Semantic Video Abstracting: Automatic Generation of Movie Trailers Based on Video Patterns

Till von Wenzlawowicz and Otthein Herzog

Technologie-Zentrum Informatik und Informationstechnik,
Universität Bremen, Am Fallturm 1, 28359 Bremen, Germany
{tillvw,herzog}@tzi.de

Abstract. Summarizing and abstracting of multimedia data is a field of great research interest, especially for multiple video platforms. In this paper we describe a system capable of generating Hollywood movie trailers automatically by using audio and video processing algorithms, combined with ontology-based knowledge and CLIPS, a rule based system. We show that the system is capable of generating convincing movie trailers for the action genre. Further work has been done to extend the results to other movie genres.

Keywords: video analysis, semantic representation of video, video abstracting, video synthesis, movie trailer generation.

1 Introduction

The recent progress in automatic video and audio processing enables multiple application fields. A classical case of video abstracting are movie trailers, used to attract people to watch a movie by summarizing the content to some extent and to create excitement and expectations. The rather artistical task of creating a trailer, nowadays perfected by the movie industry, is challenged by the automatic approach outlined in this paper. By using video and audio analysis techniques various low-level features of video files can be extracted and combined to higher level semantic features. Our approach deals with the combination and processing of these features and semantic knowledge about trailer structures to automatically generate trailers.

The paper is structured as follows: First related techniques and systems are described. Then the structure of a typical Hollywood trailer is described. The subsequent section discusses the approach and the software system used to generate trailers. The paper concludes with an evaluation of the generated trailers and the conclusion.

2 Related Work

The general problem of creating a video abstract, the summarization of the content of the video, is an emerging research field. Different ways of video abstracting are described by Truong and Venkatesh[8]. Two main approaches are

I. Maglogiannis, V. Plagianakos, and I. Vlahavas (Eds.): SETN 2012, LNAI 7297, pp. 345–352, 2012.
© Springer-Verlag Berlin Heidelberg 2012

named, keyframing and video-skimming. Keyframing summarizes the content of a video in one or more still images while video skimming provides a shorter version of the video. A summarization of multi-view videos is described and applied to surveillance camera footage by Fu et al. in [3]. The task of finding specific actions in movies is described in [4] where the action *drinking* can be recognized. The term *trailer generation* is explicitly mentioned by Lienhart et al. in [6] and Chen et al. [2], both using film theory in order to select footage without focusing on the generation of trailers.

3 Trailer Structure

In modern cinema the trailer is one of the most important ways to advertise an upcoming film. It therefore has the task to attract a broad audience and convince people to watch the movie. A common trailer introduces the setting, location and characters of the story to be told, and presents elements of the basic plot of the movie.

These characteristics can be found in the selection of scenes and their arrangement. In a typical Hollywood trailer for an action movie five different phases can be identified:

1. Intro: The people, setting and location are introduced.
2. Story: The problem or task to be solved or challenged is portrayed and the relationships between the characters are shown.
3. Break: A dramatic moment, often combined with a dramatic comment by a main character.
4. Action: Fast and loud, spectacular scenes to draw attention.
5. Outro: Mostly calm and slow footage, often showing a main character in a closeup shot and making a tough or comic comment, followed by the title, credits and a release date.

In order to be able to describe the structure in a computable way a corresponding representation was developed, the *trailer grammar*. This grammar consists of syntactic elements and semantic rules.

The basic elements of a video are usually *shots* and *transitions*, thus they can be defined also as the syntactic elements of a trailer. In order to distinguish between shots in the movie and shots used in the trailer grammar the latter ones are called *clips*.

The semantic rules for the composition of the trailer are defined in a generic hierarchical structure displayed in Figure 1. The structure consists of *patterns* and *pattern lists*. The first level is the *trailer pattern*, which consists of five *phases patterns*, corresponding to the phases described above. Each phase contains *sequence patterns* which are constructed of *clip/transition pairs*, the syntactic elements described before. This generic structure allows the modeling of a specific trailer model incorporating semantic knowledge.

