Automatic Generated Recommendation for Movie Trailers

Zhe Xu Ya Zhang

…

***Abstract*—Trailer generation is an important part of movie industry. Editing trailers manually is both time consuming and labor intensive. In this paper, we present an automatic scheme to select candidate movie clips for generating movie trailers. The candidates are well classified and ordered, allowing human editors to choose different types of clips systematically. We analyze professional trailers to guide the training process. The system leverages multimedia analysis techniques to generate candidates for movie trailers. Experiments have shown that clips chosen by the system can significantly outperform randomly picked clips.**

***Index Terms*—Video Analysis, Video Skim, Video Summarization, Face Recognition**

I. Introduction

Movie trailer, or short preview, consists of a series of selected segments from the movie. It serves as an important form of advertisement for a film. With the rapid growth of online multimedia/video hosting services such as YOUTUBE (www.youtube.com), movie trailers have become increasingly popular. According to [1], movie trailers have been observed to be the #3 most watched videos online annually, after news and user-created video. However, the production of movie trailer, as a creative process, has been mostly manual – both time consuming and costly. We want to seek for a method to ease the workload for producing trailers, either an automatic or a semi-automatic solution. Because the purpose of a movie trailer is to attract an audience to a film, trailers are usually drawn from the most exciting, funny, or otherwise noteworthy parts of the film. In another word, trailers have certain patterns which may allow (semi-)automatic generation.

Some may argue that the generation of movie trailers does not have a good feasibility for automation. Furthermore, for movie trailers, "good" and "bad" is a very subjective issue. Hence, there is no convincing ideal evaluation standard. Most importantly, movie trailers, different from standard video skim problems, have a specific characteristic called "story line", which cannot be extracted by video summarization approaches.

Considering of these problems, instead of directly compose a movie trailer, we tend to build a system to assist human editors to create the trailer more efficiently. By analyzing professional trailers, we build a classifier to indicate which kinds of clips are the most suitable ones for trailer generation. For a testing movie, the system provides candidate clips with the highest scores given by the classifier. Meanwhile, for several prototypes of movie trailers, such as the presence of main characters, dramatic camera motions and explosion scenes, the system again gives a pool of the most relevant clips. Even an un-trained person can create a movie trailer by choosing proper clips according to his script (or story line) using the proposed system.

Fig. 1 shows the flowchart of the proposed movie trailer system. First, we choose a set of professionally produced trailers as training examples, and extract multiple features. By matching each key frame in the trailer to its original position in the respective movie, we train a classifier based on the extracted features. Given a new movie, the system analyzes the movie according to its features and outputs a pool of candidate clips using the classifier. The generation of the recommending clips is highly automated.

The rest of the paper is arranged as follows. Section 2 briefly describes related work of the paper. Section 3 introduces the preprocessing process and the feature extraction process. Section 4 describes the training process, including the training data acquisition and training algorithms. Section 5 shows experimental results. Section 6 presents discussion on the result and some extension of the system.

II. Related Work

Video summarization is an extensively studied topic, a detailed discussion can be found in [2] and [3]. The basic idea of video abstracting or video skimming is that it analyzes and condenses a video into several important clips [4] [5] [6]. Much previous works in video summarization focused on sports and news videos [7]. These videos have well-defined structures and characteristics which makes it easier to summarize the main content. The most difference between automatic generation of movie trailers and sports videos summarization is that for movie trailers, the predominant characteristics mostly appear in

Zhe Xu and Ya Zhang are with Institute of Image Communication and Network Engineering & Shanghai Key Laboratory of Multimedia Processing and Transmissions, Shanghai Jiao Tong University. Ya Zhang is the corresponding author (phone: +86-21-34204468; fax: +86-21-34204155; e-mail: ya_zhang@sjtu.edu.cn).

