The ICSI RT-09 Speaker Diarization System

Gerald Friedland* *Member IEEE*, Adam Janin, David Imseng *Student Member IEEE*,
Xavier Anguera *Member IEEE*, Luke Gottlieb, Marijn Huijbregts, Mary Tai Knox, Oriol Vinyals

***Abstract*—The speaker diarization system developed at the International Computer Science Institute (ICSI) has played a prominent role in the speaker diarization community, and many researchers in the Rich Transcription community have adopted methods and techniques developed for the ICSI speaker diarization engine. Although there have been many related publications over the years, previous articles only presented changes and improvements rather than a description of the full system. Attempting to replicate the ICSI speaker diarization system as a complete entity would require an extensive literature review, and might ultimately fail due to component description version mismatches. This article therefore presents the first full conceptual description of the ICSI speaker diarization system as presented to the National Institute of Standards Technology Rich Transcription 2009 (NIST RT-09) evaluation, which consists of online and offline subsystems, multi-stream and single-stream implementations, and audio and audio-visual approaches. Some of the components, such as the online system, have not been previously described. The article also includes all necessary preprocessing steps, such as Wiener filtering, speech activity detection and beamforming.**

***Index Terms*—Speaker Diarization, Machine Learning, Gaussian Mixture Models (GMM)**

I. Introduction

THE goal of Speaker Diarization is to segment audio without supervision into speaker-homogeneous regions with the goal of answering the question "who spoke when?". Knowing when each speaker is talking in a recording is a useful processing step for many tasks; it has been used for copyright detection, video navigation and retrieval, and several branches of automatic behavior analysis. In the field of rich transcription, speaker diarization is used both as a stand-alone application that attributes speaker regions in an audio or video file and as a preprocessing task for speech recognition. As a preprocessing step, it enables speaker-attributed speech-to-text and allows for different modes of adaptation (e.g. vocal tract length normalization and speaker model adaptation [1]). The task has therefore become central in the speech community and, as a result, also in the National Institute of Standards Technology (NIST) Rich Transcription (RT) evaluation, where it has been evaluated for several years. Observing the NIST RT evaluations of past years (i.e. 2006, 2007 and 2009) [2]–[4], one can see that the state-of-the-art systems use a combination of agglomerative hierarchical clustering (AHC) with Bayesian Information Criterion (BIC) [5] and Gaussian Mixture Models (GMMs) of frame-based Mel Frequency Cepstral Coefficient (MFCC) features [6]. This article presents a comprehensive description of a set of such systems, the ICSI speaker diarization systems submitted to the NIST RT-09 evaluation, with the goal of allowing their reproduction by third parties without requiring an exhaustive literature research and considerable experimentation. We also present the current limits and discuss future improvements.

The article's structure mirrors the conceptual structure of the ICSI speaker diarization systems: After a brief overview of the system is given in Section II, Section III describes the preprocessing steps such as format normalization, noise reduction, channel selection, and so on. Beamforming, the process by which signals from multiple microphones are exploited, is outlined in Section IV. Speech activity detection is explained in Section V. Next, the batch system for segmentation and clustering of the audio data is described in Section VI. This core system is used for single-microphone diarization. Additional details on audio-visual diarization are presented in Section VII-A. The multi-stream combination algorithm which is used for multi-microphone and audio-visual diarization is described in Section VII-B. Section VII-C describes a first version of a low-latency diarization system, which was presented as an experimental condition in the NIST RT-09 evaluation. Finally, Section VIII presents and discusses some results of the systems on the RT-09 evaluation, followed by the conclusion and presentation of future work in Section IX.

II. System Overview

This section provides a broad outline of the speaker diarization approach; the following sections go into further detail. The ICSI RT-09 diarization system is derived from the Rich Transcription evaluation 2007 [4]. Figure 1 provides on overview of the Multiple Distant Microphone (MDM) and the Single Distant Microphone (SDM) basic systems.

The first step of the processing chain is a dynamic range compression, followed by Wiener filtering for noise reduction. The HTK library [7] is used to convert the audio stream into 19-dimensional Mel-Frequency Cepstral Coefficients (MFCCs) which are used as features for diarization. A frame period of 10 ms with an analysis window of 30 ms is used in the feature extraction. Prosodic features are extracted using Praat. We use the same speech/non-speech segmentation as in [4]. For the segmentation and clustering stage of

G. Friedland, A. Janin, L. Gottlieb, M. Knox, and O. Vinyals are with the International Computer Science Institute, 1947 Center Street, Suite 600, Berkeley, CA, 94704, USA, fractor,janin,luke,knoxm,vinyals@icsi.berkeley.edu

D. Imseng is with Idiap Research Institute, P.O. Box 592, CH-1920 Martigny, Switzerland and Ecole Polytechnique Fédérale, Lausanne (EPFL), Switzerland, david.imseng@idiap.ch

X. Anguera is with the multimedia research group at Telefonica Research, via Augusta 177, 08021, Barcelona, Spain, xanguera@tid.es

M. Huijbregts is with Radboud University Nijmegen, Erasmusplein 1, 6525 HT Nijmegen, The Netherlands, marijn.huijbregts@let.ru.nl

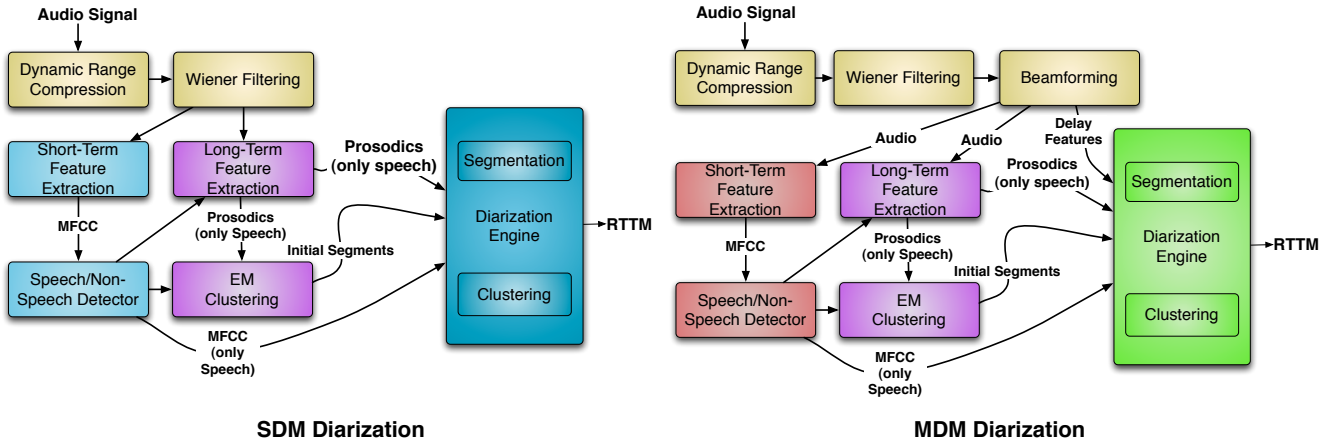

Fig. 1. Overview of the processing chain for the single distant microphone (SDM) case (left) and the multiple distant microphone (MDM) case (right)

speaker diarization, an initial segmentation is generated by our prosodic feature initialization scheme, which is described in Section VI-A.

