



# Video segmentation using a histogram-based fuzzy c-means clustering algorithm

Chi-Chun Lo<sup>\*</sup>, Shuenn-Jyi Wang

*Institute of Information Management, National Chiao-Tung University, 1001 Ta Hseuh Road, Hsinchu 300, Taiwan*

Received 7 March 2001; received in revised form 1 May 2001; accepted 18 June 2001

---

## Abstract

The purpose of video segmentation is to segment video sequence into shots where each shot represents a sequence of frames having the same contents, and then select key frames from each shot for indexing. Existing video segmentation methods can be classified into two groups: the shot change detection (SCD) approach for which thresholds have to be pre-assigned, and the clustering approach for which a prior knowledge of the number of clusters is required. In this paper, we propose a video segmentation method using a histogram-based fuzzy c-means (HBFCM) clustering algorithm. This algorithm is a hybrid of the two approaches aforementioned, and is designed to overcome the drawbacks of both approaches. The HBFCM clustering algorithm is composed of three phases: the feature extraction phase, the clustering phase, and the key-frame selection phase. In the first phase, differences between color histogram are extracted as features. In the second phase, the fuzzy c-means (FCM) is used to group features into three clusters: the shot change (SC) cluster, the suspected shot change (SSC) cluster, and the no shot change (NSC) cluster. In the last phase, shot change frames are identified from the SC and the SSC, and then used to segment video sequences into shots. Finally, key frames are selected from each shot. Simulation results indicate that the HBFCM clustering algorithm is robust and applicable to various types of video sequences. © 2001 Elsevier Science B.V. All rights reserved.

*Keywords:* Key frame; Video segmentation; Shot change detection; Clustering; Fuzzy c-means; Histogram

---

## 1. Introduction

Multimedia information is ubiquitous. Among different types of media; e.g., text, graphic, image, audio, and video, video is the most challenging one since its contents are rich and need large storage space. However, because of the decreasing cost of storage devices, improving compression techniques, and the advent of high speed networking, digital

video is becoming available at an ever-increasing rate. One important research topic on video concerns the efficiency and effectiveness of querying video database.

Video segmentation is the first step of building a video query system. The purpose here is to segment video sequence into shots where each shot represents a sequence of frames having the same content. Once shots are identified, key frames are extracted from each shot for indexing. These indexed key frames from the basis of querying video database.

In order to find the right number of shots and select the optimal set of key frames from each shot,

---

<sup>\*</sup> Corresponding author. Tel.: +886-3-573-1909; fax: +886-3-572-3792.

*E-mail address:* cclo@cc.nctu.edu.tw (C.-C. Lo).

video segmentation techniques have to detect shot changes correctly. There are two types of shot changes, abrupt and gradual. Abrupt shot change resulting from editing cuts is usually easy to be detected. Gradual shot change resulting from chromatic edits, spatial edits, and combined edits [1–3] is in general hard to be detected. Existing video segmentation methods can be classified into two groups: the shot change detection (SCD) approach for which thresholds have to be pre-assigned, and the clustering approach for 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 determines the performance of SCD. As to the clustering approach, the right number of clusters is hard to be found. Different clustering may lead to completely different results.

In this paper, we propose a video segmentation method using a histogram-based fuzzy c-means (HBFCM) clustering algorithm. This algorithm is a hybrid of the two approaches aforementioned, and is designed to overcome the drawbacks of both approaches. The HBFCM clustering algorithm is composed of three phases: the feature extraction phases, the clustering phase, and the key-frame selection phase. In the first phase, differences between color histogram are extracted as features. In the second phase, the fuzzy c-means (FCM) is used to group features into three clusters: the shot change (SC) cluster, the suspected shot change (SSC) cluster, and the no shot change (NSC) cluster. In the last phase, shot change frames are identified from the SC and the SSC, and then used to segment video sequences into shots. Finally, key frames are selected from each shot. The HBFCM clustering algorithm has the following two distinct merits: first, it does not need the threshold as required by the shot change detection approach; second, it introduces the SSC cluster, which is not considered in the clustering approach. Simulation results indicate that the HBFCM clustering algorithm is robust and applicable to various types of video sequences.

