PRESENTED AT MLSP 2016, SEPT. 13–16, 2016, SALERNO, ITALY

# SPEAKER TRACKING FOR HEARING AIDS

*Joachim Thiemann, Jörg Lücke, and Steven van de Par*

Cluster of Excellence "Hearing4All", University of Oldenburg, Oldenburg, Germany

## ABSTRACT

Modern multi-microphone hearing aids employ spatial filtering algorithms capable of enhancing speakers from one direction whilst suppressing interfering speakers of other directions. In this context, it is useful to track moving speakers in the acoustic space by linking disjoint speech segments. Since the identity of the speakers is not known beforehand, the system must match short speech segments without having a specific speaker model or prior knowledge of the speech content, while ignoring changes in acoustic conditions. In this paper, we present a method that matches each speech segment to non-specific speaker models thereby obtaining an activation pattern, and then compares the patterns of disjoint speech segments to each other. The proposed method is low in computational complexity and memory footprint and uses mel-frequency cepstral coefficients (MFCCs) and Gaussian mixture models (GMMs). We find that, when using MFCCs as acoustic features, the proposed speaker tracking method is robust to changes in the acoustic environment provided that sufficiently large segments of speech are available.

***Index Terms***— speaker tracking, speaker recognition, GMM

## 1. INTRODUCTION

Computational Auditory Scene Analysis (CASA) is the analysis of the acoustic scene in a human-oriented manner [1]. The continuing development of ever-smaller and more efficient processors allows us to use CASA methods in modern hearing aids, which have tight constraints in their size and power consumption.

Modern hearing aids typically use multiple microphones, such that spatial filtering (e.g. beamforming) can be applied to enhance speech signals relevant to the user [2]. However, in a complex acoustic scene, determining which speech signal to focus on can be a difficult problem.

In a multi-speaker context, we consider the problem of speaker tracking, in the sense that a particular speaker may be at one specific location (relative to the hearing aid user) in the acoustic scene, but after a short pause, may have moved to a different location. The task of a CASA system would then be to determine that the speaker at the new location is in fact the same individual as was previously at the old location. Thus, the hearing aid can avoid suppressing speech signals that should be enhanced.

To illustrate, one can imagine a hearing aid user listening to a speaker directly in front, who is interrupted by either another speaker or some other noise to the side. The hearing aid user then turns his head towards the interfering source. If the first speaker resumes speaking, the hearing aid should continue to enhance the speech coming from the original speaker.

From a machine learning perspective, this is a very challenging problem, as multiple aspects of the signal change: the acoustic conditions (early reflections and spectral coloration due to late reverberations and head and ear geometry), the speech content, and the background noise. Furthermore, the speaker(s) to be tracked will in general not be known to the system, thus we cannot train for specific people to be recognized, and lastly, the speech fragments originating from the speakers to be tracked could be very short.

This problem may be viewed as a speaker recognition problem where very little training data is available (e.g. the utterances immediately prior to those of the speaker or prior to the head movement). It is also similar to the task of speaker diarization, which typically concerns itself with the problem of "who spoke when" for transcription tasks of broadcast audio. However, in the context of hearing aids the task can be expected to be more challenging since the acoustic environment can be more variable than in broadcast situations, and recognition needs to be done with as little delay as possible. In addition, hearing aids have tight constraints on size and power consumption, thus it is important to consider computational complexity and memory usage.

In this article we present a method to detect the reoccurrence of a speaker based on short speech fragments, with the assumption that the speech segment boundaries are already known. Our method is robust to changes in the acoustical environment (location of the target speaker in the room or changes in the head-related transfer function (HRTF) due to head movement) while being low in computational complexity and memory footprint to allow for implementation on a hearing aid.

## 2. BACKGROUND

Speaker recognition is the process of identifying persons based on their voice. The voice of one given speaker will differ from another based on a number of physical differences (which cannot be easily altered) as well as prosodic differences (which the speakers can generally alter in some way). Considerable work has been done in the field of *text-independent* speaker recognition [3], where the recognition is attempted without constraints of the text the speaker is saying. If the goal is to recognize a particular speaker or set of speakers, the system must be trained (usually off-line) with speech samples of the speaker to be recognized.