The procedure for segmenting the audio data takes the following steps:

1) Train a set of GMMs for each initial cluster.
2) Re-segmentation: Run a Viterbi decoder using the current set of GMMs to segment the audio track.
3) Re-training: Retrain the models using the current segmentation as input.
4) Select the closest pair of clusters and merge them. At each iteration, the algorithm checks all possible pairs of clusters to see if there is an improvement in BIC scores when the clusters are merged and the two models replaced by a new GMM trained on the merged cluster pair. The clusters from the pair with the largest improvement in BIC scores, if any, are merged and the new GMM is used. The algorithm then repeats from the re-segmentation step until there are no remaining pairs that when merged will lead to an improved BIC score.

The results of the algorithm consist of a segmentation of the audio track with n clusters and an audio GMM for each cluster, where n is assumed to be the number of speakers.

To use multiple audio tracks as input (presumably from a far-field microphone array), beamforming is first performed as a preprocessing step to produce a single noise-reduced audio stream from the multiple audio channels by using a delay-and-sum algorithm. In addition, as part of its processing, beamforming also estimates time-delay-of-arrival (TDOA) between each microphone and a reference microphone in the array. The TDOA features contain information about the location of the audio source, and are used as an additional feature in the clustering system. Separate GMM models are estimated from these TDOA features. In the Viterbi decoding and in the BIC comparison, a weighted combination of the MFCC and TDOA likelihoods is used. We are using the same mechanism for audio/visual integration (see Section VII-A). The online system, described in Section VII-C is an experimental system not based on the diarization core system.

III. Preprocessing

For all the systems described in this article, the audio files are first preprocessed both to achieve uniformity of format and to mitigate the effects of noise and channel characteristics. First, each channel of multichannel audio is extracted and given a unique name. All files are then converted to 16 bit linear PCM by truncating the high order bits in files with 16 or more bits per sample. Next, files sampled at greater than 16 kHz are downsampled to 16 kHz. In our experience, diarization is not sensitive to choice of downsampling algorithm, so we use the same method as with our work in speech recognition: "Medium Sinc Interpolation" from the open source *libsamplerate* [8] package. We did not perform contrast experiments with other downsamplers.

To mitigate the effects of noise, we apply a Wiener filter [9] to each channel. The Wiener filtering software was originally developed for the Aurora project [10], which dealt with speech recognition of numbers (e.g. zip codes and phone numbers) in noisy conditions. However, we have found the technique to be widely applicable, and we have never observed it to hurt performance. It has therefore become standard practice at ICSI to apply it to all distant microphone audio tasks. We did not perform the contrast experiment of leaving out this step. The noise reduction algorithm includes a noise estimation step that uses the results of a voice activity detector. Although we experimented with various speech/non-speech detectors including the one described in Section V, the built in detector worked as well or better than the other methods. More details on the Wiener filtering can be found in [11].

The next steps differ depending on the number of microphones in the task. For the Single Distant Microphone (SDM) task, where only one signal was present, we compute prosodic features (see Section VI-A) on the noise-reduced channel. This is followed by dynamic range compression to mitigate the change in energy from nearby vs. distant participants, and consists of merely raising the signal to a small power (specifically, $s^{0.75}$). Finally, Mel Frequency Cepstral Coefficients (MFCCs) are computed using the *HCopy* program from *HTK* [7] using 19 MFCC features, a 10 ms step size and a 30 ms analysis window.

The Multiple Distant Microphone (MDM) condition in-

Condition	Arrays	Channels
MM3a	1–4	1 5 9 13 17 22 26 30 34 38 43 47 51 55 59 64
ADM	1–3	1 5 9 13 17 26 30 34 38 47 51 55 59 64
	4	1 5 9 13 17 22 26 30 34 38 47 51 55 59 64

TABLE I
CHANNELS FROM LARGE ARRAY USED FOR MM3A AND ADM CONDITIONS

cludes desktop microphones and small microphone arrays, resulting in at most a few dozen channels per meeting. Each channel is separately noise-reduced using the method described above. Next, the channels are combined into a single channel using a delay-sum technique described in detail in Section IV. Delay features are computed in this step as well. Note that dynamic range compression is not performed in this condition. Finally, MFCCs and prosodic features are computed on the combined signal as described in the previous paragraph.

The next condition, known as MM3a, consists of three meetings that used four large microphone arrays, each with 64 channels (a total of 256 channels). Because of a (since fixed) bug in the beam forming software at the time of the evaluation, we were only able to process a total of 64 channels to produce the single channel output. The first line of Table I indicates which channels were used.

Our previous experience on combining the delay features with MFCCs always used 8 microphones per meeting. Therefore, to avoid retuning, we restrict ourselves to exactly eight delay features per meeting — channel 1 and 64 from each array. The rest of the processing for the MM3a condition is identical to the MDM (Multiple Distant Microphone) condition described above.

The All Distant Microphone (ADM) condition uses all available microphones, including the large array from the MM3a condition for those meetings that were equipped with them. Since we wanted to use the desktop microphones, we dropped 7 microphones from the MM3a condition and added the 7 desktop microphones for the 3 meetings that included the large array. The microphones used are shown in the last two lines of Table I.

IV. MULTICHANNEL ACOUSTIC BEAMFORMING

Since meetings often use multiple microphones to record from several different locations within the room [2], [12], [13], application of Rich Transcription to the meeting domain required a method for handling multiple microphones (referred to as channels). We therefore developed robust acoustic beamforming algorithms to cope with such multiple channels by transforming them into a single *enhanced* channel to which we could apply speech recognition or speaker diarization algorithms. Although many alternative algorithms exist for beamforming, we focused on relatively simple algorithms that could overcome the many constraints that meetings impose, including: 1) exact microphone locations are unknown; 2) their impulse responses and quality are unknown and often differ; 3) the number of microphones per meeting vary (from 2 to more than 100); and 4) the locations and number of sound sources (i.e. the speakers) is unknown.

Our current acoustic beamforming approach for multichannel speaker diarization described below is presented in depth in [14]. This approach has been also used by many RT participants through the open-source acoustic beamforming toolkit known as BeamformIt [15].

BeamformIt is based on the weighted-delay&sum microphone array algorithm, which is a generalization of the well-known delay&sum beamforming technique [16] for far-field sound sources. The single output signal $y[n]$ is expressed as the weighted sum of the different available channels as follows:

$$y[n] = \sum_{m=1}^{M} W_m[n] x_m[n - \mathrm{TDOA}^{(m,\mathrm{ref})}[n]] \qquad (1)$$

where $W_m[n]$ is the relative weight for microphone m (out of M microphones) at frame n, with the sum of all weights being equal to 1; $x_m[n]$ is the signal for each channel at frame n, and $\mathrm{TDOA}^{(m,\mathrm{ref})}[n]$ (Time Delay of Arrival) is the number of samples that each channel should be delayed (around sample n) in order to optimally align it with the channel taken as reference. In this implementation, $\mathrm{TDOA}^{(m,\mathrm{ref})}[n]$ is estimated in steps that are 250 ms long using GCC-PHAT (Generalized Cross Correlation with Phase Transform) [17] by using an analysis window of 500 ms. This algorithm is computationally efficient (several times faster than real-time) and can cope with the constraints mentioned above.

