SPEAKER SEGMENTATION AND CLUSTERING IN MEETINGS

Qin Jin, Kornel Laskowski, Tanja Schultz, and Alex Waibel

Interactive Systems Laborotary
Carnegie Mellon University
{qjin, kornel, tanja, ahw}@cs.cmu.edu

ABSTRACT

This paper describes the issue of automatic speaker segmentation and clustering for natural, multi-speaker meeting conversations. Two systems were developed and evaluated in the NIST RT-04S Meeting Recognition Evaluation, the Multiple Distant Microphone (MDM) system and the Individual Headset Microphone (IHM) system. The MDM system achieved a speaker diarization performance of 28.17%. This system also aims to provide automatic speech segments and speaker grouping information for speech recognition, a necessary prerequisite for subsequent audio processing. A 44.5% word error rate was achieved for speech recognition. The IHM system is based on the short-time crosscorrelation of all personal channel pairs. It requires no prior training and executes in one fifth real time on modern architectures. A 35.7% word error rate was achieved for speech recognition when segmentation was provided by this system.

1. INTRODUCTION

In recent years, the study of multispeaker meeting audio has seen a surge of activity at many levels of speech processing, as exemplified by the appearance of large meeting speech corpora from several groups, important observations available in the literature [1][2], and the ground-breaking evaluation paradigm launched by NIST, the Rich Transcription Evaluation on Meetings.

The full automatic transcription of meetings is considered an AI-complete, as well as an ASR-complete, problem [3]. It includes transcription, meta-data extraction, summarization and so on. Automatic speaker segmentation and clustering is one type of meta-information extraction. NIST started the "Who Spoke When" speaker diarization evaluation (which is the speaker segmentation and clustering task) on telephone conversations and Broadcast News in 2002. However, it is more challenging to segment and cluster speakers involved in meetings with speaking overlap and with distant microphones. Therefore, NIST initiated the same evaluation on meetings in the spring of 2004 [4].

Speaker segmentation and clustering consists of identifying who is speaking and when, in a long meeting conversation. Ideally, a speaker segmentation and clustering system will discover how many people are involved in the meeting, and output clusters corresponding to each speaker. This paper describes the automatic speaker segmentation and clustering of meetings based on multiple distant microphones. For the personal close-talking microphone condition, it is actually a speech/silence detection task. However, unexpectedly, even with close-talking microphones, due to unbalanced calibration and small inter-speaker distance, each participant's personal microphone picks up significant levels of activity from the other participants, making independent energy thresholding an unviable approach. The presence of extraneous speech activity in a given personal channel leads to a high word error rate due in large part to faulty insertion. Furthermore, portable microphones are subject to low frequency noise such as breathing and speaker (head) motion. We propose an algorithm for dealing with this issue based on the short-time crosscorrelation of all channel pairs. To our knowledge, the only work which specifically addresses the simultaneous multispeaker segmentation problem is [5] at ICSI.While our conclusions are very similar to those in the ICSI study, the algorithm we propose is architecturally simpler. Specifically, it does not employ acoustic models for speech and non-speech states and thus requires no prior training.

The remainder of this paper is organized as follows. In section 2 we briefly describe the data we used for the evaluation of our systems. In section 3 we introduce the speaker segmentation and clustering system based on multiple distant microphones and show experimental results. In section 4 we describe the crosscorrelation-based multispeaker speech activity detection system for multiple personal microphones and report experimental results. Conclusions follow in section 5.

2. DATA

The experiments throughout this paper were conducted on the RT-04S meeting data. Each meeting was recorded with

Table 1. *Development dataset*

MeetingID (abbreviation)	#Skrs	cMic	#dMic
CMU_20020319-1400 (CMU1)	6	L	1
CMU_20020320-1500 (CMU2)	4	L	1
ICSI_20010208-1430 (ICSI1)	7	H	4
ICSI_20010322-1450 (ICSI2)	7	H	4
LDC_20011116-1400 (LDC1)	3	L	8
LDC_20011116-1500 (LDC2)	3	L	8
NIST_20020214-1148 (NIST1)	6	H	7
NIST_20020305-1007 (NIST2)	7	H	6

personal microphones for each participant (close-talking microphones), as well as room microphones (distant microphones) placed on the conference table. In this paper we focus on two tasks: 1) automatic speaker segmentation and clustering based on distant microphone channels only; 2) automatic segmentation of all personal microphone channels, that is, the discovery of portions where a participant is speaking in his/her personal microphone channel.

