $= 10^{-7}$, and tolr $= 0.1$, maxre $= 4$.

The microphone self-noises are included in the simulation with a 20 dB signal-to-noise ratio (SNR). The SNR used in this simulation represents the equivalent SNR measured by a microphone on the hypothetical acoustically transparent SMA positioned at 1 m distance from a sound source. For both microphone self noises and source signals, uncorrelated zero-mean white Gaussian random signals with equal variances were used. The candidate source positions were randomly populated from a uniform distribution within the $\pm 3$ m range in the $x, y, z$ direction. Sources separated from one another or the array centers by less than 0.5 m were discarded.
and repopulated. We evaluated the RMSE in distance between the estimated and the simulated source location for various numbers of sources ($Q, \tilde{Q} \in [1, 9]$). For each ($Q, \tilde{Q}$), we randomly selected 100 combinations of $Q$ source positions from the populated candidate positions. Different source signals and noises were then simulated 20 times for each source combination. Consequently, a total of 2000 independent trials were tested for each ($Q, \tilde{Q}$). We used the VEB-ESPRIT with the ad hoc JEVD for each array (12) as the baseline technique, for which DoAs from the same sources were manually associated by searching the ideal pairs giving minimal DoA errors.

The results in Fig. 3(b, c) compare the performance of MEB-ESPRIT with that of the VEB-ESPRIT baseline that we allow to use ideal association. Both methods work relatively well with the exactly matching number of sources ($\tilde{Q} = Q$). Of course, allowing ideal association exaggerates the performance that the baseline would realistically achieve with any practical association algorithm, so its performance is an idealistic benchmark. As the assumed number of sources outnumbers the actual ones ($\tilde{Q} > Q$), also invalid sources or image sources would get detected that should be removed in a post-processing step. For those sourced matching the actual source position, MEB-ESPRIT outperforms the baseline. In this case, some noise-subspace eigenvectors are also included in the identified eigenvectors, but MEB-ESPRIT is capable of suppressing the noise-subspace eigenvectors that do not satisfy a joint decomposition. This trend is more prominent for large $\tilde{Q}$ and small $Q$, showing errors on the scale of a few centimeters (Fig. 3(c)) in comparison to the scale of tens of centimeters observed for the baseline (Fig. 3(b)). Although the performance of both methods drastically decreases for an underestimated number of sources ($\tilde{Q} < Q$) due to the incomplete signal subspace, these cases are not seriously problematic for MEB-ESPRIT because for higher-order SMAs it always appears helpful to choose the maximum number $\tilde{Q} = N^2$ as the assumed number of sources. The robust performance for the overestimated number of sources is a great benefit of MEB-ESPRIT gained from the iterative search combined with the geometric projection algorithm. These results reveal that MEB-ESPRIT can localize multiple sound sources in 3D space without having to rely on DoA pairing algorithms and number of sources estimation strategies, whose adjustment requires to make assumptions that may fail.

ACKNOWLEDGEMENTS
This work was supported by the National Research Council of Science and Technology (NST) funded by the Ministry of Science and ICT (MSIT) of Korea (Grant No. CRC 21011), the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MSIT) (No.1711091575), and the BK21 FOUR program through the National Research Foundation (NRF) funded by the Ministry of Education of Korea.

REFERENCES
Understanding posture during performance and vocalization and its application in multiple Azure Kinects recordings

Arisa KURAMOTO(1), Yasuhiro OIKAWA(2)

(1)Department of Intermedia Art and Science, Waseda University, Japan, akuramoto@fuji.waseda.jp
(2)Department of Intermedia Art and Science, Waseda University, Japan, yoikawa@waseda.jp