In addition to the GCC-PHAT core module, a set of other steps are added to compute the single output channel from the multiple initial channels. This is shown in Figure 2, and is split into four main blocks described below.

A. Iterative Single Signal Block

Prior to multichannel beamforming, each channel is independently Wiener filtered (see Section III) to remove noise (assumed to be additive and of a stochastic nature). Next, a weighting factor is computed per channel in order to maximize the dynamic range of the signal and therefore reduce the output quantization errors produced by the use of the standard 16 bits per sample. The individual channel weighting is computed by averaging the maximum energy values over a sliding window of several seconds (set by default to 10 seconds).

B. Reference Channel Selection and TDOA Calculation

Several algorithms are used to extract information from the input signals. First, a coarse cross-correlation-based algorithm is used to find the system's reference channel by estimating which channel best matches the others over the entire meeting. Although NIST usually provides a reference channel, we found that computing our own generally led to improved results, particularly when estimating the time difference of arrival (TDOA) values. Next, only for meetings recorded at ICSI, a special processing step is applied to reduce the inter-channel skew present in these recordings as documented in [18]. This module reduces the skew by coarsely aligning the different channels by using long analysis windows. Finally, the aforementioned GCC-PHAT TDOA estimation algorithm is used to retrieve the top N best alignment delays per step from which we choose the final delay in the next block.

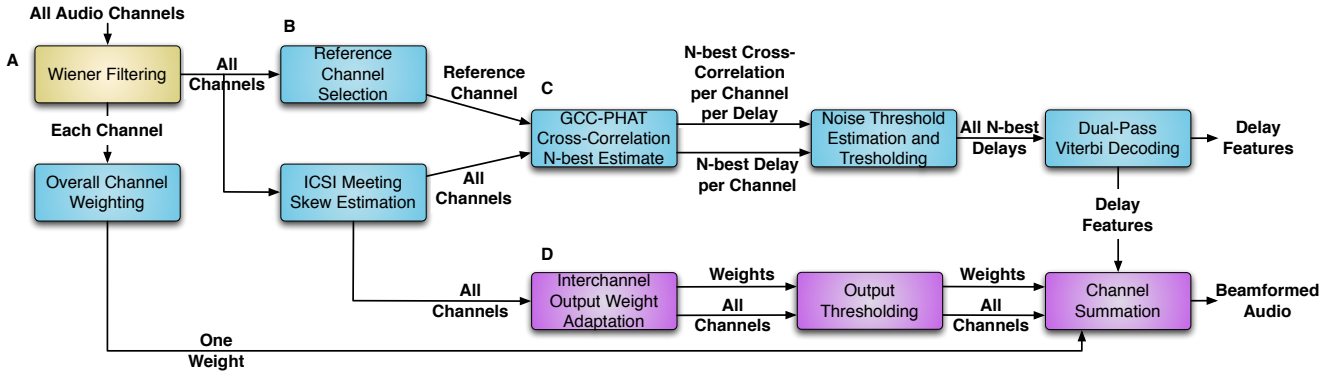

Fig. 2. *A block diagram of the BeamformIt toolkit used in the multi-microphone condition as described in Section IV.*

C. TDOA Values Selection Block

In this block, a post-processing step is applied to the obtained N-best TDOA values and the most appropriate delay is selected per time step. First, a noise threshold is applied to the signal in order to detect those regions where TDOA estimation is prone to unreliable results (e.g. during silence). When the GCC-PHAT cross-correlation result for the 1-best result is below this threshold the previously computed delays are extended to cover the current less reliable ones. Then a dual-step Viterbi decoding is executed in order to select the optimum TDOA values from the N-best available at each step. We call this *dual-step* as it first computes a discrete Viterbi decoding to select the 2-best delays among all N-best available in every single channel, and then computes the best overall combination by considering all 2-best delays from all channels. To do this within a Viterbi decoding, the GCC-PHAT correlations are used as delay probabilities, and the TDOA distances between consecutive estimations are used as transition probabilities. For more details, see [14] and [19].

D. Output Signal Generation Block

Once all information is computed from the input signals, and the optimum TDOA values have been selected, the BeamformIt outputs the enhanced signal and any accompanying information to be used by the subsequent systems. First, for each analysis window a relative channel weight W_m is computed in an adaptive manner by using cross-correlation between all channels in order to account for inter-channel differences in impulse response and overall quality. When any of the channels is below a tuned threshold, it is eliminated from the final sum. Finally, the signal sum obtains a single *enhanced* channel and stores it as a wav/sph file. Optionally, the system can also output the final computed time delays between each channel. These values are known as *delay features*, and are used in combination with MFCCs (Section III) in the later stages of the system.

V. Speech Activity Detection

Our method for Speech Activity Detection (SAD) is inspired by a model-based approach where speech and non-speech are modeled by two Hidden Markov Models (HMMs) and the speech/non-speech segmentation is obtained by performing a Viterbi search on the audio. The difference between the standard model-based approach and our method is that for our system the models are not trained on a training set, but during the classification process itself on the audio that is being processed.

In order to train the models on the audio itself, we first require a rough initial classification, called *bootstrap classification*. We use a standard model-based speech/silence classifier to obtain this initial classification. Once the bootstrap classification is available, three models are trained on the audio to be processed: a model trained on silence; a model trained on audible non-speech; and a model trained on speech. Each of these models is trained on the data to be segmented. By applying the three models, the system is able to perform high quality SAD.

Our SAD algorithm does not use any parameters that require tuning on in-domain training data. It is possible to perform SAD directly on any type of recording without the need to re-train the statistical models or fine tune parameters on in-domain training data. We used this SAD system for RT-07 and RT-09 without tuning any parameters — not even the bootstrapping models that were originally trained on Dutch broadcast news (rather than matched English meeting data).

This section provides an overview of the SAD system. An in-depth description of the system can be found in [20]. An implementation of the algorithm as well as the bootstrap speech/silence models are freely available under a GNU license in the SHoUT toolkit [21].

Step 1: Bootstrapping Speech and Silence

The recording is first segmented using a model-based bootstrapping component which segments the data into speech and silence fragments. The component consists of an HMM with two strings of parallel states. The first string represents silence and the second string represents speech. The states in each string share one Gaussian mixture model (GMM) with diagonal covariance matrix as their probability density function. Using a string of states instead of single states ensures a minimum duration of each segment. The minimum duration for silence is set to 30 states (300 ms) and the minimum duration for speech is set to 75 states (750 ms).

For feature extraction, twelve MFCCs supplemented by the zero-crossing rate are used. From these thirteen features, the

derivatives and second derivatives are calculated and added to the feature vector, creating a 39-dimensional feature vector. Each vector is calculated on a window of 32 ms audio, with a 10 ms step-size between one vector and the next.

Step 2: Training the Models for Non-Speech

Next, a silence and a (non-speech) sound model are created from the parts of the data classified as silence in the bootstrapping phase. Measures are developed to calculate the confidence that a segment is actually silence or audible non-speech. To determine these confidences, first all segments that are longer than one second are divided into evenly sized shorter segments of one second each, so that all segments are comparable in length. The confidence measures then returns a certain number of these *one-second-segments* that are most likely to be either silence or audible non-speech.

