Video Shot Boundary Detection Using Frame-Skipping Technique

Yue Gao, Jun-Hai Yong *
Institute of CG & CAD
School of Software, Tsinghua University

Fuhua (Frank) Cheng †
Graphics & Geometric Modeling Lab
Department of Computer Science, University of Kentucky

Abstract

Multimedia techniques have been used in many different applications over the past decade, sometime unmaturely. Demand for more efficient Efficient methods for video indexing and retrieval are required to confront this situation. Video temporal segmentation, aiming at finding the location of shot boundaries, including abrupt shot boundaries and gradual shot boundaries, is the first and essential step in visual data processing. In this paper we propose an algorithm for video shot boundary detection using frame-skipping technique. In our algorithm only some typical frames are selected for initial processing, which can be obtained by Frame-Skipped Module (FSM). As a visual description, we may choose 1 or 2 frames from 10 consecutive ones. To diminish the error caused by Frame-Skipped technique, we use Abrupt-Boundary Module (*ABM*) and Gradual-Boundary Module (*GBM*) to detect video shots, where the *ABM* based on global histogram comparison is used to detect abrupt shot boundaries and the *GBM* based on local histogram comparison concentrates on detecting gradual transitions. This new algorithm can reduce about 40%$\sim$70% time-consumption compared with some previous algorithms on 4 typical video clips (movie, cartoon, sports and news).

Keywords: shot boundary, Frame-Skipped, time-economizing

1 Introduction

With the advancement of video technology, multimedia techniques have found applications in nearly every aspect of our life, such as national security, broadcasting, communication, entertainment, library science, ... etc. The mushroom growth of video information, consequently, necessitates the progress of content-based video indexing and retrieval techniques. Video temporal shot boundary detection is the first and, actually, a crucial step towards automatic processing of video sequences. The aim of shot detection is to find the position of shot boundaries which can be divided into *abrupt shot boundaries* and *gradual shot boundaries*, so that key frames can be selected from each shot for subsequent processing (indexing and retrieval). Video analysis is usually based on feature comparison between such key frames.

A shot is a sequence of interrelated consecutive frames taken contiguously by a single camera which represents a continuous action in time and space [Tsai and Chen 2005] or, simply, an unbroken sequence of frames taken from one camera [Koprinska and Carrato 2001].

*e-mail: Yue.Gao.thu@gmail.com, yongjh@mail.tsinghua.edu.cn
†e-mail: cheng@cs.uky.edu

In the past decade, shot boundary detection has attracted a considerable amount of research attention, and plenty algorithms have been proposed to analyze this problem. Most of them depend on comparison between consecutive frames [Cotsaces et al. 2006]. They extract certain special features from each frame and define distance functions on the feature domain, then compare dissimilarity between adjacent images. Items that have been used as features include color histogram, luminance histogram and image edges [Kikukawa and Kawafuchi 1992][Rasheed and Shah 2005][Zabih et al. 1999].

In this paper, we present a unified algorithm for video shot boundary detection using *frame-skipping technique*. In our approach, not all frames of the video have to be calculated, only some typical ones are selected by the frame-skipping model for initial processing. As a visual description, we may choose 1 or 2 frames from 10 consecutive frames as the initial test data. To diminish inaccuracy caused by the frame-skipping technique, we use abrupt-boundary model (*ABM*) and gradual-boundary model (*GBM*) to detect video shots. *ABM*, based on global histogram comparison, is used to detect abrupt shot boundaries and *GBM*, based on local histogram comparison, focuses on detecting gradual transitions. This algorithm can achieve 40%$\sim$70% improvement on processing time than some previous algorithms on 4 typical video clips (movie, cartoon, sports and news).

The remainder of this paper is organized as follows. Section 2 provides a brief review of existing algorithms on content-based video shot boundary detection. The description of our approach for video temporal segmentation and its computational complexity is given in section 3. Experiment results and their analysis are shown in section 4. Conclusions and comments on future directions are given in section 5.