Speaker diarization is the process of segmenting a stream of audio into sections based on the identity of the speakers. It is usually described as solving the problem of "who speaks when" in the context of transcribing radio broadcasts, recordings of meetings, etc. [4]. Like the problem of text-independent speaker recognition, the goal is to recognize the speaker identity, but typically no *a priori* model of the speaker is given. Instead, speaker models are built up by finding clusters in the feature space of the audio signal, for example by starting with a Gaussian mixture model (GMM) based background model [5, 6]. One example of real-time segmentation and diarization (also called speaker tracking) can be found in [7], where speaker models are created and updated as new data arrives. The segmentation and speaker tracking are tightly linked, and the authors report that segments need to be longer than 3s for good segmentation and tracking performance.

Linking speech signals that are spoken by the same speaker is also useful in the context of hearing aids. In particular, hearing aids with multiple microphones can employ spatial filtering methods such as beamforming to enhance or suppress sounds originating from some given direction [8]. To allow for a more natural use, the spatial filtering needs to be dynamic (such that the hearing aid user as well as the target speaker are not required to be immobile). Thus, if the hearing aid is enhancing the speech originating from some direction and (possibly after a pause) new speech is detected from a different direction, the hearing aid could infer that the new speech is from a speaker that has previously been enhanced and should therefore also be enhanced.

As in the case of speaker diarization, it is unfeasible to have models of all the speakers the hearing aid user is likely to encounter. Instead, the hearing aid needs to build up a short-term database of recently encountered speakers, and build models of these speakers only from speech recorded in real-time without control over the acoustics, noise, or the content of the speech. In addition, a determination of whether to match the current speech segment to a previously encountered segment needs to be made with as little delay as possible. Furthermore, the algorithm needs to be simple enough to be implemented on a signal processing chip that fits in a hearing device and does not drain the battery.

## 3. SYSTEM DESCRIPTION

The speaker tracking system presented here is based on a speaker recognition system trained on a large variety of speakers. The speaker models are trained using a corpus that is diverse with sentences that are phonetically balanced. All speech items are pre-processed by convolving each item with a set of room impulse responses (RIR) to avoid making the models be specific to one acoustic scenario. The speaker models are deliberately kept very simple in order keep them generic.

We assume that segmentation information is available, that is, the system knows the temporal segment boundaries, and that in a given time segment all speech originates from the same speaker, as illustrated in Fig. 1. In the context of a multi-channel hearing aid, this information is available from e.g. a localization algorithm [9, 10, 11].

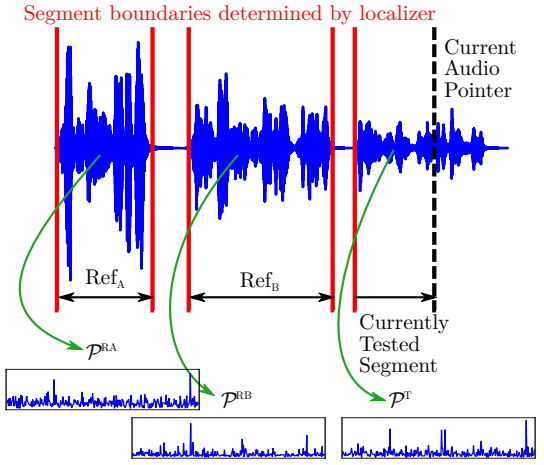


**Fig. 1**. The audio signal is segmented by a localizer, and past segments are transformed into reference patterns with which the current speech segment is compared.

The task of the speaker tracking algorithm is to link segments belonging to the same speaker across segment boundaries and multiple segments. A full system could generate labels unique to each speaker, label segments according to similarity and generate new labels as needed. Further processing can use the labeled segments eg. for speech enhancement or transcription.

Given a speech segment (or fragment thereof), the features extracted from the audio signal are used to compute an activation pattern $\mathcal{P}$ which in turn is defined based on likelihoods of the extracted features to belong to a series of speaker models. Similarity between patterns is computed using the normalized correlation coefficient, and a decision of whether two speech segments are from the same speaker is made by comparing this correlation coefficient to a threshold.

### 3.1. Feature Selection

To analyze the audio signal, we use Mel-Frequency Cepstrum Coefficients (MFCCs) [12] which are well-understood and commonly used in speaker and speech recognition tasks [3]. While a variety of other features could be used, MFCCs are attractive in this framework since they are easy to compute. An energy-based voice activity detector (VAD) is used to ensure that noise-only frames are excluded.

