SCENE CHANGE DETECTION BY AUDIO AND VIDEO CLUES

Shu-Ching Chen[1†], *Mei-Ling Shyu*[2], *Wenhui Liao*[1], *Chengcui Zhang*[1]

[1]Distributed Multimedia Information System Laboratory
School of Computer Science, Florida International University, Miami, FL 33199
[2]Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33124

ABSTRACT

Automatic video scene change detection is a challenging task. Using audio or visual information alone often cannot provide a satisfactory solution. However, how to combine audio and visual information efficiently still remains a difficult issue since there are various cases in their relationship due to the versatility of videos. In this paper, we present an effective scene change detection method that adopts the joint evaluation of the audio and visual features. First, video information is used to find the shot boundaries. Second, the audio features for each video shot can be extracted. Lastly, an audio-video combination schema is proposed to detect the video scene boundaries.

1. INTRODUCTION

More and more research groups use both audio and video information to detect scene boundary [4, 5, 6, 7]. Despite several initial successes, how to find a good way to combine audio and video information is still challenging. In [4], the candidate scene boundaries are extracted from the video data based on the extraction of visual effects such as dissolve or fade in/out. In addition, for the audio data, they only analyze the starting and ending audio frames of each shot using the average power of subbands. Since the audio features within a short time duration often cannot represent the characteristics of the whole shot, monitoring the audio changes within a short time around the video shot boundaries cannot guarantee to identify the audio changes between the neighboring video shots. For example, when an audio change occurs before the video shot change, the method in [4] cannot work well. In [5], the authors used a finite-memory model to separately segment the audio and video data into scenes, and then applied two ambiguity windows to merge the audio and video scenes. In [6], the authors did not combine the audio and video segmentation results. In [7], audio breaks were first detected in the one-second intervals. If a video shot boundary is within the one-second interval, this boundary is marked as a scene candidate that is then analyzed by a color correlation algorithm. The problem is that the audio data with one-second duration is too sensitive to represent the characteristics of the audio data.

In this paper, an integrated audio-visual framework for video scene change detection is presented. Our framework consists of two parts. First, the video shot boundaries are detected using an unsupervised segmentation algorithm together with the technique of object tracking based on the results of the segmentation [8]. The second part is to analyze the audio features based on the detected video shots. According to the distance between two shots with respect to the audio features, scene changes can be detected. Due to hard cuts, fades and dissolves in a video sequence, some of the video shots may be very short. In order to reduce these types of effects, if a shot consists of less than 5 frames, it is merged with its preceding shot before the audio processing. The main advantage of our method is that we use a very natural way to combine the audio and video data. Our audio processing is based on the video shots, which can avoid the conflicts between the audio and video data. The conflicts often occur when they are processed independently. One the other hand, in the audio processing part, there is no need to focus on what the content (music, conversation, etc.) of the audio data is. Instead, the focus is on their differences between the video shots. Using this way, our method can be simplified and at the same time it can still be very effective. Experiment results show that the proposed method is better than those that separately segment the audio and video data into scenes and then integrate them.

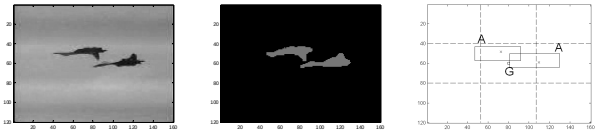
The rest of the paper is organized as follows. Shot detection using the video information is presented in Section 2. In Section 3, the method of dealing with audio data to detect scene change is described. Experiment evaluation results are given in Section 4. In Section 5, the conclusion is given.

† This research was supported in part by NSF CDA-9711582.

2. VIDEO SHOT DETECTION

2.1 Video Frame Segmentation

In our previous work [8, 9], we proposed an innovative method for video shot detection using an unsupervised segmentation algorithm and the technique of object tracking based on the results of the segmentation. As illustrated in Figures 1(a) and 1(b), the pixels belonging to an air show video frame have been partitioned into two classes of areas. In Figure 1(b), the gray regions correspond to the plane objects, while the dark region captures the blue sky. Figure 1(c) shows the bounding boxes and centroids for the plane objects.



