ACADEMIC PRESS

Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

J. Vis. Commun. Image R. 14 (2003) 358–368

JOURNAL OF VISUAL Communication & IMAGE Representation

www.elsevier.com/locate/yjvci

On the preview of digital movies

Liang-Hua Chen,[a,*] Chih-Wen Su,[b] Hong-Yuan Mark Liao,[c] and Chun-Chieh Shih[c]

[a] *Department of Computer Science and Information Engineering, Fu Jen University, Hsinchuang, Taipei, Taiwan, ROC*
[b] *Institute of Computer Science and Information Engineering, National Central University, Taiwan, ROC*
[c] *Institute of Information Science, Academia Sinica, Taiwan, ROC*

Received 11 September 2002; accepted 8 May 2003

Abstract

In this paper, a new technique is proposed for the automatic generation of a preview sequence of a feature film. The input video is decomposed into a number of basic components called shots. In this step, the proposed shot change detection algorithm is able to detect both the abrupt and gradual transition boundary. Then, shots are grouped into semantic-related scenes by taking into account the visual characteristics and temporal dynamics of video. Finally, by making use of an empirically motivated approach, the intense-interaction and action scenes are extracted to form the abstracting video. Compared with related works which integrate visual and audio information, our visual-based approach is computationally simple yet effective.

Keywords: Video content analysis; Video segmentation; Movie trailer

1. Introduction

As recent advances in computational power and storage capacity, the potential for large digital video libraries is growing rapidly. Owing to the sheer volume of data and unstructured format, efficient access to video is not an easy task. It is imperative

* Corresponding author. Fax: +886-2-2341-5838.
E-mail address: lchen@csie.fju.edu.tw (L.-H. Chen).

1047-3203/$ - see front matter
doi:10.1016/S1047-3203(03)00036-1

to give users necessary summarizing and skimming tools so that they can quickly find the content they interest. The *abstract* of a video sequence is a short synopsis which preserves the essence of the original video content. If effective abstraction tools were available, the users could evaluate ten hours of video in tens of minutes and determine quickly which portions to examine in further detail. Depending on its objective, there are two basic forms of video abstraction:

Preview: Its objective is to reduce a long video into a short sequence that is used to help the user to determine if a video program is worth viewing entirely (Hanjalic and Zhang, 1999a; He et al., 1999; Nam and Tewfik, 1999; Pfeiffer et al., 1996; Smith and Kanade, 1997; Toklu et al., 2000).

Browsing: It consists of a set of key frames which can be used to guide a user to locate specific video segments of interest (DeMenthon et al., 1998; Gong and Liu, 2000; Uchihashi et al., 1999; Zhang et al., 1997).

In our work, we focus on the preview of digital movies. In current movie marketing, it is common to produce a trailer (short summary) of a movie to get people interested. However, manual production of trailer is time-consuming and costly. There is, therefore, a need to develop procedures for the automatic summarization of movie content.

Until now, only a few techniques have been proposed for the automatic generation of preview sequences of video. One of the most straightforward approaches is based on the dynamic sampling of the underlying video sequence (Nam and Tewfik, 1999). The local sampling rate is directly proportional to the amount of visual activity in localized "sub-shot" units of the video. At playtime, linear interpolation is performed to provide the viewer a moving storyboard. However, the compression ratio is too low (6:1) and no discussion on how to handle the accompanying audio track is reported. Hanjalic and Zhang (1999a) apply multiple partitional clustering to all frames of a video sequence. Then the optimal number of extracted video segments (clusters) is determined by a cluster-validity analysis. For long video sequence such as movie, this method will result in too many clusters and make the abstracting video too long. Another common approach is the integration of speech recognition and image understanding techniques (Christel et al., 1998; Pfeiffer et al., 1996; Smith and Kanade, 1997; Toklu et al., 2000). In this approach, audio as well as visual information is used to extract important content from a video. However, it is computationally expensive. Moreover, satisfying results may not be obtained from video with audio track containing more than just speech, or video clip which is silent.