We denote the feature vectors with $\mathbf{y}[m]$, where $m$ is the index of the current observation, which is a time frame where the VAD detects speech activity. Below, we use $\mathbf{y}_R[m]$ to specifically refer to feature vectors from the reference speech and $\mathbf{y}_T[m]$ for the speech to be compared to the reference.

Using this segmentation, we define the reference speech as a temporal segment in the past that we attempt to link to the current segment of speech (see Fig. 1). There may be multiple reference speech items that could be matched, but it is not necessary to store the feature vectors, only the resulting activation patterns.

### 3.2. Speaker similarity estimation

We use a set of $K$ speaker models $\mathcal{S}_k, k = 1, ..., K$, and assume all speaker models $\mathcal{S}_k$ match the speakers in the audio signal with equal likelihood, thus $P(\mathcal{S}_k) = const, \forall k$. For each speaker model $\mathcal{S}_k$ and each time frame $m$ in which speech is detected, we compute the *a posteriori* probability

$$P(\mathcal{S}_k \,|\, \mathbf{y}[m]) = \frac{P(\mathbf{y}[m] \,|\, \mathcal{S}_k)P(\mathcal{S}_k)}{\sum_{k'} P(\mathbf{y}[m] \,|\, \mathcal{S}_{k'})P(\mathcal{S}_{k'})} \quad (1)$$

$$\sim \frac{1}{Z_m} P(\mathbf{y}[m] \,|\, \mathcal{S}_k), \quad (2)$$

where $Z_m$ is the normalization factor for the particular observation (time frame) $m$.

In the proposed method, the speaker models are represented by $K$ Gaussian mixture models (GMM) $\mathcal{S}_1, \ldots, \mathcal{S}_K$ with $V$ Gaussians each [5]. The parameters of one GMM $\mathcal{S}_k$ are given by $\{w_{k,j}, \boldsymbol{\mu}_{k,j}, \Sigma_{k,j}\}_{j=1:V}$, with $w_{k,1}, \ldots, w_{k,V}$ denoting the mixing proportions, $\boldsymbol{\mu}_{k,1}, \ldots, \boldsymbol{\mu}_{k,V}$ denoting the cluster means, and $\Sigma_{k,1}, \ldots, \Sigma_{k,V}$ denoting the cluster covariances. The mixture density is then given by

$$P(\mathbf{y}[m] \,|\, \mathcal{S}_k) = \sum_{j=1}^{V} w_{k,j} b_{k,j}(\mathbf{y}[m]), \quad \text{with} \quad (3)$$

$$b_{k,j}(\mathbf{y}) = \frac{1}{(2\pi)^{D/2}|\Sigma_{k,j}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_{k,j})^T \Sigma_{k,j}^{-1} (\mathbf{y} - \boldsymbol{\mu}_{k,j})\right\}, \quad (4)$$

where $|\Sigma_{k,j}|$ denotes the determinant of $\Sigma_{k,j}$.

Assuming we have a segment of speech from the reference speaker (only) ($\mathbf{y}_R$), we compute a reference pattern $\mathcal{P}^R$ by summing the *a posteriori* probabilities $P(\mathcal{S}_k \,|\, \mathbf{y}_R[m])$ for each speaker model $k$ over time, with

$$\mathcal{P}_k^R = \sum_{m \in \text{Ref}} P(\mathcal{S}_k \,|\, \mathbf{y}_R[m]) = \sum_{m \in \text{Ref}} \frac{1}{Z_m} P(\mathbf{y}_R[m] \,|\, \mathcal{S}_k), \quad (5)$$

where "Ref" indicates the time frames of the reference speaker speech segment. Thus $\mathcal{P}^R = (\mathcal{P}_1^R, \ldots, \mathcal{P}_K^R)^T$ is a $K$-dimensional vector that can be regarded as a "fingerprint" of the reference speaker, and as more speech segments are seen by the system, they can be stored as multiple reference patterns $\mathcal{P}^{RA}, \mathcal{P}^{RB}, \ldots$, as shown in Fig. 1.