Both the development and the evaluation datasets from the NIST RT-04S evaluation were used. The data were collected at four different sites, including CMU, ICSI, LDC, and NIST [6][7][8][9]. The development dataset consists of 8 meetings, two per site. Ten minute excerpts of each meeting were transcribed. The evaluation dataset also consists of 8 meetings, two per site. Eleven minute excerpts of each meeting were selected for testing. All of the acoustic data used in this work is of 16kHz, 16-bit quality. Table 1 gives a detailed description of the RT-04S development dataset, on which we report detailed performance numbers. "cMic" is the type of close-talking microphones used and "#dMic" is the number of distant microphones provided for each meeting. The final speaker diarization performance and speech recognition performance on the RT-04S evaluation dataset is also presented.

3. MDM SYSTEM

3.1. System Overview

The MDM system consists of following steps: 1) initial speech/non-speech segmentation for each channel; 2) unification of the initial segmentations across multiple channels; 3) best channel selection for each segment; 4) speaker change detection in long segments; 5) speaker clustering on all segments; 6) smoothing.

Initial speech/non-speech segmentation is produced based on the acoustic segmentation software CMUseg_0.5. We removed the classification and clustering components and used it as a segmenter. A detailed description of the algorithms used in this software can be found in [10].

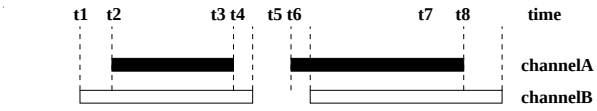

Fig. 1. Multiple Channel Unification

In the **multiple channel unification** step, the segment boundaries are unified across multiple channels. Figure 1 shows an example for two distant microphone channels. The initial segmentation produces two speech segments on channel A, (t2, t3) and (t5, t7); and two segments, (t1, t4) and (t6, t8), on channel B. After unification, the segments across the two channels are (t1, t2), (t2, t3), (t3, t4), (t5, t6), (t6, t7) and (t7, t8).

We conduct **best channel selection** for each of the segments produced during the unification step. We compute the minimum energy ($MinE_i$), maximum energy ($MaxE_i$), and the signal-to-noise ratio (SNR_i) within each segment on all channels. We select the best channel for each segment according to following criterion,

$$i^{\star} = \text{argmin}_i \left(\frac{MinE_i}{MaxE_i} \times \frac{1}{SNR_i} \right) \quad (1)$$

Speaker change detection is applied to any segment that is longer than 5 seconds. We choose 5 seconds because this was found to give optimal segmentation accuracy via cross-validation on the development set. **Speaker clustering** is then performed on all segments. We will discuss the speaker change detection and speaker clustering modules in detail in the following two sections.

In the final **smoothing** step, we merge any two segments that belong to the same speaker and have less than a 0.3 seconds gap between them. This is based on our experience in the RT-03S evaluation.

3.2. Speaker Segmentation

For any segment that is longer than 5 seconds, we use a speaker change detection procedure to check whether there exist speaker turn changes that have not been detected. The procedure is shown in Figure 2.

We first compute the distance between two neighboring windows. The window size is one second each and it is shifted every 10ms. The distance between Win_1 and Win_2 is definded as

$$D(Win_1, Win_2) = -\log \frac{P(X_C|\theta_C)}{P(X_A|\theta_A)\ P(X_B|\theta_B)} \quad (2)$$

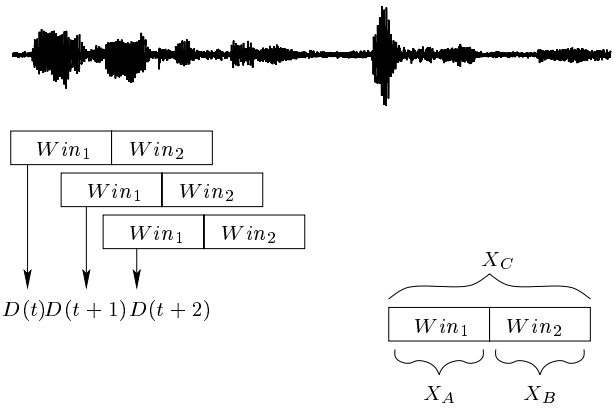

Fig. 2. Speaker Change Detection

where X_A, X_B, and X_C are feature vectors in Win_1, in Win_2, and in the contatenation of Win_1 and Win_2, respectively. θ_{A}, θ_{B}, and θ_{C} are statistical models built on X_A, X_B, and X_C, respectively. We can see from (2) that the larger the distance, the more likely a speaker turn change exists at the boundary between Win_1 and Win_2.

We assume a speaker turn change exists if the local maximum of distances satisfies

$$\begin{array}{c} D_{max} - D^L_{min} > \alpha \\ D_{max} - D^R_{min} > \alpha \\ Min(|I_{max} - I_{min}|^L_R) > \beta \end{array} \quad (3)$$

where D_{max} refers to the local maxiamum distance value and D^L_{min} and D^R_{min} refer to the left and right local minimum distance values around the local maximum. I_{max} refers to the index of the local minimum. The third inequality in (3) considers not only the value of the local maximum but also its shape. α and β are constant thresholds, for which we found optimal values via cross-validation on the development set. α is equal to the variance of all the distance values times a factor of 0.5. β is set to 5. Our approach differs from other approaches, such as [11][12], because in our implementation we build a Tied GMM (TGMM) using the entire speech segments and generate a GMM for each segment by adapting the TGMM. The advantage is that a more reliable model can be estimated with a TGMM.