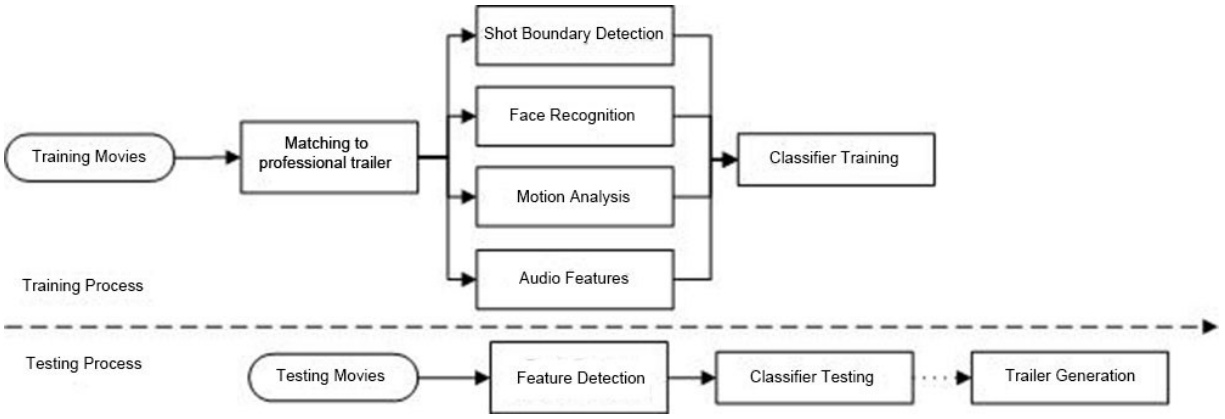

Fig. 1. Flowchart of the proposed movie trailer generating system

semantic level. To extract these characteristics, low-level image features such as color, motion is not enough. The process need to be guided by higher level of features, such as face recognition, explosion detection and some more complicated training algorithms.

Although video skimming has been studied for a while, few studies have focused on automatically generating movie trailers. Hermes et. al. design a fully annotated system for producing movie trailers[8]. The major different between their work and ours is that different feature extracting methods and video segmentation algorithms are used. In addition, we employ a machine learning framework to automatically learn the key features of the trailers. Finally, we do not tend to leave the creative part of writing scripts to computers – the proposed system only outputs a candidate pool which makes the trailer producing process easier.

III. VIDEO PROCESSING

A. Shot Boundary Detection

For shot boundary detection, we use the algorithm of HS optical flow as a measure of motion [9]. Optical flow calculates the intensity and direction of the image pixels' motion. Due to the high computation complexity, we downsample the frame images so that the operation can be performed in acceptable time cost. Moreover, the result of optical flow can be reused for motion analysis during feature extraction.

Varies types of transitions exist between shots, such as cut, fade, dissolve, and wipe. A cut is detected if there is a sudden change between two adjacent frames. One of the most difficult parts of shot boundary detection is to deal with different transition types. Here we define two thresholds and use the idea of running difference to determine the shot boundary. If the intensity falls between the upper and lower thresholds, a running difference is kept to determine whether it is a fade or just a part of a high motion scene. The detail is shown in Algorithm 1.

B. Key Frame Extraction

A key frame summarizes the content of a video sequence. By turning the whole movie into a set of key frames, feature extraction and other analysis can be performed much more efficiently. We follow the approach described in [10] for key frame extraction. Key frames are selected at the local minimal of motion. An important feature of this algorithm is that it does not always extract a fixed number of key frames per shot. The number varies according to the complexity of the shot. Shots with more changes on motion intensity tend to generate more key frames.

Key frame is the minimum unit used in the proposed system. We define a "clip" as the video sequence determined by a key frame, which means starting from the local maximum point before this key frame and ending at the local maximum right after this key frame.

Algorithm 1 Shot boundary detection

Input: threshold th_1 and th_2, where $th_1>th_2$. A set of intensity numbers given by optical flow $\{i_1,i_2,\ldots,i_n\}$. Initial as not in transition mode.
Output: the result of shot boundary detection
for each i
if not in transition mode
if $i>th_1$ *return* cut;
if $i<th_2$, do nothing;
if $th_1<i<th_2$, from this frame, calculate running difference r, enter transition mode.
if in transition mode
if $r>th_1$ *return* fade/dissolve/wipe(according to each i in transition mode), leave transition mode;
if $r<th_2$, leave transition mode;
if $th_1<r<th_2$, recalculate running difference r, according to the first frame of transition mode.

C. Feature Extraction

Given the segmented clips, we extract features in terms of actor appearance, motion and sound. What is worthy to note is that for different types of movies, the feature extraction process can be customized.

Motion Analysis

Motion of a video is calculated by optical flow as mentioned above. Each frame is assigned by a value of average intensity of optical flow among all pixels. The resulting motion measure is the average of all frames in a shot. Shots with high motion indicate action.

Moreover, we want to distinguish two kinds of motion - camera motion and object motion. That is accomplished by

analyzing the moving direction of the pixels. For camera motion such as lifting left or right and zoom in or zoom out, moving direction of all pixels follows a certain distribution in macro view. For object motion such as action scenes, only the nearby pixels move simultaneously.

