A Semi-automatic Approach to Detect Highlights for Home Video Annotation

Peng Wu
Hewlett-Packard Laboratories
1501 Page Mill Road, Palo Alto, CA 94304
peng.wu@hp.com

Abstract

This paper presents a semi-automatic highlight detection system for home video annotation. To automatically identify highlights from home videos is a challenging research issue in general. Currently home users mostly use video editing tools to manually find the highlight, which is very time consuming. To alleviate this hurdle and promote the reusability of home video material, we propose a semi-automatic user environment that aims at reducing the editing time required for users to find highlights. With a well designed user interface and using the localized visual similarity trail to estimate candidate highlight boundaries, we enable home users quickly and mostly accurately identify the highlight. The initial evaluation on a home video database demonstrates a 60% saving on editing time.

Keywords: Highlight detection, home video annotation, localized similarity trail

1 Introduction

We consider video highlights to be the most semantically preservable portions of a video file or a video segment from a viewer or a group viewers' perspective. In general, to determine a portion of a video as highlight is a highly subjective process. Many research works have been presented to detect highlights from domain specific videos, such as sports videos and news videos (see [1], for example). In these works, the prior domain knowledge, which essentially captures the most widely accepted notions of what are preservable in a video, is often introduced to define highlights. However, it is difficult to apply any prior knowledge to generic home videos to pre-define highlights, considering the uncountable life events that people may be interested. Mainly due to this reason, there has not been much work addressing highlight detection for home videos.

On the other hand, video highlight detection is of great interest in home video annotation, particularly with the proliferation of camcorders and digital cameras available to more and more home users. Since most home videos are non-professional video material, it often happens that the interesting parts are mixed with long-winded, less interesting video segments. Thus, to promote the reusability of home video material, it is highly demanded to develop efficient highlight detection technologies for home videos.

Currently, most home users rely on video editing tools, such as Adobe Premiere to manually identify highlights, which is a very time consuming and often frustrating process. In this paper, we present a semi-automatic highlight detection system to alleviate this time consuming process. With a well designed user interface and video analysis algorithms, we enable home users an interactive environment to quickly and most time accurately find video highlights. Our objective is to reduce the amount of time required for home users to find highlights, compared with using the traditional video editing tools.

This paper is organized as follows. Some related work is discussed in Section 2. In Section 3, we describe the semi-automatic highlight detection system in detail; The experimental results are presented in Section 4 and concluding remarks in Section 5.

2 Related work

The related work can be classified into two categories:
Video highlight detection: Domain specific highlight detection has been addressed by many researchers. Some recent works include [1], [2]. The work in [1] demonstrates a typical usage of domain knowledge in helping the highlight detection. In [1], a few types of highlights, such as "home run", are predefined for baseball videos as highlights. These concepts of highlights are further associated with various types of scene shots, which can be modeled and computed. Thus, by detecting the occurrence of scene shots, one can estimate the instance of highlights; In [2], the authors intend to identify the most "representative" video parts, which are closely related to the concept of "highlight", to compose the skimmed video. This work does not restrict itself in a particular domain. However, the authors rely on a certain set of predefined audio cues (noun-phrases in [2]) as the indicators of interesting video portions. We argue that for home videos, such predefined cues may not be generic enough to capture the highlights.
Home video editing: There are several research works devoted to home video annotation/management [3][6][7]. In [3], a system is presented to create custom videos from