3.3. Speaker Clustering

For speaker clustering, we use a hierachical, agglomerative clustering technique called TGMM-GLR. We first train a TGMM, θ, based on all speech segments. Adapting θ to each segment generates a GMM for that segment. The definition of the GLR distance between two segments is the same as in (2). A symmetric distance matrix is built by computing the pairwise distances between all segments. At each clustering step, the two segments which have the smallest distance are merged, and the distance matrix is updated. We use the Bayesian Information Criterion as a stopping criterion.

3.3.1. Bayesian Information Criterion

The Bayesian Information Criterion (BIC) is a model selection criterion widely used in statistics. It was introduced for speaker clustering in [11]. The Bayesian Information Criterion states that the quality of model M to represent data $\{x_1, \ldots, x_N\}$ is given by

$$BIC(M) = \log L(x_1, \ldots, x_N|M) - \frac{\lambda}{2} V(M) \log N \quad (4)$$

with $L(x_1, \ldots, x_N|M)$ representing the likelihood of model M and $V(M)$ representing the complexity of model M, equal to the number of free model parameters. Theoretically, λ should equal to 1, but it is a tunable parameter in practice.

The problem of determining if there is a speaker change at point i in data $X = \{x_1, \ldots, x_N\}$ can be converted into a model selection problem. The two alternative models are: (1) model M_1 assumes that X is generated by a multi-Gaussian process, that is $\{x_1, \ldots, x_N\} \sim N(\mu, \Sigma)$, or (2) model M_2 assumes that X is generated by two multi-Gaussian processes, that is

$$\begin{array}{c} \{x_1, \ldots, x_i\} \sim N(\mu_1, \Sigma_1) \\ \{x_{i+1}, \ldots, x_N\} \sim N(\mu_2, \Sigma_2) \end{array}$$

The BIC values for the two models are

$$\begin{aligned} BIC(M_1) &= \log L(x_1, \ldots, x_N|\mu, \Sigma) - \frac{\lambda}{2} V(M_1) \log N \\ BIC(M_2) &= \log L(x_1, \ldots, x_i|\mu_1, \Sigma_1) \\ &\quad + \log L(x_{i+1}, \ldots, x_N|\mu_2, \Sigma_2) \\ &\quad - \frac{\lambda}{2} V(M_2) \log N \end{aligned}$$

The difference between the two BIC values is

$$\begin{aligned} \Delta BIC &= BIC(M_1) - BIC(M_2) \\ &= \log \frac{L(x_1, \ldots, x_N|\mu, \Sigma)}{L(x_1, \ldots, x_i|\mu_1, \Sigma_1) L(x_{i+1}, \ldots, x_N|\mu_2, \Sigma_2)} \\ &\quad + \frac{\lambda}{2} [V(M_2) - V(M_1)] \log N \end{aligned}$$

A negative value of ΔBIC means that model M_2 provides a better fit to the data, that is there is a speaker change at point i. Therefore, we continue merging segments until the value of ΔBIC for the two closest segments (candidates for merging) is negative.

3.4. MDM Experiments

3.4.1. Speaker Segmentation Performance

A good speaker segmentation algrithm should provide only the correct speaker changes. As a result, each segment should contain exactly one speaker. There are two types of errors related to speaker change detection: insertion errors (when a speaker change is detected but it does not exist in reference) and deletion errors (an existing speaker change is not detected). These two types of errors have a different impact depending upon the application. In our system, the segmentation stage is followed by a clustering stage. Therefore, insertion errors (resulting in oversegmentation) are less critical than deletion errors, since the clustering procedure has the opportunity to correct the insertion errors by grouping the segments related to the same speaker. On the other hand, deletion errors cannot be corrected in the clustering stage.

A reference of speaker change is required for analyzing these errors. The reference was generated from a manual transcription. However, the exact speaker change point is not very accurate in the reference, since the perception of speaker change is very subjective. Therefore, we define an accuracy window around the reference speaker change point; following [13], it is set to 1 second. For example, if N_r and N_h are sample indices of reference and hypothesized speaker change points respectively, they are mapped to one-another and we call the hypothesis N_h a hit if (1) N_h is the hypothesized change point closest to N_r and (2) N_r is the reference change point closest to N_h and (3) the distance between N_r and N_h is less than 1 second. From the formed mapping between reference and hypothesis, we can determine the precision (percentage of a hit from among all the hypothesized change points) and recall (percentage of a hit from among all the reference change points). Deletion errors will directly lower the recall. Insertion errors will reduce the precision. Generally we seek systems that exhibit both high recall and high precision. However, as mentioned previously, deletion errors are more critical than insertion errors; we are more concerned about the recall value.