In Section 2, existing video segmentation algorithms are examined. The HBFCM clustering algorithm is detailed in Section 3. In Section 4, simulation results are presented and analyzed. In Section 5, we conclude this paper with possible future directions.

## 2. Literature review

A number of video segmentation algorithms has been reported in the literature [4–13]. In general, these algorithms can be classified into two major groups: the shot change detection approach and the clustering approach.

### 2.1. Shot change detection approach

All SCD algorithms are threshold-based. Inter-frame differences are obtained by measuring differences between pixels, histograms, or blocks. If the difference is higher than the pre-assigned threshold, a shot change is declared.

#### 2.1.1. Pixel-based algorithms [4]

Pixel-based algorithms compare the pixels of two frames across the same location and can be formulated as

$$D_1(f_i, f_j) = \sum_{x=0}^M \sum_{y=0}^N |f_i(x, y) - f_j(x, y)| \quad (1)$$

where the size of an  $M$  rows by  $N$  columns frame is  $M * N$ ,  $f_i(x, y)$  denotes the measurement in terms of intensity, luminous, gray, or color, of the pixel at position  $(x, y)$  for the  $i$ th frame. The inter-frame difference between frames  $f_i$  and  $f_j$  is represented as  $D_1(f_i, f_j)$ . A shot change is declared whenever  $D_1(f_i, f_j)$  exceeds the pre-assigned threshold. Pixel-based algorithms are sensitive to noise, object motion, or camera operation.

#### 2.1.2. Histogram-based algorithms [5–8]

The intensity/color histogram of a gray/color frame  $f$  is an  $n$ -dimensional vector  $\{\mathbf{H}(f, k) | k = 1, 2, \dots, n\}$  where  $n$  is the number of levels/colors, and  $\mathbf{H}(f, k)$  the number of pixels of level/color  $k$  in frame  $f$ .

Tonomura [5] has proposed a technique based on the gray-level histogram difference. The histograms of frames  $f_i$  and  $f_j$  are denoted as  $\mathbf{H}(f_i, k)$  and  $\mathbf{H}(f_j, k)$ , respectively. Histogram difference is defined as

$$D_2(f_i, f_j) = \sum_{k=1}^n |\mathbf{H}(f_i, k) - \mathbf{H}(f_j, k)|. \quad (2)$$

A cut is declared if  $D_2(f_i, f_j)$  is greater than the pre-assigned threshold. Gargi et al. [6] have investigated the performance of the histogram-based algorithm using different color spaces. In the RGB color space, formula (2) can be rewritten as

$$D_3(f_i, f_j) = \sum_{k=1}^n (|\mathbf{H}^R(f_i, k) - \mathbf{H}^R(f_j, k)| + |\mathbf{H}^G(f_i, k) - \mathbf{H}^G(f_j, k)| + |\mathbf{H}^B(f_i, k) - \mathbf{H}^B(f_j, k)|). \quad (3)$$

To illustrate the difference between two frames across a cut, Nagasaka and Tanaka [7] have proposed the use of the chi-square test to compare two histograms,  $\mathbf{H}(f_i, k)$  and  $\mathbf{H}(f_j, k)$ . The chi-square test is defined as

$$D_4(f_i, f_j) = \sum_{k=1}^n \frac{(\mathbf{H}(f_i, k) - \mathbf{H}(f_j, k))^2}{\mathbf{H}(f_j, k)}. \quad (4)$$

If the difference  $D_4(f_i, f_j)$  is larger than the pre-assigned threshold, a shot change is declared.  $D_4(f_i, f_j)$  enhances the differences between cuts as well as the changes due to object or camera motion [8]; however, the computational complexity of  $D_4(f_i, f_j)$  is greater than  $D_2(f_i, f_j)$ .