2 Previous Works

Shot boundaries in general can be divided into two categories: *abrupt shot boundaries* (ABs) and *gradual shot boundaries* (GBs). Abrupt shot boundaries have simpler form. An abrupt shot boundary (also called *cut*) is caused by the stopping or restarting of the camera. Such action usually generates great change in one single frame. Gradual transitions, on the other hand, occur in several consecutive frames. Boundaries of these transitions, consequently, can be relatively difficult to detect even though they are less frequent than cuts. Gradual shot boundaries have three distinctive forms: *fades*, *dissolves* and *wipes*. A fade is a transition with gradual diminishing (fade out) or heightening (fade in) of visual intensity [Z.Cernekova et al. 2002]. A dissolve occurs when superimposing images of one video on images of another video with the underneath frames getting dimmer and those on top becoming brighter [Koprinska and Carrato 2001]. A wipe is actually a set of shot change techniques, where the appearing and disappearing shots coexist in different spatial regions of the intermediate video frames, and the region occupied by the former grows until it entirely replaces the latter [Cotsaces et al. 2006].

As the first step of visual data processing, shot boundary detection has become an essential research area in the application of video indexing and retrieval. Most of the early algorithms focus on abrupt shot boundary detection. But detection of boundaries of gradual transitions have attracted more attention recently.

Template matching, the simplest method of video segmentation, is to evaluate the difference in intensity or color value between two consecutive frames in a pixelwise manner [Kikukawa and Kawafuchi 1992]. The main disadvantage of this approach is that it is sensitive to object and camera movements and can not distinguish a large change in a small block of the scene.

Contrary to pixelwise comparison, a method that is based on global image characteristic to perform video segmentation and, consequently, diminishes the sensitivity to object and camera movements, is to compare gray histograms between successive images [Zhang et al. 1993]. A further approach to shot boundary detection using color feature is based on local histogram comparison [Swanberg et al. 1993]. Each image is divided into k^2 blocks for some positive integer $k > 1$ so that relationship between these blocks and their counterparts in the consecutive frame can be analyzed. Shot boundaries are detected by studying the statistics of this block information. This approach is robust to smaller objects and object/camera movements for video temporal segmentation, some advanced algorithms with sliding window [Tsai and Chen 2005] and edge detection [Rasheed and Shah 2005] have been proposed recently. However, those algorithms still can not handle scenes which are very dark or very bright in the whole image.

As video segmentation methods using intensity histogram are sensitive to dramatic change of the overall frame luminance (called flashlight), edge-based detection [Zabih et al. 1999][Heng and Ngan 1999] was introduced to estimate similarity between consecutive frames. When a shot boundary occurs, the old intensity edges make some distance from the new ones, which can decrease the sensitivity of acute brightness changes effectively. The main drawback of this method is that it may cause false positives in frames with the rapid changes in the overall shot luminous and those of darkness or brightness. This algorithm also requires extensive computation because such edge change detection is rather complicated than computation of histogram.

Another interesting issue is how to eliminate inaccuracy caused by noise signals. Text edge detection [Zhong et al. 2005] can obliterate the influence of superimposed video text, which is primarily based on connected monochrome color regions of a certain size. A further improvement [Liang et al. 2004][Liang et al. 2005] is achieved by effacing the effect of those texts embedded in sophisticated background scenes with diverse colors.

3 Shot detection with Frame-Skipping Technique(FST)

The overall architecture of the proposed algorithm is shown in Figure1. This approach is aimed at providing an effective but precise shot boundary detection and is composed of 4 modules: Frame-Skipping Module (FSM), Change-Detecting Module (CDM), Abrupt-Boundary Module (ABM) and Gradual-Boundary Module (GBM). The input video data will first be processed by FSM to provide a new visual sequence for the subsequent module. This new visual sequence is then divided into 3 classes by CDM to be processed by ABM or GBM, or abandoned due to certain attributes. In this section, we analyze in detail the proposed method using FST to detect shot boundaries of visual sequence.