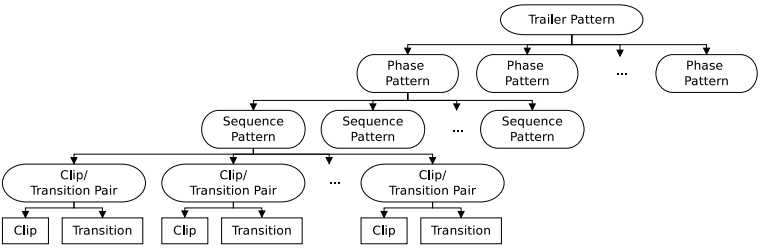

Fig. 1. The hierarchical trailer structure, shown is a branch of the generic model (taken from [1], page 149)

3.1 Action Trailers

The first kind of trailers chosen to be analyzed and generated are the trailers of a typical Hollywood action movie. A trailer of this genre usually focuses on spectacular shots and scenes like explosions, gunshots, fights and less on a story-based narrative structure. These spectacular scenes are defined by features which can be automatically detected and used as elemental parts for the decomposition of a video.

3.2 Other Genres

Trailers for other genres than action movies mostly tell more about the characters of the movie. This is very challenging for automatic detection and generation as it requires a system to research the narrative units used to tell the story of the movie. An interesting question is how a trailer for a non-action genre can be generated using an adapted model for action trailers.

4 Automatic Trailer Generation System

In order to automatically generate trailers, a movie needs to be segmented into shots, which can then be categorized into a list of pre-defined shottypes. Using the semantic rules defined in the trailer grammar these shots can then be arranged and composed to generate a new video, the trailer.

4.1 Approach

For the automatic generation the syntactic elements and the semantic rules for action trailers must be defined to build a specific trailer model. This was done in a first step by a manual shot-by-shot analysis of 11 action trailer movies and resulted in a system of 19 clip categories (see Table 1) and a set of semantic rules, describing from which category clips should be arranged. Additional seven categories for non-movie footage were defined, containing textual animations for the title of the movie, the main actors, the credits and a few others.

Table 1. Clip Categories

Footage Clip			Animation Clip
Character1CUSilent	PersonSpeaking	SlowAction	ActorName
Character1CUSpeaking	Quote	Spectacular	CompanyName
Characer1Silent	QuoteLong	Shout	Credits
Character1Speaking	Explosion	Scream	DirectorProducer
PersonCUSilent	Fire	Setting	Greenscreen
PersonCUSpeaking	Gunshot		Tagline
PersonSilent	FastAction		Title

In order to automatically classify the shots of a movie into semantic categories a two step method is used. First various audio and video processing algorithms analyze the movie for low-level features, such as the motion in a certain frame range or the loudness of the audio track. The results are then combined to improve the classification and find higher-level features. A shot classified in the category *PersonSpeaking* for example needs to show a face in small size, has speech present in the audio track and should not exceed a certain length.

Once the shots of the movie are categorized, a *trailer template*, derived from the specific trailer structure and described by the trailer grammar, can be filled with corresponding clips. The animations can be generated and and the trailer video file, together with trailer music and sound effects, can be composed.

4.2 Architecture

The software system can be split into two parts: The first part is a set of video and audio processing tools, which analyze a movie. The second part consists of the generator module. The main parts of the generator are an ontology, where the knowledge about the categories and about the semantic trailer structure are stored using the trailer grammar, and a CLIPS[1] component which performs the actual generation. Additionally the generator controls the rendering module and the final composition of a trailer.

The analyzer modules are mostly based on open-source software and described briefly below. A detailed description of the system is available in [1].

Shot detection. The shot detection is using a tool developed in[7]. It performs the basic segmentation of the movie into shots based on gray-level histogram changes.

Motion-based segmentation. Using the Lucas Kanade feature tracker provided in the OpenCV library[2] this module calculates the motion in the movie and segments it into frame ranges with similar optical flow characteristics.

[1] `http://clipsrules.sourceforge.net/`
[2] `http://opencv.willowgarage.com/wiki/`

Face detection. This module performs a face detection in the movie using the Haarcascade classifier[5] from OpenCV.

Face recognition. The results of the face detection are clustered by using PCA and k-means clustering. The output is the frame range where a certain character is detected.

Text detection. The text detection is based on a tool described in[9]. It filters frames containing text to avoid them during the generation of the trailer.

