Boosting-Based Multimodal Speaker Detection for Distributed Meetings

Cha Zhang*, Pei Yin†, Yong Rui*, Ross Cutler* and Paul Viola*
* Microsoft Research, One Microsoft Way, Redmond, USA
†College of Computing, Georgia Institute of Technology

Abstract— **Speaker detection is a very important task in distributed meeting applications. This paper discusses a number of challenges we met while designing a speaker detector for the Microsoft RoundTable distributed meeting device, and proposes a boosting-based multimodal speaker detection (BMSD) algorithm. Instead of performing sound source localization (SSL) and multi-person detection (MPD) separately and subsequently fusing their individual results, the proposed algorithm uses boosting to select features from a combined pool of both audio and visual features simultaneously. The result is a very accurate speaker detector with extremely high efficiency. The algorithm reduces the error rate of SSL-only approach by 47%, and the SSL and MPD fusion approach by 27%.**

I. Introduction

As globalization continues to spread throughout the world economy, it is increasingly common to find projects where team members reside in different time zones. To provide a means for distributed groups to work together on shared problems, there has been an increasing interest in building special purpose devices and even "smart rooms" to support distributed meetings [1], [2], [3], [4]. These devices often contain multiple microphones and cameras. An example device called RoundTable is shown in Figure 1(a). It has a six-element circular microphone array at the base, and five video cameras at the top. The captured videos are stitched into a 360 degree panorama, which gives a global view of the meeting room. The RoundTable device enables remote group members to hear and view the meeting live online. In addition, the meetings can be recorded and archived, allowing people to browse them afterward.

One of the most desired features in such distributed meeting systems is to provide remote users with a close-up of the current speaker which automatically tracks as a new participant begins to speak [2], [3], [4]. The speaker detection problem, however, is non-trivial. Two video frames captured by our RoundTable device are shown in Figure 1(b). During the development of our RoundTable device, we faced a number of challenges:

- People do not always look at the camera, in particular when they are presenting on a white board, or working on their own laptop.
- There can be many people in a meeting, hence it is very easy for the speaker detector to get confused.
- The color calibration in real conference rooms is very challenging. Mixed lighting across the room make it very difficult to properly white balance across the panorama images. Face detection based on skin color is very unreliable in such environments.
- To make the RoundTable device stand-alone, we have to implement the speaker detection module on a DSP chip with the budget of 100 million instructions per second (MIPS). Hence the algorithm must be extremely efficient. Our initial goal is to detect speaker at the speed of 1 frame per second (FPS) on the DSP.
- While the RoundTable device captures very high resolution images, the resolution of the images used for speaker detection is low due to the memory and bandwidth constraints of the DSP chip. For people sitting at the far end of the table, the head size is no more than 10×10 pixels, which is beyond the capability of most modern face detectors [5].

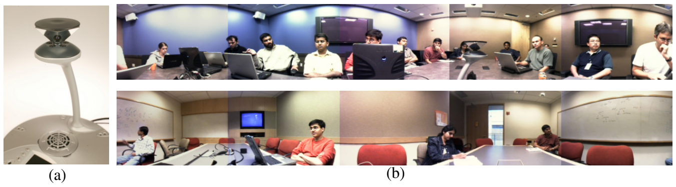

Fig. 1. RoundTable and its captured images. (a) The RoundTable device. (b) Captured images.

In existing distributed meeting systems, the two most popular speaker detection approaches are through sound source localization (SSL) [6], [7] and SSL combined with face detection using decision level fusion (DLF) [3], [4]. However, they both have difficulties in practice. The success of SSL heavily depends on the levels of reverberation noise (e.g., a wall or whiteboard can act as an acoustic mirror) and ambient noise (e.g., computer fans), which are often high in many of the meeting rooms. If a face detector is available, decision level fusion can certainly help improve the final detection performance. However, building a reliable face detector in the above mentioned environment is itself a very challenging task.