(a) (b) (c)

Figure 1. (a) a video frame in an air show video; (b) segmentation mask map for (a); and (c) the bounding boxes and centroids for the objects/segments in (b).

2.2 Object Tracking

The first step for object tracking is to identify the segments in each class in each frame. Then the bounding box and the centroid point for that segment are obtained. For example, Figure 1(b) shows the segmentation mask map of the video frame in Figure 1(a). Then the bounding box and the centroid point for that segment are obtained (as shown in Figure 1(c)). The next step for object tracking is to connect the related segments in the successive frames. The idea is to connect two segments that are spatially the closest in the adjacent frames [8]. In other words, the Euclidean distances between the centroids of the segments in the adjacent frames are used as the criteria to track the related segments. In addition, size restriction should be employed in determining the related segments in the successive frames.

2.3 Shot Change Detection Method

In order to improve the processing speed, our current framework adopts a multi-filtering architecture. The steps for our video shot change detection method are given in the following:

Step 1: Use the traditional pixel-level comparison to see if there is a significant enough difference between two consecutive frames. If yes, mark the current frame as a video shot boundary. Otherwise, go to Step 2.

Step 2: Get the segmentation maps of the two consecutive frames and compare them. If the distance is larger than a threshold, mark the current frame as a video shot boundary. Otherwise, go to Step3.

Step 3: Track the objects/segments within the two consecutive frames. If the total area of the matched objects is less than a low threshold, mark a new video shot boundary. Otherwise, continue processing the next frame.

3. DETECTING SCENE CHANGE BY AUDIO

3.1 Audio Feature Extraction

In our experiments, the audio data is sampled by 16bits, 22.05KHZ. We regard 1024 samples as a frame, and each frame is shifted by 256 samples from the previous frame. Nine different features (mentioned in [1, 2, 3]) are used to analyze the audio data: 1) volume (*V*); 2) energy (*P*); 3) sub-band energy (*Sub-P*); 4) low shot-time energy ratio (*LSTER*); 5) zero crossing rate (*ZCR*); 6) frequency centroid (*FC*); 7) frequency bandwidth (*FB*); 8) spectral flux (*SF*); and 9) cepstral flux (*CF*). Features are extracted from each frame first. In addition to that, we divide the frequency domain into four sub-bands dynamically, which are 0 – 1/16 *fs*, 1/16 *fs* – 1/8 fs, 1/8 *fs* – 1/4 *fs*, and 1/4 *fs* – 1/2 *fs* respectively, where *fs* is the sample rate. From our observations, we found that the first and the third sub-bands are more suitable for our framework, so these two are selected to obtain the sub-band energy.

3.2 Shot-level Processing

We observe that the audio changes that happen within a short time (such as one second) often cannot indicate the existence of a scene boundary. Therefore, instead of only monitoring the audio track changes around the video shot boundaries, the proposed method analyzes the audio features within the whole range of a video shot since these features may contain more useful information.

After dividing the audio data into different shots based on the video shot boundaries (obtained from Section 2), we need to extract the audio features for each shot. Since it is not necessary to extract features for a silent shot, we first identify those silent shots using the following criteria. For each frame, if its volume < 0.003 and *ZCR* > 50, we consider it is a silent frame. Within each shot, if the percentage of the silence frames is larger than 0.7, this shot is considered as a silent shot, which will be ignored in the later processing. Moreover, for any non-silent shot, if it consists of consecutive “silent frames” that last more than 0.33 second, these silent frames will also be skipped from the processing, which can prevent them from significantly affecting the audio feature value of that shot.

Another observation is that editing effects often generate very short shots, which do not have enough audio information. Hence, we regard such a shot as a part of its neighboring shots (either the preceding shot or the succeeding one). In our framework, we select the preceding shot.

