



ELSEVIER

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

Pattern Recognition Letters 24 (2003) 2209–2218

Pattern Recognition
Letters

www.elsevier.com/locate/patrec

A histogram-based moment-preserving clustering algorithm for video segmentation

Chi-Chun Lo ^{*}, Shuenn-Jyi Wang

Institute of Information Management, National Chiao-Tung University, 1001 Ta Hsueh Road, Hsinchu 30050, Taiwan

Received 12 July 2001; received in revised form 29 November 2002

Abstract

Video segmentation is the first step of creating video indices for a video retrieval system. A segmentation algorithm is used to identify shots from video data. In this paper, we propose a histogram-based moment-preserving (HBMP) clustering algorithm for segmenting video data. This algorithm is a hybrid of the shot change detection approach and the clustering approach. The computational results indicate that the proposed algorithm is both effective and efficient with respect to various types of video sequence.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Video segmentation; Clustering; Moment preserving

1. Introduction

With the advances in computer technologies, such as the increasing speed of CPU, the capacity of the storage device, and various compression methods, digital video is becoming more and more common in almost every aspect of our life, including education, entertainment, communications, etc. For the ever-increasing amount of video data, a systematic approach of retrieving video data is needed. A video retrieval system consists of two major subsystems for indexing and querying, respectively. In the indexing process, video seg-

mentation is used to segment video sequence into shots where each shot represents a sequence of frames having the same contents. Once shots are identified, key frames are extracted from each shot for indexing (Jain et al., 1999; Zhang and Lu, 2002). By using the indices, the query process provides a means of retrieving video data.

In order to find the right number of shots and select the optimal set of key frames from each shot, a video segmentation algorithm has to detect shot changes (SCs) correctly. There are two types of SC, abrupt and gradual. An abrupt SC resulting from editing cuts is usually easy to be detected. A gradual SC resulting from chromatic edits, spatial edits, or combined edits is in general hard to be detected (Idris and Panchanathan, 1997; Jiang et al., 1998; Lupatini et al., 1998). Existing video segmentation algorithms can be classified into two

^{*} Corresponding author. Tel.: +886-3-5731909; fax: +886-3-5723792.

E-mail address: ccllo@faculty.nctu.edu.tw (C.-C. Lo).

groups: the shot change detection (SCD) approach by which a threshold has to be pre-assigned, and the clustering approach with which a prior knowledge of the number of clusters is required. The major problem of SCD lies on the difficulty of specifying the correct threshold which affects the performance of SCD. As to the clustering approach, the right number of clusters is hard to be identified. Different clusters may lead to completely different results.

In this paper, we propose a histogram-based moment-preserving (HBMP) clustering algorithm for segmenting video data. This algorithm is a hybrid of the two approaches aforementioned, and is designed to overcome the drawbacks of both approaches. The HBMP clustering algorithm is composed of three phases: the feature extraction phases, the clustering phase, and the SC identification phase. In the first phase, differences between color histograms are extracted as features. In the second phase, the moment-preserving equations (Tsai, 1985) are used to group features into three clusters: the SC cluster, the suspected shot change (SSC) cluster, and the no shot change (NSC) cluster. In the last phase, the shot change frames (SCFs) are identified from the SC and the SSC, and then are used to segment video sequence into shots; finally, a key frame is selected from each shot. The computational results indicate that the proposed algorithm is both effective and efficient with respect to various types of video sequence.

In the following section, existing video segmentation algorithms are examined. The HBMP clustering algorithm is detailed in Section 3. In Section 4, the computational results are presented and analyzed. In the last section, we conclude this paper with possible research directions.

2. Literature review

A number of video segmentation algorithms have been reported in the literature (Sethi and Patel, 1995; Nagasaka and Tanaka, 1992; Zhang et al., 1993; Shahraray, 1995; Swanberg et al., 1993; Joshi et al., 1998; Gonsel et al., 1998). In general, these algorithms can be classified into two major groups: the SCD approach and the clustering approach.

2.1. Shot change detection

The SCD algorithm is based on a threshold. An inter-frame difference is obtained by measuring the differences between pixels, histograms, or blocks. If the inter-frame difference is greater than the pre-assigned threshold, a SC is declared.

The pixel-based algorithm (Sethi and Patel, 1995) compares the pixels of two frames across the same location. The pixel-based algorithm is sensitive to noise, object motion, or camera operation.

The intensity/color histogram of a gray/color frame f is a n -dimensional vector $\{H(f, k) | k = 1, 2, \dots, n\}$ where n is the number of levels/colors, and $H(f, k)$ the number of pixels of level/color k in frame f . To illustrate the difference between two frames across a cut, Nagasaka and Tanaka (1992) proposed the chi-square test to compare two histograms, $H(f_i, k)$ and $H(f_j, k)$. Zhang et al. (1993) suggested the so-called “twin-comparison” technique to detect the gradual SC. The histogram-based algorithm is sensitive to a local motion or noise.

