SPEAKER TURN SEGMENTATION BASED ON BETWEEN-CHANNEL DIFFERENCES

Daniel P.W. Ellis [1,2] *and Jerry C. Liu* [1]

[1] LabROSA, Dept. of Electrical Engineering, Columbia University, NY NY
[2]International Computer Science Institute, Berkeley CA
{dpwe,jliu}@ee.columbia.edu

ABSTRACT

Multichannel recordings of meetings provide information on speaker locations in the timing and level differences between microphones. We have been experimenting with cross-correlation and energy differences as features to identify and segment speaker turns. In particular, we have used LPC whitening, spectral-domain cross-correlation, and dynamic programming to sharpen and disambiguate timing differences between mic channels that may be dominated by noise and reverberation. These cues are classified into individual speakers using spectral clustering (i.e. defined by the top eignenvectors of a similarity matrix). We show that this technique is largely robust to precise details of mic positioning etc., and can be used with some success with data collected from a number of different setups, as provided by the NIST 2004 Meetings evaluation.

1. INTRODUCTION

In contrast to the major challenges of accurate recognition of speech recognition from meeting recordings (and how to make use of any results), we have been looking at information that can be extracted from such recordings without resorting to such heavyweight processing. One kind of information that should be relatively prominent within the data (and hence feasible to extract) is the pattern of speaker turns – i.e. when the active speaker changes, and patterns of transitions between speakers. In earlier work, we have used these patterns, derived from hand-marked speaker turns, to segment meetings into distinct dialogues [1].

Here we consider the practical problem of recovering speaker turn information directly from multichannel audio recordings, without any manual annotation. For maximum applicability, we are interested in developing algorithms that make as few assumptions as possible about the nature of the multiple audio channels; in particular, we do not employ any information about the actual locations of the different microphones. All we can assume is that the sound field has been sampled at several different points, resulting in several ambient recordings. We envisage a future scenario in which several participants in a meeting may have hand-held recording devices which they place at arbitrary points on the table; the speaker turn analysis system has access to several of these (possibly synchronized) recordings, but has no control over, or information about, where the sensors are placed.

In particular, this study takes advantage of the NIST 2004 Meeting Evaluation [2] which includes development data comprising real meetings recorded at four independent sites with a range of equipment and configurations. While our principal development has been on the ICSI meeting data [3], this dataset has allowed us to evaluate on meeting data recorded at CMU, NIST, and the LDC.

Most previous work on speaker segmentation has focused on single channel data such as mono recordings of broadcast audio, and has relied exclusively on changes in the statistical properties of the speech spectrum to detect change points (e.g. [4, 5] among many others). Here, we deliberately avoid using this conventional approach to consider instead how well the problem can be solved without modeling the source spectra, but looking only at the differences between the signals received by each sensor. Ultimately, of course, it is best to use both sources of information, but for now we consider new approaches made possible by the availability of multiple, distant-mic recordings.

The next section describes our speaker turn segmentation system based on timing differences. In section 3, we describe how the approach was tested with the NIST RT04 Meeting data, and present the results on the different meetings in that set. Finally, section 4 discusses the performance and suggests some future developments.

2. USING TIMING CUES

Figure 1 gives an overview of the speaker turn detection system based on timing difference cues. Microphone channels are compared pairwise; in our system, we arbitrarily compared two distinct pairs of mics to create a two-dimensional time-difference feature vector, but the approach could easily be extended to make comparisons between more channels.

The different distances between each microphone and a particular speaker will result in time differences in the

