ROBUST LEARNING-BASED TV COMMERCIAL DETECTION

Xian-Sheng Hua, Lie Lu, Hong-Jiang Zhang

Microsoft Research Asia
{xshua, llu, hjzhang}@microsoft.com

ABSTRACT

A robust learning-based TV commercial detection approach is proposed in this paper. Firstly a set of basic features that facilitate distinguishing commercials from general program are analyzed. Then, a series of context-based features, which are more effective for identifying commercials, are derived from these basic features. Next, each shot is classified as commercial or general program based on these features by a pre-trained SVM classifier. And last, the detection results are further refined by scene grouping and some heuristic rules. Experiments on around 10-hour TV recordings of various genres show that the proposed scheme is able to identify commercial blocks with relatively high detection accuracy.

1. INTRODUCTION

TV commercials have great influence upon our lives though sometimes they are not liked by most of the audience. There are many objectives for automatic TV commercial detection. For example, companies who make advertisements on TV generally charge other companies to verify that their TV commercials are actually broadcasted as contracted. Some companies may also want to automatically observe what their competitors are doing [1]. Companies who design advertisements would like to do research on all currently broadcasted commercials and try to accomplish more influential ones. On the consumer side, one may want to record TV programs using digital video recorder but they may not want to record commercials. However, presently we have to assign people to watch the TV programs in order to achieve the above goals. It is desirable to transfer this task to a computer system, which is able to automatically detect commercials in the broadcasted TV signal.

Some efforts have been made for automatic TV commercial detection. Previous approaches on this topic can be classified into three categories. The first category is rule-based methods, which use a set of features and rules to distinguish commercials from general programs (non-commercials) [1][2]. The second category is logo-based algorithms, which identify commercials only by the existence of TV station logos [3][4]. The third one is recognition-based methods, which can only detect known/registered commercials by video signatures [1][5].

However, for rule-based algorithms, a number of thresholds are required to be fine-tuned. The system may be sensitive to some thresholds and thus difficult to find uniform thresholds for all kinds of programs. And, currently many TV stations do not always hide the logos when broadcasting commercials, thus logo-based methods may fail in this case. Furthermore, present TV station logos are becoming more and more complex and sometimes they are even semi-transparent and animated, which make it very difficult to be detected. For recognition-based schemes, we can only detect previously known and registered commercials and thus we need a relatively large database to store the signatures for all known commercials. For every new commercial, we are required to manually label it and add its signature into the database. These shortcomings limit the applications of these kinds of detecting methods.

Except for the above problems, previous researches are mainly focused on detecting commercials from special kinds of program, such as news or films [1]. The features they selected for representing commercials sometimes are not stable enough, since many of them vary with different TV stations or different countries. It's difficult to find a uniform detecting system by these features. Furthermore, many methods highly depend on black frames or monochrome frames, whether within commercial block or at the boundaries of commercial blocks. However, black frames can be easily removed by TV stations, and other programs like movies may also contain many black frames. In addition, audio features are not sufficiently utilized in the above three kinds of methods. Many methods do not use audio features [1][3][4][5], while others only use a very few of them [2].

In this paper, we try to deal with the above issues, and propose a robust learning-based TV commercial detection scheme. This scheme is different from the above mentioned three categories of methods. In the proposed approach, the TV video is firstly segmented into shots. Six basic visual features and five basic audio features are then extracted from the shots. Next a set of context-based features are derived from these basic features, which are more stable for identifying commercials. A SVM classifier is then applied to classify each shot as a commercial shot or a general program shot. And last, to make our scheme more robust, a couple of post-processing procedures are employed to refine the classification results. Experimental results show our scheme is quite robust and accurate for many types of TV programs.

The paper is structured as follows. Section 2 introduces the features we used. The learning-based detection scheme is described in Section 3. Experimental results and conclusion remarks are presented in Section 4 and Section 5, respectively.

2. FEATURE ANALYSIS

Six basic visual features and five basic audio features are applied in our scheme. However, these basic features do not always work well for distinguishing the characteristics of commercials from those of programs. So a set of context-based features are derived from them.