In the block-based algorithm (Shahraray, 1995; Swanberg et al., 1993), each frame f_i is partitioned into a set of k blocks, called sub-frames. Rather than comparing frame i with frame j , every sub-frame of f_i is compared with the corresponding sub-frame of f_j . The difference between sub-frames can be measured by either the pixel-based or the histogram-based algorithm. Whenever the difference between a sub-frame of f_i and the corresponding one of f_j is greater than the pre-assigned threshold, it is marked as a changed sub-frame. A SC is declared whenever the number of the changed sub-frames is greater than a given lower bound. Usually, the block-based algorithm is less sensitive to a local motion or noise than the histogram-based algorithm.

2.2. Clustering

The clustering technique (Jain and Dubes, 1998) is used to organize data according to the pre-assigned criteria. The k -means clustering algorithm (Bezdek, 1981; Hanjalic and Zhang, 1999) and the fuzzy c -means clustering algorithm (Bezdek, 1981; Joshi et al., 1998) are two most noticeable clus-

tering algorithms. In the k -means clustering algorithm (Hanjalic and Zhang, 1999), a sample is assigned to one and only one cluster, so a clear partition is possible. As to the fuzzy c -means clustering algorithm, a sample is assigned a membership function for each cluster, so a fuzzy partition is made. The moment-preserving clustering algorithm (Appendix A) using an analytical approach is reported in (Tsai, 1985). In this approach, a sample is assigned to one and only one cluster according to the center of the cluster which is obtained by solving moment-preserving equations. When the number of clusters is 2, 3, or 4, this algorithm can find the center of each cluster in a linear time.

3. The HBMP clustering algorithm

The HBMP clustering algorithm is a hybrid of the SCD approach and the clustering approach. It is designed to be threshold-free and at the same time require little computing time. It first measures the histogram differences between frames, which are then used as inputs to the clustering algorithm. The number of clusters is 3 instead of 2, since a two-cluster approach (Gunsel et al., 1998) may erroneously put frames into wrong clusters while handling boundary conditions; i.e., those frames in which SC is difficult to be detected. The additional cluster suggested in the HBMP clustering algorithm contains all ambiguous SCFs. A heuristic is developed to resolve those ambiguities.

As illustrated in Fig. 1, the HBMP clustering algorithm is composed of three phases: the feature extraction phase, the clustering phase, and the SC identification phase.

3.1. Feature extraction phase

In this phase, each frame is compared with its previous frame using the color histogram differ-

ence (bin-to-bin) and the chi-square test (Nagasaki and Tanaka, 1992). Frame dissimilarities are extracted as features. We consider the red–green–blue (RGB) color coordinates, along with the YCbCr color space. In (Joshi et al., 1998), it has been shown that luminance and chrominance information contained in the YCbCr color space can be used for the SCD.

3.2. Clustering phase

In this phase, a moment-preserving clustering algorithm is used to group frame dissimilarities obtained in the feature extraction phase into three clusters: the SC cluster, the SSC cluster, and the NSC cluster. By solving the moment-preserving equations (A.1), the moment-preserving clustering algorithm derives centers z_0 , z_1 , and z_2 of the NSC cluster, the SSC cluster, and the SC cluster, respectively. A detailed description of the clustering algorithm is given as follows:

The moment-preserving clustering algorithm

// The inputs are frame dissimilarities $X(= (x_1, x_2, x_3, \dots, x_n))$.

// The outputs are the SS, SSC, and NSC clusters. /* $N = 3$ */

// X represents frame dissimilarities;

// m_i the i -th moment ($i = 0, 1, \dots, 5$); /* $2N - 1 = 5$ */

// z_0, z_1 , and z_2 the center of the NSC cluster, the SSC cluster, and the

// SC cluster, respectively.

1. For $i = 0$ to 5
2. Derive the i -th moment m_i of X using (A.2).
3. End
4. Find centers z_0, z_1 , and z_2 , where $z_0 < z_1 < z_2$.
5. Assign frame dissimilarities, x_i , to the NSC cluster, the SSC cluster, or the SC cluster according to its shortest distance from centers z_0, z_1 , and z_2 , respectively.

Since $z_0 < z_1 < z_2$, the SC cluster contains the SCFs which are easily identified; the SSC cluster contains all frames in which SCs are difficult to be determined; the NSC cluster contains frames that definitely are not the SCFs.

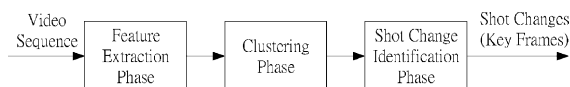


Fig. 1. The HBMP clustering algorithm.