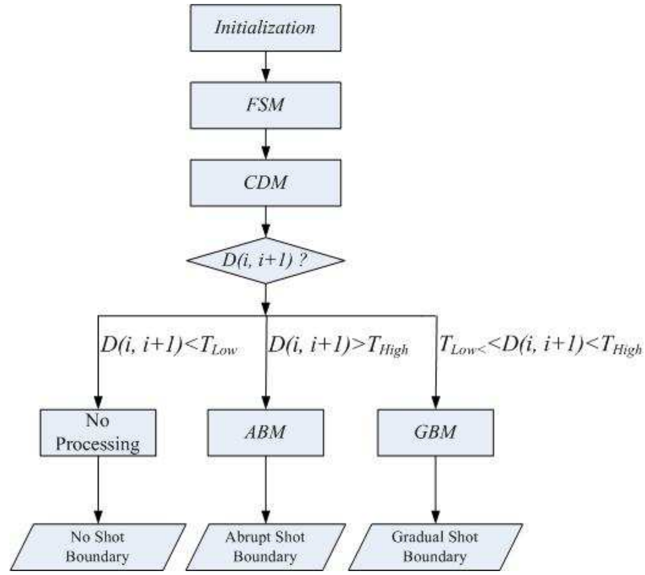

Figure 1: Framework of Shot Detection with FST.

3.1 Frame-Skipping Module

The Frame-Skipping Module (FSM) takes an original video as input and output some selected frames for further processing using Frame-Skipping Technique (FST). This module also offers detailed information on the selected video clip. The reasoning and selection process are illustrated below.

First note that multimedia always contains extensive information. For example, a five-minute video shot may contain over 10,000 frames. Hence, using the traditional, exhaustive approach to contrast dissimilarity of frames, i.e., comparing consecutive frames one pair at a time, would not be a good choice at all. While two consecutive frames in the same shot are usually too similar to distinguish one from the other, two successive images belonging to different shots may contain relatively large variation. As shown in Figure 2, the first five frames are consecutive frames from one shot and the last four images are consecutive frames from another shot. Frames in each group are quite similar to each other, but each frame in the first group is significantly different from any frame from the second group. Therefore, we have the following obvious conclusion: any frame randomly taken from the first shot would be dissimilar enough to any frame taken from the second shoe.

Figure 2: Two Continuous Shots.

An abrupt shot transition occurs between two adjacent frames while gradual transitions such as fades, wipes and dissolves may spread over several frames (10 or even more) [Boccignone et al. 2005].

The number of video frames played per second (FPS) is 24 or 30. So if 1 or 2 frames are selected from 5 or 10 consecutive images for further detection, we would significantly reduce the volume of data to be tested. The ratio of data reduction is 90% if we choose one frame in every 10 frames, and 80% if 1 in every 5. The simplest way in doing the selection is to use a constant step size, as shown in Figure 3. Once a step size, called a Key of FST (KoF), is set, simply output a frame every other KoF frames. The output is a new sequence with a length $1/KoF$ of the initial data waiting for further processing.

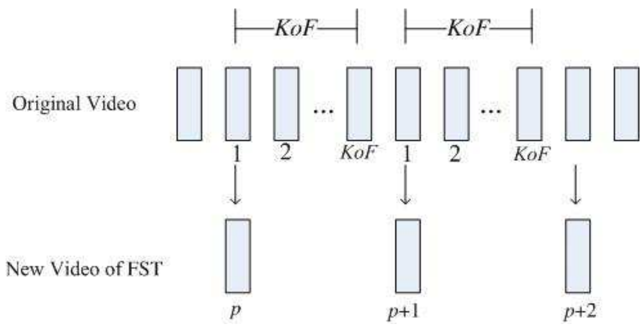

Figure 3: Select frames with FSM.

3.2 Change-Detecting Module

The purpose of the Change-Detecting Module (CDM) is to classify frames of the the input video sequence into 3 categories: C_{pass}, $C_{gradual}$ or C_{abrupt}. C_{pass} indicates this frame has no possibility to be a shot boundary and consequently can be abandoned for any further computing. C_{abrupt} means the frame has some possibility to be a cut and $C_{gradual}$ indicates the frame is likely to be part of a gradual transition and requires further processing. We use gray level histogram comparison as a way to analyze the output sequence of FSM. Within a shot, the variation of global histogram is small between two adjacent frames. The sum of gray histogram differences $D(i,i+1)$ [Koprinska and Carrato 1998] is formulated as follows:

$$D(i,i+1) = \frac{\sum_{j=1}^{n} |H_i(j) - H_{i+1}(j)|}{n} \quad (1)$$

Where $H_i(j)$ represents the value of histogram for gray level j in frame i, j is the gray value and n is the sum of gray levels. The value range of $D(i,i+1)$ is between 0 to 1. We also need to set 2 thresholds T_{High} and T_{Low} as testing metrics. The principle of classification is described as follows:

(a) If $D(i,i+1) < T_{Low}$, these two frames are similar enough to be considered consecutive and this frame is tagged C_{pass};

(b) If $D(i,i+1) > T_{High}$, the difference between the two frames is relative large and it is possible to be a abrupt shot boundary, so this frame is marked C_{abrupt} and should wait for further processing;

(c) If $T_{Low} < D(i,i+1) < T_{High}$, the difference is not large enough to be a cut, and not small enough to be a consecutive video sequence either, and the image is signed as $C_{gradual}$.

T_{High} and T_{Low}, as well as T and T_r used in ABM and GBM (to be defined shortly) are all obtained from extensive test results on real abrupt and gradual shot boundary data.

The frames labeled C_{pass} will simply be abandoned, the ones tagged C_{abrupt} will go to ABM for abrupt detection and those marked with $C_{gradual}$ will be sent to GBM for gradual transition detection.

3.3 Abrupt-Boundary Module

The Abrupt-Boundary Module (ABM) focuses on judging if abrupt boundaries occur on frames labeled C_{abrupt} and if they do, where they are. Here block-based histogram is used as feather, which is robust to object and camera movements concluding the spatial information. This algorithm first introduced in [Nagasaka1995] has been ameliorated to identify the feather. When a C_{abrupt} found in the P^{th} frame, we break each frame ($P^{th} \sim (P+KoF)^{th}$) into 16 equal size regions, and $D(i,i+1)$ is measured by comparing the histograms difference using the following equation:

$$D(i,i+1) = \frac{\sum_{k=1}^{r} DB(i,i+1,k) - max(DB(i,i+1,k))}{r-1} \quad (2)$$

$$DB(i,i+1,k) = \frac{\sum_{j=1}^{n} |H_{i,k}(j) - H_{i+1,k}(j)|}{n} \quad (3)$$

Where r is the total number of the blocks, i is from P to $(P+KoF)$. is the histogram value at gray level j for block k and n is the quantity of gray levels; $max(DB(i,i+1,k))$ is the maximum of $DB(i,i+1,k)$.The purpose of regarding the $max(DB(i,i+1,k))$ is to alleviate the influence of noise signal and effect of object and camera movements. We label the frames whose dissimilarities $D(i,i+1)$ exceeds a predefined threshold T as a cut. A large $D(i,i+1)$ implies a remarkable content change in the frame, so an abrupt shot boundary is assumed in the labeled frame pairs.

3.4 Gradual-Boundary Module

This Gradual-Boundary Module (GBM) provides a method to do effective detection of gradual transitions in those frames labeled $C_{gradual}$ in CDM. The block-based histogram mentioned in ABM is applied as testing feather and the definition of $D(i,i+1)$ is the same as that in ABM, while we choose different judgment. Gradual transitions normally occur during a series of successive frames with relative small variance between two adjacent images. So we set a diminutive threshold T_r and also use the threshold T claimed above. We monitor the variances change to detect the boundaries of gradual shot changes. When the variance decreases or increases consecutively over several frames, these frames may be possible to gradual transitions as GB candidates. As shown in [Fang et al. 2006], the duration of most gradual shot boundaries is more than one second. We propose two new criterions here:

(1) L_c: the length of these candidates;

(2) C_e: the entire changing of candidates.

If these candidates are from m^{th} frame to $(m+L_c)^{th}$ frame, we can get C_e by this equation: $C_e = D(m,m+L_c)$.

The detection principle could be described as follows:

If $(L_c > 10)$ and $(C_e > T)$, we declare a gradual transition here.