In this paper, we propose an efficient technique for video abstraction using not audio or text but visual contents of the given video. Our approach is based on the construction of high-level video structure. Since shots are marked by physical boundaries only, in our approach, shots are grouped into semantic-related scenes by taking into account the visual characteristics and temporal dynamics of video. Then, a compact representation of video content called scene transition graph is built. By analyzing the structure of scene transition graph, some important/interesting video clips are extracted and form the preview sequence of the original video.

2. The proposed approach

A video is physically formed by shots and semantically described by scenes. A shot is a sequence of frames that was continuously captured by the same camera. A scene is basically a story unit and consists of a small number of interrelated shots that are consecutive or not. In view of such underlying structures, it is desirable to automatically identify both visual and temporal relations in video and extract a compact representation of the story. Thus, the proposed approach is made up of three main steps: (1) Segmentation of video into shots. (2) Grouping of shots into scene. (3) Selection of video clips to form preview sequence. Each of these steps is described in the following subsections.

2.1. Segmentation of video into shots

Shot is the fundamental unit of a video. Shots can be joined together by either an *abrupt transition* (cut) or a *gradual transition*. In abrupt transition, two shots are simply concatenated, while in the gradual transition, additional frames may be introduced using editing operations such as fade in, fade out, dissolve and wipe. A good video segmentation technique should be able to detect shots with both types of transition. The existing shot boundary detection techniques can be classified into five categories: pixel based (Zhang et al., 1993), statistics based (Hanjalic and Zhang, 1999b), transform based (Yeo and Liu, 1995), feature based (Zabin et al., 1995), and histogram based (Swanberg et al., 1993). Several researchers claim that the histogram-based approach achieves good trade off between accuracy and speed (Gargi et al., 2000). The histogram-based algorithm is implemented by comparing the histogram of two consecutive frames. If the frame difference is greater than a threshold, an abrupt transition is detected. A problem arises when the transition is gradual; the shot does not change abruptly but over a period of few frames. The difference between two frames is not so large to declare it a shot boundary. Here, we propose an improved algorithm to detect both types of shot boundary.

The basic idea is that the frames before and after a gradual transition are usually markedly different. Instead of the difference between two consecutive frames, we compute the difference between two frames which are k frames apart. Let $H(f_m, i)$ denote the number of pixels of gray value i in the mth frame f_m, the histogram difference between the frame f_m and its kth predecessor is defined as

$$fd(m, m-k) = \frac{1}{2N} \sum_i |H(f_m, i) - H(f_{m-k}, i)|,$$

where N is the number of pixel in a frame. A similar formulation is defined for color image. Given a sequence of frames $f_1, f_2, \ldots, f_n$ and a fixed number k, the *frame difference sequence* $fd(k+1, 1), fd(k+2, 2), \ldots, fd(n, n-k)$ can be determined. Frame f_m is detected as a gradual transition boundary (see Fig. 1), if $fd(m, m-k) > \alpha$ (threshold) and $fd(m, m-k)$ is a local maximum of the frame difference sequence. Frame f_m is detected as an abrupt transition boundary (see Fig. 2), if $fd(m, m-k)$ is the first element of the subsequence $fd(m, m-k), \ldots,$

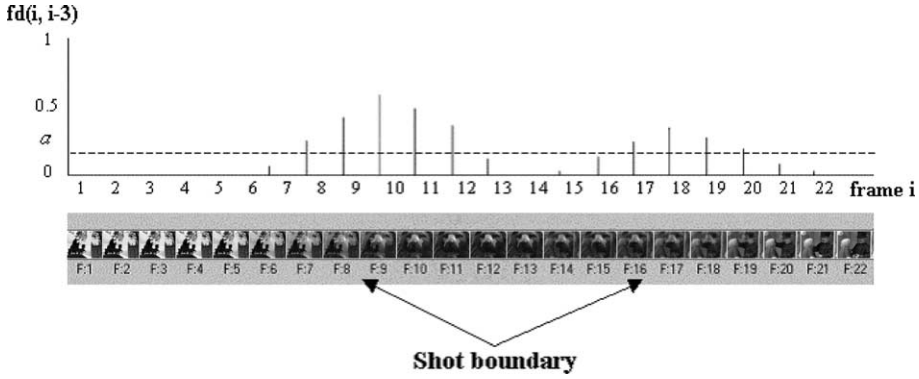

Fig. 1. The detection of gradual transition.