Table 2. *Speaker Segmentation Performance (in %)*

System Stage	Precision	Recall
Initial	86.83	11.60
Unification	87.74	19.00
Change Detection	85.17	76.41

Table 2 shows the speaker segmentation performance at different system steps. Not surprisingly, the low recall of the initial segmentation indicates high deletion errors, which means that a lot of speaker changes are missed. Multiple channel unification compensates a little for the deletion errors. Speaker change detection leads to a big improvement in recall while suffering only a small decrease in precision.

3.4.2. Speaker Diarization Performance

We use a standard performance measurement, speaker diarization error, for speaker segmentation and clustering as used in the NIST RT-03S evaluation [14]. The overall speaker segmentation and clustering performance can be expressed in terms of the miss rate (speaker in reference but not in system hypothesis), false alarm rate (speaker in system hypothesis but not in reference), and speaker error rate (mapped reference speaker is not the same as the hypothesized speaker). The speaker diarization score is the sum of these three components and can be calculated using

$$DiaErr = \frac{\sum_{allS}\{dur(S)*(max(N_{ref}(S),N_{sys}(S))-N_{correct}(S))\}}{\sum_{allS}\{dur(S)*N_{ref}(S)\}}$$

where $DiaErr$ is the overall speaker diarization error, $dur(S)$ is the duration of the segment, $N_{ref}(S)$ is number of reference speakers in the segment, $N_{sys}(S)$ is the number of system speakers in the segment, and $N_{correct}(S)$ is the number of reference speakers in the segment for which are also hypothesized by the system. This formula allows the entire audio to be evaluated, including regions of overlapping speech. In the following tables, we use abbreviations "Miss", "FA", "SpkrErr", and "DiaErr" to represent miss rate, false alarm rate, speaker error rate, and diarization error rate, respectively.

Table 3. *Speaker Diarization Performance (in %)*

Error	Development Set		Evaluation Set	
	Include	Exclude	Include	Exclude
Miss	8.7	0.0	19.8	0.4
FA	3.3	2.9	2.6	4.1
SpkrErr	25.1	26.7	17.8	23.4
DiaErr	**37.11**	**29.59**	**40.19**	**28.17**

Table 3 shows the overall speaker diarization performance on the development set and on the evaluation set, both when including the regions of overlapping speech and when excluding the regions of overlapping speech. Comparable results are achieved on both datasets. The dominant error among the three error components is speaker error.

In Table 4 we show the speaker diarization performance on individual meetings of the development set. The results exhibit large variability over meetings collected at different sites. We think that this variability may be due to unquantified meeting characteristics such as overall degree of crosstalk, general meeting geometry including room acoustics and microphone variability within a meeting. However,

Table 4. *Speaker Diarization Performance on individual meeting in dev set including overlapping speech (in %)*

Meeting	Miss	FA	SpkrErr	DiaErr	#ref	#sys
CMU1	12.6	4.3	30.3	47.12	6	4
CMU2	3.4	5.0	16.3	24.72	4	2
ICSI1	4.7	2.9	35.0	42.62	7	4
ICSI2	9.8	1.1	37.0	47.92	7	3
LDC1	6.2	2.6	9.0	17.78	3	3
LDC2	17.3	1.1	11.0	29.41	3	3
NIST1	7.2	7.1	11.7	26.01	6	2
NIST2	6.5	3.1	49.5	59.04	7	2

we noticed that our system often underestimates the number of speakers involved in a meeting. Although on meetings CMU2 and NIST1 the system underestimates the number of speakers, it still achieves better performance compared to most other meetings. This is due to the fact that both these two meetings have a dominant speaker who talks for more than 70% of the time. We compute the speaker speaking time entropy $H(Meeting)$ for each meeting,

$$H(Meeting) = -\sum_{i=1}^{M} P(S_i) * \log P(S_i)$$

$$P(S_i) = \frac{T(S_i)}{\sum_{i=1}^{M} T(S_i)}$$

where M is the number speakers invovled in the meeting. $T(S_i)$ is the total time that speaker S_i speaks. $P(S_i)$ is the percentage of time (ie. probability) that speaker S_i speaks. The lower the entropy, the more biased is the distribution of the speaker speaking time in the meeting. As $H(Meeting) \rightarrow 0$, it becomes more likely that there is only one dominant speaker in the meeting.

Figure 3 shows the speaker diarization error on each individual meeting in the development set versus its speaker speaking time entropy. We can see from the figure that our system tends to produce lower speaker diarization error on meetings that have lower speaker speaking time entropy.