According to the properties of the nine features, we divide them (except *ZCR*) into three groups, namely Volume, Power and Spectrum. In each group, we calculate different values (such as mean, standard deviation, volume dynamic range, etc.) for the features in this group and add these values together. The idea is formalized as follows:

Volume group
$Vec(shot_i) = mean(V_i) + dev(V_i) + vdr(V_i) + diff(V_i)$
Power group
$Pvec(shot_i) = dev(P_i) + dev(Sub\text{-}P_i) + lster(P_i) + lster(Sub\text{-}P_i) + diff(P_i) + diff(Sub\text{-}P_i)$
Spectrum group
$Svec(shot_i) = dev(FB_i) + dev(FC_i) + diff(SF_i) + diff(CF_i)$

Where:
mean: the mean value of a feature in the *ith* shot;
dev: the standard deviation of a feature in the *ith* shot;
vdr: the volume dynamic range in the *ith* shot;
diff: the standard deviation of the frame-to-frame difference of a feature in the *ith* shot.

Because different features may have big differences in their values, the values are normalized by dividing them by their maximal value, and the normalized values are used in the above equations to obtain the three values (*Vec*, *Pvec*, and *Svec*) for the three groups in each shot. These three values will be used to detect scene changes.

3.3 Scene Change Detection

First, we determine the distances between two neighboring shots with respect to the *Vec*, *Pvec*, and *Svec* values. For each value, a threshold can be obtained from the following equation.

$T_{sv} = (mean(dist(sv_i, sv_{i+1})) + dev(dist(sv_i, sv_{i+1}))) / \sqrt{2}$,

where *sv* can be either *Vec*, *Pvec*, or *Svec*, sv_i and sv_{i+1} are their values in two neighboring shots (the *ith* shot and the *(i+1)th* shot), and *dist* is the Euclidean distance.

Based on these threshold values, the following rule is used to determine whether there is a scene change.

> *If* $dist(Vec(shot_i), Vec(shot_{i+1})) > T_{Vec}$ *or*
> $dist(Pvec(shot_i), Pvec(shot_{i+1})) > T_{Pvec}$ *or*
> $dist(Svec(shot_i), Svec(shot_{i+1})) > T_{Svec}$
> then a scene boundary is recorded

If the distance of any one of the three values is larger than its corresponding threshold value *T*, we consider there is a scene boundary. We do not combine the three distance values into one distance measure because of the following two reasons. First, each value contains different audio semantic meanings, and simply combining them will destroy these meanings. Second, it is difficult to give them appropriate weights if we combine them. Different audio feature may have different levels of importance for different audio data. A static combination to them will reduce the flexibility. Hence, we consider the above three values are complementary to each other.

4. EXPERIMENTAL RESULTS

A series of experiments on a long movie video and several TV news videos are conducted. The performance is measured in terms of the precision (Pre.) and recall (Rec.) parameters. Table 1 lists the video features and the experiment results.

Table1: Scene detection performance using joint audio and video clues.

	shot	scene	correct	miss	fault	pre.	rec.
V1	68	14	13	1	1	0.93	0.93
V2	29	13	12	1	1	0.92	0.92
V3	27	9	7	2	0	1	0.78
V4	18	11	10	1	1	0.91	0.91
V5	514	144	129	15	21	0.86	0.90
Average						0.92	0.89

In this table, we use V1 to V5 to denote the video sequences that we used in our experiments. The second and third columns indicate the numbers of shots and scenes in each video sequence. The fourth and fifth column give the numbers of scenes that our method correctly identifies and misses. The sixth column indicates the number of scenes that our method misidentifies. The last two columns give the precision and recall values for each video sequence using our method. For example, the number of scenes in V1 should be 14. Our method correctly identifies 13 out of 14 of them (i.e., one miss), and misidentifies one scene, which result in 0.93 in precision and 0.93 in recall values. As can be seen from the last row in this table, our method can achieve 0.92 in precision and 0.89 in recall in average, which demonstrates that our method works very well. Particularly, it works very well for the following two situations, where most of the existing approaches have difficulty to deal with them.

1. At a shot boundary that happens to be a scene boundary, the visual feature changed but the audio feature did not change (as shown in Figure 2).

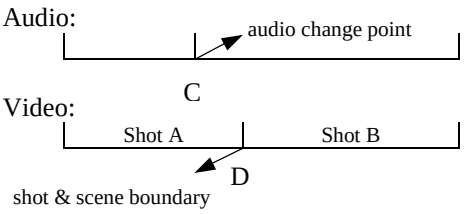


Figure 2: A scene boundary where audio does not change around shot boundary.