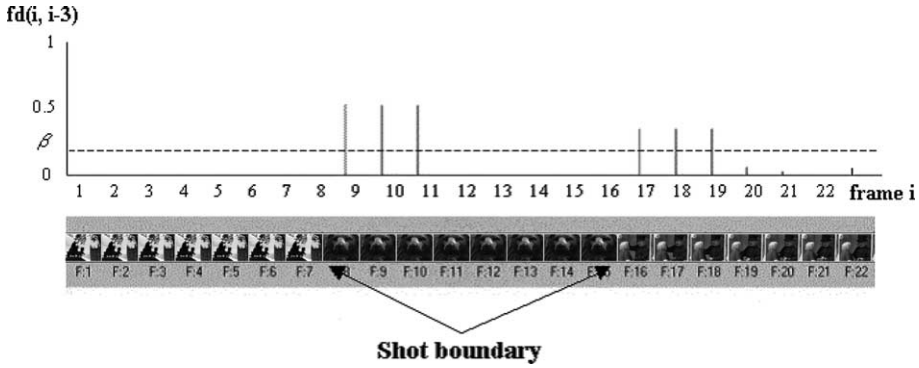

Fig. 2. The detection of abrupt transition.

$fd(m+s, m+s-k)(s>0)$ that are greater than β (threshold). If k is too large, the algorithm will miss some short shot boundary. To determine appropriate k value, we consider the minimal length of a video shot. In movie making, each shot lasts at least for 1/3 s to impress audience. Therefore, for 30 frames/s video, k should not be larger than 10.

Like most of the reported work in the literature, our approach still can not avoid the over-segmentation (i.e., false detection of shot boundary) problem resulting from the following factors:

(I) Object moving close to the camera and covering most of the frame surface.
(II) Significant camera/object movement.
(III) Sharp lighting change such as flashing lights.

However, by some post-processing, we can remedy this problem partially.

After video is segmented into shots, key frames can be extracted from each shot. Key frame is the frame which can represent the salient content of the shot. For any shot after an abrupt transition, a natural and easy way is to choose the first frame of that shot as the key frame. However, for a shot after a gradual transition, it is

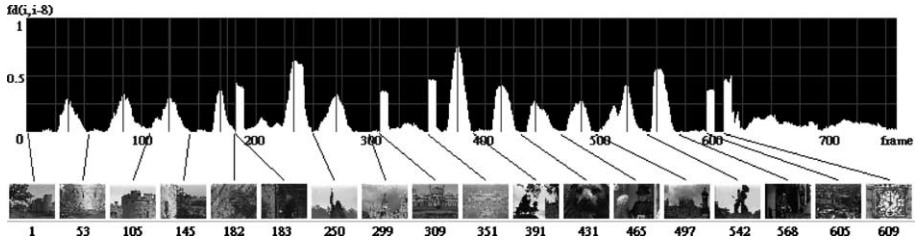

Fig. 3. An example of shot boundary detection and key frame extraction.

possible that the first frame is part of a dissolve effect at the shot boundary, which strongly reduces its representative quality. Therefore, for a shot after a gradual transition, we choose frame f_m as the key frame if $fd(m, m-k)$ is the first element of the subsequence $fd(m, m-k), \ldots, fd(m+s, m+s-k)(s > 0)$ that are smaller than a threshold. If no such f_m exists, we choose a frame f_m with the minimum $fd(m, m-k)$ as the key frame. Next, we remedy the over-segmentation problem resulting from factor (I). For two shots before and after a gradual transition, if the histogram difference of their respective key frames is smaller than a threshold, these two shots should be merged together into a single one. Finally, we test the proposed shot boundary detection algorithm on a 760-frame video sequence with five abrupt transitions and 12 gradual transitions. As shown in Fig. 3, both types of shot boundaries are all correctly detected and the extracted key frames are displayed at the bottom.