Similarly, in the *current* speech segment, the observations from the most recent segment boundary up to the current observation are analyzed by defining the target pattern:

$$\mathcal{P}_k^T = \sum_{m=\text{B}}^{t} \frac{1}{Z_m} P(\mathbf{y}[m] \,|\, \mathcal{S}_k), \quad (6)$$

where $B$ is the last segment boundary and $t$ is the current audio frame. Note that $\mathcal{P}^T$ can be updated online by simple accumulation when new observations are made.

We can now compare the pattern computed from the current speech segment $\mathcal{P}^T$ to any reference pattern $\mathcal{P}^R$, by using the normalized correlation coefficient as a similarity measure:

$$D(\mathcal{P}^R, \mathcal{P}^T) = \frac{\sum_k \mathcal{P}_k^R \mathcal{P}_k^T}{\|\mathcal{P}^R\| \|\mathcal{P}^T\|}, \quad (7)$$

which is at its maximum of 1 if the two patterns match exactly. Note that because of the normalization of $D(\mathcal{P}^R, \mathcal{P}^T)$, the patterns $\mathcal{P}^R$ and $\mathcal{P}^T$ do not have to be normalized by the number of used time frames. If the system is tracking multiple speakers, only (7) needs to be repeated for each $\mathcal{P}^R$ belonging to the speakers being tracked.

### 3.3. Speaker Model Training

A key aspect to the method presented here is the training of the $K$ speaker models $\mathcal{S}_k$. It is advantageous to have a large number of speaker models, such that diverse speakers can be differentiated, however the training data also needs to be well-balanced phonetically. Should, for example, a phoneme appear in the training data of only one model speaker, the occurrence of that phoneme in the speech being tested would bias towards that model, causing a false negative detection. Furthermore, the model speakers should all be recorded using similar acoustic conditions.

When using the speaker tracking algorithm, an exact match of a given speaker to the models in the database is neither required nor expected. Thus, the speaker models can be very simple, which reduces computational complexity and memory footprint, but also helps tp avoid biases in the models since they become very general.

Training the speaker models can be very computationally expensive, but can be done off-line. Only the trained parameters of each speaker model, $\{w_{k,j}, \boldsymbol{\mu}_{k,j}, \Sigma_{k,j}\}_{j=1:V}$ need be stored in read-only memory, requiring at worst the storage of $SV(1 + D + (D(D + 1)/2))$ floating-point values.

## 4. EXPERIMENTS

The speaker tracking system presented above was tested using the TIMIT database [13], which consists of a large number of speech items spoken by a diverse set of speakers. This database is split into a training and a test set, with no speakers present in both sets. While the amount of speech for each speaker is low (10 sentences), there is good coverage of phonemes by all speakers and the recording conditions are uniform.

To simulate a realistic acoustic environment for a hearing aid user, we used a set of room impulse responses (RIR) recorded with a hearing aid model [14]. Speech items were convolved with the RIR from the front left and right microphones of the hearing aid, then turned back into a single-channel signal by summation.

The signal analysis used short-time frames of 20 ms, with a frame advance of 10 ms. A simple VAD discarded frames with overall energy less than 30 dB below the maximum observed frame energy. From each valid frame, 12 MFCCs (1-13, the 0th coefficient was discarded) were extracted using a 40-channel filterbank. All processing was done with a sampling rate of 16 kHz.

In our experiment, for both the training and test set, we used only male speech samples to avoid a bias due to the lower number of female speakers in the TIMIT database. The set of speaker models was composed of $K = 326$ speakers from the training set of the TIMIT database, and we used GMMs with $V = 4$ components for all speakers. The GMMs were trained by rendering all sentences spoken by each speaker in every direction ($-90°$ to $90°$ in $5°$ steps) relative to the head in one room ("office_I"), repeated by also adding speech-shaped isotropic noise at 20 dB SNR.

We evaluated the algorithm by randomly creating 1000 triplets containing 3 different sentences. Each triplet consisted of two sentences spoken by one randomly selected male speaker (the reference) and one sentence spoken by another randomly selected male speaker. There was no overlap in speakers between the testing set (containing 112 individual speakers) and the training set described above. Using a different room model of the database ("office_II", $T_{60} \approx 300$ms), each sentence was rendered at a different position/head rotation combination. The reference pattern $\mathcal{P}^R$ was obtained from the first sentence of the reference speaker, and portions of the two other sentences were used to compute test patterns $\mathcal{P}^T$ which were then compared to the reference pattern. A match was declared if $D(\mathcal{P}^R, \mathcal{P}^T)$ exceeded a threshold $\theta$.