4 Experiment results

To evaluate the proposed temporal segmentation algorithm, we have tested this method on 4 typical video clips: *movie*, *cartoon*, *sports* and *news*. These video clips are segmented into shots using FST with the Key of FST (KoF) set to $2, 3, ..., 15$ and 20. To obtain a reference for comparison, we also detect shot boundaries using regular method without FST. The performance of the FST algorithm presented in section 3 is assessed in terms of *precision rate*, *recall rate*, and especially *processing time*. These terms are defined as follows:

$$\begin{aligned} \textit{Precision rate} &= N_c/N_d \times 100\% \\ \textit{Recall rate} &= N_c/N_t \times 100\% \\ \textit{Processing Time} &= \textit{CPU time taken to process} \\ &\quad\ \textit{shot detection} \end{aligned} \tag{4}$$

where N_c is the number of correctly detected shots, N_d is the number of total shots detected and N_t is the actual total number of shots. We also define T_S as the time saved with FST. The experiment environment is described in Table I.

Table I: Experimental Environment

Processor	Dothan 1.86GHz
Memory	512M DDR533 RAM
Operating System	Microsoft Windows XP Professional
DirectX Version	DirectX 9.0c
AGP	ATI Radedon X300, 64M, 400MHz

4.1 Shot detection using FST on movie video

This visual clip is taken from the celebrated movie *Star War Episode 6: Return of the Jedi*. The length of this video sequence is 9 minutes 22 seconds with 86 shots and a frame size of 500×210 in pixels. The number of total frames is 13492 and fps is 24. Table II summarizes the performance of shot detection on this movie clip using FST method. The results show that the precision rates are approximatically the same for all the operating conditions, and improvement on processing time is maximized when *KoF* is between 8 and 10.

Table II: Detection Result on Movie Video by FST

KoF	$R_T(s)$	$T_S(\%)$	N_t	N_c	Recall(%)	Precision(%)
1(no *KoF*)	501	0	86	84	100	97.67
2	287	42.71	86	84	100	97.67
3	216	56.89	86	84	100	97.67
4	180	64.07	86	84	100	97.67
5	159	68.26	86	84	100	97.67
6	151	69.86	86	84	100	97.67
7	141	71.76	86	84	100	97.67
8	138	72.46	86	84	100	97.67
9	133	73.45	86	84	100	97.67
10	133	73.45	86	84	100	97.67
11	138	72.46	86	84	100	97.67
12	135	73.09	86	84	100	97.67
13	139	72.26	86	84	100	97.67
14	143	71.46	86	84	100	97.67
15	146	70.86	86	84	98.84	97.65
20	162	67.66	86	84	97.67	98.81

4.2 Shot detection using FST on cartoon video

The selected cartoon clip is one of the Oscar optional shorts with 77 shots in it. The length is 5 minutes and 41 seconds with a frame size of 568×272 in pixels. The number of total frames is 8195 and fps is 24. Table III summaries the shot boundary detection results on this cartoon clip with the FST method. The results again show that the precision rates are approximatically the same for all the operating conditions, and improvement on processing time is maximized when *KoF* is between 8 and 10.

Table III: Detection Result on Cartoon Video by FST

KoF	$R_T(s)$	$T_S(\%)$	N_t	N_c	Recall(%)	Precision(%)
1(no *KoF*)	437	0	77	74	98.70	97.37
2	274	37.30	77	74	98.70	97.37
3	215	50.80	77	73	97.40	97.33
4	194	55.61	77	74	98.70	97.37
5	181	58.58	77	74	98.70	98.67
6	175	59.95	77	72	93.51	98.63
7	175	59.95	77	73	97.40	97.33
8	172	60.64	77	73	97.40	97.33
9	175	59.95	77	74	98.70	97.37
10	181	58.58	77	74	97.40	98.67
11	178	59.27	77	73	97.40	97.33
12	177	59.50	77	72	93.51	98.63
13	183	58.12	77	73	97.40	97.33
14	186	57.44	77	72	93.51	97.30
15	190	56.52	77	72	93.51	98.63
20	213	51.26	77	74	97.40	98.65

4.3 Shot detection using FST on sports video

This video is taken from a sports program about the reputed soccer player *Zidane*. The clip lasts 9 minutes 33 seconds with 138 shots. The number of frames is 17205 and the frame size is 400×294 in pixels with a fps of 24. The results of video segmentation are represented in Table IV. From the chart, we find that when the *KoF* is chosen between 8 and 10, we get rather faster speed of computation with better accuracy.