2.2. Grouping of shots into scenes

Since people watch the video by its semantic scenes not the physical shots, shots cannot convey meaningful semantics unless they are purposely grouped into semantic-related scenes (story units). In the segmentation of a still image, pixels are grouped into the same cluster if they are homogeneous with respect to some characteristics. Likewise, we want to group shots into several clusters, each of which represents a collection of interrelated shots that are unified by some common characteristics. In other words, each cluster composes a meaningful unit in the story. The common characteristics of shots we seek are the mutual interactions in terms of visual similarities and temporal localities. Therefore, in our approach, shots are grouped on the basis of their visual contents and temporal localities. Two shots are *similar* (or semantic-related) if they are visually similar and temporally close. On the other hand, two shots that are far apart in time but similar in visual content should belong to two different scenes. Given two shots S_i and S_j with respective key frames f_{k_i} and f_{k_j}, a similarity measure between these two shots is defined as

$$D(S_i, S_j) = \begin{cases} fd(k_i, k_j) & \text{if } |k_i - k_j| < T, \\ \infty & \text{otherwise.} \end{cases}$$

With this similarity measure, we apply classic data clustering technique (such as complete-link method Jain and Dubes, 1988) to group shots that are similar together into a cluster (scene). The choice of threshold T is also critical. A too large T can

render two distinct scenes to be grouped into one story unit, while a too small T can cause a scene to be broken into several story units. For application likes video abstraction, we prefer over-segmentation rather than under-segmentation. It is less detrimental to have several story units represent a scene than to have one story unit represent several scenes—these scenes cannot be recovered in subsequent analysis. In our experiment, we set T to be 90 s (or 2700 frames for 30 frames/s video).

To represent the video content in a compact way, we adopt the "scene transition graph" proposed by Yeung and Yeo (1997). A scene transition graph is a directed graph, such that each node represents a scene consisting of a cluster of visually similar shots, and a directed edge is drawn from node A to B if there is a shot in node A that immediately precedes a shot in node B. Therefore, the clustering results and the temporal relationships of the shots in the clusters are used to derive such a representation. This representation allows some form of analysis of video through the analysis of graph.

2.3. Selection of video clips

To choose video clips (segments) for inclusion in the abstract, some heuristic criteria are needed. Obviously, the criteria would vary depending on the type of video: documentary film abstract should give a good overview of the contents of the entire video whereas feature film abstract should be entertaining in itself. In our work, we concentrate on abstracting feature films. For feature films, the criteria used include: (1) important contents and (2) attractive to the viewer.

In most feature movies, important content is characterized by intense-interaction scenes. This is a consequence of the montage presentation of film—that movie director often emphasizes the important content by controlling the camera to repeatedly alternate among several scenes. On the other hand, action clips are often more interesting and carry more content in a short time than calm clips. Action scenes such as gunfire, explosions and car chases, attract attention and make viewers curious.

To extract intense-interaction scene, we analyze the structure of scene transition graph. Since the edges depict the temporal flow of the story, the degree of interactions among scenes is determined by the number of edges connecting to the corresponding nodes. In our approach, two nodes (scenes) are extracted if there are more than three edges between them. Fig. 4 is a portion of the scene transition graph of the movie "Four Weddings and a Funeral," where the shown image is the key frame of each video shot. The number below each key frame indicates the temporal order. Unfortunately, because of copyright consideration, we cannot show the content of each image. Fig. 5 shows the extracted intense-interaction scenes. One important issue is the determination of the length of selected video clip. Based on some psychological experiments, Pfeiffer et al. argued that a scene must be at least 3.25 s long to get completely analyzed (Pfeiffer et al., 1996). Therefore, for each extracted scene, we select the shortest shot which length is also larger than 3.25 s. If no component shot is larger than 3.25 s, the longest shot is chosen as video abstract.