3.3. Shot change identification phase

In this phase, the SCFs are first identified from the SC and the SSC, and are used to segment video sequence into shots. Then, the centroid frame of each shot is selected as the key frame. The SC identification algorithm is stated as follows:

The shot change identification algorithm

// The input values are the SC, the SSC, and a video sequence.

// The output values are **shots** (key frames)

1. Label all frames in SC as the SCFs.
2. Select possible SCFs from the SSC cluster using heuristic.
3. Segment the video sequence into shots according to the SCFs obtained in steps 1 and 2.
4. For each shot, select its centroid frame as the key frame.

In the second step, a heuristic is developed to resolve the uncertainty existing in the frames of the SSC cluster. As shown in Fig. 2, for every two consecutive frames in SC, $SC(i)$ and $SC(i+1)$, all SSC frames; namely, $SSC(k)$ $k = j, j+1, \dots, j+n-1$, between $SC(i)$ and $SC(i+1)$, are checked. An $SSC(k)$ is declared as a SCF if its histogram difference satisfies the following inequality:

$$H_SSC(k) \geq param * [0.5 * (H_SC(i) + H_SC(i+1))], \quad (1)$$

where $H_SSC(k)$ represents the histogram difference of $SSC(k)$; $H_SC(i)$, the histogram difference of $SC(i)$; $H_SC(i+1)$, the histogram difference of $SC(i+1)$; and $param$, the weight factor.

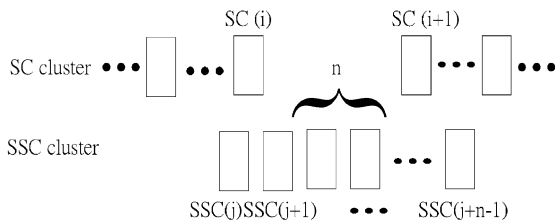


Fig. 2. SSC frames.

Furthermore, to reduce error detection due to a local motion or noise, we assume that the phenomenon of having two SCFs adjacent to each other is not possible. This assumption is based on the finding that two SCFs side-by-side usually occur due to video editing. In (1), we assign $param$ to be equal to 0.3. In fact, a fuzzy number instead of a constant could be used. From the computational results, we notice that this assignment is acceptable. Also, in (1), the constant 0.5 is used to calculate the average of $H_SC(i)$ and $H_SC(i+1)$.

4. Computational results and analyses

The computational experiments were done by using an IBM PC with the Intel Pentium III processor and 256 MB RAM. The MATLAB toolbox for image processing is used to develop the HBMP clustering algorithm. For comparison, Zhang's algorithm (Zhang et al., 1993) and Nagasaka's algorithm (Nagasaka and Tanaka, 1992) were simulated.

4.1. Performance metrics

Two performance metrics, the hit ratio (HR) and the fault ratio (FR), are used to evaluate the HBMP clustering algorithm. The HR and the FR are expressed as N_d/N_t and $(N_m + N_e)/N_t$, respectively, where N_d represents the number of the correct detections; N_m , the number of the missing detections; N_e , the number of the erroneous detections; and $N_t (= N_d + N_m)$, the total number of the SCFs in the video sequence being examined. A well-performed video segmentation algorithm should have a high hit ratio and at the same time a low fault ratio.

4.2. Assumptions

In the experiments, the following is assumed:

1. For a gradual shot change, dissolve, fade-in, or fade-out introduces only one SCF; pan or zoom does not produce any SCF.
2. Since an improper editing may cause several abrupt shot changes within two or three contig-

uous frames, it is assumed that the time interval between two abrupt shot changes covers at least two frames so as to eliminate the effect of an improper editing.

- The ground-truth shot frame is identified by manually examining the test sequence by five persons.
- Since frame difference obtained by using histogram interaction and the bin-to-bin histogram difference are the same (Ralph et al., 2000), experiments are conducted only in terms of the bin-to-bin color histogram difference and the chi-square color histogram difference.

4.3. Test cases

The performance of a video segmentation algorithm is sensitive to the shot change ratio (SCR), where the SCR is equal to the number of the SCFs divided by the total number of frames in the video sequence. Since human vision requires at least 30 frames per second; therefore, we consider that the SCR with one SCF for every thirty frames; i.e., the SCR is greater than or equal to 3.3% ($=1/30$), is high. We also consider that a video sequence without shot change for more than 10 s has a low SCR; i.e., the SCR is less than or equal to 0.33%. For completeness, different types of video sequence; e.g., animation, soap opera, movie, advertisement, and sport are considered. Three test cases are chosen as follows: In the first test case, 14 video sequences are selected from movie, animation, advertisement, and soap opera. The SCRs of these test sequences are 0.35%, 0.42%, 0.54%, 0.67%, 0.76%, 0.85%, 0.88%, 1.35%, 1.39%, 1.66%, 1.91%, 2.24%, 3.40%, and 3.89%, respectively. Action movie and advertisement usually have high SCRs. On the contrary, romantic movie and soap opera have low SCRs. Each test sequence contains a number of abrupt shot changes coupled with a few gradual shot changes (on the average of 2.6 out of all shot changes). In the second test case, nine video sequences are selected from animation only. The SCRs of these test sequences are 0.5%, 0.6%, 0.78%, 0.93%, 1.58%, 2.37%, 2.5%, 2.96%, and 3.15%, respectively. In the third test case, 10 video sequences are selected from soap opera only. The SCRs of these test sequences are 0.11%, 0.24%,

0.34%, 0.54%, 0.61%, 0.75%, 0.85%, 0.86%, 0.96% and 1.06%, respectively.