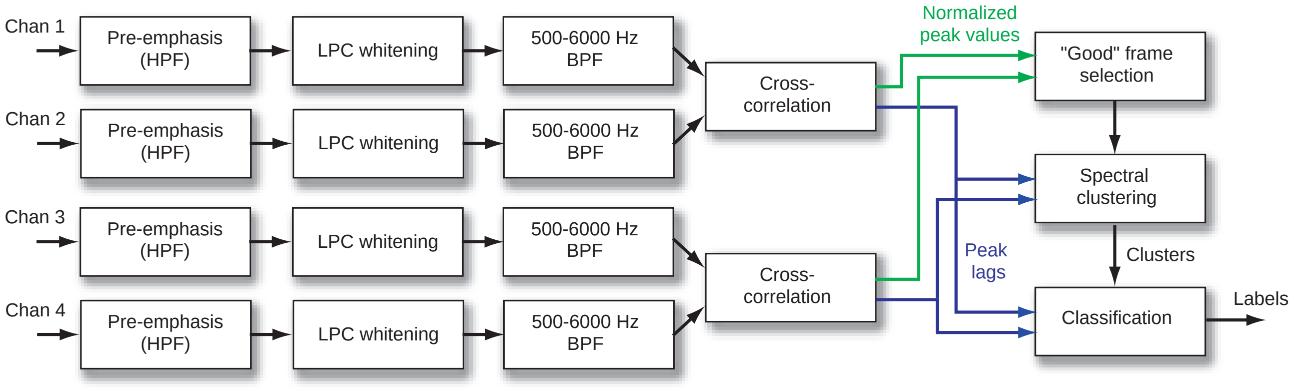

Fig. 1. Overview of the timing difference classification system.

direct-path voice detected by those mics, and the time difference can be estimated by finding the 'lag' ℓ that maximizes the (normalized) short-time cross-correlation centered at time M:

$$\rho_{ij}[\ell, M] = \frac{\sum_{n=-N/2}^{N/2} m_i[M+n] \cdot m_j[M+n+\ell]}{\sqrt{\sum m_i^2 \sum m_j^2}} \quad (1)$$

where $m_i[n]$ is the sample sequence from microphone i etc., the cross-correlations are calculated over an $N+1$ point window, and the denominator, summing over the same range, normalizes the results to lie within -1..1 with 1 indicating a perfect match. The best lag for each mic pair $\{i, j\}$ at each time M is simply:

$$\ell^*_{ij}[M] = \underset{\ell}{\text{argmax}} \ \rho_{ij}[\ell, M] \quad (2)$$

Rather than cross-correlating the raw waveforms, we found the preprocessing stages shown in fig 1 to give significant improvements in robustness and precision. First, the recorded waveforms are pre-emphasized by filtering with the 2-point FIR filter $\{1 - 1\}$ – i.e. a zero at d.c. – to strongly attenuate low frequencies, where background noise such as air conditioning tends to dominate, and to flatten the natural spectral rolloff of speech.

To further flatten the signals, a 12th order LPC model is fit to a 32 ms (512 sample) window every 16 ms, then the signal is filtered by the inverse of this model to obtain the residual, in which the broad spectral structure captured by the LPC model has been removed. This eliminates strong signal resonances that would otherwise appear as strong 'ringing' in the cross-correlations, and equalizes the influence of all frequency bands.

However, since energy in the low frequency region (below about 500 Hz) and in the very highest regions (e.g. above 6 kHz, where the mic may disappear due to anti-aliasing filters) holds little information about the target voices, it is filtered out of the residual with 16th order Chebyshev-2 filter providing 60 dB of rejection outside the passband.

Figure 2 compares the short-term cross-correlations for a short segment with and without the preprocessing. The combination of low-frequency noise and strong resonances in the raw signal lead to broad peaks and ambiguous peaks in the cross correlation; the preprocessed signal leads to very precise timing difference peaks, with a clear distinction between the two dominant speakers indicated in the ground-truth speaker labels shown below each pane.

Cross-correlation results are more robust when the correlation window size is increased, provided the underlying time difference remains constant. We found a 250 ms window to be good compromise. Then the lag corresponding to the overall peak in the cross correlation is recorded as the time delay estimate. (We have tried a fractional-sample resolution estimator, based on fitting a linear slope to the phase of the Fourier transform of the cross-correlation in the vicinity of the peak, but sample-level quantization was adequate.) We experimented with exploiting temporal constraints to remove outliers from the sequence of time estimates: we tried both median filtering, and a dynamic programming approach that selected among the top 10 cross-correlation peaks observed at each time frame to find a path that optimized a quality metric comprising both peak height and lag continuity, but both approaches seemed to introduce more errors than they were able to resolve.