Action scene is often characterized by the fast object/camera movement, or sharp change in lighting. Thus, to extract action scene, we make use of the consequence of

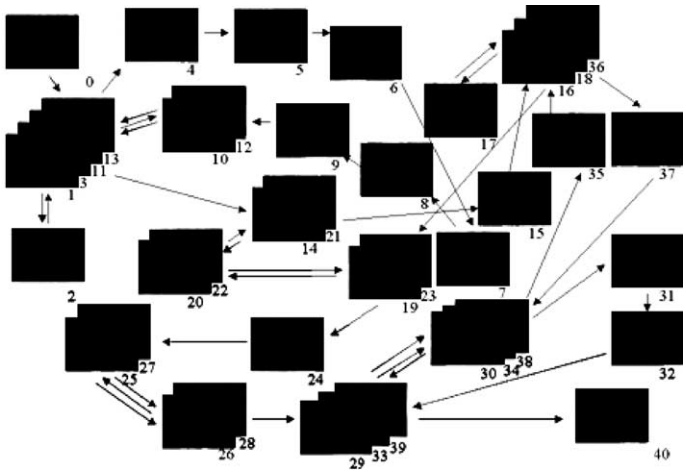

Fig. 4. Scene transition graph for "Four Weddings and a Funeral." The images cannot be shown because of copyright consideration.

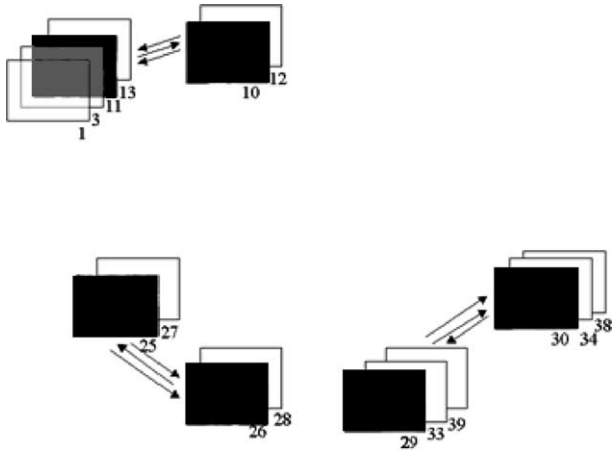

Fig. 5. The extracted intense-interaction scenes.

over-segmentation resulting from factors (II) and (III). Such scene is most likely found in the shot sequence where there is a consecutive subsequence of short shots. Then, a 3-s video clip is selected from the identified scene to form the preview sequence. Figs. 6 and 7 show the frames difference of video sequences with significant camera movement and sharp light change respectively. As indicated in these two figures, there are many short shots in the video.

Another feature of our technique is that the end of the movie is not revealed. To keep the suspense of story, we do not include clips from the last 10% of the movie. Finally, the selected video clips and their respective audio tracks are composed into

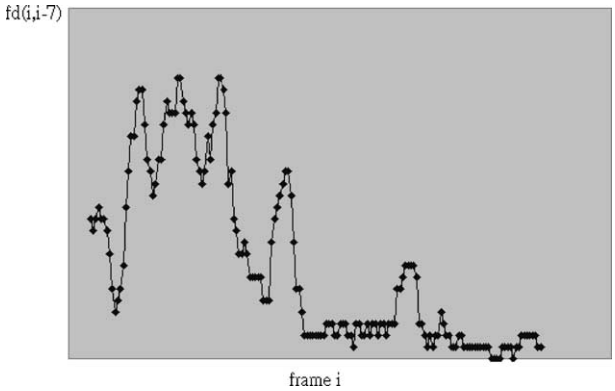

Fig. 6. Frames difference of a video sequence with significant camera movement.

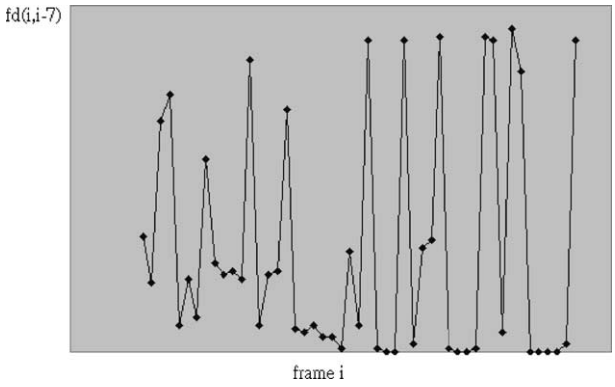

Fig. 7. Frames difference of a video sequence with sharp lighting change.

the final form of preview sequence. It is also noted that the selected video clips should be concatenated in such a way that the temporal order is preserved.