In this paper, we propose a novel boosting-based multimodal speaker detection (BMSD) algorithm, which attempts to address most of the challenges listed above. The algorithm does not try to locate human faces, but rather heads and upper bodies. By integrating audio and visual multimodal information into a single boosting framework at feature level, it explicitly learns the difference between speakers and non-speakers. Specifically, we use the output of an SSL algorithm to compute features for windows in the video frame. These features are then placed in the same pool as the appearance and motion visual features computed on the gray scale video frames, and selected by the boosting algorithm automatically. The BMSD algorithm reduces the error rate of SSL-only solutions by 47% in our experiments, and the SSL and person detection DLF approach by 27%. The BMSD algorithm is super-efficient. It achieves the above performance with merely 20 SSL and Haar basis image features. With pruning, we show the average number of features computed for each detection window drops further to less than two. Lastly, BMSD does not require high frame rate video analysis or tight AV synchronization, which is ideal for our application.

The paper is organized as follows. Related work is discussed in Section II. The BMSD algorithm is described in Section III. Experimental results and conclusions are given in Section IV and V, respectively.

II. Related Work

Audio visual information fusion has been a popular approach for many research topics including speech recognition [8], [9], video segmentation and retrieval [10], event detection [11], [12], speaker change detection [13], speaker detection [14], [15], [16], [17] and tracking [18], [19], etc. In the following we describe briefly a few approaches that are closely related to this paper.

Audio visual synchrony is one of the most popular mechanisms to perform speaker detection. Explicitly or implicitly, many approaches measure the mutual information between audio visual signals and search for regions of high correlation and tag them as likely to contain the speaker. Representative works include Hershey and Movellan [16], Nock et al. [20], Besson and Kunt [14], and Fisher et al. [15]. Cutler and Davis [21] instead learned the audio visual correlation using a time-delayed neural network (TDNN). Approaches in this category often need just a single microphone, and rely on the synchrony only to identify the speaker. Most of them require a good frontal face to work well.

Another popular approach is to build graphical models for the observed audio visual data, and infer the speaker location probabilistically. Pavlović et al. [17] proposed to use dynamic Bayesian networks (DBN) to combine multiple sensors/detectors and decide whether a speaker is present in front of a smart kiosk. Beal et al. [22] built a probabilistic generative model to describe the observed data directly using an EM algorithm and estimated the object location through Bayesian inference. Brand et al. [11] used coupled hidden Markov models to model the relationship between audio visual signals and classify human gestures. Graphical models are a natural way to solve multimodal problems and are often intuitive to construct. However, their inference stage can be time-consuming and would not fit into our tight computation budget.

Audio visual fusion has also been applied for speaker tracking, in particular those based on particle filtering [18], [19], [23], [24]. In the measurement stage, audio likelihood and video likelihood are both computed for each sample to derive its new weight. It is possible to use these likelihoods as measures for speaker detection, though such an approach can be very expensive if all the possible candidates in the frame need to be scanned.

In real-world applications, the two most popular speaker detection approaches are still SSL-only and SSL combined with face detection for decision level fusion (DLF) [2], [3], [4]. For instance, the iPower 900 teleconferencing system from Polycom uses an SSL-only solution for speaker detection [6]. Kapralos et al. [3] used a skin color based face detector to find all the potential faces, and detect speech along the directions of these faces. Yoshimi and Pingali [4] took the audio localization results and used a face detector to search for nearby faces in the image. Busso et al. [1] adopted Gaussian mixture models to model the speaker locations, and fused the audio and visual results probabilistically with temporal filtering.

As mentioned earlier, speaker detection based on SSL-only is sensitive to reverberation and ambient noises. The DLF approach, on the other hand, has two major drawbacks in speaker detection. First, when SSL and face detection operate separately, the correlation between audio and video, either at high frame rate or low frame rate, is lost. Second, a full-fledged face detector can be unnecessarily slow, because many regions in the video can be skipped if their SSL confidence is too low. Limiting the search range of face detection near SSL peaks, however, is difficult because it is hard to find a universal SSL threshold for all conference rooms. Moreover, this can introduce bias towards the decision made by SSL. The proposed algorithm uses a boosted classifier to perform feature level fusion of information in order to minimize computation time and maximize robustness. We will show the superior performance of BMSD by comparing it with the SSL-only and DLF approaches in Section IV.