2.1 Basic Visual Features

Four of the six basic visual features are shot-based features. That is, we firstly segment the video into shots and then extract these four features from each shot. These four features, listed as below, are based on the observation that typically the visual content within commercial shots change much more rapidly compared with typical program shots.

- Average of Edge Change Ratio (A-ECR)
- Variance of Edge Change Ratio (V-ECR)
- Average of Frame Difference (A-FD)
- Variance of Frame Difference (V-FD)

Edge Change Ratio represents the amplitude of edge changes between two consecutive frames [6], which is defined as

$$ECR_m = \max\left(\frac{X_m^{in}}{\sigma_m}, \frac{X_{m-1}^{out}}{\sigma_{m-1}}\right) \quad (1)$$

where σ_m is the number of edge pixels in frame m, X_m^{in} and X_{m-1}^{out} are the number of *entering* and *exiting* edge pixels in frame m and m-1, respectively. A-ECR and V-ECR of shot C are then defined by

$$AECR(C) = \frac{1}{F-1}\sum_{m=1}^{F-1} ECR_m \quad (2)$$

$$VECR(C) = \frac{1}{F-1}\sum_{m=1}^{F-1} (ECR_m - AECR(C))^2 \quad (3)$$

where F is the number of frames in the shot.

Frame Difference (FD) is defined by

$$FD_m = \frac{1}{P}\sum_{i=0}^{P-1} \left| F_i^m - F_i^{m-1} \right| \quad (4)$$

where P is the pixel number in one video frame, F_i^m is the intensity value of pixel i in frame m. And A-FD and V-FD are obtained as the same way as A-ECR and V-ECR.

The other two basic visual features are time-based features. One is Shot Frequency (SF), i.e., the number of shots per second; the other is Black Frame Ratio (BFR), i.e., the number of black frames per second. **Figure** 1 shows an example of these six basic features and the corresponding commercial/program ground-truth. In the figure, A-ECR, V-ECR, A-FD and V-FD are the corresponding values of each shot, while BFR and SF are of each second. This figure shows these features are distinct from each other for commercials and programs.

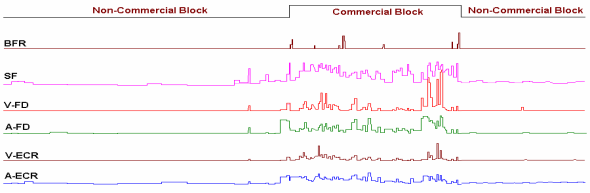

Figure 1. Feature curves of a sample video.

However, sometimes these features sometimes are not so capable of distinguishing commercials. **Figure** 2 shows another example. Here, we cannot clearly distinguish commercials and general programs only by these basic features. That is reason that we introduce a set of context related features that derived from these basic features, as to be presented later in Section 2.3.

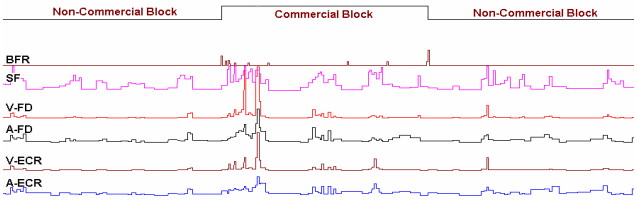

Figure 2. Feature curves of another sample video.

2.2 Basic Audio Features

Audio information also provides useful information for TV commercials detections. In the previous research works, audio features are not utilized sufficiently (only a feature called "*silence*" was applied). More audio features are adopted in our implementation including *audio break* and *audio types*. These audio features are extracted with a constant time step (0.5 second). Audio break frequency and the confidences of the four audio types are taken as the five basic audio features.

(1) *Audio Break Detection*

There are always audio transitions between commercial and other programs or between different commercials. Audio break detection is based on our previous work on speaker change detection [7]. The audio stream is first equally divided into sub-segments of 3 seconds with 2.5s overlapping. Each sub-segment is further divided into non-overlapping 25ms-long frames. Mel-frequency Cepstral Coefficient (MFCC) and short-time energy are extracted from each frame. K-L distance is applied to measure the dissimilarity of MFCC and energy between every two sub-segments.