Face recognition

Character introduction is a key factor for movie trailers. A good trailer should contain all the leading characters. To recognize these characters, the easiest way is to recognize their faces. For face detection, we use the pre-trained Adaboost classifier with Haar feature available in OpenCV. Experiments show that most frontal perspective faces with proper scales are detected. Failure detection occurs when persons are too far from camera or with back perspective. The false-positive rate is about 10 to 15 percent.

Face recognition is achieved with Principle Components Analysis (PCA). First, a set of faces detected in the previous procedure are chosen as prototypes and labeled by human editors. Eigenfaces are extracted from the set of labeled prototypes. Other faces are labeled by the most similar prototype.

Notice that in a movie, even the faces of the same character can vary a lot due to complicated factors such as illumination, perspective, expression, and makeup (Consider a movie which tells a story that lasts 20 years, all characters are growing old). In order to achieve better recognition accuracy, instead of character images found in the poster, we use a human annotation step to pick up the prototypes. 5 percent of the detected faces are chosen according to different time slot as prototypes. They are identified by a human annotator so that prototypes containing different characters driven by different factors are available.

Sound Volume Detection

For audio features, average and highest sound volume are calculated. High volume together with high frequency sound indicates scream or accident.

Fig. 2. Samples of recognized faces for the hero of the movie “Source Code”. Faces bounded by the red frames are false positives.

Speech and Music Detection

There are a lot of features feasible for speech and music detection. We use zero-crossing rate and percentage of low energy frame. These features are used together with other visual and audio features to train a classifier. This procedure also gains information in frequency domain, so we omit the beat detection part.

Other features

Other features of brightness, contrast, a shot's frame length are extracted to support the visual and audio features. These low level features together can convey valuable information. For example, long shot with low brightness near the end of the movie mostly belong to the credit part, which is not suitable for trailers.

IV. Training Process

Having obtained the segmentation results and multiple features, the next thing to do is to determine what kinds of patterns are preferred for movie trailers.

A. Training Data Acquisition

We propose a supervised learning approach for trailer generation. Therefore, the first thing to do is to acquire a training set. The optimal approach to build such a training set is to manually analyze movies and trailers and store the correspondence in a database [8]. Unfortunately, it requires heavy labor work and is very time consuming. Therefore, we design an alternate approach. Key frames of both trailers and movies are extracted, after which a matching process is performed by using the method Speeded Up Robust Features (SURF) [11]. When a match is found, the key frame together with other key frames in the same shot are labeled as positive training examples.

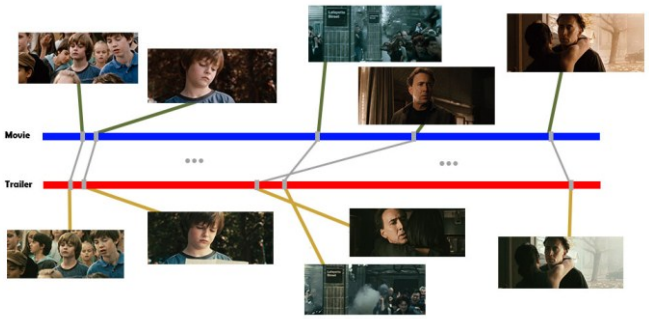

Fig. 3. Sample demonstration of SURF matching process (Movie: *Knowing*).

Although SURF matching may suffer from low efficiency, this acquisition process is needed only in training, which means once done then it's no longer needed in testing process.

B. Training algorithm

We train a global classifier to judge whether a clip should be a part of the final trailer, results in a candidate pool of movie clips.

Several algorithms can be used for training a classifier. Hidden Markov Model (HMM) [12] and Support Vector Machine (SVM) [13] are the most popular methods. In the

analysis of movie trailers, temporal relationship between different clips plays an important part. Adjacent clips usually have an effect on each other, such as a sequence of clips with high motion and high sound volume indicates action scene. In that opinion, HMM is more appropriate for modeling temporal feature. However, due to the severe sparsity of training data (most of clips in movies are not part of the relative trailer), the performance of SVM outperforms HMM significantly, which encourage us to use SVM as the training model.

Considering the sparsity problem, to make full use of information in adjacent clips, we introduce a sliding window trick in SVM training process. Features of not only the current clip, but also the N clips before and after it contribute to the final feature vector. We empirically assign N=3. Therefore, the context features are included in training the classifier by SVM. Different features are organized by feature concatenation and fed to the classifier. The finally vector length is 48. We use SVM with RBF kernel and use professional trailers as training examples. The process trains a global classifier to give each key frame a score indicating the probability to appear in the trailer.