ABSTRACT

Ideally, the position and orientation of the vocalist and microphone should be invariant during voice recording in order to keep the conditions unchanged. This is because the transfer function changes with the position and orientation of the vocalist and microphone. A microphone fixed to the head, such as a head-mounted microphone, allows the position and orientation of the vocalist to be invariant, but microphones used for recording are generally large and mounted on microphone stands. If the vocalist’s movements could be completely still, the position and orientation of the vocalist and microphone could be made invariant, but this is impossible due to the structure of the human body. Currently, sound engineers perform sound compensation based on their experience, which takes a lot of time. Therefore, there is a need for a system that automatically compensates for the effects of the vocalist’s body movements. In this study, we developed the system that uses multiple Azure Kinect units to acquire data on the vocalist’s position and orientation, and estimate the directivity of sound source to correct speech based on the acquired data.

Keywords: Recording, Body Tracking, Directivity Estimation

1 INTRODUCTION

Ideally, when recording audio in a studio, the position and orientation of the vocalist and microphone should be invariant in order to keep the conditions unchanged. This is because the transfer function changes with the position and orientation of the vocalist and microphone. If the vocalist’s motion could be completely stationary, it would become an ideal recording environment, but in practice this is impossible [1]. Currently, engineers manually correct this problem based on their experience, but this takes a lot of time, thus automation is desired.

In recent years, motion capture has been used to measure body movements in a variety of fields. There are various types of motion capture systems, such as those in which markers are attached to the body and the position of the markers are tracked by a camera, and those in which human body parts are extracted from images captured by a camera without using markers. Azure Kinect, sold by Microsoft, is relatively inexpensive at approximately 50,000 yen per unit, which can track body movements without using markers. It is equipped with a depth sensor, 7-microphone array, accelerometer, and gyroscope.

In a previous study, a system was developed to acquire the vocalist’s head movements during voice recording using a single Azure Kinect [2]. The developed system was found to be able to track the vocalist’s movements when the vocalist was at least 0.5 m away in front of the Azure Kinect. However, when the vocalist’s body rotates, only one side of the vocalist’s body was visible on the Azure Kinect. When the vocalist was at a distance from the Azure Kinect, correct recognition was not possible. In this study, two Azure Kinects were used simultaneously to achieve more extensive and accurate body tracking compared to the previous system. A system was constructed to simultaneously perform body tracking and voice recording. In addition, directivity estimation was performed from sound recorded by using Azure Kinect.
2 RECORDING SYSTEM OF VOICE AND BODY MOVEMENT

2.1 System overview

Figure 1 shows the system constructed in this study. Audio data acquired by using Azure Kinect is stored via a driver (ASIO4ALL) in a PC, and body tracking data is also stored at the same time. Audio acquired by the recording microphone is stored together with the time code using digital audio workstation (DAW). Both data acquired using the recording microphone and the Azure Kinect could be synchronized with the time code. Two Kinect units were connected to a PC, and the body motion of the vocalizer obtained from each Kinect was integrated to become one body motion in Unity 2019. Figure 2 shows the positional relationship between the two Azure Kinect units and the subject, and the head coordinate system. As a result, we considered that a wider range of body tracking was possible than using a single Azure Kinect.

In this study, it is assumed that each Azure Kinect can capture only one person who is identical in the image; body tracking with the Kinect is based on the position and orientation of each of the 32 joints, including the shoulders and knees. Because there are one subject and two cameras at different positions, the position and orientation values acquired by each Kinect are different. Therefore, they are continuously processed to the integrated values of the joint position and orientation in real time. The spherical linear interpolation method is applied to integrate the joint data acquired by each Kinect into the integrated joint data in the proposed system.

![Figure 1. Recording system of voice and body movement.](image)

2.2 Comparison of body tracking by vocalist position

In order to compare the range of the joint data acquisition between the proposed body tracking system with two Azure Kinect units and the previous system with one Azure Kinect unit, we conducted an experiment in a seminar room 415 of Bldg. 59, Waseda University. The location of Azure Kinect units and the subject in the experiment is shown in Fig. 3. The experiment was conducted by moving the subjects to 14 locations. ♦ is the position of Azure Kinect when measured with two units. ▲ is the position of Azure Kinect in the case of measurement with one unit. The subject was seated on a chair.