We also conducted an experiment as follows. We assume a one-to-one mapping between channel and speaker. We use the best channel information only, which was provided in the channel selection step described in section 3.1. We do not perform speaker clustering. For any two segments, if the channel selection process produces the same best channel for them, we assume these two segments belong to the same speaker. This yields 55.45% and 52.23% speaker diarization error under the conditions of including and excluding overlapping speech, respectively. It indicates that there is rich information that can be used to aid in speaker segmentation and clustering from the multi-channel recordings. Our current system utilizes such information implicitly by doing best channel selection. In future work, we plan to explore more efficient use of the information provided by multi-channel recordings, such as timing information, which relates to speaker location.

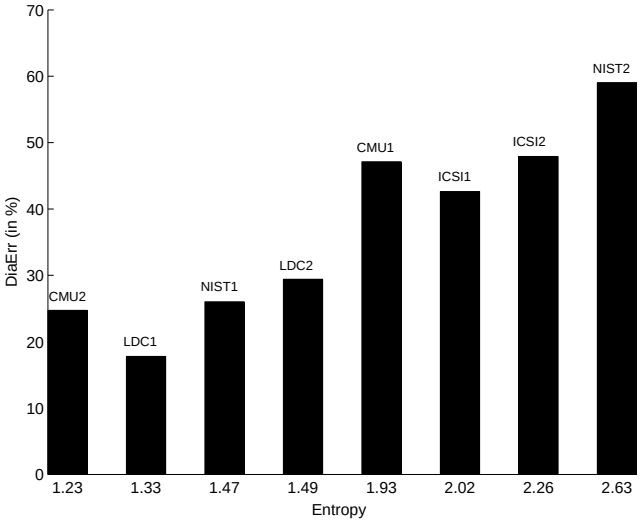

Fig. 3. Speaker speaking time entropy vs. diarization error.

3.4.3. Speech Recognition Performance

Our speech recognition system achieved a 44.5% word error rate on the evaluation set when using segments provided by this system; refer to [15] for details. We have noticed that speech recognition has a different requirement for speaker segmentation and clustering. In speech recognition, the goal of speaker segmentation and clustering is to provide clean single speaker segmetns for speaker adaptation. Speaker adaptation is concerned more with the regression of speakers, not the strict classification of speakers. So if two speakers sound similar, they can be considered as equal and grouped into one cluster. It actually would be rather desirable for speech recognition to group similar speakers together, so that it can get more data for adaptation. Therefore, a specific speaker segmentation and clustering system tuned for speech recognition may achieve better word error rate even if speaker diarization performance is worse.

4. IHM SYSTEM

4.1. Algorithms

4.1.1. Conceptual Framework

In contrast to the MDM condition, the audio for a single meeting consists of N time-aligned mono channels, where

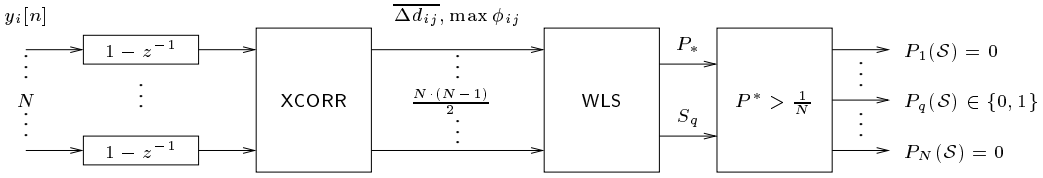

Fig. 4. Architectural depiction of the IMTD algorithm

N is the number of speakers.

The response at microphone M_i, $y_i[n]$, is a combination of signals $x_j[n]$ from every acoustic source S_j in the room, both delayed and attenuated. We restrict our attention to exactly N possible sources, namely the vocal apparata of the N speakers wearing the microphones; we ignore the existence of other potential sound sources which we group at each microphone into a white noise term η_i. Furthermore we assume that the mouth-to-microphone distance for each speaker is negligible compared to the minimum inter-microphone distance; ie. $M_i \approx S_i$. This assumption is patently false but it allows for a simplified analysis involving the relative positions of only N points in a two-dimensional plane.

Each $x_j[n]$ is delayed and attenuated as a function of the distance d_{ij} between its source S_j and microphone M_i. The delay Δn_{ij}, measured in samples, is linearly proportional to the distance,

$$\Delta n_{ij} = \frac{f_s \, d_{ij}}{c} \tag{5}$$

where f_s is the sampling frequency and c is the speed of sound. For simplicity, we assume that $y_i[n]$ is a linear combination

$$y_i[n] = \sum_{j=1}^{N} \alpha_{ij} x_j[n - \Delta n_{ij}] + \eta_i \tag{6}$$

where η_i is a noise term.

In the general case, all α_{ij} are positive, ie. all microphones pick up all speakers to some extent.

4.1.2. Baseline

The straightforward approach to this problem is obviously to use energy thresholding on each personal microphone channel. Our baseline system uses this approach. The energy threshold is equal to the average of the 200 lowest energies multiplied by a factor of 2. Any frame that has energy beyond the threshold will be considered as the participant's speech in that channel. As we will show in the experimental results section, the baseline system yields very poor performance.