Let $D(i, j)$ denote the distance between the i-th and j-th audio sub-segments. Thus, an audio transition break is detected between i-th and $(i+1)$-th sub-segments if the following conditions are satisfied:

$$D(i,i+1) > D(i+1,i+2), D(i,i+1) > D(i-1,i), D(i,i+1) > Th_i \quad (5)$$

The first two conditions guarantee that a local dissimilarity peak exists, and the last condition prevents relatively low peaks from being detected. Th_i is a threshold, which is automatically set according to the previous N successive distances. That is,

$$Th_i = \alpha \cdot \frac{1}{N}\sum_{n=0}^{N} D(i-n-1, \quad i-n) \quad (6)$$

where α is a predefined coefficient (working as an amplifier).

(2) *Audio Type Discrimination*

Silence has been used in previous work [2], however, more audio types are more helpful for commercials detection. For example, in general, there is more background music in TV commercials than other programs. In the proposed system, four audio types are applied: speech, music, silence and background sound. The classification method is based on our previous research works [8][9]. The confidence of each audio type is taken as feature in our system, thus totally we have a four-dimensional feature representing audio type.

2.3 Derived Context-Based Features

As well-known, if we only watch a few seconds of TV but do not know any contextual information, it is even difficult for human being to identify whether it is a commercial block or general

program. While after watching for some more seconds or minutes, we can then recognize it. This phenomenon enlightens us to use contextual information to automatically distinguish commercials from general programs.

The contexts of a shot can be represented by the features on its neighborhoods. As shown by Figure 3, in our implementation, we consider left and right neighborhoods of a shot separately instead of regard them as one neighborhood. This method can eliminate *boundary effects* when extracting features, which may lead classification errors for the shots on the boundaries of commercial and non-commercial blocks. The final feature set is then extracted based on this contextual information as follows.

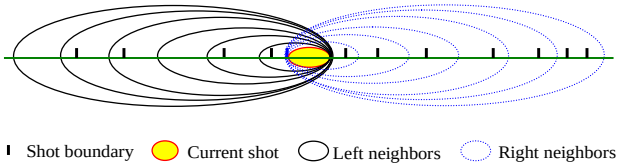

Figure 3. Neighborhood of current shot.

Let $[s_i, e_i]$ denote the start and end frame number of current shot C_i. For convenient, $[s_i, e_i]$ also represents the start and end time (in seconds) of the shot (for the time-based features we count them by time). The (2*n*+1) neighborhoods (including left *n* neighborhoods, right *n* neighborhoods and itself) of C_i are then defined as

$$N^k = [N_s^k, N_e^k] = \begin{cases} [\min(e_i + \alpha k, 0), e_i] & k < 0 \\ [s_i, e_i] & k = 0 \\ [s_i, \min(s_i + \alpha k, L)] & k > 0 \end{cases} \quad (7)$$

where L is the duration or total frame number of the TV program, and $k \in Z, |k| \le n$, β is the time step of the neighborhoods. In our experiments, we set $n = 6$, and $\beta = 5$.

Let S^k represent the set of shots that are partially or totally included in N^k, that is

$$S^k = \{C_j^k : 0 \le j < M^k\} = \{C_i : C_i \cap N^k \ne \Phi\} \quad (8)$$

where M^k is the number of shots in S^k.

The derived context-feature set is the average value of basic features on S^k (for shot based features) or N^k (for time based features). For example, A-ECR on S^k and BFR on N^k are obtained by

$$AECR_{S^k} = \frac{1}{\sum_{j=0}^{M^k-1}(e_j^k - s_j^k)} \sum_{j=0}^{M^k-1}(e_j^k - s_j^k) AECR(C_j^k) \quad (9)$$

$$BFR_{N^k} = \frac{1}{N_e^k - N_s^k} \sum_{j=N_s^k}^{N_e^k-1} BFR(j) \quad (10)$$

where $[e_j^k, s_j^k]$ is the start and end frame number of shot C_j^k. $BFR(j)$ is the black frame ratio in [*j*, *j*+1] (counted by second).

Thus, from the 11 basic features and the above derivation procedure, we can get $11 \times (2n+1)$ context-based features. In our experiments, it will be a 143-dimensional feature.

3. COMMERCIAL DETECTION SCHEME

3.1 SVM-Based Classification

In Section 2, we have got $11 \times (2n+1)$ dimensional feature for each shot. To detect commercials, Support Vector Machine (SVM) [8] is applied to classify every shot.