2.1. "Good" peak selection

The whitened example in figure 2 is fairly clean, but in general the cross correlation may not find any time lags that make the two channels very similar – for instance, if one channel is dominated by a noise that barely registers in the other channel, or if there are multiple conflicting sound sources. The next stage of the system seeks to identify the separate speakers present in the recording by searching for clusters in the multidimensional space defined by the lag

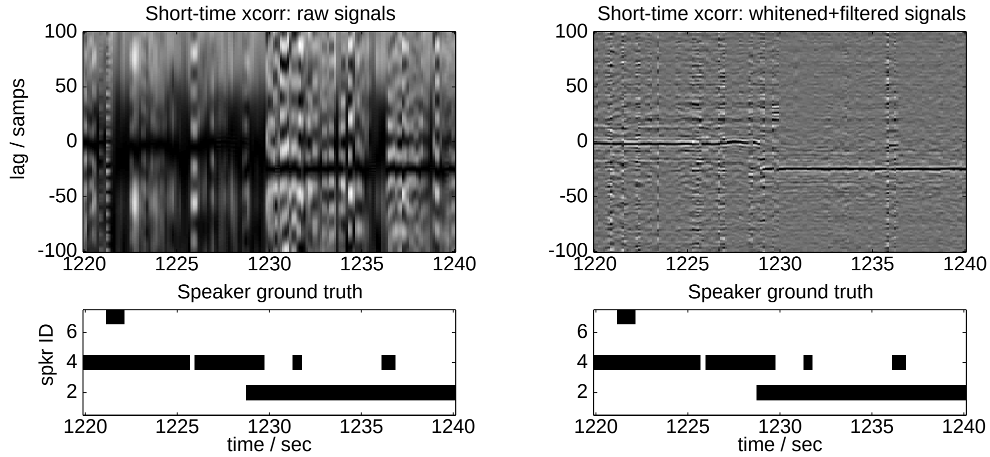

Fig. 2. Short-time cross-correlations on raw signals (left panel) and after LPC whitening and filtering (right panel). Vertical axis is lag; horizontal axis is time; darker pixels indicate higher normalized cross-correlation. The ground-truth speaker labels are shown below each pane; there is a visible transition between speaker 2 and speaker 4 around t = 1229 sec.

values of the cross-correlation peaks, but spurious points resulting from poorly-matching frames will pollute this process. Thus, we use the peak *value* of the normalized cross correlation $\rho_{ij}[\ell^*]$ at each time frame to decide if a particular lag value is 'reliable'. Although the absolute value of this peak varies considerably depending on the recording conditions, we discard, for each mic pair, the time frames whose normalized cross correlation peak value is below the median over the whole meeting. Clustering is then based only on time frames for which every mic pair returns a 'good' value, but this is event strongly correlated between channels so almost half the time frames are selected as 'good'. Since these points are being used only to define the clusters, it is acceptable to be conservative in selecting these points; we will subsequently go back and classify all time frames based on the clusters we derive.

2.2. Spectral clustering

Clustering of these 'good' points in two-dimensional lag space is performed by a simplified form of "spectral clustering" [6]. We form an affinity matrix of the similarities between every point based on a Gaussian local similarity function. If $\mathbf{L}[M]$ is the vector of time differences from time frame M, then the affinity matrix $\mathbf{A}$ has elements:

$$a_{mn} = exp\{-\|\mathbf{L}[m] - \mathbf{L}[n]\|^2/2\sigma^2\} \qquad (3)$$

where σ is a 'cluster radius' parameter indicating how distant two points can be before their affinity; we used $\sigma = 10$ samples (or about 600 μs) based on our initial investigations; for well-separated clusters, the results should be stable over a wide range. Points lying within a single cluster will tend to have affinities close to 1 with all their cluster mates, and close to zero for points in other clusters.

Spectral clustering considers the eigenvectors of this affinity matrix. An intuitive understanding comes from imagining a finite state system with each point as a state, and the affinity matrix as proportional to the transition probabilities between the states. A particular point/state will tend to make transitions only to other points within its same cluster, so the eigenvectors of the transition matrix, which correspond to 'steady-state' state occupancy distributions, will tend to identify the separate clusters [7][1].

The eigenvalues of the unnormalized affinity matrix indicate roughly the number of points 'involved' in each eigenvector, so we consider only the eigenvectors whose eigenvalues are larger than 1% of the total number of points. Then for each eigenvector we decide if it is defining a single cluster (most values either close to zero or close to one) or 'splitting' an existing cluster (values spread between -1 and 1) by looking at the ratio of smallest to largest values. (Eigenvectors with many values close to -1 are flipped in sign). If this ratio is less than 0.5, then all points for which the eigenvalue dimension is greater than the mean over the whole vector are assigned to a new cluster; otherwise, the eigenvector is ignored. While this procedure is highly heuristic and has not been carefully optimized, in the ideal conditions when clusters are well defined and separated, it gives good results.