4.1.3. Inter-microphone Time Differences (IMTD)

In our first experiment, we consider the use of inter-microphone time differences much as humans use interaural time differences to lateralize sources of sound [16]. In contrast to a single interaural lag in the latter, the meeting scenario offers an ensemble of $N \cdot (N-1)/2$ lags given N microphones/speakers, whose magnitudes are governed by much larger distances than head diameter as well as arbitrary seating arrangement.

Consider the general case with exactly one person S_q speaking during the current analysis frame. Then for each pair of microphone signals $\{y_i[n], y_j[n]\}$, $i \neq j$, the short-time crosscorrelation

$$\phi_{ij}[\Delta n] = \sum_n y_i[n] \cdot y_j[n + \Delta n] \tag{7}$$

exhibits a distinct peak at a lag corresponding to the difference in distance $\Delta d_{ij}^{(q)} = d_{iq} - d_{jq}$.

Given N points, we can compute $N \cdot (N-1)/2 > N$ distance differences. If the noise term, η, is both small and white, then this overdetermined system of equations will nevertheless be consistent, that is, for any three microphones $\{y_i[n], y_j[n], y_k[n]\}$,

$$\Delta d_{ik}^{(q)} = \Delta d_{ij}^{(q)} + \Delta d_{jk}^{(q)} \tag{8}$$

This defines an implicit transformation into polar coordinates, with speakers arranged radially around a single sound source, and in particular their projection onto the radial direction, spaced apart by the corresponding distance differences. After placing the origin arbitrarily in this single dimension, we solve for the positions of the listeners' microphones relative to that origin using a weighted least squares approximation, with the normalized crosscorrelation as the weight. The magnitude of the approximation error E indicates the degree to which the system of $N \cdot (N-1)/2 > N$ distance difference equations is consistent, and therefore the degree to which the hypothesis that a single speaker is speaking holds. We posit the probability that a single speaker is speaking (in a somewhat ad hoc fashion) as

$$P_* = e^{-E/\sqrt{N}} \tag{9}$$

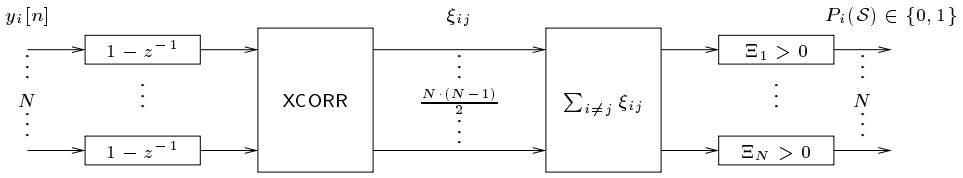

Fig. 5. Architectural depiction of the JMXC algorithm

which we can threshold as desired. Furthermore, the microphone whose abscissa is smallest is hypothesised as being worn by the speaker.

In situations where multiple speakers are speaking, maxima in the crosscorrelation spectra will not in general lead to a consistent system of distance difference equations; therefore E will be high. Likewise, during pauses, maxima in the spectra will occur at random lags since the microphone signals will be uncorrelated under the assumptions of our framework; likewise in this case, E will tend to be high.

The three main functional blocks of this algorithm, computation of all crosscorrelations, weighed least squares approximation and probability thresholding, are shown in Figure 4. In addition, we apply preemphasis to all channel signals, using a simple IIR filter ($1 - z^{-1}$), to reduce their low frequency contribution. Microphone motion and breathing both exhibit significant activity at low frequencies, and this method leads to significant reduction in the miss rate due to these phenomena on channels other than the foreground speaker's.

4.1.4. Joint Maximum Crosscorrelation (JMXC)

In a second competing algorithm, we employ the peak magnitude of the crosscorrelation between microphone signals as opposed to the lag at which it occurs.

After locating the peak in the crosscorrelation spectrum $\max \phi_{ij}$ between two microphone signals $\{y_i[n], y_j[n]\}$, we compute the quantity

$$\xi_{ij} = \log_{10} \frac{\max \phi_{ij}}{\phi_{jj}} \tag{10}$$

where the ϕ_{jj} is the power of $y_j[n]$ in the current analysis frame. If speaker S_i is speaking and speaker S_j is silent, then ξ_{ij} will be positive, since $\max \phi_{ij}$ will be due to the power in $y_i[n]$, not the distant, attenuated copy $y_j[n]$. If both S_i and S_j are speaking, then their crosscorrelation spectrum will exhibit two peaks (symmetric about zero), but our search for a single peak will miss this bimodality and will only locate that which is higher. Under circumstances where the microphone gains are approximately equal, ξ_{ii} will be positive if S_i is the dominant speaker in the current analysis frame.