Zhang et al. [8] suggested the so-called “twin-comparison” technique to detect gradual shot change. Two thresholds,  $T_b$  and  $T_s$ , where  $T_s < T_b$ , are set for abrupt shot change and gradual shot change, respectively. Whenever the histogram difference value exceeds threshold  $T_b$ , an abrupt shot change is declared. Any frame whose histogram difference satisfies  $T_s < D_2(f_i, f_j) < T_b$  is considered as a potential starting frame for gradual shot change. This frame is then compared to subsequent frames. This is called the accumulated comparison. The last frame of the gradual shot change is detected when the difference between consecutive frames decreases to a value less than  $T_s$ , while the accumulated comparison increases to a value larger than  $T_b$ .

### 2.1.3. Block-based algorithms [9,10]

Pixel by pixel difference and intensity/color histogram are used as global attributes in pixel-based algorithms and histogram-based algorithms, respec-

tively. Blocked-based algorithm [9,10] use local attributes instead to reduce the effect of noise or camera flash. Here, 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 of each sub-frame 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 shot change is declared whenever the number of changed sub-frames is greater than a given lower bound.

All SCD algorithms are threshold-based. If the threshold is too low, too many key frames will be extracted, resulting to a high redundancy of their visual contents. As for a high threshold, the set of key frames may be very sparse. Above all, most studies on the statistical behavior of frame difference [4] clearly indicate that a threshold that is appropriate for one type of video data will not yield acceptable results for other types of inputs.

## 2.2. Clustering approach

Clustering techniques [15] in general are used to organize and categorize data according to pre-assigned criteria. The K-means clustering and the fuzzy c-means (Appendix A) clustering are two most noticeable clustering algorithms. In a K-means clustering algorithm, a sample is assigned to one and only one cluster, so a clear partition is possible. However, as to a fuzzy c-means clustering algorithm, a sample is assigned a membership function for each cluster, so a fuzzy partition is made.

### 2.2.1. K-means clustering algorithms

Hanjalic and Zhang [13] apply the partitioning clustering technique [16] to all frames of a video sequence, and select the most suitable clustering option using an unsupervised procedure for cluster-validity analysis. Key frames are selected from centroids of optimal clusters obtained. Although they have proposed a cluster-validity analysis procedure to determine the optimal number of clusters, there still have some unanswered questions. The video sequences of interest are rather constrained to those

having short, well-defined, and reasonably structured contents. That is, to efficiently apply their approach to a long video sequence, it is necessary to first segment the video sequence into well-structured fragments. Therefore, their approach may not be directly applicable to long video sequences, especially for those video sequences with high shot change rates. Furthermore, some parameters; e.g., the number of clusters, the measurement of reference dispersion, are not clearly defined. Also, they did not explain how to derive key frames when discontinuous frames occur within one cluster; this situation is quite possible when the video sequence contains lots of shot change frames.

### 2.2.2. Fuzzy c-means clustering algorithms

Joshi et al. [11] apply fuzzy clustering to short video sequences with gradual shot change only. For each shot, its centroid frames are selected as key frames. Their approach has two major problems. First, the number of clusters has to be pre-assigned. In real situation, the actual number of clusters may not be known in advance. Second, the more abrupt shot changes in a long video sequence, the more erroneous clustering. Therefore, their method is not suited for long video sequences with abrupt shot changes.

## 3. The HBFCM clustering algorithm

The HBFCM clustering algorithm is a hybrid of the shot change detection algorithm and the clustering algorithm. It is designed to be threshold-free. It first measures histogram differences between frames, which are then used as inputs to the clustering algorithm. The number of clusters chosen is three instead of two. Two-cluster approaches [12] may erroneously put frames into wrong clusters while handling boundary conditions; i.e., those frames in which shot change is difficult to be detected. The

additional cluster suggested that the HBFCM clustering algorithm contains all ambiguous frames. Heuristic is developed to resolve those ambiguities.

### The algorithm

As illustrated in Fig. 1, the HBFCM clustering algorithm is composed of three phases: the feature extraction phase, the clustering phase, and the key-frame selection phase. Following are detailed discussions of each phase.

