SPORTS PROGRAM BOUNDARY DETECTION

Regunathan Radhakrishnan†, Ajay Divakaran†and Isao Otsuka‡

†Mitsubishi Electric Research Laboratory, Cambridge, MA 02139
‡Advanced Technology R & D Center, Mitsubishi Electric Corporation, Kyoto, Japan
E-mail: †{regu, ajayd}@merl.com ‡Otsuka.Isao@cw.MitsubishiElectric.co.jp

ABSTRACT

In the recent years, consumer devices that can record broadcast video have become prevalent. Such devices rely on the program guide information about a program's start time and end time for recording. However, sports broadcasts can run over the specified time sometimes. In this paper, we propose a framework based on an audio classification framework to correctly detect the end of a sports broadcast so as to enable complete recording of the game. Our experimental results show that the proposed algorithm can detect sports program boundaries with a high accuracy.

1. INTRODUCTION

With increasing number of consumer electronic devices like TiVo that can record broadcast videos, content analysis technologies have made their way to these devices so as to enable efficient browsing of the stored content. Such devices rely on Electronic Program Guides (EPGs) for information regarding the start time and end time of programs. It is worth noting here that the EPG information is updated only 4 times a day. When the end-user chooses to record a program that will be broadcast later, the device uses this information to record and store the complete program. However, for recording sports programs this will not work sometimes as the game can run over the specified duration in EPGs. In such a scenario, one should be able to utilize the available content analysis technology in the device to correctly record the whole program without completely relying on the EPG information.

In this paper, we propose one such solution for a summarization enabled personal digital video recorder system proposed in [1]. The recorder uses an audio classification framework to analyze and produce highlights of sports programs while recording.

In this paper, we propose a solution to detect the correct endings of sports broadcasts that uses the built-in audio classification framework. The problem of detecting the program boundaries can be formulated as that of audio scene segmentation. Refer to [2] for a overview on existing methods of audio scene segmentation. In [2], an audio scene is defined as a semantically consistent sound segment characterized by a few dominant sound sources. The proposed method in [2] uses multiple features and models to characterize these dominant sources in the scene. This is computationally intensive. In our framework, we get an idea of the audio scene from the output of the audio classification framework in terms of existing types of sound sources efficiently and use it to perform the segmentation.

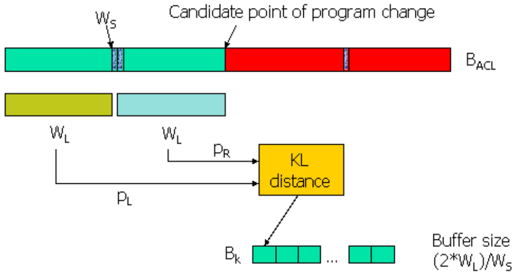

Figure 1: Steps 1 and 2 of the proposed algorithm

The rest of the paper is organized as follows. In section 2, we describe the proposed algorithm to detect sports program boundaries. In section 3, we present some experimental results of the proposed framework. And finally conclude with possibilities for further research in section 4.

2. PROPOSED FRAMEWORK

We formulate the problem of detecting sports program boundary detection as that of detecting a change in the generating process for the observed features. This is motivated by the observation that the audio class composition and the audio background during a sports broadcast is distinctively different from, say, a news program that follows it.

The proposed algorithm follows these steps:

- **step 1:** We create two buffers of appropriate sizes to hold audio classification labels and audio features corresponding to $4W_L$ minutes of content. Let us denote the audio classification labels buffer by B_{ACL}.

Let us denote the audio features buffer by B_{AF}. $2W_L$ is the candidate point of program change according to the program guide information.

- **step 2:** Let us start with a window of audio class labels corresponding to $2W_L$ minutes. Initially, this window is centered at W_L. Starting at W_L and stepping by W_S every time, we compare the audio class composition within the first W_L against the audio class composition within the second W_L. The comparison is performed using Kullback-Leibler(KL) distance between the two distributions. We slide this window $\frac{2W_L}{W_S}$ times and evaluate whether the generating process for the first W_L is same as that for the second W_L. The computed KL distance is stored in a buffer, B_K. Figure 1 shows the first two steps of the analysis so far.
- **step 3:** After $\frac{2W_L}{W_S}$ steps, the buffer B_k is full. If there was a program change at time $2W_L$, one would expect the KL distance to peak at that time. We run a peak detection algorithm on the KL distance values in the buffer B_K to mark candidate points of program change.
- **step 4:** Then, for every candidate point suggested in step 3, we verify if there is a change in generating process using low-level features stored in the buffer B_{AF}. The verification is done by first modelling the low-level features to the left of candidate point using a Minimum Description Length Gaussian Mixture Model(GMM). Let us denote this GMM by G_L.Then, we compute a similar GMM to model the low-level features to the right of the candidate point. Let us denote this GMM by G_R. We compute the distance $(D(G_L, G_R))$ between G_R and G_L as shown in the equation 1. Here 0_L and O_R are low-level features to the left and to the right of the candidate point respectively. And, # represents the cardinality operator. By using a threshold on this distance, one can declare a candidate point to be a point of program boundary. Figure 2 illustrates the analysis steps 3 and 4 just described.
- **step 5:** In case, none of the candidate points qualify as a program boundary, remove old data corresponding to $2W_L$ minutes and wait for the buffers B_{ACL}, B_{AF} and B_K to be filled and repeat steps 1 through 4.