It is determined if a one-second-segment is silence by measuring the energy for each frame and calculating the mean energy of the segment. This calculation is performed for all candidate segments (all segments classified as non-speech by the bootstrap classification component) and the resulting values are histogrammed. By using the histogram, it is possible to return the segments with the lowest mean energy.

For determining the number of one-second-segments that are most likely audible non-speech, a similar approach is taken as for silence segments: segments are picked with the highest average energy. From these segments, the segments with the highest mean zero-crossing rates are returned. In other words, this algorithm returns the segments with the highest mean energy and zero-crossing rates. Although audible non-speech segments will have high mean energy values, it is possible that speech segments even have higher average energy values. It is assumed that for these speech segments, the average zero-crossing rates will be lower than for the audible non-speech.

In the first training iteration, a small part of the non-speech data that is marked with the highest silence confidence score is used to train an initial silence model. A small amount of data that is labeled with high audible non-speech confidence scores is used to train the initial "sound" model.

Using these silence and sound models and the primary speech model, a new classification is created. This classification is used to train silence and sound models that fit the audio very well simply because they are trained on it. All data assigned to the sound and silence models by the new classification are merged and any samples that were originally assigned to the speech model in the first iteration are subtracted from the set. This is done to avoid having the sound model pull the data from the speech model. This risk is present because although the sound model is already trained on the data that is being processed, the speech model applied is still the old model trained on outside data. Therefore, the sound model may fit *all* of the data better (including speech segments) so that during the Viterbi alignment, speech segments may be assigned to the sound model.

The remaining data is divided over the silence model and the sound model as before. The silence model receives data with high silence confidence scores and the sound model receives data with high audible non-speech confidence scores. This time though, the confidence threshold is not set as high as the first time, and consequently more data is available to train each model and therefore one more Gaussian can be used to train each GMM. This procedure is repeated three times. Note that the confidence threshold is a system parameter that could potentially be tuned according to the audible non-speech prior. In our experiments we have observed that tuning this parameter in not needed for the algorithm to perform well on various types of audio [4]. Although the silence and sound models are initialized with silence and sound respectively, there is no guarantee that sound is never classified as silence. Energy is not used as a feature (see Section III) and some sound effects appear to be modeled by the silence GMM very well. Because the goal is to find all speech segments and discard everything else, this is not considered a problem.

Step 3: Training All Three Models

After the silence and sound models are trained, a new speech model is trained using all data classified as speech. By now, the non-speech will be modeled well by the sound and silence models so that a Viterbi alignment will not assign any non-speech to the speech model. This makes it possible to train the speech model on all data assigned to it rather than only on the high confidence regions. Once the new speech model is created, all models are iteratively retrained with increasing the number of Gaussians by one in each step until a threshold is reached. At each training iteration the data is re-segmented. Note that in this phase, all data is being used to train the models. During the earlier iterations, the data assigned to the speech class by the bootstrap classification component was not used to train the silence and sound models, but because now the speech model is being retrained, it is less likely that using this data will cause the sound model to pull speech data away from the speech model.

Step 4: Training Speech and Silence Models

The algorithm works for audio of various domains and with a range of non-speech sounds, but it is not well suited for data that contains speech and silence only. In that case, the sound model will be trained solely on the speech that is misclassified at the first iteration (because the initial models may be trained on data not matching the audio being processed, the amount of misclassified speech can be large). During the second training step the sound model will subtract more and more speech data from the speech model and finally instead of having a silence, sound and speech model, the system will contain two competing speech models. Therefore as a final check, the Bayesian Information Criterion (BIC, see Equation 4 in Section VI-B) is used to check if the sound and speech model are the same. If the ΔBIC score is positive, both models are trained on speech data and the speech and sound models need to be replaced by a single speech model. Again, a number of alignment iterations is conducted to obtain the best silence and speech models.

Category	Short description
pitch	median of the pitch
pitch	minimum of the pitch
pitch	mean of the pitch tier
formants	standard deviation of the 4th formant
formants	minimum of the 4th formant
formants	mean of the 4th formant
formants	standard deviation of the 5th formant
formants	minimum of the 5th formant
formants	mean of the 5th formant
harmonics	mean of the harmonics-to-noise ratio
formant	mean of the formant dispersion
pitch	mean of the pointprocess of the periodicity contour

TABLE II
THESE 12 LONG-TERM ACOUSTIC FEATURES HAVE GOOD SPEAKER DISCRIMINATION ACCORDING TO THE RANKING METHOD PROPOSED IN [24]. THE FEATURES ARE EXTRACTED WITH THE HELP OF PRAATLIB, A LIBRARY USING PRAAT [25], ON ALL THE SPEECH REGIONS OF THE RECORDINGS. FEATURES ARE THEN USED TO ESTIMATE THE NUMBER OF INITIAL CLUSTERS TO PERFORM THE AGGLOMERATIVE CLUSTERING. FOR MORE INFORMATION ON THE FEATURES REFER TO THE PRAAT DOCUMENTATION.

VI. SEGMENTATION AND CLUSTERING

A. Initialization

The segmentation and clustering starts with an adaptive initialization scheme that can be applied to most state-of-the-art Speaker Diarization algorithms. More specifically, the initialization is a combination of the recently proposed "adaptive seconds per Gaussian" (ASPG) method [22] and a new pre-clustering and number of initial clusters estimation method based on long-term features [23]. This initialization method results in an AHC (agglomerative hierarchical clustering) approach where the two most sensitive parameters, namely the number of initial clusters k and the number of Gaussians per Gaussian Mixture g, are estimated without the need for supervision.

1) Pre-clustering: The pre-clustering method estimates the number of initial clusters and also provides a non-uniform initialization for the AHC procedure based on the long-term feature study and ranking presented in [24], where 70 different suprasegmental features have been studied according to their speaker discriminability. Derived from the ranking in [24], the 12 top-ranked features (listed in Table II) are extracted on all the speech regions in the recording.

Temporally slow features are computed as statistics based on (noisy) pitch and formant values across time. In our configuration, we use the Praat library [25] to compute 100 pitch values and 80 formant values per second. For the feature extraction procedure, we use a Hamming window function with a minimum window size of 1000 ms. The minimum window size parameter is used as follows: Every segment output from the speech/non-speech detector of less than 2000 ms (2 times the minimum) is untouched and segments larger than 2000 ms are split into segments of at least 1000 ms, thus yielding an effective window length $w \in [1000, 2000]$, i.e. a minimum window size of 1000 ms. The concept of a minimum window size is a trade-off between using longer windows, allowing accurate estimates of statistical features, and using smaller windows, providing a larger number of feature vectors for good clustering (an appropriate estimation of k) and a reasonable non-uniform initialization. The minimum window size is not a very sensitive initialization parameter because even if the initial segmentation and k vary, we can still interpolate g accordingly [26].

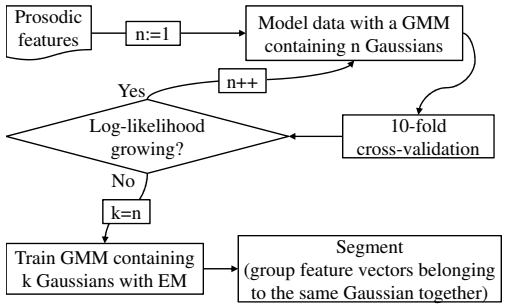

Fig. 3. A schematic view of the pre-clustering procedure to estimate k and perform a non-uniform initialization.