#### 3.1. Feature extraction phase

In this phase, each frame is compared with its previous frame using the color histogram difference as shown in Eq. (3). Frame dissimilarities are extracted as features. We have considered the red–green–blue (RGB) color coordinates, along with the YCbCr color space. In Ref. [11], it has been shown that luminance and chrominance information contained in the YCbCr color space can be used for shot change detection.

#### 3.2. Clustering phase

In this phase, the fuzzy c-means algorithm is used to group frame dissimilarities obtained in the feature extraction phase into three clusters: the shot change (SC) cluster, the suspected shot change (SSC) cluster, and the no shot change (NSC) cluster. The SSC suggested that this phase contains all frames in which shot changes are difficult to be detected. The FCM-based clustering procedure iteratively minimizes the criterion function as shown in Eq. (A1). Detailed description is given as follows.

#### The FCM clustering procedure

```
// The input values are frame dissimilarities.
// The output values are SS, SSC, and NSC clusters.
```

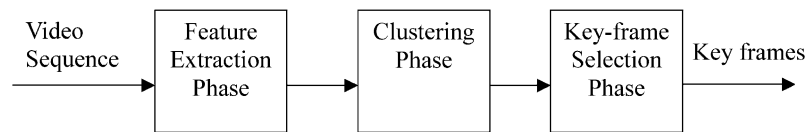


Fig. 1. The HBFCM clustering algorithm.

//  $c$  represents the number of clusters,  $w$  the exponential weight, and  
 //  $\mu_{ik}$ 's ( $i = 1, \dots, c, k = 1, \dots, n$ ) the membership values

1. Initialize parameters  $c$  and  $w$ ; and then assign values to  $\mu_{ik}$ 's using either a random function or an approximation method.
2. Do
3. For each cluster  $c$ , update center using Eq. (A6) and
4.  $\mu_{ck}$ 's using Eq. (A5);
5. Until (all centers are stabilized)
6. Assign frame dissimilarities to the SC cluster, the SSC cluster, or the NSC cluster according to  $\mu_{ik}$ 's.

In the simulation,  $c$  and  $w$  are set to 3 and 1.5, respectively. Different initial values of  $\mu_{ik}$ 's and  $w$  may result in different clustering of SC and SSC. Nevertheless, the SC and SSC always contain all shot change frames.

### 3.3. Key-frame selection phase

In this phase, shot change frames are first identified from the SC and SSC, and then are used to segment video sequences into shots. Finally, centroid frames of each shot are selected as key frames. The key-frame selection procedure is stated as follows.

#### The key-frame selection procedure

// The input values are the SC, the SSC, and video sequence.  
 // The output values are key frames

1. Label all frames in SC as shot change frames (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 frames as key frames.

In the second step, a heuristic is developed to resolve the uncertainty existing in 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,  $SSC(k) \ k = j, j + 1, \dots, j + n - 1$ , containing between  $SC(i)$  and  $SC(i + 1)$ , are checked.  $SSC(k)$  is declared as a shot change frame if its histogram difference satisfies the following inequality

$$H\_SSC(k) \geq \text{param} * [0.5 * (H\_SC(i) + H\_SC(i + 1))] \tag{5}$$

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  $\text{param}$  the weight factor.

Furthermore, because shot change will not occur between two consecutive frames, some  $SSC(k)$  frames will be discarded although their histogram differences satisfy (Eq. (5)). In Eq. (5), we assign  $\text{param}$  to be equal to 0.5. In fact, a fuzzy number instead of a constant for  $\text{param}$  should be used. From simulation results, we notice that this choice is rea-