III. Boosting-Based Multimodal Speaker Detection

Our speaker detection algorithm adopts the popular boosting algorithm [25], [26] to learn the difference between speakers and non-speakers. It computes both audio and visual features, and places them in a common feature pool for the boosting algorithm to select. This has a number of advantages. First, the boosting algorithm explicitly learns the difference between a speaker and a non-speaker, thus it targets the speaker detection problem more directly. Second, the final classifier can contain both audio and visual features, which implicitly explores the correlation between the audio and visual information if they coexist after the feature selection. Third, thanks to the cascade pruning mechanism introduced in [5], audio features selected early in the learning process will help eliminate many non-speaker windows, which greatly improves the detection speed. Lastly, since all the audio visual features are in the same pool, there is no bias toward either modality.

In the following we first introduce the visual and audio features, then present the boosting learning algorithm. We also briefly discuss the SSL-only and SSL and multi-person detector (MPD) DLF algorithms, which will be used in Section IV to compare against BMSD.

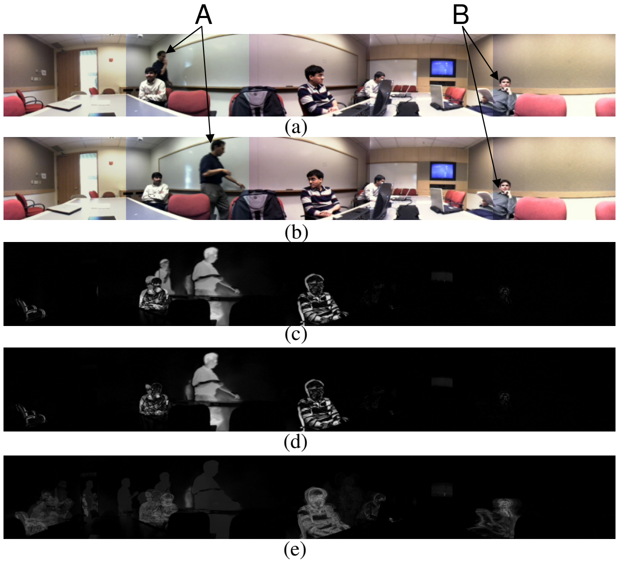

Fig. 2. Motion filtering in BMSD. (a) Video frame at $t-1$. (b) Video frame at t. (c) Difference image between (a) and (b). (d) Three frame difference image. (e) Running average of the three frame difference image.

A. Visual Features

Appearance and motion are two important visual cues to tell a person from the background [27]. The appearance cue is generally derived from the original video frame, hence we focus on the motion cue in this section.

The simplest motion filter is to compute the frame difference between two subsequent frames [27]. When applied to our testing sequences, we find two major problems, demonstrated in Figure 2. Figure 2(a) and (b) are two subsequent frames captured in one of the recorded meetings. Person A was walking toward the whiteboard to give a presentation. Because of the low frame rate the detector is running at, the difference image (c) has two big blobs for person A. Experiments show that the boosting algorithm often selects motion features among its top features, and such ghost blobs tend to cause false positives. Person B in the scene shows another problem. In a regular meeting, often someone in the room stays still for a few seconds, hence the frame difference of person B is very small. This tends to cause false negatives.

To address the first problem, we introduce a simple three frame difference mechanism to derive the motion pattern. Let I_t be the input image at time t, we compute:

$$M_t = min\Big(|I_t - I_{t-1}|, |I_t - I_{t-2}|\Big). \tag{1}$$

As shown in Figure 2(d), Equation 1 detects a motion region only when the current frame has large difference with the previous two frames, and can thus effectively remove the ghost blobs in Figure 2(c). Note three frame difference was used in background modeling before [28]. Equation 1 is a variation that does not require a future frame I_{t+1}, hence it does not incur 1 second extra delay.

We add another frame as Figure 2(e), which is the running average of the three frame difference images:

$$R_t = \alpha M_t + (1-\alpha)R_{t-1}. \tag{2}$$

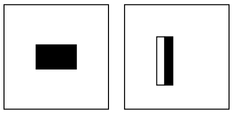

Fig. 3. Example rectangle features shown relative to the enclosing detection window. Left: 1-rectangle feature; right: 2-rectangle feature.

The running difference image accumulates the motion in the history, and captures the long-term motion of people in the room. It can be seen that even though person B moved very slightly in one particular frame, the running difference image is able to capture his body clearly.

Despite their simplicity, the two added images reduce the detection error significantly. We also experimented by replacing Equation 1 with a background subtraction module such as the one in [29]. Only marginal improvement was observed with a relatively high computational cost (for our application).