The 12-dimensional feature vectors are then clustered with the help of a GMM with diagonal covariances. As this clustering serves only as initialization for an agglomerative clustering algorithm, it is desirable for the model selection to over-estimate the number of initial clusters; the agglomerative clustering algorithm merges redundant clusters but it is not able to split them. To determine the number of Gaussians per Gaussian Mixture, we train GMMs with different number of Gaussians (using the EM algorithm [27]), evaluate the log-likelihood of the obtained GMMs and choose the number of Gaussians based on the maximal log-likelihood result. To avoid overfitting, we apply 10-fold cross-validation (see [28, page 150]), i.e. we divide the set of feature vectors into ten subsets, train a GMM on each subset and evaluate the log-likelihood on the corresponding other nine subsets. Then, expectation maximization is used to train the GMM (consisting of the previously determined number of Gaussians) on all the feature vectors. Finally, every feature vector is assigned to one of the Gaussians in the GMM. We can group all the feature vectors belonging to the same Gaussian into the same initial segment. The clustering thus results in a non-uniform initialization where the number of initial clusters is automatically determined. A schematic view of the pre-clustering can be seen in Figure 3.

2) Adaptive seconds per Gaussian (ASPG): An appropriate estimate for the number of seconds of data available per Gaussian for training, $secpergauss$, is crucial for good Speaker Diarization performance. We found a general estimated optimal $secpergauss$ based on a linear regression on the duration of speech in a meeting [22]. $secpergauss$ relates the two initialization parameters k and g. Anecdotal evidence suggests that optimal k is best chosen in relation to the number of different speakers in the meeting, whereas optimal g is more related to the total amount of available speech; therefore, we use the pre-clustering to estimate k. Having an estimate for k, linear regression can then be used to determine g as summarized in Equation 2 and Equation 3.

$$secpergauss = 0.01 \cdot \text{speech in seconds} + 2.6 \quad (2)$$

$$g = \frac{\text{speech in seconds}}{secpergauss \cdot k} \quad (3)$$

B. Core Algorithm

Our core segmentation/clustering system uses an agglomerative hierarchical clustering approach based on a Hidden Markov Model (HMM), which models the temporal structure of the acoustic observations, and Gaussian Mixture Models (GMMs) as emission probabilities to model the multimodal characteristics of the data.

The main tasks involved in the core system are as follows:

- Step 0. Initialization, as discussed above.
- Step 1. Model retraining and re-segmentation using Expectation Maximization (EM).
- Step 2. Model merging based on the Bayesian Information Criterion (BIC).
- Step 3. Stopping condition (if there are more models to merge, go to step 1; otherwise, go to 4).
- Step 4. Final segmentation and output.

Step 1: Model Retraining and Resegmentation: After preprocessing acoustic observations that contain speech (as previously described in Section V), the main challenge is to segment the data and generate speaker models where no a priori information is known. This process is done iteratively, in an EM fashion, where models are trained based on current temporal segmentation, and a new segmentation is recomputed using the newly trained models. These two steps are iterated through three times before moving to Step 2.

For model retraining, we assume we are given a segmentation and the goal is to retrain the acoustic models for each of the states (each state models different speech characteristics of each speaker and, after the agglomerative clustering has converged, they model all speech found in a meeting for a single speaker). Since the segmentation is given by the Viterbi path (and not by the forward-backward algorithm), each frame is uniquely assigned to a single state. The update on the k-th state emission model, which is a GMM, is performed on the frames that belong to state k given by the segmentation, and trained using the standard EM procedure to update the parameters for each mixture within the GMM as described in [4]. We consider diagonal covariance matrices, so each mixture has a total of 38 parameters to be updated (19 for the mean and 19 for the covariance).

During resegmentation, we assume that models are given (in the form of a GMM), and the task is to find the segmentation for the dual purpose of retraining the models, and to give an output that will yield the desired information that diarization provides (i.e. identifying who spoke when).

Since the best front-end features that were found for this task are spectral features in the form of MFCC with 19 coefficients (Section III), we need to ensure that the clusters that we find are modeling speakers instead of smaller acoustic units such as phones (since similar features are used for speech recognition). To achieve this, we force the topology of our HMM to remain in the same state for at least 2.5 seconds (i.e. we set a minimum duration of speech of 250 samples). This step is critical for the core algorithm to work. The choice of 2.5 seconds seems reasonable, as it assumes that each speaker takes the floor for at least that amount of time. Smaller numbers yield worse performance on a development set, and state persistence shorter than 1.0 seconds yielded very poor performance. Lastly, the HMM model assumes that from a given boundary state, we can jump to any other speaker (including itself) with equal probability.

Given the HMM structure described above, and the emission probability models obtained from the GMM, the segmentation is performed using the Viterbi algorithm for efficiency.[1]

Step 2: Model Merging Based on BIC: Given that our approach agglomerates clusters, a metric for which two clusters should be merged at any given point is needed. The hypothesis from which we start is that the large number of clusters at the beginning will align with some acoustic characteristic for a single speaker (i.e. each cluster maps to one speaker only, but the mapping is many to one), and the goal is to find which set of clusters correspond to the same speaker, to merge them, and to reduce the total number of clusters by one. Given this, one should answer the question: which two clusters (if any) correspond to the same speaker and thus should be merged?

This reduces to a model selection problem for any pair of clusters, and can be reformulated as the question: given these two clusters, are the two separate models better than a joint model? To answer the question, we use the Bayesian Information Criterion (BIC), which is a model selection technique, and the two hypotheses as follows:

For each cluster pair (i, j), test the two hypotheses:
-H_0: cluster i and j should be merged
-H_1: cluster i and j should not be merged

The merging score S is given by the change in the BIC score (called delta BIC) [5], where the number of parameters of the hypothesized merged cluster is the sum of the parameters that the initial clusters i and j had, which reduces the delta BIC to a simple likelihood computation (as the total number of parameters of our model remains constant across iterations):

$$S(i,j) = \mathcal{L}(x_{i\cup j}|\Theta_{i\cup j}) - \mathcal{L}(x_i|\Theta_i) - \mathcal{L}(x_j|\Theta_j) \quad (4)$$

where x_i and x_j are the data from clusters i and j, $x_{i\cup j}$ is the data that belongs to either i or j, and Θ_i, Θ_j, and $\Theta_{i\cup j}$ are the GMM parameters of clusters i, j, and $i \cup j$. The number of parameters in $\Theta_{i\cup j}$ is the sum of the number of parameters in Θ_i and Θ_j.

Note that if $S(i,j) > 0$, H_0 is selected, and otherwise H_1 is selected, as $S(i,j) = \log \frac{p(H_0)}{p(H_1)}$.

Finally, we merge only one pair of clusters at a time (before returning to Step 1), selecting the pair of clusters (i, j) such that $S(i, j)$ is the maximum. The newly created GMM has as many mixtures as the sum of the two merged GMMs, and we initialize each mixture to have the same mean and variance as the original, merged, model, with mixture weights re-scaled so that the sum is one.