Figure 2 shows the histogram of $D(\mathcal{P}^R, \mathcal{P}^T)$. The blue line (labeled a) shows the distribution for the patterns derived from two segments originating from the same speaker if the content and acoustic transfer functions are varied. The green line show the distribution if in addition the speaker is different from the reference. For the given test data, the first curve has a mean of 0.79 and deviation of 0.07; the second has a mean of 0.60 and deviation of 0.14.

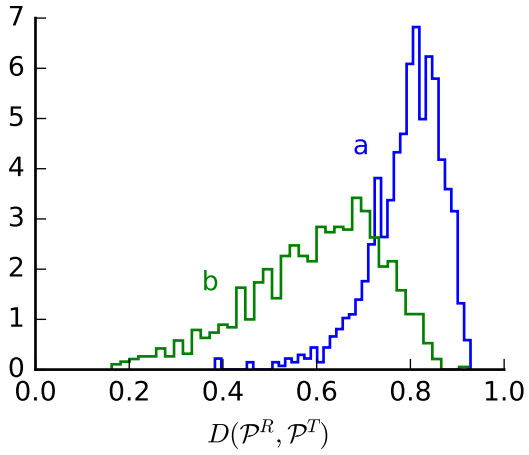


**Fig. 2**. Normalized histogram of $D(\mathcal{P}^R, \mathcal{P}^T)$ for segments that are full sentences. The histogram for $S_R = S_T$ is shown in blue and labeled "a" and the histogram for $S_R \neq S_T$ is shown in green and labeled "b".

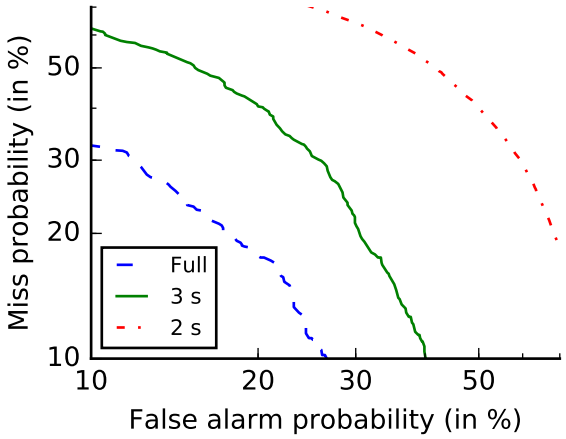


**Fig. 3**. Detection error trade-off figure presenting the effect of truncating the tested segment. Best performance is achieved if the full speech segment is available.

The performance of the proposed algorithm for online speaker tracking is shown in Fig. 3, showing the trade-off between the false alarm probability (a different speaker is falsely labeled as a match) and the miss probability (a matching speaker is not detected). The dashed blue line shows that

reasonably good performance can be achieved if the complete speech segment is available, however, if only 3 seconds of audio is available, performance degrades significantly. This mirrors the performance reported in [7], where for speaker tracking at e.g. a false-detection rate of 20%, the authors reported a recall of 65% (or 35% miss rate) at the speaker change detection point (for 3–6 s segments), to a recall of 77% (23% miss rate) at the next speaker boundary (the full speech segment). As in [7], performance degrades even further with shorter segments. Using only the first 2 s of speech in a segment, performance is near a random guess.

An informal (due to the gender-imbalance of the TIMIT database) test using the same male-speaker trained models described above considered the case where both the reference and contrasting speech was spoken by female speakers; we found a decrease in performance, where the detection error trade-off curve for full sentence segments was shifted by 10% to the right. This suggests that a universal speaker tracker requires a gender-balanced model set.