Given the three frames I_t, M_t and R_t, we use two kinds of simple visual features to train the classifier, as shown in Figure 3. Similar to [5], these features are computed for each detection window of the video frame. Note each detection window will cover the same location on all three images. The 1-rectangle feature on the left of Figure 3 is computed on the difference image and running difference image only. Single rectangle features allow the classifier to learn a data dependent and location dependent difference threshold. The 2-rectangle feature on the right is applied to all three images. This arrangement is to guarantee that all the features have zero-mean, so that they are less sensitive to lighting variations. For our particular application, we find adding more features such as 3-rectangle or 4-rectangle features gives very limited improvements on the classifier performance.

B. Audio Features

The raw output from the microphone arrays is a multi-channel audio signal. To derive speaker location information from the audio signal, we use the SSL algorithm developed in [7], which combines the advantages of the steered beam SSL and the one-step time-delay-of-arrival SSL. In the case of RoundTable, the microphone array is shared between the SSL and sound capture, and the geometry is circular because it provides significantly superior sound quality. Given this circular array geometry, the SSL only provides 1D azimuth of the sound source location through hypothesis testing. We obtain a 1D array of numbers between 0 and 1, which represents the likelihood of the sound source coming from each tested horizontal angle [7], denoted as $L_a(\theta), \theta = 0, \alpha, \cdots, 360-\alpha$. The hypothesis testing is done for every α degrees. In the current implementation, $\alpha = 4$ gives good results. We perform SSL at 1 FPS, which is synchronized to video within 100 milliseconds. For computing audio features for detection windows in the video frames, we map $L_a(\theta)$ to the image coordinate as:

$$L_a(x) = L_a\big(\theta(x)\big), x = 1, 2, \cdots, X, \tag{3}$$

where X is the width of the panoramic images, and $\theta(x)$ is the mapping function.

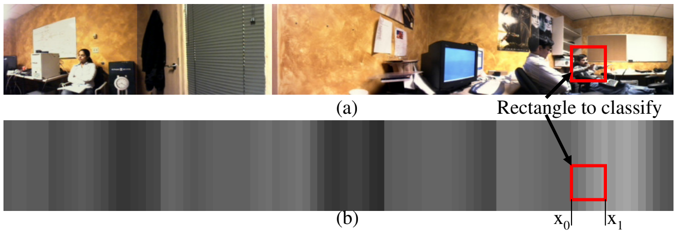

Fig. 4. Compute SSL features for BMSD. (a) Original image. (b) SSL image. Bright intensity represents high likelihood. Note the peak of the SSL image does not correspond to the actual speaker (the right-most person), indicating a failure for the SSL-only solution.

1. $\frac{L^l_{\max} - L^g_{\min}}{L^g_{\max} - L^g_{\min}}$	2. $\frac{L^l_{\min} - L^g_{\min}}{L^g_{\max} - L^g_{\min}}$	3. $\frac{L^l_{\text{avg}} - L^g_{\min}}{L^g_{\max} - L^g_{\min}}$
4. $\frac{L^l_{\text{mid}} - L^g_{\min}}{L^g_{\max} - L^g_{\min}}$	5. $\frac{L^l_{\max}}{L^l_{\min}}$	6. $\frac{L^l_{\max}}{L^l_{\text{avg}}}$
7. $\frac{L^l_{\min}}{L^l_{\text{avg}}}$	8. $\frac{L^l_{\text{mid}}}{L^l_{\text{avg}}}$	9. $\frac{L^l_{\max} - L^l_{\min}}{L^l_{\text{avg}}}$
10. $\frac{L^l_{\max}}{L^g_{\max}}$	11. $\frac{L^l_{\min}}{L^g_{\max}}$	12. $\frac{L^l_{\text{avg}}}{L^g_{\max}}$
13. $\frac{L^l_{\text{mid}}}{L^g_{\max}}$	14. $\frac{L^l_{\max} - L^l_{\min}}{L^g_{\max}}$	15. $L^g_{\max} - L^l_{\max} < \epsilon$

Fig. 5. Audio Features extracted from the SSL likelihood function. Note the 15th feature is a binary one which tests if the local region contains the global peak of SSL.