Step 3 and 4: Stopping Criterion and Final Output: If $S(i, j)$ is negative for all possible cluster pairs, no more merging is required and a final segmentation is performed using the current cluster models. During the final segmentation, the

[1]Full forward-backward for retraining models using Baum-Welch did not improve the performance versus "hard" assignments given by the Viterbi path.

clusters should be more accurate and should, in theory, match a single speaker. The HMM used to produce the final output is set with a minimum duration of 1.5 seconds instead of 2.5 seconds to suffer fewer quantization errors on the evaluation metric used for diarization.

VII. New Directions

The RT-09 evaluation incorporated several optional tasks for the first time that are described in the following.

A. Audiovisual Diarization

The audiovisual diarization system incorporates the single distant microphone and the close-up camera views to perform speaker diarization. The ICSI RT-07 multi-stream engine was used to combine MFCC, prosodic, and video features. In this subsection, we describe the audio and video features used.

Three types of features are used in the audiovisual diarization system: MFCC, prosodic, and video. We describe these features below.

We extract 19^{th} order MFCC features computed over a 30 ms window with a step size of 10 ms. These are standard features that were also used in our audio-only speaker diarization systems (see Section III).

Prosodic features are also computed over the single distant microphone recording. We extract 10 prosodic features which perform well on our development set. The prosodic features are median pitch, mean pitch, minimum pitch, mean pitch tier, mean pitch tier number of samples, mean formant dispersion, mean long term average spectrum energy, minimum 5th formant, mean 5th formant, and mean pointprocess periodicity contour.

We include compressed domain based video features that were shown to work well for audiovisual speaker diarization in [29]. These features are obtained from the MPEG-4 video encoding, making them extremely fast to extract.

The video features are average motion vector magnitudes over estimated skin blocks for each of the close-up cameras. Motion vector magnitudes are used to estimate activity levels of the participants [30]. By averaging the motion vector magnitudes over skin blocks, we focus our attention to salient regions of the video and reduce the effect of scale variation [29].

The motion vectors are block-based and computed during video compression. Further post-processing is performed for the motion vectors; namely, motion vectors for blocks with low confidence λ values (blocks with a small amount of texture) are considered not reliable and thus set to 0. For more information regarding the motion vector confidence, see [30].

The skin blocks were determined based on the chrominance Discrete Cosine Transform (DCT) DC coefficients. We use a GMM to model the chrominance DCT DC coefficients of skin regions [31], and blocks for which the likelihood exceeded a threshold were classified as containing skin.

Since the video features are computed for the close-up camera views, we compute these for the meetings held at the Idiap Research Institute and Edinburgh only. The meetings recorded at NIST did not contain the close-up camera view, so we use our audio-only speaker diarization submission for those meetings.

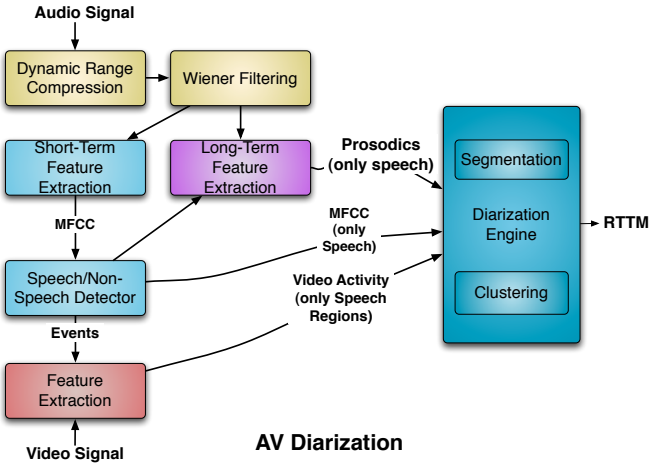

Fig. 4. The audio-visual speaker diarization system is an extension of the SDM system.

B. Multi-stream Algorithm

The ICSI RT-09 diarization engine is able to combine the clustering information from the various components—MFCCs, prosodic features, delay features, and video features—into a single optimal choice of clustering. The configuration that combines MFCC, prosodic, and audio/visual features is shown in Figure 4 and described below.

After the initialization, GMM parameters for each type of feature (Θ_{MFCC}, Θ_{pros}, and Θ_{vid}) are trained for each cluster and the input stream is resegmented using the hard Expectation Maximization (EM) algorithm. In the E-step, segmentation is performed such that the "joint log-likelihood" $\hat{\mathcal{L}}$ of the data is maximized based on the current parameters of the GMM. In the M-step, the GMM parameters for each type of feature are updated based on this new segmentation. The "joint log-likelihood" for cluster k and frame i is defined as:

$$\begin{aligned}\hat{\mathcal{L}}(x[i]|\Theta_k) \doteq\ & \alpha \cdot \mathcal{L}(x_{MFCC}[i]|\Theta_{MFCC,k}) \\ & + \beta \cdot \mathcal{L}(x_{pros}[i]|\Theta_{pros,k}) \\ & + (1-\alpha-\beta) \cdot \mathcal{L}(x_{vid}[i]|\Theta_{vid,k})\end{aligned} \tag{5}$$

where $x_{MFCC}[i], x_{pros}[i]$, and $x_{vid}[i]$ are the MFCC feature vector, prosodic feature vector, and video feature vector at frame i, $\mathcal{L}$ is the log-likelihood, Θ_k is the parameters for the joint model for cluster k, and $\alpha, \beta \in [0, 1]$ are weights for the MFCC and prosodic log-likelihoods. Empirically we found that $\alpha = 0.75$ and $\beta = 0.1$ worked well for our development set.

The merging steps proceed as described in Section VI-B but replacing the standard likelihood by the one represented in Equation 5.

C. Low-Latency Diarization

The goal of low-latency diarization is to create a system that minimizes the sample processing latency, defined as an average

of the amount of sensor data (in seconds) an algorithm needs to process for each sample.

Our online diarization system for both the SDM and MDM are fundamentally the same, comprised of a training step and an online recognition step. We used our offline SDM and MDM (Section VI) for the training step, which generated models for use in the online recognition step. In this section, we will describe the training tools and the operation of the online recognition system.

1) Training: For the training step, we take the first 1000 seconds or the entire meeting file before the testing region (whichever is larger) and perform a regular offline speaker diarization using the system we submitted as our primary SDM and MDM conditions. We then train speaker models and a speech/non-speech model from the output of the system. This is done by concatenating 60 random seconds of each speaker's segmented data and the non-speech segments. We then train a GMM for each speaker and for the non-speech model with 20 Gaussians per mixture using expectation maximization on a diagonal-only covariance matrix.

2) Online Diarization: For the online diarization step, we use a GMM-SVM system as fully described in [32], the use of which is briefly described here. After the training data is extracted, we perform online recognition of the remaining portion of the meetings using the trained models. For every frame, the likelihood for each set of features is computed against each set of Gaussian Mixtures obtained in the training step, i.e. each speaker model and the non-speech model. A total of 250 frames is used for a majority vote on the likelihood values to determine the classification result. Therefore the latency totals $t + 2.5$ seconds per decision (plus the portion of the offline training that overlaps with the testing region). The online recognition step does not take advantage of delay features, although the MDM system uses beamformed audio (beamforming has a latency of 0.5 seconds).