Table IV: Detection Result on Sports Video by FST

KoF	$R_T(s)$	$T_S(\%)$	N_t	N_c	Recall(%)	Precision(%)
1(no *KoF*)	726	0	138	126	91.30	92.65
2	533	26.58	138	127	92.03	94.07
3	432	40.50	138	127	92.03	94.07
4	382	47.38	138	127	92.03	94.78
5	354	51.24	138	127	92.03	94.07
6	333	54.13	138	126	91.30	94.03
7	335	53.86	138	125	90.58	93.98
8	331	54.41	138	125	90.58	93.98
9	331	54.41	138	126	91.30	94.03
10	325	55.23	138	125	90.58	93.98
11	362	50.14	138	127	92.03	94.07
12	356	50.96	138	125	90.58	93.98
13	371	48.90	138	125	90.58	93.98
14	380	47.66	138	124	89.56	93.94
15	391	46.14	138	127	92.03	94.07
20	415	42.84	138	125	90.58	93.98

4.4 Shot detection using FST on news video

A news clip is selected as the 4^{th} test case, which was a section of English news broadcast on $CCTV-9$(*China Central Television International*). The length is 9 minutes 13 seconds with 77 shots and 11886 frames in it. The video is of size 400×292 in pixels and fps is 24. Table V shows the shot detection performance of this news video sequence with FST method. Again, we achieve minimal computation cost with a *KoF* between 8 and 10.

Table V: Detection Result on News Video by FST

KoF	$R_T(s)$	$T_S(\%)$	N_t	N_c	Recall(%)	Precision(%)
1(no *KoF*)	432	0	77	64	83.11	95.52
2	245	43.29	77	64	83.11	95.52
3	180	58.33	77	64	83.11	95.52
4	153	64.58	77	62	80.52	95.38
5	136	68.52	77	64	83.11	95.52
6	132	69.44	77	64	83.11	95.52
7	123	71.53	77	61	79.22	95.32
8	119	72.45	77	63	81.83	95.45
9	118	72.69	77	63	81.83	95.45
10	116	73.15	77	61	79.22	95.32
11	117	72.93	77	63	81.83	95.45
12	120	72.22	77	64	83.11	95.52
13	120	72.22	77	61	79.22	95.32
14	119	72.45	77	62	80.52	95.38
15	123	71.53	77	62	80.52	95.38
20	132	69.44	77	62	80.52	95.38

4.5 Discussion

To get an evaluation strategy to assess the synthetical performance of shot boundary detection using FST, we define a new parameter P (shorthand for Performance) to do comprehensive analysis, which considers both precision and processing time factors. The definition of P is shown below:

$$P = T_S \times F1 \tag{5}$$

$$F1 = 2 \times Recall \times Precision / (Recall + Precision) \tag{6}$$

As shown in Table VI, a larger P can be achieved when using a *KoF* between (8∼12) on movie, (7∼9) on cartoon, (6∼10) on sports and (8∼12) on news.

Table VI: Synthetical Performance of Shot Detection using FST

KoF	$P_{movie}(\%)$	$P_{Cartoon}(\%)$	$P_{Sports}(\%)$	$P_{News}(\%)$
1(no *KoF*)	0	0	0	0
2	42.21	36.57	24.73	38.48
3	56.22	49.46	37.68	51.85
4	63.31	54.51	44.25	56.39
5	67.46	57.81	47.67	60.69
6	69.04	57.55	50.15	61.72
7	70.91	58.37	49.69	61.89
8	71.61	59.04	50.19	63.84
9	72.58	58.77	50.41	64.05
10	72.58	57.43	50.95	63.29
11	71.61	57.71	46.65	64.26
12	72.23	57.12	47.01	64.19
13	71.41	56.58	45.11	62.49
14	70.62	54.78	43.70	63.27
15	69.61	54.26	42.93	62.46
20	66.47	50.25	39.52	60.64

Based on the results presented in Tables II∼VI, we come up with the following key points.

(1) Primarily, the proposed shot detection algorithm with FST can reach a 40%∼70% improvement rate conspicuously. In particular, we also achieved high accuracy (recall rate 94.81%∼100% and precision rate 97.37%∼98.81%) compared with those algorithms without FST on the tested video sequences. The improvement on processing time is especially significant on movie and news video categories with a ratio of nearly 70%.

(2) This method is robust for these 4 typical categories of visual sequences, indicating the proposed approach is effective and insensitive to noise and various video structures.