Sound volume-based segmentation. This module assists the visual detection of loud events like explosions and action sequences, and indicates dialogues and calm clips with a lower volume.

Sudden volume change detection. A sudden change in volume indicates surprising and spectacular clips.

Speech detection. It is important to know whether a character is speaking or not, because seeing a person speaking while not hearing it is irritating. The speech detection uses the CMU Sphinx speech recognition system[3] together with the included HUB4 acoustic model and AN4 3gram language model. The module determines the frame ranges with speech, while content of the speech is not important.

Speech recognition. The speech recognition module finds famous quotes by a character in the movie. It takes a set of quotes, taken from the Internet Movie Database[4] and converts them into a word-phoneme dictionary by using addttp4[5]. This allows the module to search the audio track of the movie for the quote and enables the system to include the corresponding clip into the trailer later on.

Shout detection. This module looks for loud passages in the results of the speech detection, which are categorized as shouts.

Music detection. Using stable power spectrum peaks as an indication for music this module looks for parts of the movie where music is present to avoid mixing movie music and trailer music.

Sound event detection. The sound event detection module is used to find pre-trained sound-samples for gunshots, explosions, crashes and screams. This is done by using a Support Vector Machine[6] approach on feature vectors extracted from the soundtrack.

[3] `http://cmusphinx.sourceforge.net/`
[4] `http://www.imdb.com`
[5] `http://www.nist.gov/speech/tools/addttp4-11tarZ.html`
[6] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

The output of the detection modules is merged into one xml file per movie. The file contains the results from all detectors and additional metadata such as resolution and length of the video file. It also contains data about the movie which is obtained from the Internet Movie Database like the director, year of production and the most important actors.

The generator part of the movie takes the movie file and the corresponding analyzer data as input and assigns the movie shots to their clip categories (see Table 1). In the ontology the categories are defined via *category parameters*. These parameters determine the properties the footage needs to have in order to be classified into the category, for example a maximum length or volume.

After the categorization process is finished the CLIPS-based system determines a suited trailer template from the hierarchical trailer model in the ontology by selecting a valid instance using the predefined rules. Then the system fills the leafs of the hierarchy with categorized clips. In the case of a requested category without clips left alternatives are searched in a similar category.

Requested animation clips are generated using information from the metadata and the open source 3D-modeling software Blender[7].

Music and sound effects are selected from a pool of audio files containing typical trailer music and sound effects. The music is arranged according to the phases described above and selected randomly while following certain rules in the ontology to guarantee a harmonic soundtrack.

The movie and the additional media files are then merged into the final trailer video file using avisynth[8] and virtualdub[9]. Figure 4.2 shows an excerpt of a automatically generated trailer for the movie *Terminator 2*.

Fig. 2. 18 of 56 clips showing parts of our automatically generated *Terminator 2* trailer (complete Intro Phase: 1-6, middle part of the Action Phase: 35-41, and complete Outro Phase: 53-56). The corresponding type of category is given below each clip.

[7] http://www.blender.org
[8] http://avisynth.org/
[9] http://www.virtualdub.org/

4.3 Applying to Different Genres

We applied the automatic trailer generation to movies of genres other than action as well. The nature documentary *Earth* was chosen to look how our action trailer model would apply on such a movie. First we started generating trailers using the action trailer template together with the action music pool. In a second step we chose a different set of music files containing classical music which would be a better fit to the genre. Finally we developed a new trailer model by only using the categories *slow action*, *fast action* and *setting* and omitting categories which focus on human characters and action specific features like explosions and gunshots.

5 Evaluation

The automatically generated action trailers were evaluated by showing a test set of seven different trailers to a user group of 59 people. The participants of the study should then give a rating for the quality of the trailer. The test set consisted of two professional trailers, *War Of The Worlds* and *Miami Vice*, a trailer for *The Transporter* generated by the video generation software *movee*, two trailers generated by our system, *Bad Boys* and *Blade*, with a random selection of shots and finally two trailers, *Transporter 2* and *Terminator 2* based on the trailer patterns defined in the system.