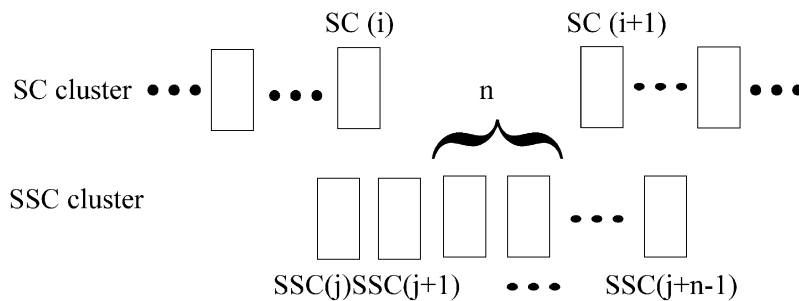


Fig. 2. SSC frames.

sonable. Also, in Eq. (5), the constant 0.5 is used to take the average of  $H\_SC(i)$  and  $H\_SC(i + 1)$ .

The HBFCM clustering algorithm has the following two distinct merits: first, it does not need the threshold as required by shot change detection algorithms; second, it introduces the SSC cluster, which is not considered in other clustering algorithms.

#### 4. Experimental result and analyses

We have simulated the HBFCM clustering algorithm. Simulations were done by using an IBM PC with the Intel Pentium III processor and 128 MB RAM. The MATLAB toolbox for fuzzy and image processing is used to develop the HBFCM clustering algorithm. For comparison, Zhang et al.'s algorithm [8] is simulated.

##### 4.1. Performance metrics

Two-performance metrics, the hit ratio (HR) and the fault ratio (FR), are used to evaluate the HBFCM clustering algorithm. HR and FR are expressed as  $N_d/N_t$  and  $(N_m + N_e)/N_t$ , respectively, where  $N_d$  represents the number of correct detection,  $N_m$  the number of missing detection,  $N_e$  the number of erroneous detection, and  $N_t (= N_d + N_m)$  the total number of real shot change frames in the video sequence being examined. A well-performed video segmentation algorithm should exert itself with both a high hit ratio and a low fault ratio.

##### 4.2. Assumptions

In the simulations, the following assumptions are assumed.

(1) For a gradual shot change, we consider that dissolve, fade in, or fade out introduces only one shot change frame, and pan or zoom produces no shot change frame.

(2) Since unclear video editing may cause several abrupt shot changes within two or three continue frames, we claim that the interval between two abrupt shot changes should cover at least two frames so as to eliminate the effect of unclear video editing.

(3) It is necessary to identify the ground-truth shot change frames, for measuring the performance of a video segmentation algorithm. We obtain the ground-truth shot change frames by manually examining each test video sequence frame by frame from five individuals.

(4) In our experiments, we find that using either the summation of RGB color space or the luminance of YCbCr space produces the same simulate results. From the aspect of frame difference, the results obtained by using histogram interaction and the bin-to-bin histogram differences are the same [14]. This phenomenon has been proven by Ford et al. [14]. We also notice whether the bin-to-bin histogram difference is normalized or has no effect on the simulation results; so does the chi-square histogram difference. From the observations aforementioned, we conduct the simulations 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 video segmentation algorithm is sensitive to the shot change ratio (SCR), where the SCR is equal to the number of shot change frames 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

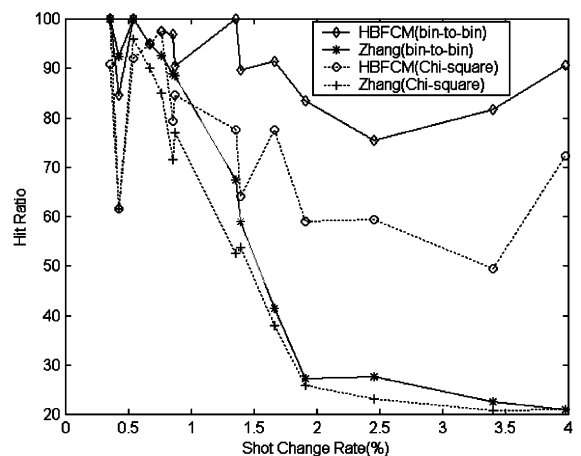


Fig. 3. Comparison between hit ratios of the HBFCM clustering algorithm and those of Zhang et al.'s algorithm [8].