It is not immediately clear what kind of audio features can be computed for each detection window from the above 1D likelihood array. One possibility is to create a 2D image out of the 1D array by duplicating the values along the vertical axis, as shown in Figure 4(b) (an similar approach was taken in [30]). One can treat this image the same as the other ones, and compute rectangle features such as those in Figure 3 on this image. However, the local variation of SSL is a very poor indicator of the speaker location. We instead compute a set of audio features for each detection window with respect to the whole SSL likelihood function. The global maximum, minimum and average SSL output are first computed as $L^g_{\max} = \max_x L_a(x)$, $L^g_{\min} = \min_x L_a(x)$ and $L^g_{\text{avg}} = \frac{1}{X}\sum_x L_a(x)$, respectively. Let the left and right boundaries of a detection window be x_0 and x_1. Four local numbers are computed as follows: local maximum $L^l_{\max} = \max_{x_0 \le x \le x_1} L_a(x)$; local minimum $L^l_{\min} = \min_{x_0 \le x \le x_1} L_a(x)$; local average $L^l_{\text{avg}} = \frac{1}{x_1 - x_0}\sum_{x_0 \le x \le x_1} L_a(x)$ and middle output $L^l_{\text{mid}} = L_a(\frac{x_0 + x_1}{2})$. We then extract 15 features out of the above values, as shown in Figure 5.

It is important to note that the audio features used here have no discrimination power along the vertical axis. Nevertheless, across different columns, the audio features can vary significantly, hence they can still be very good weak classifiers. we let the boosting algorithm decide if such classifiers are helpful. From the experiments in Section IV, SSL features are among the top features selected by the boosting algorithm.

C. The Boosting Algorithm

We adopt the Logistic variant of AdaBoost developed by Collins, Schapire, and Singer [31] for training the BMSD detector. The basic algorithm is to boost a set of decision "stumps", decision trees of depth one. In each round a single rectangle feature or audio feature is selected. In addition a threshold and two weights α and β are computed. During classification the score of an example is updated by α if the feature is below the threshold, and β otherwise.

As suggested in [25] importance sampling is used to reduce the set of examples encountered during training. Before each round the boosting weight is used to sample a small subset of the current examples. The best weak classifier is selected with respect to this sample. The implementation of the boosting training process differs little from [5], and we refer the readers to [5] for more details.

D. Alternative Speaker Detection Algorithms

1) SSL-Only: The most widely used approach to speaker detection is SSL [7], [6]. Given the SSL likelihood as $L_a(x), x = 1, 2, \cdots, X$, we simply look for the peak likelihood to obtain the speaker direction:

$$\hat{x} = \arg\max_x L_a(x). \tag{4}$$

This method is extremely simple and fast, though its performance varies significantly across different conference rooms, as shown in Section IV.

2) SSL and MPD DLF: The second approach is to design a multi-person detector, and fuse its results with SSL output probabilistically. We designed an MPD algorithm similar to that in [27], with the same visual features described in Section III-A. The MPD output is a list of head boxes. To fuse with the 1D SSL output, a 1D video likelihood function can be created from these boxes through kernel methods, i.e.:

$$L_v(x) = \sum_{n=1}^{N} e^{-\frac{(x - x_n)^2}{2\sigma^2}}, \tag{5}$$

where N is the number of detected boxes; x_n is the horizontal center for the n^{th} box; σ is $\frac{1}{3}$ of the average head box width. Assuming the audio and visual likelihoods are independent, the total likelihood is computed as:

$$L(x) = L_a(x) * L_v(x), \tag{6}$$

and we pick the highest peak in $L(x)$ as the horizontal center of the active speaker. The height and scale of the speaker is determined by its nearest detected head box.

IV. Experimental Results

A. Test Data

Experiments were performed using a set of 8 video sequences captured by the RoundTable device in different conference rooms, each about 4 minutes long. A total of 790 frames were sampled from these videos. The active speakers' heads are manually marked with a box as the ground truth.

Since the human body can provide extra cues for speaker detection, we expand every head box with a constant ratio to include part of upper body, as shown in Figure 6(b). Rectangles that are within a certain translation and scaling limits of the expanded ground truth boxes are used as positive examples (Figure 6(c)). The remaining rectangles in the videos are all treated as negative examples.