3. Experimental results

The proposed approach has been implemented on a Pentium III computer. Fig. 8 is the user interface of our system, where

- Area "1" shows the frames difference of test video and the detected shot boundary is indicated by the red line.
- Area "2" shows RGB histogram of the current frame which is being processed.
- Area "3" lists the detailed information of each shot, including: the starting, ending and key-frame positions, belonging group and a flag indicates whether the shot is selected to form the preview sequence.
- Area "4" shows the setting of all threshold values.

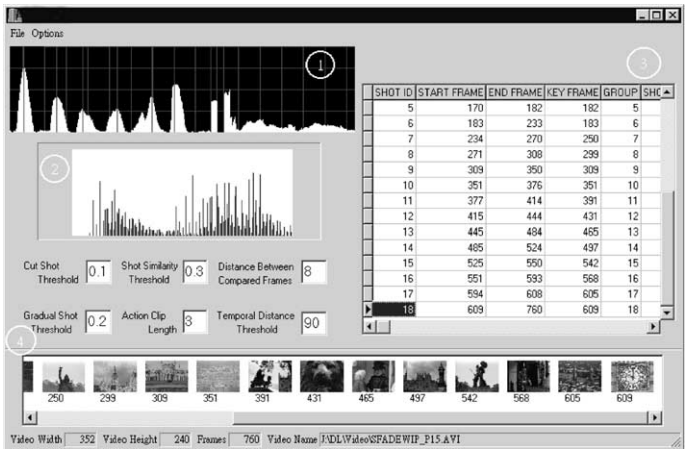

Fig. 8. User interface of video preview.

Table 1
Results of video abstraction

Original video	Original video length	Number of shots	Abstracting video length	Number of shots	Compression ratio
Forrest Gump	2 h 22 min	1006	8 min 52 s	92	16:1
The Rock	2 h 17 min	3541	6 min 44 s	75	20:1
Major League	1 h 47 min	1350	8 min 55 s	158	12:1

Meanwhile, the extracted key frames are shown at the bottom. Three movies are used in our experiment: ''Forrest Gump,'' ''The Rock,'' and ''Major League.'' The experimental results are shown in Table 1.

The performance of the proposed technique is evaluated in two aspects: shot/scene boundary detection and video abstraction. To measure the correctness of shot/scene boundary detection, two metrics are used:

$$\text{Recall} = \frac{D}{D + MD}, \quad \text{Precision} = \frac{D}{D + FD},$$

where D is the number of shot/scene boundary detected correctly, MD is the number of missed detection and FD is the number of false detection. The test video set is obtained by randomly selecting a 10-min video clip from each movie. It also take an experienced person roughly 9 h to get the ground truth of test set. Experimental results show that the average recall and precision for our shot boundary detection algorithm are 92 and 80%, respectively. According to the benchmark reported in Gargi et al. (2000), such metric values are sufficient to capture the basic video structure. However, the average recall and precision for scene boundary detection (or shots grouping) algorithm are 84 and 70%, respectively. Because scene is a group of shots that are semantically correlated and is a subject concept to reflect human

perception, different human subject has different rules to extract scenes. Therefore, the experimental results are reasonable and acceptable.

As to video abstraction, a direct and mathematically precise quality measure does not exist. Therefore, we decide to measure the quality of our abstracts by user questioning. After viewing the abstracting video, 10 test person have to answer the following questions on a scale of 1–5, corresponding to strong disagreement or strong agreement, respectively:

(1) Does the video abstract give you a quick overview of the underlying contents?
(2) Is the video abstract of good quality?
(3) Is the video abstract interesting?

The average rating for the three questions are 4.75, 4.12, and 4.58, respectively. The rating for question (2) is a little bit low, because we do not concatenate the selected video clips with editing technique such as dissolves or wipes. However, the overall opinions of test people is encouraging.