(3) Table VI establishes a relationship between the effect of shot detection and KoF. It can be seen that the best performance is achieved when *KoF* is between 8 and 10.

5 Conclusion and future work

We have proposed a new algorithm for video shot boundary detection using Frame-Skipped technique. The improvement of operation efficiency is achieved by selecting typical frames for processing. Both local histogram comparison and global histogram comparison have been applied into this checking system and we use a judging module to classify various kind of frames, which can improve the accuracy and efficiency significantly, especially when dealing with gradual transitions. This approach is robust for both abrupt shot boundaries and gradual shot boundaries with *ABM* and *GBM*. Experiment results show that this method can improve processing time by 40%∼70% with high precision of detection, and the method achieves better effectiveness with a *KoF* between 8 and 10.

To improve the performance of this algorithm on shot detection, a further study of *ABM* and *GBM* is necessary. Another research direction in the future is to apply the Frame-Skipping technique in video indexing, video retrieval and other related video applications.

Acknowledge

The research was supported by the National Science Foundation of China (60533070).

References

BOCCIGNONE, G., CHIANESE, A., MOSCATO, V., AND PICARIELLO, A. 2005. Foveated shot detection for video segmentation. *IEEE Transactions on Circuits and Systems for Video Technology 15*, 3, 365 – 377.

COTSACES, C., NIKOLAIDIS, N., AND PITAS, L. 2006. Video shot detection and condensed representation:a review. *IEEE Signal Processing Magazinen 23*, 3, 28 – 37.

FANG, H., YIN, Y., NORHASHIMAH, P., AND JIANG, J. M. 2006. A hibrid scheme for temporal video segmentation. In *Processing of Third IEEE International Workshop on Electronic Design, Test and Applications*.

HENG, W. J., AND NGAN, K. N. 1999. Post shot boundary detection technique: flashlight scene determination. In *Processing of SPIE Conference*, vol. 1, 447 – 450.

KIKUKAWA, T., AND KAWAFUCHI, S. 1992. Development of an automatic summary editing system for the audio-visual resources. *Transactions on Electronics and Information*, 204 – 212.

KOPRINSKA, I., AND CARRATO, S. 1998. Video segmentation of mpeg compressed data. In *Processing of IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, 243 – 246.

KOPRINSKA, I., AND CARRATO, S. 2001. Temporal video segmentation: a survey. *Signal Process.: Image Communication 16*, 5, 477 – 500.

LIANG, L. H., LIU, Y., XUE, X. Y., LU, H., AND TAN, Y. P. 2004. Improved shot boundary detection method based on text edges. In *Processing of Control, Automation, Robotics and Vision Conference*, vol. 1, 115 – 119.

LIANG, L. H., LIU, Y., LU, H., XUE, X. Y., AND TAN, Y. P. 2005. Enhanced shot boundary detection using video text information. *IEEE Transactions on Consumer Electronics 51*, 2, 580 – 588.

RASHEED, Z., AND SHAH, M. 2005. Detection and representation of scenes in videos. *IEEE Transactions on Multimedia 7*, 6, 1097 – 1105.

SWANBERG, D., SHU, C. F., AND JAIN, R. C. 1993. Knowledge guided parsing in video databases. In *Processing of SPIE Conference*, vol. 1908, 13 – 24.

TSAI, T. H., AND CHEN, Y. C. 2005. A robust shot change detection method for content-based retrieval. In *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 5, 4590 – 4593.

ZABIH, R., MILLER, J., AND MAI, K. 1999. A feature-based algorithm for detecting and classifying production effects. *Multimedia Systems 7*, 119 – 128.

Z.CERNEKOVA, NIKOU, C., AND PITAS, I. 2002. Shot detection in video sequences using entropy based metrics. In *Proceedings of International Conference on Image Processing*, vol. 3, 421 – 424.

ZHANG, H. J., KANKANHALLI, A., AND S-W.SMOLIAR. 1993. Automatic partitioning of full-motion video. *Multimedia Systems 1*, 1, 10 – 28.

ZHONG, Y., KARU, K., AND JAIN, A. K. 2005. Locating text in complex color images. *Pattern Recognition 5*, 1523 – 1535.