The rationale behind the system is that meetings happen repeatedly in the same room with the same people. In the beginning of the first meeting, one would train speaker models using the offline system and then be able to compute the "who is speaking now" information after 1000 seconds (plus runtime) every 2.5 seconds. Unfortunately, the system currently does not detect any speakers who were not present in the initial training phase. We experimented with different "unknown speaker detection" methods, but all of them decreased our total score significantly on the development set.

VIII. Results

Table III shows the official NIST Rich Transcription 2009 evaluation results for the various conditions [33]. ADM (all-distant microphones), MM3A, and MDM are different microphone array processing tasks, with MDM being considered the most important task. We used the Diarization Error Rate, which is defined by NIST, as evaluation measure. The Diarization Error Rate expresses the percentage of time that is not attributed correctly to a speaker or to non-speech. As in previous years, the results show that adding more microphones does not necessarily increase the accuracy of the system.

System	Condition	Speech Non-Speech Error Rate	Diarization Error Rate
Batch Audio	adm	6.43	28.52
	mm3a	6.29	28.32
	mdm	4.92	17.24
	sdm	5.92	31.30
Online Audio	mdm	7.94	39.27
	sdm	15.03	44.61
Audiovisual	sdm	6.89	32.56

TABLE III
Results on the Eval09 set for the batched (offline) audio system, the low-latency online system, and the audiovisual system.

The RT-09 dataset differs from previous datasets in that it is more challenging because it has more speakers and also more overlapped speech. Therefore the biggest challenge in RT-09 was to detect the correct number of speakers and to create overlap-robust methods. The results shown in Table III reflect this, as the speech/non-speech errors are quite high due to mishandling of overlapped speech.

The experimental online system performs reasonably well given its ad-hoc construction. The novel audio-visual system was not yet able to improve over the audio-only SDM system in this evaluation. Although the reasons are yet to be analyzed, it is not clear that we should even expect the strength of audio-visual integration to be increased accuracy. In fact, there is evidence that the primary strength of audio/visual integration is increased robustness against different noise conditions [34] — something that is not measured in NIST evaluations.

IX. Conclusion and Future Work

This article presents the state of the ICSI speaker diarization system as of the NIST Rich Transcription evaluation for 2009. The system consists of many components, from preprocessing, feature extraction, speech activity detection and beamforming, to initialization and segmentation and clustering. In addition, several variants of the system competed in the evaluation: many microphones, single microphones, audiovisual, online, and offline systems. Future efforts for improving the system will most likely put more emphasis on robustness against overlap as well as the estimation of the correct number of speakers. With the rising trend towards parallelization, speed gains will most likely lead to better online systems.

The ICSI speaker diarization has been applied in many domains, from telephone conversations within the speaker recognition evaluations, to broadcast news and meeting recordings in the NIST Rich Transcription evaluations. Furthermore, it has been used in many applications such as a front-end for speaker and speech recognition, as a meta-data extraction tool to aid navigation in broadcast TV, lecture recordings, meetings, and video conferences and even for applications such as media similarity estimation for copyright detection. We conclude that speaker diarization is an essential fundamental technology that will be used for and adopted to even more application domains as more and more people acknowledge the usefulness of audio methods for many tasks that have traditionally been thought to be exclusively solvable in the visual domain. The ICSI speaker

diarization engine should serve as a good starting point for exploring the area and we therefore encourage researchers to try our system[2].

ACKNOWLEDGMENTS

This work was sponsored by the Swiss NSF through the National Center of Competence in Research (NCCR) on "Interactive Multimodal Information Management" (IM2, www.im2.ch) and the European Integrated Project on "Augmented Multiparty Interaction with Distance Access" (AMIDA, www.amidaproject.org).

REFERENCES

[1] J. Ajmera, "A robust speaker clustering algorithm," in *Proceedings of IEEE Workshop on Automatic Speech Recognition Understanding*, 2003, pp. 411–416.

[2] A. Janin, J. Ang, S. Bhagat, R. Dhillon, J. Edwards, J. Macias-Guarasa, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters, and B. Wrede, "The ICSI meeting project: Resources and research," in *Proceedings of ICASSP Meeting Recognition Workshop*, 2004.

[3] X. Anguera, C. Wooters, B. Peskin, and M. Aguilo, "Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system," in *Proceedings of the NIST MLMI Meeting Recognition Workshop*. Edinburgh: Springer, 2005.

[4] C. Wooters and M. Huijbregts, "The ICSI RT07s speaker diarization system," *Second International Workshop on Classification of Events, Activities, and Relationships (CLEAR 2007) and the Fifth Rich Transcription 2007 Meeting Recognition (RT 2007)*, pp. 509–519, 2007.

[5] S. S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, USA, February 1998. [Online]. Available: http://www.nist.gov/speech/publications/darpa98/pdf/bn20.pdf

[6] D. Reynolds and P. Torres-Carrasquillo, "Approaches and Applications of Audio Diarization," *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 5, pp. 953–956, March 2005.

[7] "HMM Toolkit Web Page." [Online]. Available: http://htk.eng.cam.ac.uk

[8] "Secret Rabbit Code (aka libsamplerate) Web Page." [Online]. Available: http://www.mega-nerd.com/SRC

[9] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley, 1949.

[10] D. Pearce and H.-G. Hirsch, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, October 2000, pp. 29–32.

[11] A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas, "Qualcomm-ICSI-OGI features for ASR," in *Seventh International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, Colorado, USA, September 2002, pp. 4–7.

[12] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner, "The AMI meeting corpus," in *Proc. Measuring Behavior*, 2005.

[13] D. Mostefa, N. Moreau, K. Choukri, G. Potamianos, S. M. Chu, A. Tyagi, J. R. Casas, J. Turmo, L. Cristoforetti, F. Tobia, A. Pnevmatikakis, V. Mylonakis, F. Talantzis, S. Burger, R. Stiefelhagen, K. Bernardin, and C. Rochet, *The CHIL audiovisual corpus for lecture and meeting analysis inside smart rooms*. Language Resources and Evaluation, December 2007, vol. 41.

[14] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2023, September 2007.

[15] X. Anguera, "BeamformIt (the fast and robust acoustic beamformer)." [Online]. Available: http://www.xavieranguera.com/beamformit/

[16] J. Flanagan, J. Johnson, R. Kahn, and G. Elko, "Computer-steered microphone arrays for sound transduction in large rooms," *Journal of the Acoustic Society of America*, vol. 78, pp. 1508–1518, November 1994.

[17] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-24, no. 4, pp. 320–327, August 1976.

[18] "ICSI meeting recorder project: Channel skew in ICSI-recorded meetings," 2006. [Online]. Available: http://www.icsi.berkeley.edu/~dpwe/research/mtgrcdr/chanskew.html

[19] X. Anguera, "Robust Speaker Diarization for Meetings," Ph.D. dissertation, Universitat Politecnica de Catalunya, 2006.

[20] M. Huijbregts and F. de Jong, "Robust speech/non-speech classication in heterogeneous multimedia content," *Speech Communication*, accepted for publication.

[21] "SHoUT Toolkit Web Page." [Online]. Available: http://shout-toolkit.sourceforge.net

[22] D. Imseng and G. Friedland, "Robust speaker diarization for short speech recordings," in *Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding*, December 2009, pp. 432–437.