Despite of global candidate pool, for each significant feature, the system also provides a candidate pool. This is done by recommending the top 20 scored key frames among which satisfied the given feature on prior. Consider a case as below, an editor wants to find a piece of clip that contains the leading actor. He can do the job easily because the system provides appropriate key frames and orders them according to the score given by classifier.

V. EXPERIMENTS

The proposed algorithm is performed on 8 movies. 4 of them are selected for cross-validation, the rest are used for testing. Video segmentation and feature extraction process are done for each of the movies. For training movies and their official trailers, SURF matching process is done. As a result, vary among different movies, 500-1500 shots, 5000-8000 key frames are extracted, 5-15 percent of the key frames are labeled as positive, others are negative.

Testing movies are analyzed via the same procedure of segmentation and feature extraction. We use SVM with RBF kernel and probabilistic estimation and find the top 100 clips as a pool of recommenders. All these clips are shown in a displaying canvas. A human editor goes through these clips and then chooses some of them by clicking the submit button. The choice can be based on his own script, which introduces a a story line in the trailer. A demo of the displaying canvas is shown in Fig. 4.

There is no single criterion to determine whether a trailer is good or not. We propose to use both statistical analysis and manual evaluation to evaluate the trailers. In the end, we introduce a user study to find out the advantages and disadvantages of the movie trailer generated by the proposed system.

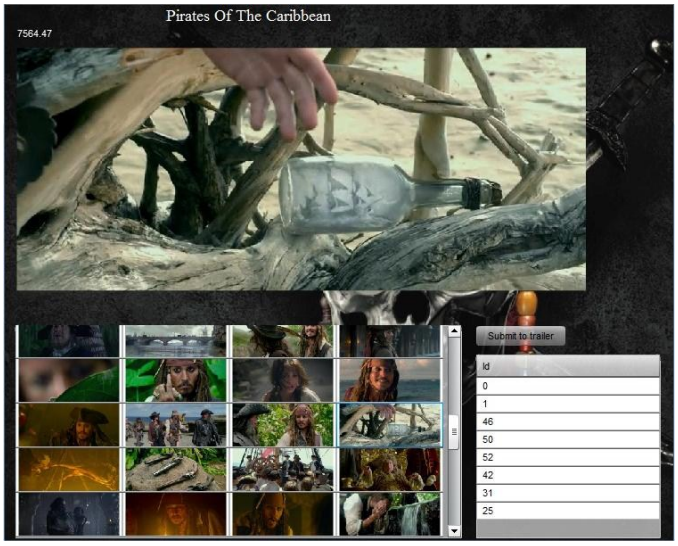

Fig. 4. Result displaying canvas for *Pirates of the Caribbean 4*. Click a thumbnail and the relative clip will be on play at the top. A human editor can select some clips and click submit button, the selections are shown in the right.

Statistical Analysis

Based on the result of SURF matching process, we compared the result of the proposed system and random chosen clips, which is calculated by the number of matches found.

TABLE 1

Precision (%)	FF	PC	K	SC
Proposed	7.8	24.4	17.6	5.8
Random	3.8	12.0	10.0	4.2

Comparing our result with random selection based on the matches found by SURF. For movies, FF for *Fast Five*, PC for *Pirates of the Caribbean 4*, K for *Knowing* and SC for *Source Code*.

Table 1 shows that the proposed system outperforms random selection by about 100% in precision. Furthermore, we manually compare our trailer and the official trailer shot by shot, and calculate the recall of first 100 candidates, we find that the recall is 15%, which is significantly better than the random selection, which is about 1%.

Manual Analysis

Furthermore we analyze the overall performance of the proposed system on several aspects. First, as the result shows, the algorithm tends to give high scores on continuous clip sequences. In *Fast Five*, a gunfire scene and a car driven scene at the beginning of the movie get high scores. In *Knowing*, clips in a plane crashing scene have the highest scores. Indeed the same scenes also appear in the official trailer, in the form of long shots. It can be explained that long shots in the trailer should be a combination of multiple key features, which will receive higher scores by the classifier.

According to the official trailer, most misses of the proposed system are flash-and-off clips and introduction for supporting characters. Indeed the flash-and-offs can be replaced by other clips recommended by our system. As we have already got low level features, customized amendments such as introducing each character individually can be done straightforwardly.

Fig. 5. Top 15 recommenders given by SVM for *Fast Five*. Notice that some of them come from the same scene, indicating that this scene should get a long shot in the final trailer.