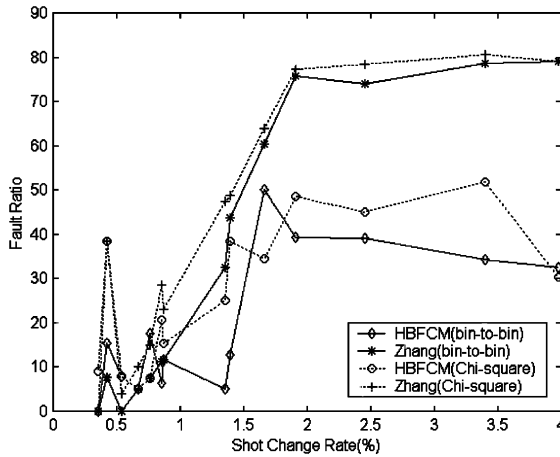


Fig. 4. Comparison between fault ratios of the HBFCM clustering algorithm and those of Zhang et al.'s algorithm [8].

SCR with one shot change frame 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%. Considering high and low SCRs, we have chosen 14 test video sequences with SCRs 0.35%, 0.42%, 0.54%, 0.67%, 0.76%, 0.85%, 0.87%, 1.35%, 1.39%, 1.66%, 1.91%, 2.45%, 3.40%, and 3.98%, respectively. Test video sequences are selected from various categories; e.g., movie, animation, advertisement, and soap opera. Action movie and advertisement usually have high SCRs. On the contrary, romantic movie and soap opera have low SCRs. Each test video sequence contains a number of abrupt shot changes coupled with few gradual shot changes (on the average of 2.6% out of all shot changes). The

bin-to-bin and the chi-square color histogram differences are considered.

#### 4.4. Results and analyses

Fig. 3 compares hit ratios of the HBFCM clustering algorithm with those of Zhang et al.'s algorithm. Fig. 4 presents the comparison between fault ratios of the HBFCM clustering algorithm and those of Zhang et al.'s algorithm. By examining Figs. 3 and 4, we notice that, for medium and high SCRs ( $\geq 2$ ), the HBFCM clustering algorithm has better performance than Zhang et al.'s algorithm; and the higher the shot change rate, the larger the difference. For low SCRs ( $\leq 0.75$ ), we find that Zhang et al.'s algorithm obtains better hit ratio and fault ratio than those of the HBFCM clustering algorithm. However, the differences are not significant. Tables 1 and 2 are derived from Figs. 1 and 2, respectively. Table 1 indicates that the HBFCM clustering algorithm improves on average the hit ratio by 30% to 38%. Table 2 indicates that the HBFCM clustering algorithm improves on average the fault ratio by 37% to 43%. The HBFCM clustering algorithm using the bin-to-bin color histogram difference has the best performance of all algorithms tested. 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. Therefore, some shot change frames might be excluded from the SC cluster and the SSC cluster when using the chi-square histogram difference. This phenomenon is especially noticeable at 0.42% SCR; thus, causes the peaks in Figs. 3 and 4. In all test cases, the computation time of Zhang et al.'s algorithm is in the order of seconds while that of the HBFCM clustering algorithm is in the order of min-

Table 1  
Average improvement with respect to the hit ratio

	Bin-to-bin			Chi-square		
	HBFCM (a)	Zhang (b)	Improvement ( $ a - b /b$ )	HBFCM (a)	Zhang (b)	Improvement ( $ a - b /b$ )
Average hit ratio (%)	91.2	66.0	38.2	75.7	58.3	29.8

Table 2  
Average improvement with respect to the fault ratio

	Bin-to-bin			Chi-square		
	HBFCM (a)	Zhang (b)	Improvement ( $ a - b /b$ )	HBFCM (a)	Zhang (b)	Improvement ( $ a - b /b$ )
Average fault ratio (%)	19.8	34.8	43.1	27.0	42.5	h836.5

utes. The HBFCM clustering algorithm spends more computation time than Zhang et al.'s algorithm due to the clustering and the selection of shot change frames from the SSC cluster. This is an inherited trade-off between efficiency and effectiveness. But, in the case of video segmentation, the effectiveness prevails.