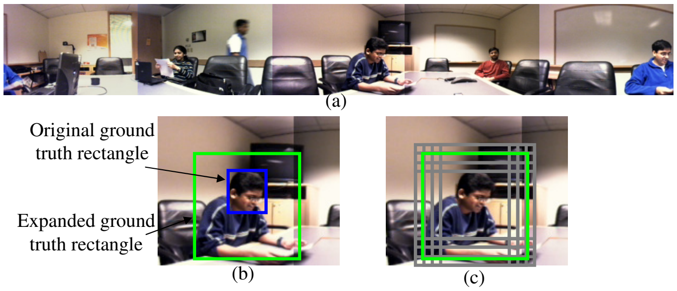
(a)
Original ground truth rectangle
Expanded ground truth rectangle
(b)
(c)

Fig. 6. Create positive examples from the ground truth. (a) Original video frame. (b) Close-up view of the speaker. The blue rectangle is the head box; the green box is the expanded ground truth box. (c) All gray rectangles are considered positive examples.

In the following experiments, we will compare the performance of the SSL-only, SSL+MPD DLF and BMSD algorithms described in Section III. The MPD algorithm is trained with exactly the same boosting algorithm in Section III-C, except that we use all visible people as positive examples, and restrict learning to include only visual features. Note in the BMSD training process, the negative examples include people in the meeting room that were not talking. We expect BMSD to learn explicitly the difference between speakers and non-speakers.

B. The Detection Process and the Matching Criterion

Once the classifier has been trained, detection is straightforward and efficient. We pass all the rectangles at different locations and scales to the classifier, similar to Viola and Jones' popular face detector [5]. Overlapping positive rectangles are merged into a single rectangle. In order to visualize the results, we also shrink the rectangle to reverse the expansion performed during training.

To measure if the detected rectangle is a true positive detection, we use the following criterion. Let the ground truth face be $\{x_g, y_g, w_g, h_g\}$, where x_g and y_g are the center of the head box, and w_g and h_g are the width and height. Let the detected box be $\{x_d, y_d, w_d, h_d\}$. A true positive detection must satisfy:

$$\begin{aligned} |x_d - x_g| < w_g; \qquad & |y_d - y_g| < h_g; \\ \frac{w_g}{2} < w_d < 2w_g; \qquad & \frac{h_g}{2} < h_d < 2h_g. \end{aligned} \tag{7}$$

Because SSL-only does not output a detected rectangle, we only check its horizontal accuracy $|\hat{x} - x_g| < w_g$, where $\hat{x}$ is computed as $\hat{x} = \arg\max_x L_a(x)$. $L_a(x)$ was defined in Equation 3.

C. Detection Performance

We ran a leave-one-out experiment on the 8 testing sequences. The MPD and BMSD detectors are both trained on 7 sequences with 20 and 60 features, and tested on the remaining one. Since in our test data most frames have only one active speaker, the true detection rate (TDR) and false positive rate (FPR) satisfies TDR $\approx 1-$FPR. Hence we report the peak true detection rate of various approaches in Figure 7. Note in the case of SSL+MPD DLF, the MPD thresholds were adjusted so that the fused detection rate is the best.

	Seq1	**Seq2**	**Seq3**	**Seq4**	**Seq5**	**Seq6**	**Seq7**	**Seq8**	**Total**
Number of frames	98	159	71	98	84	47	129	104	790
SSL-only	99.0%	97.5%	94.4%	99.0%	66.7%	87.2%	41.1%	68.3%	80.6%
SSL+MPD DLF (20 features)	100%	97.5%	95.8%	96.9%	72.6%	97.9%	61.2%	74.0%	85.9%
SSL+MPD DLF (60 features)	100%	97.5%	97.2%	99.0%	79.8%	97.9%	60.5%	74.0%	87.0%
BMSD (20 features)	100%	97.5%	97.2%	99.0%	84.5%	95.7%	67.4%	83.7%	89.7%
BMSD (60 features)	100%	96.9%	97.2%	98%	88.1%	95.7%	77.4%	83.7%	91.0%

Fig. 7. Peak detection rate for various algorithms.