4.4. Results and analyses

4.4.1. Hit ratio and fault ratio

Fig. 3 compares the hit ratios of the HBMP clustering algorithm with those of Zhang's algorithm and Nagasaka's algorithm, for test case 1. As to the fault ratios, Fig. 4 presents the comparison between the HBMP clustering algorithm, Zhang's algorithm, and Nagasaka's algorithm, for test case 1. By examining Figs. 3 and 4, we notice that, for medium and high SCRs (≥ 2), the HBMP clustering algorithm has better performance than Zhang's and Nagasaka's algorithms; and the higher the shot change rate, the larger the difference. This fact indicates both Zhang's algorithm and Nagasaka's algorithm are sensitive to the threshold; but, the HBMP clustering algorithm is threshold-free. For low SCRs (≤ 0.75), we find that both the hit and fault ratios of Zhang's algorithm and Nagasaka's algorithm as well are close to those of the HBMP clustering algorithm, since the threshold can be easily determined.

By further examining Figs. 3 and 4, we find that the HBMP clustering algorithm using the bin-to-bin color histogram difference obtains the best hit ratio among all algorithms tested. The fault

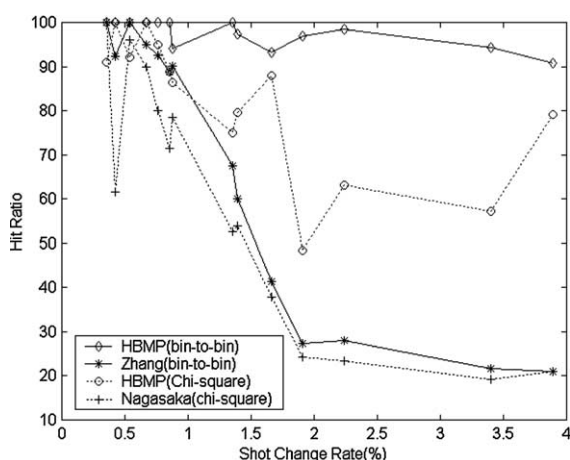


Fig. 3. Comparison between the HBMP clustering algorithm, Zhang's algorithm, and Nagasaka's algorithm with respect to the *hit ratio*, for test case 1.

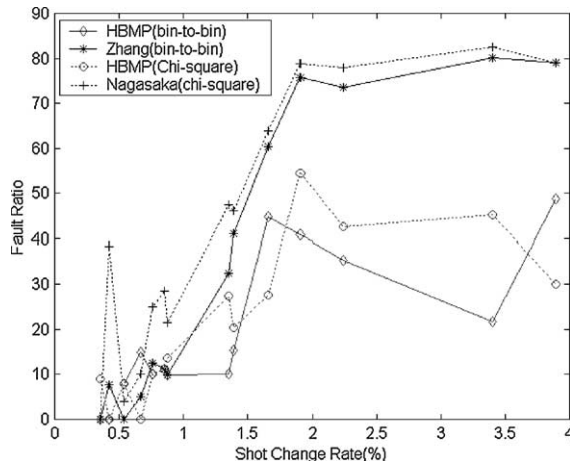


Fig. 4. Comparison between the HBMP clustering algorithm, Zhang's algorithm, and Nagasaka's algorithm with respect to the *fault ratio*, for test case 1.

ratios of the HBMP clustering algorithm using the bin-to-bin histogram difference are usually lower than those obtained using the chi-square histogram difference; however, there are circumstances where chi-square is better. The reason is that the measurements derived from the bin-to-bin histogram difference are much bigger than those derived from the chi-square histogram difference. This leads to the phenomenon that more ambiguous frames might be assigned to the SSC cluster when using the bin-to-bin histogram difference; consequently, increase the fault ratio.