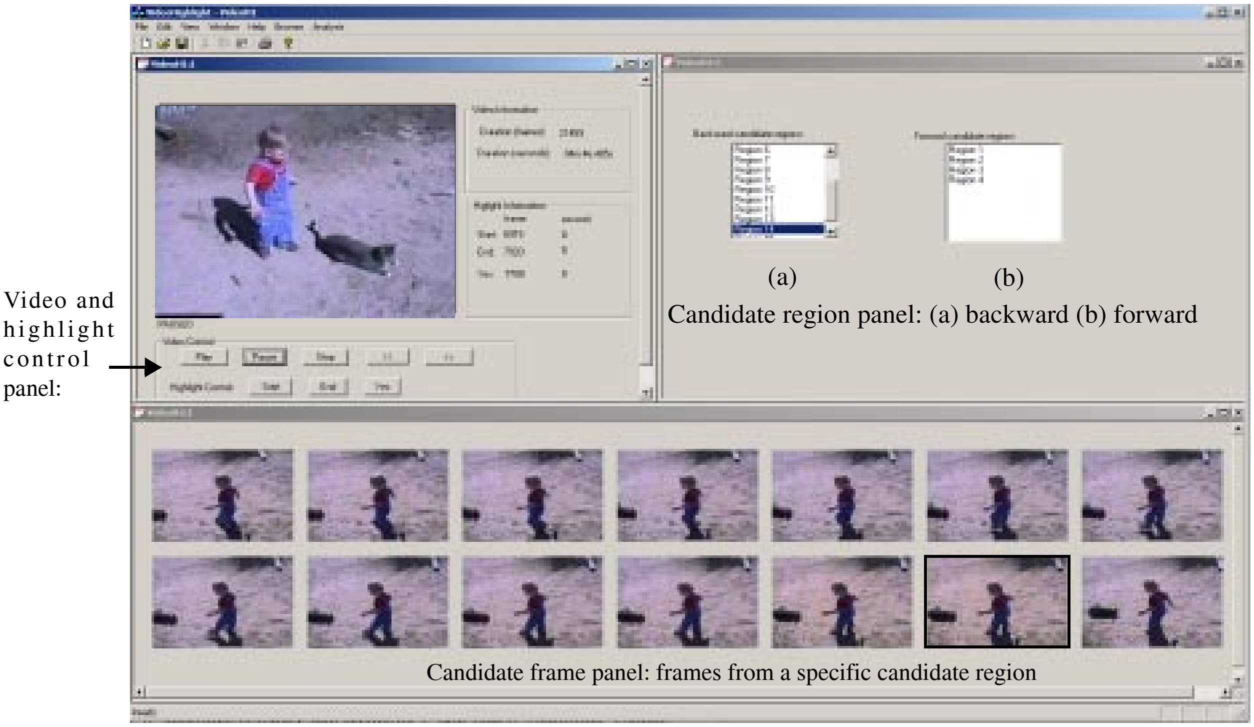

Video and highlight control panel:
(a)
(b)
Candidate region panel: (a) backward (b) forward
Candidate frame panel: frames from a specific candidate region

Fig 1. User interface for semi-automatic highlight detection.

the raw video material. The main idea of this work is to compute the unsuitability "score" for video segments to be contained in the final cut by analyzing the erratic camera motions. Then combined with a set of editing rules and user's input, the system generates the custom video; The work in [6] utilizes the time-stamp to create time-scale clustering at different levels for home video browsing; A home video browsing and indexing system is described in [7]. It organizes videos using the conventional notions of keyframe, shot and scene. This system also implements a face tracking/recognition functionality to index videos. Apparently, these works have different objectives than the focus of our work.

One interesting work, [4], aims at learning personalized highlights given that a video is pre-labeled with metadata. Though our work shares the same interest as creating personalized highlights, we address this challenge at a different level, as we focus on how to generate the metadata labels from the unedited video.

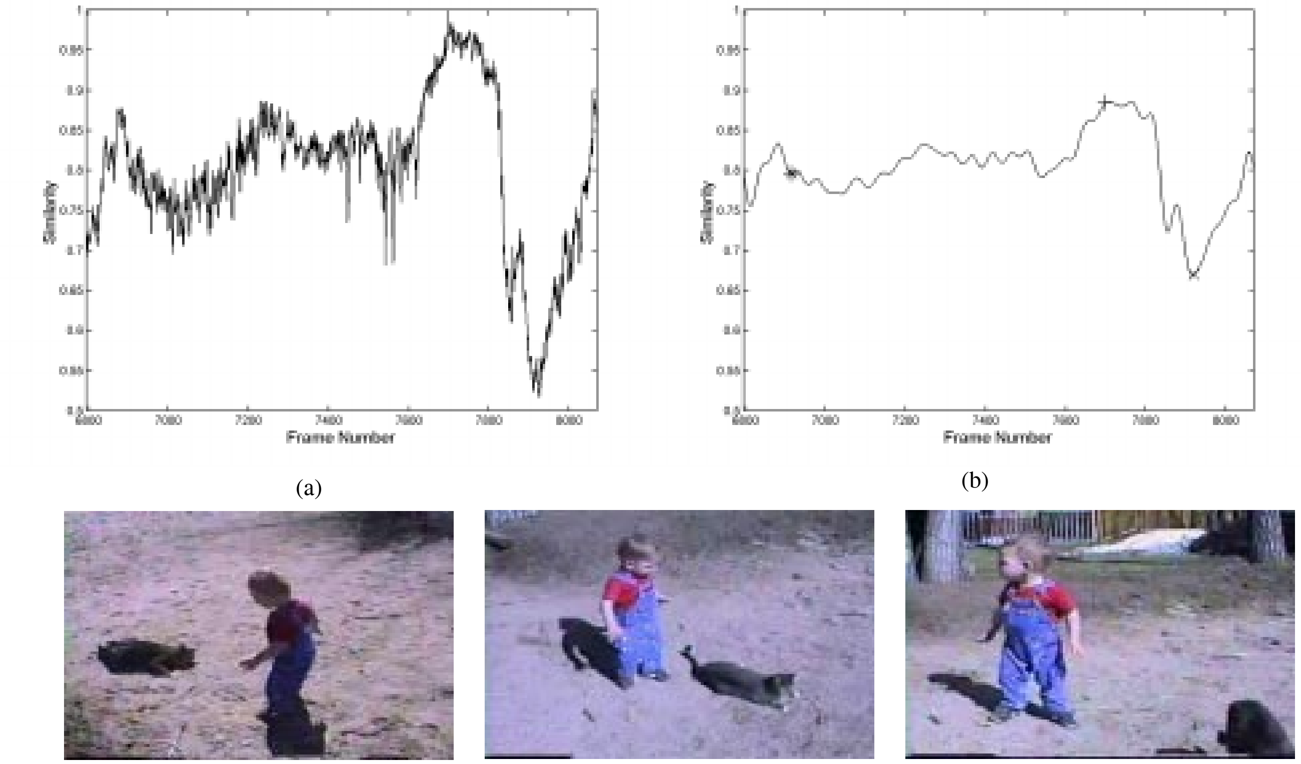
In the following, we introduce the semi-automatic home video highlight detection system.