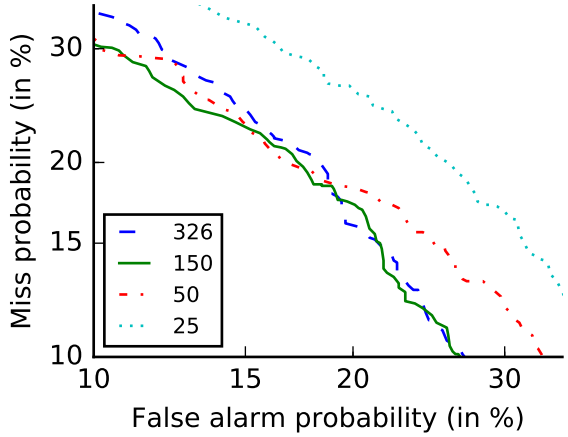


**Fig. 4**. Detection error trade-off figure presenting the effect of reducing the number of stored speaker models. Performance begins to decrease once the speaker database is reduced to 50 speakers, and is significantly lower with only 25 speakers.

In the proposed algorithm, the computational complexity and memory footprint are linearly dependent on the number of speaker models in the database. In Fig. 4, we show that the performance only degrades slightly as the size of the speaker database is reduced from $K = 326$ to $K = 50$, but that further reduction of the number of speaker models causes a noticeable loss of performance.

## 5. DISCUSSION

A hearing aid using spatial filtering can benefit from tracking speakers as the acoustic conditions change. Tracking speakers in this fashion could prevent accidentally suppressing a speaker that the hearing aid user is interested in, or enhancing an interfering speaker that should remain in the background.

Speaker tracking and diarization algorithms have been studied primarily in the context of analyzing broadcast audio, where the acoustic environment tends to be well controlled; furthermore, computational complexity, memory usage, and algorithmic delay are of less concern than accuracy. In the context of hearing aids, the acoustic path from a speaker can change significantly due to head movement or movement of the speaker. Additionally, computational complexity and memory usage directly impact battery life, and delay needs to be minimized, otherwise the hearing aid user could miss important portions of speech.

In this article, we present a method to link speakers' speech segments given differences in the acoustic path. Our method is of low complexity and memory footprint. While our approach still requires a substantial portion of speech to make an accurate decision, this may be a limitation of the selected features (MFCC), as previous MFCC-based methods also reported needing a similar amount of data (about 3 s of speech). Further research efforts should examine acoustic features that can obtain a more rapid decision, while staying within the constraints of the resources in a hearing aid.

## 6. REFERENCES


[1] D. Wang and G. J. Brown, "Fundamentals of computational auditory scene analysis," in *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, DeLiang Wang and Guy J. Brown, Eds., chapter 1. John Wiley and Sons, Hoboken, NJ, 2006.

[2] S. Doclo, S. Gannot, M. Moonen, and A. Spriet, "Acoustic beamforming for hearing aid applications," in *Handbook on Array Processing and Sensor Networks*, Simon Haykin and K. J. Ray Liu, Eds., chapter 9, pp. 269–302. Wiley-IEEE Press, Hoboken, NJ, 2010.

[3] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12 – 40, 2010.

[4] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, Feb 2012.

[5] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, Jan 1995.

[6] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19 – 41, 2000.

[7] L. Lu and H.-J. Zhang, "Unsupervised speaker segmentation and tracking in real-time audio content analysis," *Multimedia Systems*, vol. 10, pp. 332 – 343, 2005.

[8] V. Hamacher, J. Chalupper, J. Eggers, E. Fischer, U. Kornagel, H. Puder, and U. Rass, "Signal processing in high-end hearing aids: State of the art, challenges, and future trends," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 2915–2929, Jan. 2005.

[9] T. May, S. van de Par, and A. Kohlrausch, "A probabilistic model for robust localization based on a binaural auditory front-end," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 1–13, 2011.

[10] J. Thiemann, S. Doclo, and S. van de Par, "Features for speaker localization in multichannel bilateral hearing aids," in *Proc. EUSIPCO*, Nice, France, Aug 2015, pp. 1266–1270.

[11] M. Hu, D. Sharma, S. Doclo, M. Brookes, and P. A. Naylor, "Speaker change detection and speaker diarization using spatial information," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 5743–5747.

[12] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "Timit acoustic-phonetic continous speech corpus," Tech. Rep. NISTIR 4930, National Institute of Standards and Technology, Gaithersburg, MD, 1993.

[14] H. Kayser, S. D. Ewert, J. Anemüller, T. Rohdenburg, V. Hohmann, and B. Kollmeier, "Database of multichannel in-ear and behind-the-ear head-related and binaural room impulse responses," *EURASIP J. on Applied Sig. Proc.*, 2009.