Fig. 5 compares the hit ratios of the HBMP clustering algorithm with those of Zhang's algorithm and Nagasaka's algorithm, for test case 2. Fig. 6 presents the comparison between the fault ratios of the HBMP clustering algorithm and those of Zhang's algorithm and Nagasaka's algorithm, for test case 2. By examining Figs. 5 and 6, we have the following observations: for medium and high SCRs, the HBMP clustering algorithm with the bin-to-bin histogram difference obtains the best performance in terms of the hit ratio; for low SCRs, the performance of the HBMP clustering algorithm with the bin-to-bin histogram difference is close to that of Zhang's algorithm and Nagasaka's algorithm. The fault ratio of the HBMP clustering algorithm with either the bin-to-bin or the chi-square histogram difference is better than

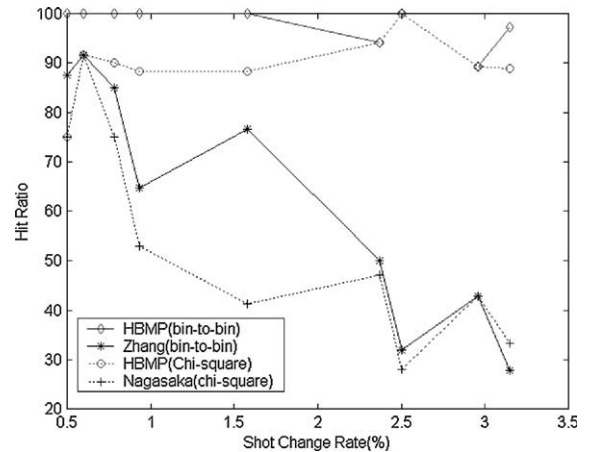


Fig. 5. Comparison between the HBMP clustering algorithm, Zhang's algorithm, and Nagasaka's algorithm with respect to the *hit ratio*, for test case 2.

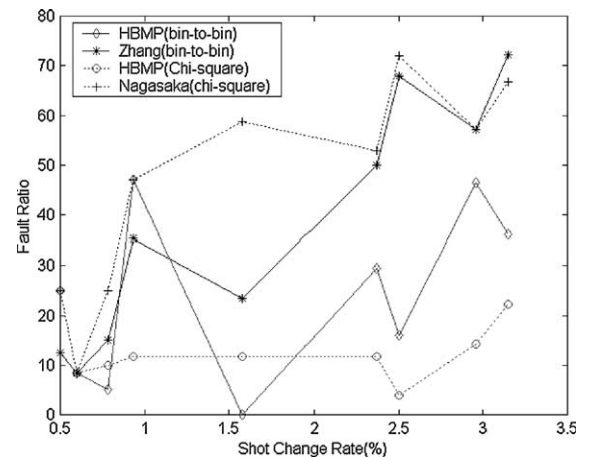


Fig. 6. Comparison between the HBMP clustering algorithm, Zhang's algorithm, and Nagasaka's algorithm with respect to the *fault ratio*, for test case 2.

that of Zhang's algorithm and Nagasaka's algorithm.

Fig. 7 compares the hit ratios of the HBMP clustering algorithm with those of Zhang's algorithm and Nagasaka's algorithm, for test case 3. Fig. 8 presents the comparison between the fault ratios of the HBMP clustering algorithm and those of Zhang's algorithm and Nagasaka's algorithm, for test case 3. By examining Figs. 7 and 8, we have the following observations: the HBMP clustering

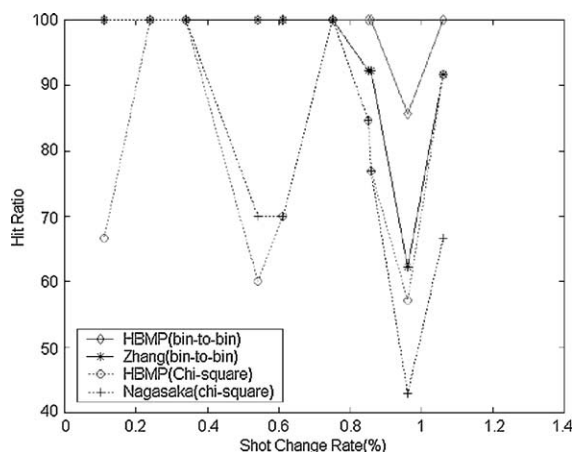


Fig. 7. Comparison between the HBMP clustering algorithm, Zhang’s algorithm, and Nagasaka’s algorithm with respect to the hit ratio, for test case 3.

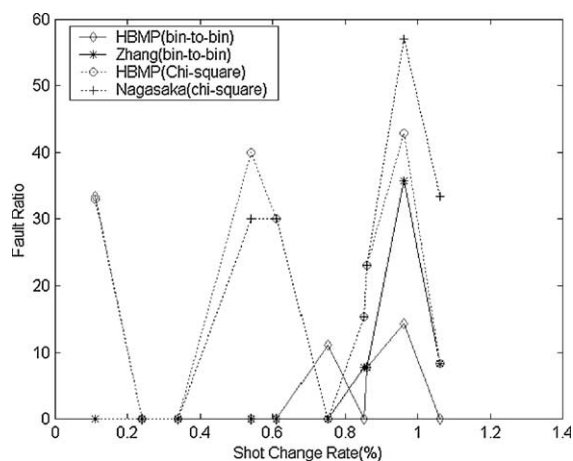


Fig. 8. Comparison between the HBMP clustering algorithm, Zhang’s algorithm, and Nagasaka’s algorithm with respect to the fault ratio, for test case 3.

algorithm with the bin-to-bin histogram difference obtains the best performance in terms of the hit ratio. In general, the HBMP clustering algorithm with the bib-to-bin histogram difference has a better fault ratio than both Zhang’s algorithm and Nagasaka’s algorithm, but the difference is small.