4. Conclusion

We have presented a method for automatically generating preview sequence of digital movies. Our approach is based on the analysis of high-level video structure. To derive such high-level video structure representation, the input video is firstly decomposed into a number of basic components called shots. The proposed shot change detection algorithm is able to detect both the abrupt and gradual transition boundary. Then, shots are grouped into semantic-related scenes by taking into account the visual characteristics and temporal dynamics of video. Although the selection of intense-interaction and action scenes as abstract is a heuristic, experimental results show that this heuristic generally captures well the relative importance of video content. Compared with the related works (Smith and Kanade, 1997; Toklu et al., 2000) at conceptual level, our approach is simple yet effective. As the production of movie trailer is a very subjective process, it is highly difficult to automatically generate a preview sequence that is very similar to the manually produced trailer. However, there is still hope in determining an empirical basis for video abstraction. The proposed approach takes a stride toward this difficult problem.

References

Christel, M.G., Smith, M.A., Taylor, C.R., Winkler, D.B., 1998. Evolving video skims into useful multimedia abstractions. In: ACM Conference on Human Factors in Computing Systems. Los Angeles, CA, April 1998, pp. 171–178.

DeMenthon, D., Kobla, V., Doermann, D., 1998. Video summarization by curve simplification. In: ACM International Conference on Multimedia, pp. 212–218.

Gargi, U., Kasturi, R., Strayer, S.H., 2000. Performance characterization of video-shot-change detection methods. IEEE Transactions on Circuits and Systems for Video Technology 10 (1), 1–13.

Gong, Y., Liu, X., 2000. Generating optimal video summaries. In: International Conference on Multimedia and Expo, New York, pp. 1559–1562.

Hanjalic, A., Zhang, H.J., 1999a. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. IEEE Transactions on Circuits and Systems for Video Technology 9 (8), 1280–1289.

Hanjalic, A., Zhang, H.J., 1999b. Optimal shot boundary detection based on robust statistical models. In: IEEE International Conference on Multimedia Computing and Systems, pp. 710–714.

He, L., Sanocki, E., Gupta, A., Grudin, J., 1999. Auto-summarization of audio-video presentations. In: ACM International Conference on Multimedia, Orlando, FL, October, pp. 489–498.

Jain, A.K., Dubes, R.C., 1988. Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs, NJ.

Nam, J., Tewfik, A.H., 1999. Video abstraction of video. In: IEEE International Workshop on Mutimedia Signal Processing. Copenhagen, Denmark, September, pp. 117–122.

Pfeiffer, S., Lienhart, R., Fischer, S., Effelsberg, W., 1996. Abstracting digital movies automatically. Journal of Visual Communication and Image Representation 7 (4), 345–353.

Smith, M.A., Kanade, T., 1997. Video skimming and characterization through the combination of image and language understanding techniques. In: IEEE Conference on Computer Vision and Pattern Recognition. St. Thomas, US Virgin Islands, June 1997, pp. 775–781.

Swanberg, D., Shu, C.F., Jain, R., 1993. Knowledge guided parsing and retrieval in video database. In: SPIE Conference on Storage and Retrieval for Image and Video Databases, February, pp. 173–187.

Toklu, C., Liou, S.P., Das, M., 2000. Videoabstract: a hybrid approach to generate semantically meaningful video summaries. In: International Conference on Multimedia and Expo., New York, pp. 1333–1336.

Uchihashi, S., Foote, J., Girgensohn, A., Boreczky, J., 1999. Video manga: generating semantically meaningful video summaries. In: ACM International Conference on Multimedia, October, pp. 383–392.

Yeo, B.L., Liu, B., 1995. Rapid scene analysis on compressed videos. IEEE Transactions on Circuits and Systems for Video Technology 5 (6), 533–544.

Yeung, M.M., Yeo, B.L., 1997. Video visualization for compact presentation and fast browsing of pictorial content. IEEE Transactions on Circuits and Systems for Video Technology 7 (5), 771–785.

Zabin, R., Miller, J., Mai, K., 1995. A feature-based algorithm for detecting and classifying scene breaks. In: ACM International Conference on Multimedia, November, pp. 189–200.

Zhang, H.J., Kankanhalli, A., Smoliar, S.W., 1993. Automatic partitioning of full-motion video. Multimedia System 1 (1), 10–28.

Zhang, H.J., Wu, J., Zhong, D., Smoliar, S.W., 1997. An integrated system for content based video retrieval and browsing. Pattern Recogn. 30 (4), 643–658.