Cluster models are given by the means of each set of points resulting from this procedure, and a single variance for each lag dimension is calculated from the distance between each clustered point and its cluster mean. The left pane of figure 3 shows the "good" points for an example meeting in a two-dimensional lag space; the coloring indicates the separate clusters found.

[1] Thanks to Brendan Frey for this explanation

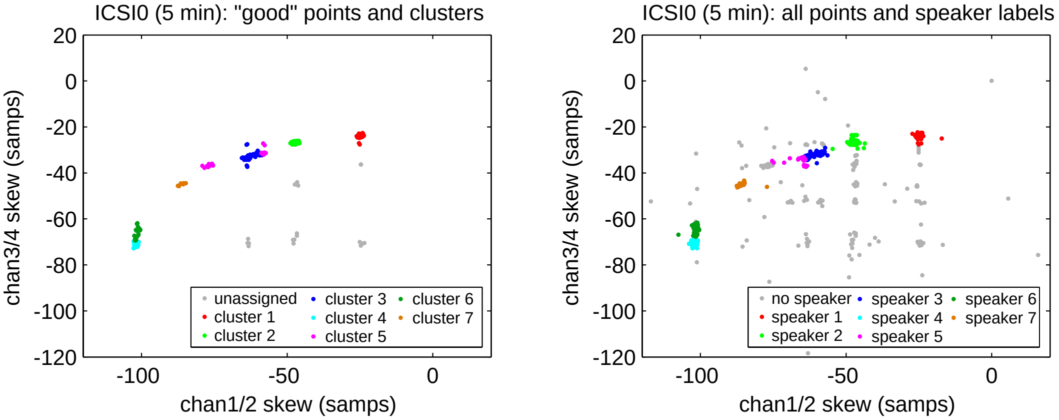

Fig. 3. Clustering of time-skew values. Left pane shows the 'good' points (high peak cross-correlation) extracted from a 5 minute meeting excerpt, colored according to the clusters they seeded; the right pane shows the lag values from all time frames of the excerpt, colored according to their final cluster assignments. About 20% of these points lie outside the range shown in the plot.

2.3. Classification

The final speaker turn assignments are obtained by classifying every time frame as belonging to one of the clusters; points will low cross-correlation peaks that were excluded from the cluster formation may still, in fact, show the appropriate best lag times, and will be correctly labeled by this stage. However, genuinely spurious lags will not give meaningful results, and may actually arise from conditions in which nobody is speaking. Thus, points whose Mahalanobis distance from any cluster is greater than 4 result in time frames labeled with no speaker active; all remaining frames are labeled as having a single speaker active, corresponding to the closest cluster. The right pane of fig 3 shows the final cluster assignments of all the frames.

3. EVALUATION

The NIST development data consisted of eight meeting excerpts, each of around 10 min duration, recorded at ICSI, CMU, NIST, and LDC [2]. Each set except CMU had multiple distant mics; we arbitrarily chose four from each for our evaluation, and for the CMU sets, we formed two pairs by comparing the single mic to two of the lavalier mics (i.e. the single distant mic participated in both pairs). Each meeting was described by a hand-marked annotation file giving the start time, duration, speaker identity, and words of each utterance, derived from headset or lavalier mic channels. We converted this into a binary speaker activity matrix, where each row described one speaker, each column referred to one time step, and each cell indicated whether that speaker was active at that time. We used a time quanta of 250 ms to match our timing analysis frames. Because of speaker overlaps, it is possible for several cells to be set in each column. We represented the output of our speaker turn detection system in the same format; however, since we find only one speaker label at each time step, there can be at most one cell active in each column for our system's output.

The scoring this task is complicated because of the ambiguity in associating the anonymous distinct speakers identified by the system with the speakers in the reference transcript. We implemented a greedy scheme, where the overlap (number of cells correctly labeled active) was calculated for all system speakers against all reference speakers, then correspondences were made by choosing the largest overlap, removing that reference/system pair from the table, then repeating. Any unassigned speakers (i.e. when the system and reference differed in the number of speakers) were matched to all-silent channels. The rows of the system output matrix were then permuted to match the reference matrix.