4.4.2. Computing time

In all test cases, the computing time of Zhang’s algorithm is in the order of *seconds* while that of the HBMP clustering algorithm is in the order of *minutes*. The HBMP clustering algorithm requires more computing time than Zhang’s algorithm due to the clustering and the selection of the SCFs from the SSC cluster. This is an inherited trade-off between the efficiency and the effectiveness. But, in the case of video segmentation, the effectiveness prevails. Table 1 compares the computing time of the HBMP clustering algorithm with that of Joshi’s algorithm (an iterative scheme). The computation time is measured in terms of the feature extraction time, the clustering time, and the shot change identification time. As shown in Table 1, the HBMP clustering algorithm is much faster than Joshi’s algorithm (an iterative scheme).

4.4.3. Discussions

4.4.3.1. *Validity and applicability of Eq. (1)*. Eq. (1) is used to identify the SCFs from the SSC. In essence, Eq. (1) is a histogram-based heuristic. Its validity and applicability need to be further analyzed. To achieve this purpose, we used the block-based algorithm (Shahraray, 1995) to resolve the ambiguity associated with the frames in SSC. Comparisons between the performance of Eq. (1)

Table 1
Comparison between the average the computing time of the HBMP clustering algorithm and that of Joshi’s algorithm

Number of frames in video sequence	Computing time (s)					
	Feature extraction time (A)		Clustering and shot change identification time (B)		Total time (A + B)	
	HBMP	Joshi	HBMP	Joshi	HBMP	Joshi
88	9.29	88.62	1.76	0.33	11.05	88.95
192	21.42	202.8	5.72	0.55	27.14	203.35
314	33.23	320.25	1.85	4.22	35.08	324.47
378	41.75	385.64	0.58	2.04	42.33	387.68
539	58.02	562.55	10.82	5.38	68.84	567.93

and that of the block-based algorithm are made with respect to a set of general video sequences and a set of special video sequences. As for the block-based algorithm, the block size is 8×8 , and the feature is obtained via the luminance, Y, of YCbCr. The computational results are shown in Tables 2 and 3.

By examining Table 2, we notice that Eq. (1) obtains the highest hit ratio and the lowest fault ratio among all SSC identification methods tested. Note that in Table 2, we also considered the two-cluster (without the SSC) approach for verifying the validity of introducing the SSC.

In Table 3, we tested the video sequences with lighting, smoking, objects moving across or near the lens of a camera, or many objects moving at the background. As for the video sequences with lighting and smoking, the block-based algorithm is better than Eq. (1) in terms of both the hit ratio and the fault ratio. However, as to the case of

many objects moving at the background, Eq. (1) is better than the block-based algorithm. Table 3 indicates that the block-based algorithm is suitable to the video sequences with noise.

In summary, we can say that Eq. (1) is good for the general video sequence and the block-based algorithm is a better choice for the video sequences with noise. However, a block-based algorithm usually requires more computing time than a histogram-based algorithm. Moreover, how to select the threshold of the block-based algorithm is a very difficult problem.

4.4.3.2. *The value of 'param' of Eq. (1).* Figs. 9 and 10 show that the assignment of 0.3 to *param* of (1) is reasonable. By examining Figs. 9 and 10, we have the following observations: for value 0.1, *param* has the highest (best) hit ratio; however, non-stable fault ratios; i.e., singular points appear at SCRs of values 0.67% and 1.66%; for value 0.3,

Table 2
Comparison between Eq. (1), the block-based algorithm, and no SSC with respect to the general video sequence

SCR	SSC identification method					
	Eq. (1)		Block-based algorithm		No SSC (only two clusters)	
	Hit ratio (%)	Fault ratio (%)	Hit ratio (%)	Fault ratio (%)	Hit ratio (%)	Fault ratio (%)
0.35	100	0.0	72.7	27.3	91.0	9.0
0.42	100	0.0	53.9	46.2	61.5	61.5
0.54	100	8.0	100	8.0	92.0	38.5
0.67	100	10.0	100	45.0	95.0	5.0
0.76	100	10.0	87.5	17.5	95.0	15.0
0.87	94.2	9.6	82.4	17.7	90.2	9.6
1.35	97.5	10.0	77.5	25.0	90.0	15.0
1.39	97.4	15.4	66.7	35.9	82.1	18.0
2.45	94.2	39.1	76.5	41.2	69.1	36.8
3.40	90.9	25.6	49.1	57.1	70.2	32.3