At each of the 14 locations, the subject’s body was rotated from side to side, tilted back and forth, etc. It’s rated on a 3-point scale. The ratings were "Tracking is possible," "Tracking may stop," and "Tracking is impossible". Figure 4 shows the evaluation results. It can be confirmed that the proposed system can track a wider area than the previous single Kinect system. Even when two Kinect units are used, there is one position that body tracking is impossible. This is because it is too close distance to Kinect.
Figure 2. Positioning and head coordinate system of two Azure Kinects.

Figure 3. Measurement position.

Figure 4. Differences in tracking area by the previous system and the proposed one. The previous system has one Kinect unit and the proposed system has two units used.
2.3 Comparison of body tracking by vocalist orientation

We compared the accuracy of the acquired body data against the rotation of a subject. Figure 5 shows the positioning of Azure Kinect units and the subject during experiment. One or two Azure Kinect units were placed at a height of 90 cm from the ground. The subject was seated on a chair and rotated his/her body as shown in Fig. 6. The orientation of the subject was measured by the proposed system. Table 1 shows the orientation of the subject measured by the system, root mean square error (RMSE) and standard deviation (SD) when one and two Azure Kinect units were used. When the number of Kinect used increased to two, both RMSE and SD were reduced by half compared to when one Kinect was used. The further out from the frontal plane of Azure Kinect, such as 0° or 180°, the greater the error. This is because Azure Kinect perceives multiple joints as if they are overlapping.

![Figure 5. Positioning during experiment.](image)

![Figure 6. Orientation of vocalist.](image)

<table>
<thead>
<tr>
<th>Number of units</th>
<th>Angle of rotation [°]</th>
<th>RMSE</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 30 60 75 90 105 120 150 180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>23.96 53.13 63.43 69.44 90.00 98.13 102.09 107.35 116.56 30.33 28.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>23.00 47.06 57.37 70.46 90.00 112.21 146.13 153.17 162.77 15.36 14.08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3 DIRECTIVITY ESTIMATION USING AZURE KINECT AND MICROPHONE

3.1 Theory

When the sound intensity of a plane wave in a diffuse sound field is \( I [W/m^2] \) and the speed of sound is \( c [m/s] \), the sound energy density \( E [J/m^3] \) is expressed as

\[
E = \frac{I}{c}.
\]  

(1)

Consider a unit sphere \( S_1 \) centered at a point \( P \) in the interior. The energy density \( E \) at point \( P \) due to plane waves arriving from all directions is

\[
E = \int_{S_1} \frac{I}{c} d\Omega = \frac{4\pi I}{c},
\]  

(2)

and then

\[
I = \frac{cE}{4\pi}.
\]  

(3)

Next, consider a small area \( dS \) on the wall in a diffuse sound field: if we take the origin of the coordinate system at the center of \( dS \) and the normal of \( dS \) through the origin as the principal axis, the solid angle element \( d\Omega \) can be expressed as

\[
d\Omega = \sin \theta \cdot d\theta \cdot d\phi,
\]  

(4)
where $\theta$ is the zenith angle and $\phi$ is azimuth angle. Since the projection of $dS \cdot \cos \theta$ on the $\phi$ and $\theta$ direction is $dS$, the energy incident from $d\phi$ to $dS$ per unit time is

$$I \cos \theta \cdot dS \cdot d\Omega = I \cos \theta \cdot dS \cdot \sin \theta \cdot d\theta \cdot d\phi.$$  \hspace{1cm} (5)

Integrating this for all solid angular elements, the energy $E_W$ incident on $dS$ per unit time is

$$E_W = I dS \int_0^{2\pi} d\phi \int_0^{\pi/2} \cos \theta \cdot \sin \theta \cdot d\theta = \pi I dS.$$  \hspace{1cm} (6)

The incident acoustic energy $I_W$ on a unit area of the wall surface is

$$I_W = E_W \frac{dS}{dS} = \pi I = \frac{cE}{4}.$$ \hspace{1cm} (7)

from Eq. (3) and Eq. (6).