First a SVM classification model is trained from a pre-labeled training dataset. We then use this model to do shot classification. For a shot C_j, we denote the SVM classification output as $Cls(C_j)$, and $Cls(C_j) \ge 0$ indicates C_j is a commercial shot.

Generally commercial shots appear in groups in the broadcasting timeline, so as program shots. And commercial and program blocks/groups appear alternately. Based on these observations, two post-processing rules are applied to refine the detection results, as to be explained.

3.2 Post-processing

A series of commercial blocks and non-commercial blocks can be formed by merging consecutive commercial shots and consecutive non-commercial shots. However, some shots are misclassified and therefore the merged (non)commercial blocks sometimes are not correct. As mentioned above, commercial and program shots appear in groups, thus *scene grouping* can be applied to improve the classification accuracy.

It is observed that there is a big difference between general program blocks and commercial blocks in color. Thus color-based scene grouping is applied to recover the errors in the classification procedure. That is to say, in a certain scene, most likely all shots are commercial shots or all are non-commercial shots. Let

$$\begin{aligned} &Shot = \{C_0, C_1, \cdots, C_{N-1}\},\ N: number\ of\ all\ shots \\ &Scene = \{S_0, S_1, \cdots, S_{M-1}\}, M: number\ of\ all\ scenes \\ &S_k = \{C_0^k, C_1^k, \cdots, C_{N_k-1}^k\},\ \sum_{k=0}^{M-1} N_k = N \end{aligned} \quad (11)$$

represent all shots of the TV program, and the scene grouping results. The scene grouping algorithm used here is similar to the one in [11], while anchor person detection is excluded.

The post-processing procedure by scene grouping then is described as follows.

(1) *Scene Classification*: Classify each scene using the following rule

$$Cls(S_k) = sign\left(\sum_{j=0}^{N_k-1} sign(Cls(C_j^k))\right) \quad (12)$$

where $sign(x)$ is a sign function which returns 1 when $x \ge 0$ and -1 when $x < 0$. This rule indicates that if the number of commercial shots in S_k is not less than half of N_k, we regard this scene as commercial scene, and otherwise, it is classified as non-commercial scene.

(2) *Merging*: Merge consecutive non-commercial scenes and consecutive commercial scenes.

After applying the above rules, we obtain the initial (non)commercial blocks. For convenient, we still name these blocks commercial scenes and non-commercial scenes. However, the scene grouping result is not always correct. Some heuristic rules are employed to further correct obvious errors. Totally there are four rules applied in our system.

(1) *Remove Short Scene*: If a scene is too short, we merge it into the shorter scene of its two neighbor scenes.

(2) *Check Long Commercial Scene*: Since commercials will not last a long period, a long (detected) commercial scene may possibly include a non-commercial segment. And the mis-merged boundaries most likely lie in the boundary between two shots C_i and C_{i+1} if the following two constraints are satisfied at the same time (T_l is a predefined threshold),

$$Cls(C_i) \cdot Cls(C_{i+1}) < 0 \quad (13)$$

$$|Cls(C_i) - Cls(C_{i+1})| > T_l \quad (14)$$

Then the scene is split at this kind of locations.

(3) *Separate Long Commercial Part in Non-Commercial Scene*: Sometimes there are several consecutive commercial shots in a non-commercial scene. To separate them, the scene is split at the beginning and end of this long commercial part, if the number of the consecutive commercial shots is larger than a threshold T_c (Figure 4).

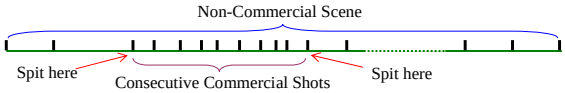

Figure 4. Separate Long Commercial Part in Non-Commercial Scene.

(4) *Refine Boundaries*: This is a user biased rule. If the user trends to keep all non-commercial/commercial shots, we need to check several shots in the beginning and end of each commercial/non-commercial scene. If a shot C_j of this kind is too *long/short* and $Cls(C_j)$ is *smaller/bigger* than a threshold, it is transferred to its closest non-commercial/ commercial scene. In our experiments, we choose to keep non-commercials.