User Study

We invite 10 audiences to rate the results of the proposed system and random selected clip pools. Most of the audiences are college students who like watching movies. We consider character, scene and plot as three critical criteria. Using the recommending pool given by the proposed system, we choose the top scored clips and string them up in time sequence to generate a trailer. Results show that proposed algorithm received higher ratings on aspects of "key characters introduction", "magnificent scene" and "overall impression", but lower ratings on "support characters introduction", since we focus more on leading characters, respectively ignore other characters. Both proposed and randomly selected clips received low scores on "plot introduction" which depended on semantic analysis.

VI. DISCUSSION

In this section, we explicitly describe the use case of the system. Firstly, a human editor writes a script for the trailer. For each plot, he or she checks clips around the time slot in the recommender pool and pick up a desired one. As a result of the discussion above, similar scenes that appeared several times in the recommender pool can be considered to form the main part of long shots in the trailer. Other recommending clips together with clips generated according to man-make rules like high action and character introduction, fill in the remaining part of the trailer.

A natural extension of the system lies in the combination of multimedia analysis. By analyzing the relationship between natural language and image or audio features, a smarter and higher automatic system can be made. The main difficulty is how to gain enough training data for determining the relationship between multimedia features and natural language. Fortunately, there are many online datasets feasible for the use of image-text co-clustering. The ideal solution is that the system outputs a trailer automatically according to a script written by the editor. That is regarded as the future work.

VII. CONCLUSION

In this paper, we presented an approach to generate movie trailers semi-automatically. By analyzing available official trailers, utilizing video segmentation, visual and audio feature extraction and model training algorithms, clips with the highest rating score are presented as a pool of recommenders. The test result shows that the proposed system gives reasonable recommenders for movie trailers, which outperforms random selection significantly. As a consequence, the system enables human editors to produce high quality trailers without reviewing the whole movie several times. We believe that the system can be used in practice with some customized amendments.

ACKNOWLEDGMENT

This work was supported in part by the High Technology Research and Development Program of China (2011AA01A107, 2012AA011702)，National Natural Science Foundation of China (61221001), the 111 Project (B07022), and the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

REFERENCES

[1] “Awfj opinion poll: All about movie trailers,” AWFJ, vol.2008-05-09.
[2] B. Truong and S. Venkatesh, “Video abstraction: A systematic review and classification,” ACM Trans. Multimedia Comput., Commun., Appl. (TOMCCAP), vol. 3, no. 1, p. 3, 2007.
[3] A. Money and H. Agius, “Video summarisation: A conceptual framework and survey of the state of the art,” J. Vis. Commun. Image Represent., vol. 19, no. 2, pp. 121–143, 2008.
[4] M. G. Christel, A. G. Hauptmann, A. S. Warmack, and S. A. Crosby, “Adjustable filmstrips and skims as abstractions for a digital video library,” in Proc. IEEE Forum Research and Technology Advances in Digital Libraries ADL ’99, 1999, pp. 98–104.
[5] Changick Kim and Jenq-Neng Hwang, “Object-based video abstraction for video surveillance systems,” vol. 12, pp. 1128–1138, 2002.
[6] Ying Li, Shih-Hung Lee, Chia-Hung Yeh, and C.-C. J. Kuo, “Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques,” vol. 23, pp. 79–89, 2006.
[7] B. Li and I. Sezan, “Semantic sports video analysis: Approaches and new applications,” in Proc. ICIP, 2003, vol. 1.
[8] T. Hermes and C. Schultz, “Automatic generation of hollywood-like movie trailers,” in cat1.netzspannung.org, 2006.
[9] B.K.P. Horn and B.G. Schunck, “Determining optical flow,” in Artificial Intelligence, 1981, vol. 17, pp. 185–20.
[10] W. Wolf, “Key frame selection by motion analysis,” in Proc. Conf. IEEE Int Acoustics, Speech, and Signal Processing ICASSP-96, 1996, vol. 2, pp. 1228–1231.
[11] Bay H, Tuytelaars T, Van Gool L (2006) Surf: Speeded up robust features. Computer Vision - ECCV 2006 pp 404-417
[12] L.E.Baum and T.Petrie, “Statistical inference for probabilistic functions of finite state markov chains,” vol. 37, pp. 1554–1563, 1966.
[13] Joachims T. Hofmann T. Tsochantaridis, I. and Y. Altun, “Large margin methods for structured and interdependent output variables,”.

(a) *If You Are the One 2*

(b) *The Battle of Los Angeles*

(c) *Knowing*

(d) *Pirates of The Caribbean 4*

Fig. 6. Typical result of the proposed system, top 10 key frames scored by SVM classifier.