3 Semi-automatic highlight detection

3.1 System Overview

Traditionally, using the existing software video editing package, a user has to go through the following step to find a highlight and its boundaries in a video: 1) browsing the video to identify an interesting video segment; 2) play the video forward and backward to locate the boundary regions; 3) zoom in frame by frame to find the boundary frames. Usually, step (2) and (3) take a large amount of editing time.

In the proposed approach, a new user environment is constructed so that the editing time can be greatly reduced (see Figure 1). The user interface is composed by three panels: 1) Video and highlight control panel; 2) Candidate region panel; 3) Candidate frame panel. Specifically, a user, while browsing the video, clicks on the "Yes" button in the video and highlight control panel whenever he or she considers the currently displayed frame belongs to a highlight; We call such a frame "reference frame". Using the reference frame, the system estimates a set of small video segments that are likely to be the boundary regions. The estimated candidate boundary regions are presented on candidate region panel. Note that there are two listboxes, namely "Backward candidate regions" and "Forward candidate regions" respectively. Separated by the reference frame, each region in the backward (forward) candidate region list provides a possible start (end) boundary region of the highlight. For any candidate region the user selected from the candidate region list, a set of frames contained in that region will be presented on the candidate frame panel to let user determine if a boundary frame can be found or not. As compared with the traditional approach, the approach, mainly by estimating a set of possible boundary regions based on a simple user's feedback,



(a)

(b)

(c)

(d)

(e)

Fig 2. (a) Localized similarity trail; (b) Localized similarity trail after low-pass filtering ("+" indicates the reference frame provided by a user; "*" indicates the highlight's start frame and "x" indicates the highlight's end frame which are both identified by the home video user.) (c) start frame; (d) reference frame; (e) end frame.

relieves user's involvement in finding the boundary regions and boundary frames.

In the following, we first introduce the estimation of candidate boundary regions and secondly, the user environment.

3.2 Boundary region estimation

3.2.1 Video pre-processing

Feature extraction: We compute two generic features, color histogram and edge energy to characterize each frame. For each frame $f_j, j = 1, \ldots, L$, where L is the total number of frames of a video, we denote $C(f_j)$ and $E(f_j)$ to be the color and edge energy feature descriptors of frame f_j respectively.

The color descriptor is a color histogram of 256 bins constructed in YUV color space. The edge energy descriptor is the standard deviation of the edge energy of a frame. To compute the edge energy, we apply the "Prewitt" edge operator to the Y component of the image frame. The two gradients used in the edge detection are defined in (1), where G_R and G_C are the row and column gradients respectively.

$$G_R = \frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \text{ and } G_C = \frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad (1)$$

Shot detection: After the feature computation, the video is segmented into shots using a color histogram based shot detection algorithm (see [5]). Denote the detected shots to be $Shot_i$, $i = 1, \ldots, N$, where N is the total number of shots. We assume that a highlight resides within a shot.

3.2.2 Localized similarity trail

As mentioned in Section 3.1, a user is asked to provide a indication if he or she considers a currently displayed frame is part of a highlight. We denote such frame to be f^*. This feedback is used to construct a localized similarity trail. Let $Shot_i$ to be the shot that contains f^* and $Shot_i$ begins at frame f_m and ends at f_n. Then the similarity trail is an one dimensional function defined as

$$T(j) = 1 - (d_{color}(C(f_j), C(f^*)) + d_{edge}(E(f_f), E(f^*)))/2 \qquad (2)$$

where $m \le j \le n$ and $C(\ )$ and $E(\ )$ denote the color and edge energy feature descriptors described in Section 3.2.1. d_{color} and d_{edge} are the dissimilarity measures for color and edge energy features. In this paper, we use normalized L-1 distance to compute both d_{color} and d_{edge}.