The eight sequences in Figure 7 can be categorized into three classes:

1) Seq 1, 2 and 4. In these sequences the reverberation and ambient noises are small. The SSL performs extremely well, and both DLF and BMSD perform equally well.
2) Seq 3 and 6. In these sequences there is occasional reverberation noise. The SSL-only solution does a reasonable job but the detection rate is below 95%. Integrating visual information helps improve the performance for both SSL+MPD DLF and BMSD, though there is no significant difference between SSL+MPD DLF and BMSD.
3) Seq 5, 7 and 8. Due to severe reverberation (people talking to the whiteboard), SSL-only has a very poor average detection rate of 56.8%, SSL+MPD DLF with 20 features has an average detection rate of 68.5%, while BMSD with 20 features achieves 77.3%. We believe this is because BMSD explores the low frame rate correlations between the audio and visual signal much better than the other two alternatives.

From Figure 7, the total detection error rate of SSL-only, SSL+MPD DLF and BMSD are 19.4%, 14.1% and 10.3%, respectively. The BMSD algorithm reduces the detection error rate of SSL-only by 47%, and the error rate of SSL+MPD DLF by 27%. These are very significant improvements. It is worth noting that while BMSD is significantly better than SSL in the third category, it does not degrade any performance in category 1.

D. Pruning

The number of features needed in our BMSD detector is surprisingly small. The BMSD detector with 20 features can easily run on our DSP processor at 1 FPS, and achieve nearly 90% average detection rate. In this section we examine the possibility to further speed up the detector by pruning, as was done in [5], [27].

We use a separate validation data set to compute the pruning thresholds. We first run the full classifier on the validation set, and obtain a number of examples that are classified as positive. We then set the pruning threshold at each node as the minimum score of these detected rectangles. Note this whole process does not require ground truth information. We simply guarantee that on the validation dataset the pruned classifier will have the same results as the full classifier.

Figure 8 shows the average number of nodes visited when pruning is enabled in MPD and BMSD. It can be seen that

	MPD (20 features)	MPD (60 features)	BMSD (20 features)	BMSD (60 features)
Number of nodes	1.66	2.43	1.32	1.88

Fig. 8. Average number of nodes visited when pruning is enabled. The improvement is significant in the case of BMSD – 15 times (20/1.32) saving in computation for 20 features and 32 times (60/1.88) saving for 60 features.

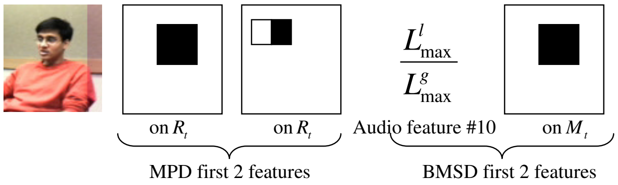

Fig. 9. Top features for MPD and BMSD.

in both cases pruning reduces the average number of features to be calculated significantly. The first two features appear to be very critical since the average number of nodes visited is only between 1 and 3. Figure 9 shows the first two features of MPD and BMSD. For MPD, both features are on the running difference image. The first feature favors rectangles where there is motion around the head region. The second feature describes that there is a motion contrast around the head region. In BMSD, the first feature is an audio feature, which is the ratio between the local maximum likelihood and the global maximum likelihood. The second feature is a motion feature similar to the first feature of MPD, but on the 3-frame difference image. It is obvious that although audio features do not have discrimination power along the vertical axis, they are still very good features, and helps the BMSD to reduce the average number of computed features by 20-25% according to Figure 8 (compare MPD and BMSD). In practice, the gain is even bigger, because the audio features are based on a 1D SSL curve, which can be pre-computed only once for all rectangles that share the same horizontal span.

V. Conclusions

This paper proposes a boosting-based multimodal speaker detection algorithm that is both accurate and efficient. We compute audio features from the output of SSL, place them in the same pool as the video features, and let the logistic AdaBoost algorithm select the best features. To the best of our knowledge, this is the first multimodal speaker detection algorithm based on boosting.

References

[1] C. Busso, S. Hernanz, C. Chu, S. Kwon, S. Lee, P. Georgiou, I. Cohen, and S. Narayanan, "Smart room: participant and speaker localization and identification," in *Proc. of IEEE ICASSP*, 2005.

[2] R. Cutler, Y. Rui, A. Gupta, J. Cadiz, I. Tashev, L. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverbert, "Distributed meetings: a meeting capture and broadcasting system," in *Proc. ACM Conf. on Multimedia*, 2002.