$$\begin{aligned}&(\frac{1}{\#(O_L)}\log P(O_L|G_L)) + (\frac{1}{\#(O_R)}\log P(O_R|G_R))\\ -&(\frac{1}{\#(O_L)}\log P(O_L|G_R)) - (\frac{1}{\#(O_R)}\log P(O_R|G_L))\end{aligned} \tag{1}$$

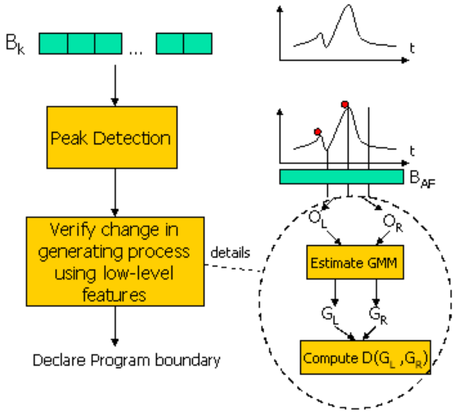

Figure 2: Steps 3 and 4 of the proposed algorithm

In essence, steps 1 and 2 of the proposed algorithm checks for generating process change at a coarser level using audio class composition statistics (P_L and P_R in Figure 1). Steps 3 and 4 verify the change of process using low-level features. In the following section, we present some experimental results to show the effectiveness of the proposed algorithm.

3. EXPERIMENTAL RESULTS

3.1. Data Set

We tested the proposed algorithm with content of total duration 3.5hrs consisting of 6 sequences. The details of each of the sequences is listed in table 1. Sequence 1 is a baseball game (with not so exciting finish) followed by a news program. Sequence 2 is the same content except for a 30s commercial between the programs. Sequence 3 is the last 10min of second half of world cup final soccer game followed by a news program. Sequence 4 is same as Sequence 3 except for a 36s commercial between the programs. Sequence 5 is 80min of baseball game with many commercial breaks but no program changes. Sequence 6 is 50min of second half of world cup final soccer game with no commercials and program changes. These clips were carefully chosen to test the proposed algorithm. Sequences 1 and 2 are challenging as the audio class composition change is not that prominent across program boundary. On the other-hand, Sequences 3 and 4 have a very distinct change in audio class composition across the program boundary that can be attributed to the exciting finish to the games. While sequences 1-4 were chosen to test the detection rate of the algorithm, sequences 5 and 6 were chosen to test the false alarm rate of the algorithm.

Sequence	Duration	Program boundary at
1	20min 10s	10min 0s
2	20min 40s	10min 30s
3	20min 0s	10min 0s
4	20min 36s	10min 36s
5	80min 38s	no change
6	50min 3s	no change

Table 1: Details about the data set

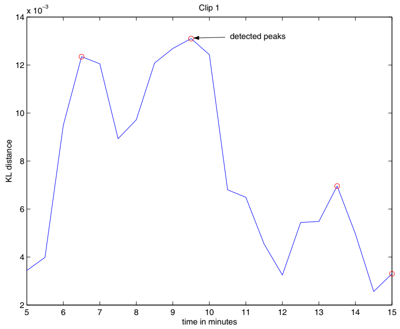

Figure 3: Results for Sequence 1

3.2. Results

The parameters of the proposed algorithm were set to the following and were not changed for each of the sequences:

- W_L = 5 minutes
- W_S = 30 secs
- Threshold on $D(G_L, G_R)$ set to 5.0

The value of W_L was chosen to be 5 mins to allow for a reliable estimate of the audio class histogram. The value of W_S determines the required granularity of program boundary detection and was chosen to be 30s for the experiments here. A threshold value of 5.0 on $D(G_L, G_R)$ gave us the best performance. We extracted the 12 MFCC coefficients to be used as the low-level features, to train G_L and G_R.

Figures 3 shows the results of peak detection on the KL distance buffer (B_K) for sequence 1. The results for sequence 2 were also similar to those of sequence 1. Note that there is a peak detected at 10min. The program changes to news at that point. Furthermore, note the value of $D(G_L, G_R)$ at that point from tables 2 and 3. The value of $D(G_L, G_R)$ is not so high for other candidate points. Thus, using the difference in audio class composition followed by verification using low-level features we are able to correctly detect the end of a sports program for these two sequences. Sequences 1 and 2 are particularly challenging because of the similar audio class composition on either side of program boundary. Since the baseball game did not have an exciting last 10 min, we had to use both cues (audio class composition & low-level features) to detect the program boundary.

time	$D(G_L, G_R)$
7.00	2.144176
9.00	1.521864
10.00	**6.438550**
14.00	2.137635

Table 2: $D(G_L, G_R)$ for each of the peaks in Sequence 1

time	$D(G_L, G_R)$
7.00	2.738464
8.50	2.020878
10.00	**6.485522**
12.50	1.946460
12.50	1.946460
15.00	1.247302