Let us assume a diffuse sound field and consider the acoustic energy in the room. From Eyring’s reverberation formula, the reverberation time $T_{60}$ is

$$T_{60} = -\frac{24 \ln 10 \cdot V}{cS \ln(1 - \bar{\alpha})},$$ \hspace{1cm} (8)

where $V$ is the volume of the room, $S$ is the surface area of the room and $\bar{\alpha}$ is the average sound absorption coefficient. The average sound absorption coefficient $\bar{\alpha}$ can be expressed as

$$\bar{\alpha} = 1 - \exp \left( -\frac{24 \ln 10 \cdot V}{cST_{60}} \right).$$ \hspace{1cm} (9)

From Eq. (7), the total energy $E_a$ absorbed by all wall surfaces in a unit time is

$$E_a = I_W \bar{\alpha}S = \frac{cE\bar{\alpha}S}{4} = \frac{cEA}{4},$$ \hspace{1cm} (10)

where $A$ is an equivalent sound absorption area and $A = \bar{\alpha}S$.

The energy change in the room over unit time is $V \frac{dE}{dt}$, which is equal to the energy supplied to the room $W(t)$ minus the total energy $E_a$ absorbed by the walls. Thus,

$$V \frac{dE}{dt} = W(t) - \frac{cEA}{4}.$$ \hspace{1cm} (11)

Since $W(t)$ is a constant $W$ and $\frac{dE}{dt} = 0$ in the steady state, the energy density in the steady state $E_0$ is

$$E_0 = \frac{4W}{cA}.$$ \hspace{1cm} (12)

Let $Q$ be the directivity coefficient, a parameter representing the directivity of the sound source, and $r$ be the distance from the sound source to the sound-receiving point. Then the sound energy density $E_D$ due to direct sound is

$$E_D = \frac{QW}{4\pi r^2c}.$$ \hspace{1cm} (13)

The directivity coefficient $Q$ is the ratio $I_D/I_0$ of the sound intensity $I_D$ in the direction of the directional source to the sound intensity $I_0$ in the case of an omnidirectional source.

Considering the sound energy density $E_R$ due to reflected sound, the energy due to reflected sound in unit time is equal to the energy that is not absorbed by all walls in unit time. Therefore, $E_R$ is

$$E_R = \frac{4W(1 - \bar{\alpha})}{cS\bar{\alpha}}.$$ \hspace{1cm} (14)
Generally, multiple microphones are placed around a sound source for directivity measurement [3–7], whereas in this study, microphones were placed at just two locations, one near and one far from the sound source. Instead, the sound source is rotated to obtain data in many directions. Assuming a diffuse sound field, the energy due to reflected sound in a room is constant regardless of the distance, while the energy due to direct sound attenuates with distance [8]. Therefore, it can be assumed that a microphone close to the sound source captures both direct sound and reverberation, while a microphone far from the sound source captures only reverberation.

\[ p^2 = \rho c^2 E, \]

where \( p \) is the sound pressure and \( \rho c \) is the characteristic impedance of air.

Let \( p_{\text{near}} \) be the sound pressure obtained by a microphone close to the sound source and \( p_{\text{far}} \) be the sound pressure obtained by a microphone far from the sound source, then the energy density \( E_D \) due to direct sound and \( E_R \) due to reverberant sound are

\[ E_D = \frac{p_{\text{near}}^2 - p_{\text{far}}^2}{\rho c^2}, \]

\[ E_R = \frac{p_{\text{far}}^2}{\rho c^2}. \]

The directivity coefficient \( Q \) can be obtained by using equations Eqs. (13), (14), (16) and (17),

\[ Q = \frac{16\pi r^2 (1 - \bar{\alpha}) (p_{\text{near}}^2 - p_{\text{far}}^2)}{S\alpha p_{\text{far}}^2}, \]

where \( r \) is the distance between the sound source and the microphone near the source.