[3] B. Kapralos, M. Jenkin, and E. Milios, "Audio-visual localization of multiple speakers in a video teleconferencing setting," York University, Canada, Tech. Rep., 2002.

[4] B. Yoshimi and G. Pingali, "A multimodal speaker detection and tracking system for teleconferencing," in *Proc. ACM Conf. on Multimedia*, 2002.

[5] P. Viola and M. Jones, "Robust real-time face detection," *Int. Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[6] H. Wang and P. Chu, "Voice source localization for automatic camera pointing system in videoconferencing," in *Proc. of IEEE ICASSP*, 1997.

[7] Y. Rui, D. Florencio, W. Lam, and J. Su, "Sound source localization for circular arrays of directional microphones," in *Proc. of IEEE ICASSP*, 2005.

[8] A. Adjoudani and C. Benoît, "On the integration of auditory and visual parameters in an HMM-based ASR," in *Speechreading by Humans and Machines*. Springer, Berlin, 1996, pp. 461–471.

[9] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *IEEE Trans. on Multimedia*, vol. 2, no. 3, pp. 141–151, 2000.

[10] W. Hsu, S.-F. Chang, C.-W. Huang, L. Kennedy, C.-Y. Lin, and G. Iyengar, "Discovery and fusion of salient multi-modal features towards news story segmentation," in *SPIE Electronic Imaging*, 2004.

[11] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *Proc. of IEEE CVPR*, 1997.

[12] M. Naphade, A. Garg, and T. Huang, "Duration dependent input output Markov models for audio-visual event detection," in *Proc. of IEEE ICME*, 2001.

[13] G.Iyengar and C.Neti, "Speaker change detection using joint audio-visual statistics," in *The Int. RIAO Conference*, 2000.

[14] P. Besson and M. Kunt, "Information theoretic optimization of audio features for multimodal speaker detection," Signal Processing Institute, EPFL, Tech. Rep., 2005.

[15] J. Fisher III, T. Darrell, W. Freeman, and P. Viola, "Learning joint statistical models for audio-visual fusion and segregation," in *NIPS*, 2000, pp. 772–778.

[16] J. Hershey and J. Movellan, "Audio vision: using audio-visual synchrony to locate sounds," in *Advances in Neural Information Processing Systems*, 2000.

[17] V. Pavlović, A. Garg, J. Rehg, and T. Huang, "Multimodal speaker detection using error feedback dynamic Bayesian networks," in *Proc. of IEEE CVPR*, 2001.

[18] D. N. Zotkin, R. Duraiswami, and L. S. Davis, "Logistic regression, adaboost and bregman distances," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 11, pp. 1154–1164, 2002.

[19] J. Vermaak, M. Gangnet, A. Black, and P. Pérez, "Sequential Monte Carlo fusion of sound and vision for speaker tracking," in *Proc. of IEEE ICCV*, 2001.

[20] H. Nock, G. Iyengar, and C. Neti, "Speaker localisation using audio-visual synchrony: an empirical study," in *Proc. of CIVR*, 2003.

[21] R. Cutler and L. Davis, "Look who's talking: speaker detection using video and audio correlation," in *Proc. of IEEE ICME*, 2000.

[22] M. Beal, H. Attias, and N. Jojic, "Audio-video sensor fusion with probabilistic graphical models," in *Proc. of ECCV*, 2002.

[23] Y. Chen and Y. Rui, "Real-time speaker tracking using particle filter sensor fusion," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 485–494, 2004.

[24] K. Nickel, T. Gehrig, R. Stiefelhagen, and J. McDonough, "A joint particle filter for audio-visual speaker tracking," in *ICMI*, 2005.

[25] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," Dept. of Statistics, Stanford University, Tech. Rep., 1998.

[26] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[27] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. of IEEE ICCV*, 2003.

[28] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proc. of IEEE ICCV*, 1999.

[29] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Trans. on PAMI*, vol. 19, no. 7, pp. 780–785, 1997.

[30] S. Goodridge, "Multimedia sensor fusion for intelligent camera control and human computer interaction," Ph.D. dissertation, Department of Electrical Engineering, North Carolina Start University, 1997.

[31] M. Collins, R. Schapire, and Y. Singer, "Logistic regression, adaboost and bregman distances," *Machine Learning*, vol. 48, no. 1-3, pp. 253–285, 2002.