#### 4.5. Discussions

Figs. 5 and 6 show that the assignment of 0.5 to param in Eq. (5) is reasonable. By examining Figs. 5 and 6, we have the following observations: for value 0.3, param has the highest (best) hit ratio; however, the highest (worst) fault ratio; for value 0.5, param has the second highest hit ratio and the lowest (best) fault ratio; for value 0.7, param has the lowest (worst) hit ratio and the second highest fault ratio. Apparently, param of value 0.5 has the most stable performance in terms of both hit ratio and fault ratio.

For video sequences with many abrupt shot changes and few gradual shot changes, the HBFCM clustering algorithm works well in terms of both the hit ratio and the fault ratio. For completeness, we further analyze the performance of the HBFCM clustering algorithm with respect to video sequences with a number of gradual shot changes. Five types of video sequence are examined. Types 1–5 contain video sequences of 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 video sequences are simulated. For comparison, ground truth is manually obtained and Joshi et al.'s algorithm [11] is simulated. Table 3 compares the average number of key frames obtained by ground truth, the HBFCM clustering algorithm, and Joshi et al.'s algorithm. By examining Table 3, we notice that the performance of the

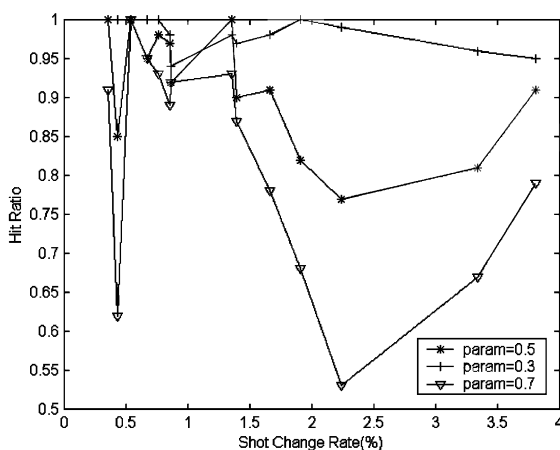


Fig. 5. Hit ratios with respect to param of assigned values 0.3, 0.5 and 0.7, respectively.

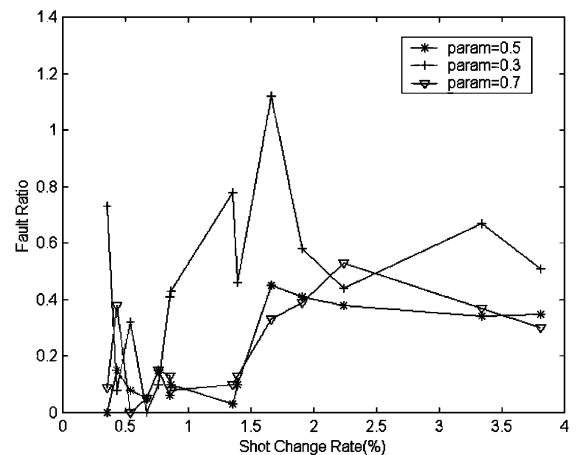


Fig. 6. Fault ratios with respect to param of assigned values 0.3, 0.5 and 0.7, respectively.



Table 3

Comparison between the average number of key frames obtained by ground truth, the HBFCM clustering algorithm, and Joshi et al.'s algorithm

Video type	Average number of key frames		
	Ground truth	HBFCM algorithm	Joshi algorithm
1 (1 gradual)	2.00	2.00	2.00
2 (2 graduals)	3.00	4.33	N/A
3 (1 gradual + 1 abrupt)	3.00	2.33	3.00
4 (2 graduals + 1 abrupt)	4.00	3.67	N/A
5 (1 gradual + 2 abrupts)	4.00	3.33	5.67

N/A: not available; the algorithm failed.

HBFCM clustering algorithm is close to that of the ground truth, and is much better than Joshi et al.'s algorithm.