Table 3
Comparison between Eq. (1) and the block-based algorithm with respect to the special video sequence

Video sequences	SSC identification method			
	Eq. (1)		Block-based algorithm	
	Hit ratio (%)	Fault ratio (%)	Hit ratio (%)	Fault ratio (%)
4 Shots with lighting	100	75.0	100	0.0
3 Shots with smoking	66.7	66.7	100	33.3
2 Shots with objects moving across or near the lens of a camera	100	35.0	50.0	50.0
14 Shots with objects moving at the background	95.0	15.0	50.0	50.0

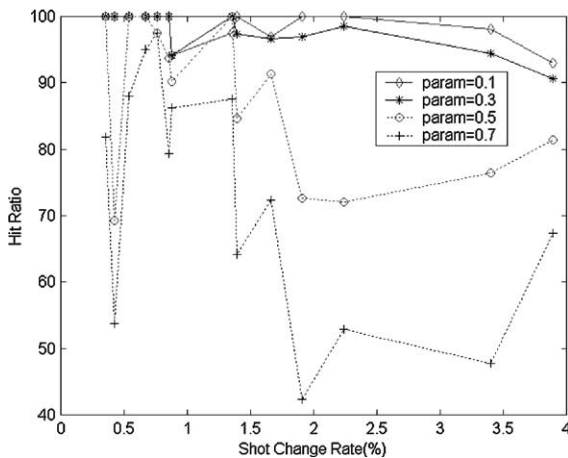


Fig. 9. The hit ratios with respect to *param* of assigned values 0.1, 0.3, 0.5, and 0.7, respectively.

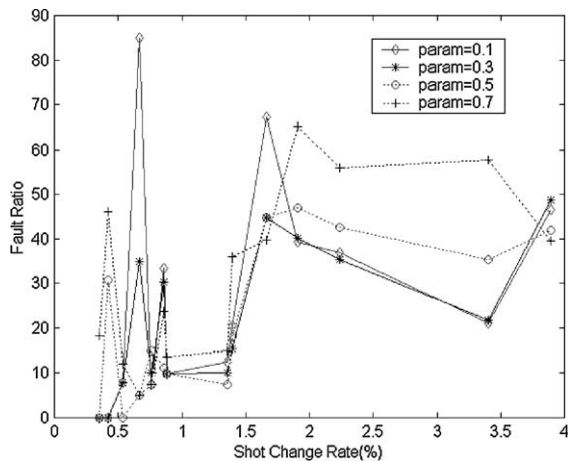


Fig. 10. The fault ratios with respect to *param* of assigned values 0.1, 0.3, 0.5, and 0.7, respectively.

param has the second highest hit ratio and the best fault ratio. Nevertheless, the second highest hit ratio is very close to the highest one; for values 0.5 and 0.7, *param* has the third best hit ratio and the worst hit ratio, respectively. As to the fault ratio, *param* performs the worst. Apparently, *param* of value 0.3 has the most stable performance with respect to the hit ratio and the fault ratio.

4.4.3.3. Gradual shot changes. For video sequences with many abrupt shot changes and a few gradual shot changes, the HBMP clustering algorithm works well with respect to both the hit ratio and the fault ratio. For completeness, we further analyze the performance of the HBMP clustering algorithm in terms of the video sequences having a number of gradual shot changes. Five types of video sequence are examined. Types 1–5 video sequences contain one gradual shot change, two gradual shot changes, one gradual shot change plus one abrupt shot change, two gradual shot changes plus one abrupt shot change, and one gradual shot change plus two abrupt shot changes, respectively. For each type, three test sequences are simulated. For comparison, the ground truth was manually obtained and Joshi's algorithm (Joshi et al., 1998) was simulated. Table 4 compares the average number of key frames obtained by the ground truth, the HBMP clustering algorithm, and Joshi's algorithm. By examining Table 4, we notice that the performance of the HBMP clustering algorithm is close to that of the ground truth, and is much better than that of Joshi's algorithm.

Table 4

Comparison between the number of the key frames obtained by the ground truth, the HBMP clustering algorithm, and Joshi's algorithm

Video type	Average number of key frames		
	Ground truth	HBMP algorithm	Joshi algorithm
1. (1 gradual)	2.00	2.00	2.00
2. (2 graduals)	3.00	4.67	N/A
3. (1 gradual + 1 abrupt)	3.00	3.00	3.00
4. (2 graduals + 1 abrupt)	4.00	4.00	N/A
5. (1 gradual + 2 abrupts)	4.00	3.67	5.67

N/A: not available (the algorithm failed).