Differences between these matrices represent errors; cells set in the reference but clear in the system are deletions (false rejects), and cells clear in the reference but set in system are insertions (false detects). The total number of insertions and deletions across all channels is calculated at each time step, and the total error at that time step is the larger of the two (i.e. an insertion-plus-deletion pair counts as a single 'substitution' error). The overall error rate is the total errors summed across all time expressed as a proportion of the total number of active cells in the reference. This is a quantized approximation to the error metric proposed in [2].

Table 1 shows the results for the eight recordings from the development set. The overall error rates are high, exceeding 50% for seven of the eight cases, and reaching 85% for one of the CMU sets. Note, however, that random guessing all cells independently results in error rates above 100%

Meeting	Ins	Del	Err	Guess1
LDC_20011116-1400	10.5%	65.4%	66.5%	52.7%
LDC_20011116-1500	19.9%	76.3%	77.3%	60.1%
NIST_20020214-1148	3.9%	56.8%	58.0%	37.3%
NIST_20020305-1007	17.6%	46.1%	**49.1%**	77.9%
ICSI_20010208-1430	23.6%	55.2%	**59.6%**	69.8%
ICSI_20010322-1450	29.6%	60.1%	63.3%	62.4%
CMU_20020319-1400	15.3%	84.3%	85.3%	54.0%
CMU_20020320-1500	11.7%	65.0%	65.8%	37.4%

Table 1. Speaker turn detection error rates for the NIST RT04 development set. Highlighted are the two cases where the error rate improves on the "Guess1" result obtained by hypothesizing that a single speaker speaks for the whole duration.

due to the many opportunities for insertions. On the other hand, the optimal correspondence between system and reference speakers found in scoring can make some random choices appear quite good. As a baseline, the "Guess1" column in the table gives the error rate that results from a system guessing that the meeting consists of just one speaker active the whole time, who, in scoring, is aligned with the single most active speaker in the reference. Meeting excerpts that are dominated by a particular speaker (such as the first NIST example and the second CMU example) have a particularly low error rate on this measure. In fact, it is only in two of the examples (second NIST and first ICSI) that our system beats this uninformed baseline. We note, however, that the insertion rates are mostly quite low (much lower than "Guess1" would obtain), and there may be situations requiring low false alarms where our approach has more value than the bottom-line error rates suggest.

Figure 4 gives some insight into the system performance by comparing the speaker activity matrices for reference and system output for a single dataset. First, the system found only three speakers although seven were present in the transcript; speakers 2 to 5 in the system output are 'silent' speakers inserted by the scoring. Second, system speaker 1 appears to be detecting reference speaker 6 i.e. this reference speaker has been 'split' into two speakers (1 and 6) by the system; since scoring disallows mapping two speakers to one, system speaker 1 has been aligned to reference speaker 1 who happens to have a small overlap. However, the missed reference speakers have very little activity (particularly speakers 3 and 5), so it is not so surprising that they were missed by the system. Also visible in the figure is the general tendency of the system to under-produce. Even when speaker 6 is being reliably identified during his long monologue around $t = 400$ sec, the system marks many frames as silent (i.e. the extracted lags were implausible for those frames).

4. DISCUSSION AND CONCLUSIONS

Our results show that the between-channel timing information revealed by cross-correlation is sufficient to gain some information about speaker activity and turns, although clearly this first effort leaves much room for improvement. We are encouraged that an approach developed on a small amount of data from a particular recording setup (at ICSI) seems to perform equally well on recordings from other sites; indeed, our lowest error rate was achieved on one of the NIST recordings. This underlines the point that our approach makes very few assumptions about the geometry of the microphone arrangements – or indeed the signal characteristics of the speakers and recording equipment.