For every speaker S_i, we compute the sum

$$\Xi_i = \sum_{i \neq j} \xi_{ij} = \sum_{i \neq j} \log_{10} \frac{\max \phi_{ij}}{\phi_{jj}} \tag{11}$$

Per analysis frame, we hypothesize that S_i is speaking only if $\Xi_i > 0$. Otherwise, we assume that the power in $y_i[n]$ is due entirely to some other distant speaker(s) $S_{j \neq i}$, whose microphone signal $y_j[n]$ contains more power.

This algorithm is depicted in Figure 5.

4.1.5. Smoothing

The purpose of smoothing is to fill in the gaps between segments as we found that there is a high fraction of very short segments with short gaps between them. Therefore, we merge any two segments which have less than a 1.2s gap between them; this was found to give optimal segmentation accuracy. Also, since it is hard to detect the exact beginning and ending points for each segment, we padded each segment with 0.5s at the start and end.

4.2. IHM Experiments

In this section, we present our segmenation results and the speech recognition results based on segments provided by our algorithms. We use the miss rate (MS) and false alarm rate (FA) to measure segmentation performance. Given the hypothetical confusion matrix over segment durations for one channel M_i in Table 5, $\text{MS}_i = T_i^{(MS)}/(T_i^{(S)} + T_i^{(MS)})$ and $\text{FA}_i = T_i^{(FA)}/(T_i^{(S)} + T_i^{(FA)})$. Generally we seek systems which exhibit both a low miss rate and a low false alarm rate.

Table 5. *Hypothetical confusion matrix*

System Output	Reference	
	Speech	Non-speech
Speech	$T^{(S)}$	$T^{(FA)}$
Non-speech	$T^{(MS)}$	$T^{(N)}$

When reporting results for an entire meeting, we com-

pute the overall miss rate

$$\mathrm{MS} = \frac{\sum T_i^{(MS)}}{\sum T_i^{(S)} + \sum T_i^{(MS)}} \quad (12)$$

and the overall false alarm rate

$$\mathrm{FA} = \frac{\sum T_i^{(FA)}}{\sum T_i^{(S)} + \sum T_i^{(FA)}} \quad (13)$$

The run-time performance for both algorithms is approximately 0.2 times real-time, as measured on a 2.8GHz Pentium 4 machine.

4.2.1. Segmentation Experiments

Segmentation results are shown in Table 6. As mentioned earlier, the performance of the baseline suffers from a high false alarm rate due to other speaker pickup. Our initial explorations were guided primarily by a desire to lower the false alarm rate.

Table 6. *Segmentation performance on devset (in %)*

System	no smoothing		smoothing	
	MS	FA	MS	FA
baseline	7.2	66.2	—	—
IMTD	54.8	23.8	38.0	30.6
JMXC	33.2	4.2	**16.9**	**13.0**

IMTD with smoothing significantly reduces the false alarm rate, but at the expense of a large increase in the miss rate. This is due to the algorithm's inability to postulate simulataneous speakers. In addition, meetings which exhibit very little channel crosstalk result in high errors because there are no clear peaks in the crosscorrelation.

JMXC significantly decreases both types of error relative to IMTD. This is due to its ability to postulate multiple speakers speaking simultaneosly. Also, the peak crosscorrelation value is a more robust feature than the sample lag at which it occurs.

In Table 7, we show the performance of the JMXC system on individual meetings. This data exhibits large variability, which appears uncorrelated with the microphone type and number of speakers. We think that this variability may be due to unquantified meeting characteristics such as overall degree of crosstalk, general meeting geometry including room acoustics, mean and standard deviation of signal-to-noise ratios and/or microphone variability within a meeting.

We have tabulated the segmentation performance separately for lapel and headset microphone meetings in Table 8.

Table 7. *JMXC segmentation performance (in %)*

Meeting ID	no smoothing		smoothing	
	MS	FA	MS	FA
CMU_20020319-1400	41.9	2.2	19.8	13.5
CMU_20020320-1500	28.8	5.7	11.8	17.4
ICSI_20010208-1430	22.3	4.8	11.1	16.1
ICSI_20010322-1450	22.1	8.7	9.0	17.2
LDC_20011116-1400	18.9	3.5	8.8	8.8
LDC_20011116-1500	36.1	3.1	23.1	13.3
NIST_20020214-1148	45.0	0.9	22.5	7.5
NIST_20020305-1007	47.0	3.2	25.5	9.1

Table 8. *JMXC segmentation performance per mic type (in %)*

Meeting ID	no smoothing		smoothing	
	MS	FA	MS	FA
lapel	32.0	3.5	16.5	13.1
headset	34.4	4.9	17.2	12.9

The numbers suggest that the difference in performance is negligible if at all significant.