5. Conclusions

In this paper, we proposed the HBMP clustering algorithm for identifying shots from video data. Some distinct properties of the proposed algorithm are: there is no need for finding a proper threshold as required by the SCD approach; its computing time is much lower than that of an iterative algorithm (the k -means or the fuzzy c -means algorithm) due to the moment-preserving clustering; and an exact solution (clustering) can be obtained, since there are no initial values are required in the moment-preserving equations.

Here we would like to mention the following areas of investigation which may merit further study.

- (1) Apply the HBMP clustering algorithm to compressed video sequences; e.g., MPEG 4 videos.
- (2) Develop a video indexing algorithm; subsequently, couple with the HBMP clustering algorithm, build a video retrieval system.
- (3) Use other information, such as spatial or temporal information, to improve the performance of the HBMP clustering algorithm.
- (4) Give a comprehensive study on using the block-based algorithm to reduce erroneous detections due to a local motion or noise.

Appendix A. Moment-preserving clustering

In order to group n data samples, $X(= (x_1, x_2, x_3, \dots, x_n))$ into N clusters, Tsai (1985) solves the first $2N$ moment-preserving equations as follows:

$$\begin{aligned} p_0 z_0^0 + p_1 z_1^0 + \dots + p_N z_N^0 &= m_0 \\ p_0 z_0^1 + p_1 z_1^1 + \dots + p_N z_N^1 &= m_1 \\ &\vdots \end{aligned} \quad (\text{A.1})$$

$$p_0 z_0^{2N-1} + p_1 z_1^{2N-1} + \dots + p_N z_N^{2N-1} = m_{2N-1}$$

$$m_i = \left(\frac{1}{n} \right) X^i \quad (\text{A.2})$$

where z_i represents the center of cluster i ; p_i the fraction of data samples in the i th cluster; m_i , the i th moment of data samples. Z_i ($i = 0, 1, 2, \dots, N$) can be obtained in terms of m_i ($i = 0, 1, 2, \dots, 2N - 1$), and p_i ($i = 0, 1, 2, \dots, N$) can be obtained

in terms of z_i and m_i . The moment-preserving clustering algorithm has the following distinct feature: for $N = 2, 3$, or 4 , it can find the center of each cluster in a liner time.

References

- Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.
- Gunsel, B., Ferman, M., Murat, A.T., 1998. Temporal video segmentation using unsupervised clustering and semantic object tracking. *J. Electron. Imaging* 7 (3), 592–603.
- Hanjalic, A., Zhang, H.J., 1999. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Trans. Circuits and Systems for Video Technology* 9 (8), 1280–1289.
- Idris, F., Panchanathan, S., 1997. Review of image and video indexing techniques. *J. Vis. Commun. Image Rep.* 8 (2), 146–166.
- Jain, A.K., Dubes, R.C., 1998. Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs, NJ.
- Jain, A.K., Vailaya, A., Wei, X., 1999. Query by video clip. *Multimedia Systems* (7), 369–384.
- Jiang, H., Helal, A., Elmagarmid, A.K., Joshi, A., 1998. Scene change detection techniques for video database systems. *Multimedia Syst.*, 186–195.
- Joshi, A., Auephanwiriyakul, S., Krishnapuram, R., 1998. On fuzzy clustering and content based access to networked video database. In: *IEEE Conference, Eighth International Workshop on Continuous-Media Databases and Applications. Research Issues in Data Engineering*, pp. 42–49.
- Lupatini, G., Saraceno, C., Leonardi, R., 1998. Scenebreak detection: a comparison. In: *IEEE Conference, Eighth International Workshop on Continuous-Media Databases and Applications. Research Issues in Data Engineering*, pp. 34–41.
- Nagasaka, A., Tanaka, Y., 1992. Automatic video indexing and full video search for object appearance. *IFIP: Vis. Database Syst. II*, 113–127.
- Ralph, M.F., Robson, C., Temple, D., Gerlach, M., 2000. Metrics for shot boundary detection in digital video sequences. *Multimedia Syst.* 8, 37–46.
- Sethi, I.K., Patel, N., 1995. A statistical approach to scene change detection. *SPIE* 2420, 329–338.
- Shahraray, S., 1995. Scene change detection and content-based sampling of video sequence. *SPIE* 2419, 2–13.
- Swanberg, D., Shu, C.F., Jain, R., 1993. Knowledge guided parsing in video database. *Proc. SPIE*, 13–24.
- Tsai, W.S., 1985. Moment-Preserving Thresholding: A New Approach. *Comp. Vision Graphics Image Process.* 29, 377–393.
- Zhang, Y.J., Lu, H.B., 2002. A hierarchical organization scheme for video data. *Pattern Recognit.* (35), 2381–2387.
- Zhang, H.J., Kankanhalli, A., Smoliar, S.W., 1993. Automatic partitioning of full-motion video. *ACM Multimedia Syst.* 1 (1), 10–28.