Several directions towards improving the results are suggested by these results. The problem of undergeneration is best explained by the right panel of fig 3. Despite our preprocessing, many frames result in peak cross-correlation lags that are meaningless, due to noise, interference between simultaneous sources, or chance correlations in the source signal. However, notice the 'streaks' below the points labeled as speakers 1 and 2; in this case, the chan1/2 skew is (presumably) correct, but the chan3/4 skew is inconsistent. This may be because chans 3 and 4 were more distant, and hence corrupted by noise, or they may even be reflecting a second, simultaneous speaker, since there seem to be significant numbers of frames aligned with the speaker 4/speaker 6 cluster on that axis. This suggests a classification approach in which each lag dimension is classified separately, with majority reporting, or even multiple labels (simultaneous speaker outputs) in some cases. Clearly, using more than two lag calculations will improve reliability for such a scheme, and perhaps make it feasible to use the peak cross-correlation value as a 'feature reliability' index for classification in addition to its current use in cluster formation.

Even if only a small number of lag dimensions are to be used, some mic pairs will work better than others, in terms of giving reliable cross-correlation results more often. We

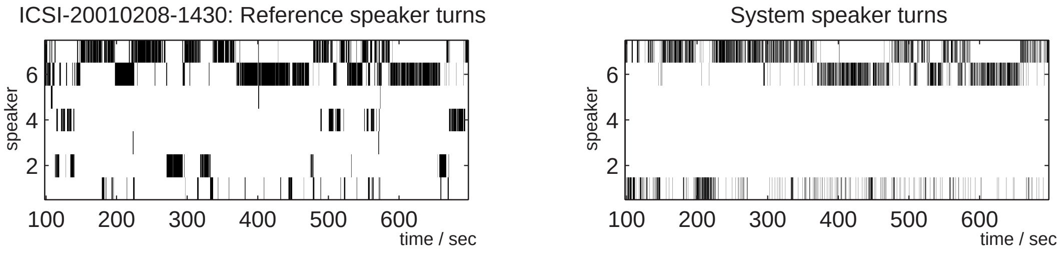

Fig. 4. Comparison of reference and speaker activity matrices for a single meeting excerpt. In this case, the system found three speakers although seven were present in the transcript; additional 'silent' speaker rows were inserted into the system output by the scoring correspondence algorithm. The overall error rate is 59.6% for this example.

chose mic pairs arbitrarily, but a simple approach of choosing mic pairs with the highest average peak cross correlation value should improve system performance.

The issue of splitting single speakers into multiple clusters, visible in the speaker 4/6 continuum mentioned above, and also shown by the division of reference speaker 6 into system speakers 1 and 6 in fig 4 suggests some genuine underlying ambiguity, since a given speaker is not rigidly fixed during a meeting, but may shift in her seat, turn her head, etc., resulting in a spread of spatial cue values. This presents a good opportunity to exploit the source spectral characteristics mentioned in the introduction as the basis of previous work in speaker segmentation: By calculating the cepstral distributions of each system speaker, it should be possible to identify speakers whose spectral behavior is nearly identical, and merge the labels. This approach could even track a speaker who shifts seats during a meeting.

Finally, this paper has considered only timing differences. We have also investigated the use of between-channel level differences (ratios), which in theory factor out source spectral characteristics and thus should also be characteristic of particular source locations. Level differences have the attraction that small timing uncertainties or drift between channels have no influence, whereas cross-correlation analysis relies on good synchrony for the duration of the meeting. However, our preliminary results (to be reported later) show that level differences, even for the high-energy portions of the signal, have very high variances, possibly reflecting the complex spectral responses of the reverberant coupling between speakers and distant mics. We will, however, further investigate this cue as another possible source of information for speaker turn estimation.

5. ACKNOWLEDGMENTS

This work was supported in part by the NSF under grant IIS-0212396 "Mapping Meetings", and by the European Union under the EU IST project "M4: multi-modal meeting manager" (IST-2001-34485).

6. REFERENCES

[1] Steve Renals and Daniel P.W. Ellis, "Audio information access from meeting rooms," in *Proc. ICASSP*, Hong Kong, 2003.

[2] John Garofolo et al., "NIST rich transcription 2004 spring meeting recognition evaluation," 2004, `http://nist.gov/speech/tests/rt/rt2004/spring/`.

[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," in *Proc. HLT*, 2001, pp. 246–252.

[4] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA Broadcast News Workshop*, 1997.

[5] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion," in *Proc. DARPA Broadcast News Workshop*, 1998.

[6] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in NIPS*. MIT Press, Cambridge MA, 2001.

[7] M. Meila and J. Shi, "A random walks view of spectral segmentation," in *Proc. AI and Statistics (AISTATS)*, 2001.