### 3.2 Measurement experiment

We conducted an experiment to estimate the directivity coefficient in a seminar room at Waseda University. The shape of the seminar room used for the experiment was a rectangle with a width of 8.6 m, depth of 6.4 m, and height of 2.3 m. A loudspeaker, an Azure Kinect unit which is far from the loudspeaker, and a microphone which is close to the loudspeaker were all placed at a height of 90 cm. Their positions are shown in Fig. 7. Time-stretched-pulse (TSP) signals is radiated from the loudspeaker as a sound source. The direction in which the microphone and Azure Kinect were aligned was set to 0 °, and the loudspeaker was rotated by 10 ° to record all 360 °. The microphone and the Azure Kinect unit remained fixed.

Figure 8 shows the positional relationship of seven microphones built into Azure Kinect. The microphones are located at the apex and the center of a regular hexagon. All seven microphones were used for recording in this measurement.

---

**Figure 7. Location during directivity estimation experiment.**

**Figure 8. Location of the microphone built into the Kinect.**
3.3 Directivity estimation

The impulse response was obtained by convolving the reverse TSP signal with the recorded TSP response and applying one octave band-pass filters at center frequencies fm 500 Hz, 1000 Hz and 2000 Hz. This yielded a bandwidth-specific reverberation decay curve as shown in Fig. 9, and an average curve was obtained from all reverberation decay curves, which is a black line in Fig. 9. The reverberation time and average sound absorption coefficient were calculated from the range of -25 dB to -5 dB of the averaged reverberation decay curve. Table 2 shows the reverberation time and the average sound absorption coefficient. Figure 10 shows the directivity coefficient Q calculated using Eq. (18). The channel number corresponds to the microphone position number of Kinect in Fig. 8.

It can be confirmed that there is the strong directivity in the frontal direction of the loudspeaker which is the 0° direction. It can be also confirmed that the directivity is widely in the low-frequency band and sharply in the high-frequency band. As for the size of the plotted circles, 6ch is plotted larger than others in all three graphs. However, for the other channels, the order of magnitude of Q differs depending on the center frequency. This may due to the position of the Azure Kinect measuring the reverberation being closed to the sound source. It is thought that only reverberation can be obtained by measuring at a sufficient distance from the sound source. However, when the Azure Kinect was placed more than 1 m away from the sound source, both the direct and the reverberation sound could be recorded. We also attributed this to the fact that all microphones of the Azure Kinect are installed upward and embedded in the body.

<table>
<thead>
<tr>
<th>Center frequency [Hz]</th>
<th>Reverberation time [s]</th>
<th>Average absorption coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.566</td>
<td>0.182</td>
</tr>
<tr>
<td>1000</td>
<td>0.404</td>
<td>0.245</td>
</tr>
<tr>
<td>2000</td>
<td>0.326</td>
<td>0.295</td>
</tr>
</tbody>
</table>

Figure 9. All energy decay curves (fm = 1000).
4 CONCLUSIONS

In this paper, we constructed a system for body tracking and voice recording using two Azure Kinects, and confirmed that the use of two Azure Kinects enables tracking over a wide area and with high accuracy. We also estimated directivity using a Azure Kinect unit and a microphone. Although it was possible to obtain a rough directivity, the detailed directivity estimation could not be achieved yet. We thought this was due to the characteristics of Azure Kinect and the experimental setting. Now we are using MATLAB for the directivity estimation process. We would like to enable similar processing in Unity, which is used for the body tracking process, and display it in real time in the same way as body tracking. In the future, we intend to apply directivity estimation to the body motion recording system to automate voice correction.

REFERENCES