There will be some split operations on scenes in the above processing. Please note after every split, we re-assign commercial properties for the affected scenes by equation (12), followed by a merging operation mentioned above. There are a few thresholds in the above rules. However, experiments have shown that the algorithm is not sensitive to them.

4. EXPERIMENTAL RESULTS

We evaluate our method on 10.57 hours of TV program, which includes 6.57 hours of NBC, 2 hours of ESPN2 and 2 hours of CNN. The types of the programs include news, sports, movie, entertainment and some integrated program. 1/4 shots of each videos are selected for training (2.34 hours in total). And we then test our algorithm on the remaining shots (8.23 hours). The evaluation results with or without post-processing are listed in Table 1 and Table 2.

In our experiments, we suppose the application is Digital Video Recorder for home users, which prefer to keep all the general programs while filter out commercials to the utmost. We use recall and precision to evaluate the performance of the algorithm, as showed in Table 1 and Table 2.

Table 1. Evaluation results of commercials

Post-processing	Total	Detect	False	Recall	Precision
Without	2.06	1.82	0.22	88.21%	89.39%
With	2.06	1.89	0.17	91.77%	91.65%

Table 2. Evaluation results of non-commercials

Post-processing	Total	Detect	FA	Recall	Precision
Without	6.17	5.95	0.24	96.60%	96.08%
With	6.17	6.00	0.17	97.72%	97.25%

From the above two tables, it can be seen that recall and precision are very high whether for commercials or non-commercials. It is up to about 92% and 97%, respectively. **Table** 3 shows the classification accuracy for both commercials and non-commercials: 94% before post-processing and 96% after it.

Table 3. Evaluation results of all TV programs

Post-processing	All Program	Correct	Accuracy
Without	8.23	7.77	94.42%
With	8.23	7.89	95.84%

5. CONCLUSION

In this paper, we have proposed a robust TV commercial detection approach based on learning from contextual features. In the scheme, each shot is firstly classified into either commercial or general program, and then commercial blocks are merged based on scene grouping and some heuristic rules. Performance evaluation on about 10.6 hours of data showed that our algorithm performs quite well for different kinds of programs.

As future work, text information and color saturation can be applied to enhance the accuracy of classification. PCA can be employed to reduce the feature dimension. More efficient post-processing procedures may also be adopted.

6. REFERENCES

[1] R. Lienhart, et al. "On the Detection and Recognition of Television Commercials," *Proc of IEEE Conf on Multimedia Computing and Systems*, Ottawa, Canada, pp. 509-516, June 1997.

[2] D. Sadlier, et al, "Automatic TV Advertisement Detection from MPEG Bitstream," *Intl Conf on Enterprise Information Systems*, Setubal, Portugal, 7-10 July 2001.

[3] T. Hargrove, "Logo Detection in Digital Video," http://toonarchive.com/logo-detection/, Mar 2001.

[4] R. Wetzel, et al, "NOMAD," http://www.fatalfx.com/nomad/, 1998.

[5] J.M. Sánchez, X. Binefa. "AudiCom: a Video Analysis System for Auditing Commercial Broadcasts," *Proc of ICMCS'99*, vol. 2, pp. 272-276, Firenze, Italy, June 1999.

[6] R. Zabih, J. Miller, K. Mai, "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks," *Proc of ACM Multimedia 95*, San Francisco, CA, pp. 189-200, Nov. 1995.

[7] L. Lu, H. J. Zhang, H. Jiang, "Audio Content Analysis for Video Structure Extraction," *Submitted to IEEE Trans on SAP*.

[8] L. Lu, H. Jiang, H. J. Zhang. "A Robust Audio Classification and Segmentation Method," 9th *ACM Multimedia*, pp. 203-211, 2001.

[9] L. Lu, Stan Li, H, J. Zhang, "Content-based Audio Segmentation Using Support Vector Machines," *Proc of ICME 2001*, pp. 956-959, Tokyo, Japan, 2001

[10] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp 121-167, 1998.

[11] X.Y.Lu, Y.F.Ma, H.J.Zhang, L.D. Wu, "A New Approach of Semantic Video Segmentation," *Submitted to ICIP2002*, New York, USA, Sept. 2002.