Table 3: $D(G_L, G_R)$ for each of the peaks in Sequence 2

Now, let us look at the results for sequences 3 and 4. Figures 4 shows the detected peaks from the KL distance buffer for sequence 3. The results for sequence 4 are similar to those of sequence 3. Unlike for the cases of sequences 1 and 2, the peak at program boundary (10min) for sequences 3 and 4 is very dominant. This is because the last 10 min of the soccer game were exciting. Consequently, there was a high percentage of cheering labels. It is then easy to detect the program change when a program follows with a different audio class composition (news program in this case). Also, note the value of $D(G_L, G_R)$ from tables 4 and 5. The value is quite high at the program boundary further verifying the program change using low-level features.

Figure 5 shows the results for sequence 5 (a 80 min sequence of a baseball game with several commercial breaks). The value of $D(G_L, G_R)$ is higher than the chosen thresh-

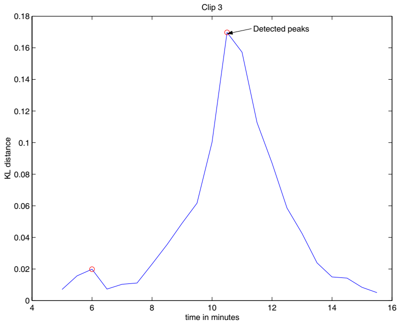

Figure 4: Results for sequence 3

time	$D(G_L, G_R)$
6.00	2.047802
10.50	**12.810868**

Table 4: $D(G_L, G_R)$ for each of the peaks in sequence 3

time	$D(G_L, G_R)$
6.00	2.619183
7.00	1.124283
10.50	**11.651732**

Table 5: $D(G_L, G_R)$ for each of the peaks in sequence 4

old at five of the detected peaks (at 13.5, 21.5, 47, 52 and 73 min). They were verified to be commercial segments of 3 min duration. Since the size of window is only 5 min on either side of the peak, it is reasonable that $D(G_L, G_R)$ will be high at these points.

Figure 6 shows the results for clip 6 (a 50 min clip of a soccer game with no commercial breaks). There are peaks detected from the KL distance buffer but none of the detected peaks had a high $D(G_L, G_R)$. There are no false program boundary detections for this clip. The dominant peak at the end of the soccer game is the portion where the crowd cheers for the winning Brazil team. In summary, the results of the proposed algorithm with sequences 1-4 show the high detection accuracy of the proposed approach while the results with sequences 5 and 6 show the low false alarm rate.

4. CONCLUSION & FUTURE WORK

We proposed an audio analysis framework to automatically detect sports program boundaries in personal digital video recorders. This would be useful when sports program exceed the EPG specified time limit. The algorithm is based on the observation that there is a generating process change at the program boundary that can be measured in terms audio class composition and low-level features. The experimental results are promising as all program boundaries have been detected from the clips and there were only few false alarms. The false alarms were verified to be 3 min long commercial segments that can be eliminated by other robust means.

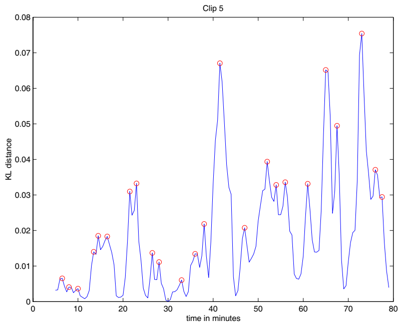

Figure 5: Results for sequence 5

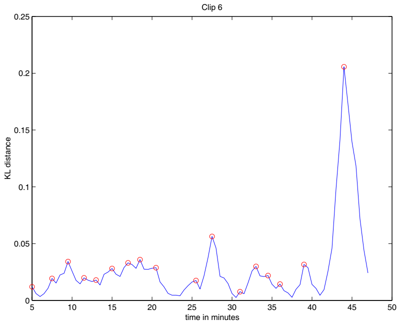

Figure 6: Results for sequence 6

In this paper, we have mainly tested the approach with "sports program to news" transitions. This is probably the most likely scenario for which the EPG information may fail. We would also test the approach for other transitions such as "sports to talk-show", "movie to news", "sports to sports" etc. It is also possible to use video cues that are semantically more meaningful than audio cues, for the purpose of program detection. By recognizing key video markers proposed in [3] such as Baseball catchers, Goal posts one can detect the end of a sports program by observing the absence of these markers across the program boundary.

5. REFERENCES

[1] I.Otsuka, K.Nakane, A.Divakaran, K.Hatanaka, and M.Ogawa, "A highlight scene detection and video summarization system using audio feature for a personal video recorder," *Proc. of ICCE*, 2005.

[2] H.Sundaram and S.F.Chang, "Audio scene segmentation using multiple features, models and time scales," *Proc. of ICASSP*, 2000.

[3] Z. Xiong, R. Radhakrishnan, and A. Divakaran, "Highlights extraction from sports video based on an audiovisual marker detection framework," *Proc. of ICME*, 2005.