Considering the wide variety of the content of home videos, we choose these two rather generic features than some context dependent features, such as features related with face, skin color and so on.

3.2.3 Boundary region estimation

Given the localized similarity trail, we identify the candidate boundary regions to be the valleys on the trail. To remove the small variations on the trail, a low pass filter is applied to the original trail. Figure 2(a) and (b) demonstrate the original trail and the trail after filtering. Figure 2(b) also demonstrates that the valleys provides a well coverage of the boundary frames.

From the filtered trail, a set of candidate regions are identified, denoted as R_k, $k = 1, \ldots, K$, where K is the total number of the candidate regions. Each region contains a fixed number of frames, which are centered at the valley frame of that region. The candidate region frame panel in Figure 1 shows such frames from a selected region. We further arrange R_ks such that R_{k-1} appears before R_k for any k.

3.3 User interface

The composition of the user interface has been introduced in Section 3.1. Several further notes regarding this user environments are:

1) In the video and highlight control panel, there are “start” and “end” buttons available for users to indicate a frame displayed on the media player to be the start and end frames of the highlight. These controls enable users use the conventional approach to find the boundaries if the estimated candidate regions do not contain the desired boundary frames;

2) In our current implementation, each candidate region contains 14 frames, as shown in Figure 1, which is mainly determined by the display devices.

4 Experimental Results

We conduct our evaluation on a video database that contains approximately 70 hours of home videos collected from 7 contributors. These videos cover a wide variety of life events and shot by non-professional customers. The main purpose of the evaluation is to measure how much time saved in finding the boundary frames using the proposed approach.

For experiment purpose, the contributors are asked to label the highlights on about 20 hours of videos using Adobe Premier and our system. There are about 50 highlights identified. In average, using Adobe Premier, it takes 2.5 minutes to find the boundaries. Using our system, we observe a 60% saving on editing time.

To evaluate the accuracy, we use the boundaries found using Adobe Premier as the ground truth. For 41 out of 50 highlights, the exact boundaries are found in one of the candidate regions. There are 6 cases the users identified different frames as the boundaries, but those frames are recognized as acceptable replacement boundaries based on user’s evaluation. Meanwhile, we observe that for 3 out of 50 highlights home users can not find the boundary frame from the candidate regions but have to use the “start” and “end” button to locate the boundary frames. Two of the 3 cases are because that the shot detection algorithm categorizes the boundary frames into a different shot. And for the other case, the valley that contains the boundary frame is flatten out after the low pass filtering.

Overall, the semi-automatic highlight detection system enables both a time efficient and accurate environment to find highlights in home videos.

5 Conclusions

An interactive system is described in this paper for highlight detections in home videos. The initial experiment results demonstrate the editing efficiency of the system. Currently, we adopts two simple and generic features to characterize video frames. It is possible to include other generic features so that the frames are better characterized, for example, texture feature. We expect to report such progress in the near future.

References

[1] P. Chang, M. Han, and Y. Gong, “Highlight detection and classification of baseball game video with Hidden Markov Models,” Proceedings of the International Conference on Image Processing (ICIP '02), 2002.

[2] Michael G. Christel, Michael A. Smith, C. Roy Taylor, and David B. Winkler, “Evolving video skims into useful multimedia abstractions,” Proc. Human factors in computing systems, Los Angeles, California, United States, pp. 171-178, 1998.

[3] Andreas Girgensohn, John Boreczky, Patrick Chiu, John Doherty, Jonathan Foote, Gene Golovchinsky, Shingo Uchihashi, and Lynn Wilcox, “A semi-automatic approach to home video editing,” Proc. the 13th annual ACM symposium on User interface software and technology, San Diego, California, United States, pp. 81-89, 2000.

[4] A. Jaimes, T. Echigo, M. Teraguchin, and F. Satoh, “Learning personalized video highlights from detailed MPEG-7 metadata,” Proc. International Conference on Image Processing, Volume: 1, pp. 133-136, 2002.

[5] Ying Li, W. Ming and C.-C. Jay Kuo, "Semantic video content abstraction based on multiple cues,” Proc. ICME2001, Japan, August 2001.

[6] Rainer Lienhart, “Abstracting home video automatically,” Proc. the 7th ACM international conference on Multimedia, Orlando, Florida, United States, pp. 37-40, 1999.

[7] Wei-Ying Ma and HongJiang Zhang, “An Indexing And Browsing System For Home Video”, Invited paper, EUSIPCO'2000, 10th European Signal Processing Conference, 5-8, Sept. 2000, Tampere, Finland.