The results are shown in Figure 3. The automatically generated trailers with a given score of 7.29 and 7.26 compete well in comparison to the professional Hollywood trailers (scored 7.86 and 4,92 by the participants). The evaluation shows that our trailer model is well superior to a random selection of shots.

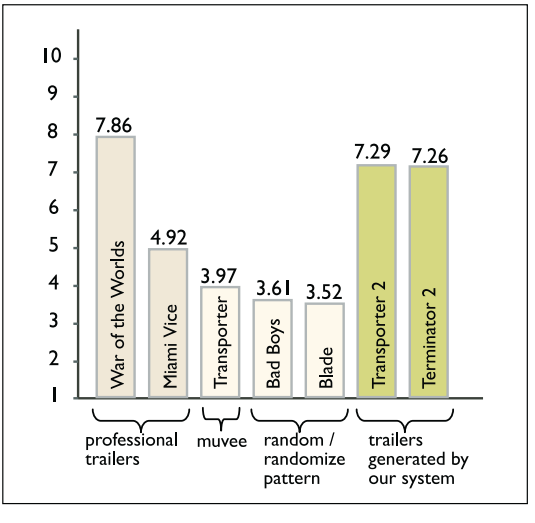

Fig. 3. Mean score of the evaluated trailers

Preliminary results for the application of our action model the genre of nature documentary appear promising. A spontaneous evaluation in our group rated the trailer with classical music convincing and pointed out that the choice of music is important for the impression of the trailer.

6 Conclusion

In this paper we presented a fully automatic system capable of generating Hollywood like trailers for action movies. We described the basic structure of such a trailer and our implementation of the knowledge using an ontology-based trailer model. The evaluation shows that the quality of the automatically generated trailers can compete with professional composed ones.

Acknowledgements. We would like to thank the Graduate School*Advances in Digital Media*, funded by the *Klaus Tschira Foundation*, and the participants of the master project *Semantic Video Patterns* on which results this paper is based:

Christoph Brachmann, Hashim Iqbal Chunpir, Silke Gennies, Benjamin Haller, Philipp Kehl, Astrid Paramita Mochtarram, Daniel Möhlmann, Christian Schrumpf, Christopher Schultz, Björn Stolper, Benjamin Walther-Franks, Arne Jacobs, and Thorsten Hermes.

References

1. Brachmann, C., Chunpir, H.I., Gennies, S., Haller, B., Kehl, P., Mochtarram, A.P., Möhlmann, D., Schrumpf, C., Schultz, C., Stolper, B., Walther-Franks, B., Jacobs, A., Hermes, T., Herzog, O.: Automatic Movie Trailer Generation Based on Semantic Video Patterns, Media Upheavals, vol. 27, pp. 145–158. Transcript Verlag (2009)
2. Chen, H.W., Kuo, J.H., Chu, W.T., Wu, J.L.: Action movies segmentation and summarization based on tempo analysis. In: MIR 2004: Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval, pp. 251–258. ACM Press, New York (2004)
3. Fu, Y., Guo, Y., Zhu, Y., Liu, F., Song, C., Zhou, Z.H.: Multi-view video summarization. IEEE Transactions on Multimedia 12(7), 717–729 (2010)
4. Laptev, I., Perez, P.: Retrieving actions in movies. In: IEEE 11th International Conference on Computer Vision, ICCV 2007, pp. 1–8 (October 2007)
5. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: IEEE ICIP, vol. 1, pp. 900–903 (September 2002)
6. Lienhart, R., Pfeiffer, S., Effelsberg, W.: Video abstracting. Communications of the ACM 40(12), 54–62 (1997)
7. Miene, A., Dammeyer, A., Hermes, T., Herzog, O.: Advanced and adapted shot boundary detection. In: Fellner, D.W., Fuhr, N., Witten, I. (eds.) Proc. of ECDL WS Generalized Documents, pp. 39–43 (2001)
8. Truong, B.T., Venkatesh, S.: Video abstraction: A systematic review and classification. ACM Trans. Multimedia Comput. Commun. Appl. 3 (February 2007), `http://doi.acm.org/10.1145/1198302.1198305`
9. Wilkens, N.: Detektion von Videoframes mit Texteinblendungen in Echtzeit. Master's thesis, Universität Bremen (2003)