In Figure 2, the audio data after point C corresponds to Shot B. For example, the scene changed between Shot A and Shot B. Near the end point of Shot A, the audio belonging to Shot A has been muffled while the audio belonging to shot B appears (for example, the voice of a speaker who belongs to Shot B). In other word, the audio changes corresponding to scene changes occur before the video changes. In such a situation, most of the existing approaches cannot detect it correctly because there are no audio changes in point D. On the other hand, our method can detect it correctly since the extraction of the audio features is based on the whole shot.

2. Audio feature changed within a short time around a shot boundary that is not a scene boundary (as shown in Figure 3).

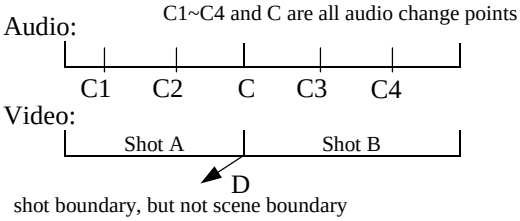


Figure 3: A shot boundary (but not a scene boundary) where audio changed around it.

In Figure 3, Shot A and Shot B are in the same scene. Several speakers exist in both of the two shots. However, at the shot boundary, the speaker changed. In most of the existing approaches, such situation is misidentified as a scene change since both audio and video features changed. The experiments show that our method does not misidentify this as a scene change because the distance between the overall audio features of Shot A and Shot B is not that significant.

Moreover, our method of shot detection tends to detect a little more (still reasonable) shots instead of missing the correct hits [8], which is appropriate for the later audio feature extraction. Also, the extracted audio feature in turn can reduce the false hits in shot detection.

5. CONCLUSIONS

In this paper, we presented an innovative scene change detection method using joint audio and video clues. Unlike the traditional methods that first analyze audio and video data separately and then combine them, we analyze them at different phases. The audio feature extraction is based on the detected video shots, which tends to be more stable and more reliable in charactering the audio data. The experimental results demonstrate that our method performs very well in terms of precision and recall values.

6. REFERENCES

[1] R. Wang, Z. Liu, and J.-C. Huang, "Multimedia Content Analysis," *IEEE Signal Processing Magazine*, pp.12-36, Nov. 2000.

[2] L. Lu, H. Jiang, and H.J. Zhang, "A Robust Audio Classification and Segmentation Method," *ACM Multimedia 2001*.

[3] H. Sundaram and S.-F. Chang, "Audio Scene Segmentation using Multiple Models, Features and Time Scales," *ICASSP*, pp. 2441-2444, 2000.

[4] A. Yoshitaka, and M. Miyake, "Scene Detection by Audio-Visul Features," *IEEE International Conference on Mulatimedia and Expo* (*ICME01*), pp. 49-52, 2001.

[5] H. Sundaram and S.-F. Chang, "Video Scene Segmentation Using Video and Audio Features," *IEEE International Conference on Mulatimedia and Expo (ICME00)*, pp. 1145-1148, 2000.

[6] T. Muramoto and M. Sugiyama, "Visual and Audio Segmentation for video streams," *IEEE International Conference on Mulatimedia and Expo (ICME00)*, pp 1547-1550, 2000.

[7] H. Jiang, T. Lin and H.J. Zhang, "Video Segmentation with the assistance of audio content analysis," *IEEE International Conference on Mulatimedia and Expo (ICME00)*, pp. 1507-1510, 2000.

[8] S.-C. Chen, M.-L. Shyu, C. Zhang, and R.L. Kashyap, "Video Scene Change Detection Method Using Unsupervised Segmentation and Object Tracking," *IEEE International Conference on Multimedia and Expo (ICME01)*, pp. 57-60, 2001.

[9] S.-C. Chen, M.-L. Shyu, and R. L. Kashyap, "Augmented Transition Network as a Semantic Model for Video Data," *International Journal of Networking and Information Systems, Special Issue on Video Data*, vol. 3, no. 1, pp. 9-25, 2000.