[23] ——, "An adaptive initialization method for speaker diarization based on prosodic features," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 4946–4949.

[24] G. Friedland, O. Vinyals, Y. Huang, and C. Muller, "Prosodic and other long-term features for speaker diarization," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 5, pp. 985–993, Jul 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5067417

[25] P. Boersma and D. Weenink, "Praat: doing phonetics by computer (version 5.0.32) [computer program]," retrieved August 12, 2008. [Online]. Available: http://www.praat.org/

[26] D. Imseng and G. Friedland, "Tuning-robust initialization methods for speaker diarization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2028–2037, November 2010.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[28] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann, San Francisco, 2005.

[29] G. Friedland, H. Hung, and C. Yeo, "Multi-modal speaker diarization of real-world meetings using compressed-domain video features," *Proc. IEEE ICASSP*, 2009.

[30] C. Yeo and K. Ramchandran, "Compressed domain video processing of meetings for activity estimation in dominance classification and slide transition detection," UC Berkeley, Tech. Rep. UCB/EECS-2008-79, June 2008.

[31] S. McKenna, S. Gong, and Y. Raja, "Modelling facial colour and identity with gaussian mixtures," *Pattern Recognition*, vol. 3, no. 12, pp. 1883–1892, 1998.

[32] O. Vinyals and G. Friedland, "Towards semantic analysis of conversations: A system for the live identification of speakers in meetings," *Proceedings of IEEE International Conference on Semantic Computing*, pp. 426–431, Aug 2008.

[33] "NIST rich transcription 2009 evaluation web page." [Online]. Available: http://www.itl.nist.gov/iad/mig/tests/rt/2009

[34] G. Friedland, C. Yeo, and H. Hung, "Dialocalizaton: Acoustic Speaker Diarization and Visual Localization as Joint Optimization Problem," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 6, no. 4, p. 27, 11 2010.

[2]Visit our website http://diarization.icsi.berkeley.edu for more information.

Gerald Friedland Gerald Friedland is a staff research scientist at the International Computer Science Institute (ICSI), an independent non-profit research lab associated with the University of California at Berkeley where he, among other functions, is currently leading the Speaker Diarization research. Apart from speech his interests also include image and video processing and multimodal machine learning. As a member of the IEEE, IEEE Computer Society, and IEEE Communication Society, Dr. Friedland is involved in the organization of various ACM and IEEE conferences, including the IEEE International Conference on Semantic Computing (ICSC2009) where he serves as co-chair and the IEEE International Symposium on Multimedia (ISM2009) where he serves as program co-chair. He is also co-founder and program director of the IEEE International Summer School for Semantic Computing at UC Berkeley. Dr. Friedland is the recipient of several research and industry recognitions, among them the Multimedia Entrepreneur Award by the German government and the European Academic Software Award. Most recently, he won the first prize in the ACM Multimedia Grand Challenge 2009. He received his "diplom" and doctorate ("summa cum laude") in computer science from Freie Universität Berlin in 2002 and 2006, respectively.

David Inseng received the B.Sc. and M.Sc. degrees from Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2006 and 2009, respectively. He is currently pursuing the Ph.D. degree at the Idiap Research Institute, Martigny Switzerland, where he is working on multilingual speech recognition. From September 2008 to April 2009, he was a Visiting Scholar at the International Computer Science Institute (ICSI), Berkeley, CA. His research interests include speech and speaker recognition and machine learning.

Adam Janin Dr. Adam Janin is a senior researcher at the International Computer Science Institute (ICSI) in Berkeley, California with more than 14 years of experience in audio processing and speech recognition. Current work focuses on improving robustness to noise through novel neural network architectures and features and on exploiting the next generation of parallel hardware to improve audio processing. Dr. Janin coordinated ICSI's activities in AMI (a large collaborative project related to meeting analysis funded by the European Union), led ICSI's efforts in the NIST Rich Transcriptions evaluations in 2006 and 2007, and was also heavily involved in the collection of the ICSI Meeting Corpus, one of the first open corpora of natural meetings. He also works closely with the multimedia and language researchers at ICSI, providing expertise in audio processing. Prior work includes pioneering research in Augmented Reality as part of Boeing's Research and Technology group.

Mary Knox Mary Tai Knox received the B.S. degree in Electrical Engineering from Purdue University and the M.S. degree in Electrical Engineering and Computer Sciences from the University of California, Berkeley. She is currently pursuing her Ph.D. in Electrical Engineering and Computer Sciences at the University of California, Berkeley. Her research interests include machine learning, specifically multimodal speaker diarization.

Luke Gottlieb Luke Gottlieb has been a research assistant at ICSI for over 5 years. Highlights of his research contributions include data structure design for the ICSI Multi-Lingual Sentence Segmentation System, experiment design for the use of artificial conversation data for speaker recognition, and the use of artistic markers for narrative theme navigation in sitcoms. The latter project having garnered him, along with Dr. Gerald Friedland and Dr. Adam Janin the first prize in the ACM Multimedia Grand Challenge 2009. He is currently assisting with research in net security and multimodal location estimation.

Xavier Anguera Xavier Anguera Miro was awarded Telecommunications Engineering and European Masters in Language and Speech [MS] degrees in 2001 from UPC (Barcelona, Spain). In 2006 he obtained his PhD degree at UPC, with a thesis on Robust Speaker Diarization for Meetings. Between 2001 and 2003 he was with Panasonic Speech Technology Lab in Santa Barbara, CA. Between 2004 and 2006 he was a visiting researcher at the International Computer Science Institute (ICSI) in Berkeley, CA, where he pursued research on speaker diarization for meetings, contributing to ICSIs participation in the NIST RT evaluations in 2004 (broadcast news) and 2005-2007 (meetings), obtaining state-of-the-art results. He briefly joined LIMSI, in Paris, in 2006. He has been with Telefonica Research in Barcelona, Spain, since 2007, pursuing research in multimedia. His current research interests include speaker characterization (including diarization, recognition, etc.), language identification (including a participation in NIST LRE07 evaluation) and several topics in multimodal multimedia analysis (e.g. video copy detection, involving the participation in NIST TRECVID 2009 and 2010 evaluations). Xavier Anguera has authored or co-authored over 50 peer-reviewed research articles. He is a member of ISCA, ACM, and IEEE Signal Processing Society and has been involved in the organization of several ACM and IEEE conferences. He has been a reviewer for many conferences, as well as for several journals in the multimedia domain. He is the main developer of the BeamformIt toolkit, extensively used by the RT community for processing multiple microphone recordings.

Marijn Huibregts Marijn Huijbregts is a postdoc researcher at Radboud University Nijmegen. He received his Ph.D. degree from University of Twente in 2008. His research interests include speaker diarization, automatic speech recognition and spoken document retrieval.

Oriol Vinyals Oriol received his double degree from the Polytechnic University of Catalonia (Barcelona, Spain) in Mathematics and Telecommunication Engineering, and a Master in Computer Science from the University of California, San Diego in 2009. He is currently a PhD student at the University of California, Berkeley, and is one of the 2011 Microsoft Research PhD Fellowship recipients. Oriol interests include artificial intelligence, with particular emphasis on machine learning, speech, and vision. He was a visitor scholar at the Computer Science department of the Carnegie Mellon University in 2006, where he worked in computer vision and robotics.