We note that both of the explored algorithms actually perform non-silence detection; this includes speech as well as non-verbal sounds such as laughter. Other sources may also be picked up provided their acoustic distance to one microphone is much smaller than to any of the others. We expect that to some degree, non-verbal phenomena coming from the speaker may appear in the transcription and be useful to subsequent components of a meeting transcription system.

4.2.2. Application to Speech Recognition

Table 9 compares the first pass speech recognition performance based on different segmentation systems with the "ideal" segmentation using human labels. We also compute the performance gap in word error rate relative to the ideal.

Table 9. *Speech recognition performance.*

System	Word Error Rate	Performance Gap
baseline	49.6%	25.3%
IMTD	68.6%	73.2%
JMXC	**43.6%**	**10.1%**
human	39.6%	—

JMXC was used to provide segmentation under the Individual Headset Microphone (IHM) condition for the ISL

speech recognizer [15] in the NIST RT-04s evaluation. This system produced a 35.7% word error rate on the evaluation set in the final pass; refer to [15] for details.

5. CONCLUSIONS

We described our automatic speaker segmentation and clustering system for natural, multi-speaker meeting conversations based on multiple distant microphones. The performed experiments show that the system is capable of providing useful speaker information on a wide range of meetings. The system achieved 28.17% speaker diarization error in the NIST RT-04S evaluation. The speech recognition system achieved a 44.5% word error rate when using segments provided by this system in RT-04S.

We also presented a simple, fast algorithm, which requires no prior training, for detecting speech vs non-speech for personal microphone channels. The experiments performed show that the algorithm significantly improves the quality of audio usable for speaker adaptation in speech recognition; our results show only a minor increase in word error rates relative to manually prepared segmentations. The speech recognition system achieved a 35.7% word error rate when using segments provided by this system in RT-04S.

6. ACKNOWLEDGEMENTS

We would like to thank Michael Dambier for providing the best channel selection scripts, Hua Yu for helping with the use of CMUseg-0.5 as a segmenter, and Florian Metze and Christian Fügen for providing feedback about segmentation impact on speech recognition and useful discussion.

7. REFERENCES

[1] S. Burger, V. MacLaren, and H. Yu, "The ISL Meeting Corpus: The Impact of Meeting Type on Speech Style," *ICSLP 2002*, Denver, USA.

[2] E. Shriberg, A. Stolcke, and D. Baron, "Observations on overlap: Findings and implications for automatic processing of multi-party conversation," *Eurospeech 2001*, Aalborg, Denmark.

[3] N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke, "The Meeting Project at ICSI," *HLT 2001*, San Diego, March 2001.

[4] NIST, Rich Transcription 2004 Spring Meeting Recognition Evaluation, `http://www.itl.nist.gov/iad/894.01/tests/rt/rt2004/spring/`

[5] T. Pfau, D. Ellis, and A. Stolcke, "Multispeaker Speech Activity Detection for the ICSI Meeting Recognizer," *ASRU 2001*, Madonna di Campiglio, Italy.

[6] S. Burger and Z. Sloane, "The ISL Meeting Corpus: Categorical Features of Communicative Group Interactions," *NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.

[7] A. Janin, J. Ang, S. Bhagat, R. Dhillon, J. Edwards, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters, B. Wrede, "The ICSI Meeting Project: Resources and Research," *NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.

[8] S. Strassel, M. Glenn, "Shared Linguistic Resources for Human Language Technology in the Meeting Domain," *NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.

[9] V. Stanford, J. Garofolo, "Beyond Close-talk — Issues in Distant speech Acquisition, Conditioning Classification, and Recognition," *NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.

[10] M. Siegler, U. Jain, B. Raj, and R. Stern, "Automatic Segmentation, Classification and Clustering of Broadcast News Audio," *DARPA Speech Recognition Workshop*, Chantilly, Virginia, 1997.

[11] S.S. Chen and P.S. Gopalakrishnan, "Clustering via the Bayesian Information Criterion with Applications in Speech Recognition," *ICASSP 1998*, Seattle, USA, 1998.

[12] P. Delacourt and C.J. Wellekens, "DISTBIC: A Speaker-based Segmentation for Audio Data Indexing," *Speech Communications*, 32, 111-126, 2000.

[13] A. Vandecatseye, J. Martens, "A Fast, Accurate Stream-based Speaker Segmentation and Clustering Algorithm," *Eurospeech 2003*, Geneva, Switzerland, 2003.

[14] The Rich Transcription Spring 2003 Evaluation Plan, `http://www.nist.gov/speech/tests/rt/rt2003/spring/docs/rt03-spring-eval-plan-v4.pdf`.

[15] F. Metze, C. Fügen, Y. Pan, T. Schultz, and H. Yu, "Issues in Meeting Transcription — The ISL Meeting Transcription System," *NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.

[16] B. C. J. Moore, "An Introduction to the Psychology of Hearing," Academic Press, 1997.