## 5. Conclusion

In this paper, we propose the HBFCM clustering algorithm for selecting key frames from a video sequence. The HBFCM clustering algorithm has the following two distinct merits: first, it does not need the threshold as required by the shot change detection algorithms; second, it introduces the SSC cluster which is not considered in other clustering algorithms. Here, we would like to mention the following areas of investigation, which may merit further study.

1. Extend the feature extraction method to compressed video sequences; e.g., MPEG four videos.
2. Develop video-indexing method; subsequently, combine it with the proposed HBFCM clustering algorithm to build a video query system.

## Appendix A. Fuzzy c-means

The fuzzy c-means (FCM) has been used extensively in pattern recognition and computer vision. The purpose of FCM [16] is to minimize the object function  $J(\mathbf{U}, \mathbf{V})$

$$J(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^c \sum_{k=1}^n \mu_{ik}^w |\mathbf{X}_k - \mathbf{V}_i|^2 \quad (\text{A1})$$

where,  $c$  represents the number of clusters,  $n$  is the number of data items,  $w$  is the exponential weight,  $\mathbf{X} = \{x_1, x_2, x_3, \dots, x_n\}$  an  $n$ -dimensional data vector,  $\mathbf{V} = \{v_1, v_2, \dots, v_c\}$  a vector of dimension  $c$ ,  $\mathbf{U} = (\mu_{ik})$  a  $c * n$  matrix, where  $\mu_{ik}$  represent the membership value of vector  $\mathbf{X}_k$  in cluster  $i$ , and

$$0 \leq \mu_{ik} \leq 1 \quad i = 1, 2, \dots, c; k = 1, 2, \dots, n \quad (\text{A2})$$

$$\sum_{k=1}^n \mu_{ik} = 1 \quad k = 1, 2, \dots, n \quad (\text{A3})$$

$$0 \leq \sum_{k=1}^n \mu_{ik} \leq n \quad i = 1, 2, \dots, c. \quad (\text{A4})$$

The minimization of the objective function with respect to membership values leads to

$$\mu_{ik} = \frac{\left[ \frac{1}{|\mathbf{X}_k - \mathbf{V}_i|^2} \right]^{1/(m-1)}}{\sum_{j=1}^c \left[ \frac{1}{|\mathbf{X}_k - \mathbf{V}_j|^2} \right]^{1/(m-1)}} \quad i = 1, 2, \dots, c; k = 1, 2, \dots, n. \quad (\text{A5})$$

The minimization of the objective function with respect to the center of each cluster gives rise to the following equality

$$\mathbf{V}_i = \frac{\sum_{k=1}^n \mu_{ik}^m \mathbf{X}_k}{\sum_{k=1}^n \mu_{ik}^m} \quad i = 1, 2, \dots, c. \quad (\text{A6})$$

## References

- [1] F. Idris, S. Panchanathan, Review of image and video indexing techniques, *Journal of Visual Communication and Image Representation* 8 (1997) 146–166.
- [2] H. Jiang, A. Helal, A.K. Elmagarmid, A. Joshi, Scene change detection techniques for video database systems, *Multimedia Systems* 6 (1998) 186–195.
- [3] G. Lupatini, C. Saraceno, R. Leonardi, Scene break detection: a comparison, In: *IEEE conference, Eighth International Workshop on Continuous-Media Databases and Applications* 8 (1998) 34–41.
- [4] I.K. Sethi, N. Patel, A statistical approach to scene change detection, In: *Proc. SPIE'95*, 2420 (1995) 329–338.
- [5] Y. Tonomura, Video handling based on structured information for hypermedia systems, *ACM Proceedings '91, International Conference on Multimedia Information Systems* (1991) 333–344.

- [6] U. Gargi, S. Oswald, D. Kosiba, S. Devadiga, R. Kasturi, Evaluation of video sequence indexing and hierarchical video indexing, In: Proc. SPIE'95, Storage Retrieval Image Video Databases III, (1995) 144–151.
- [7] A. Nagasaka, Y. Tanaka, Automatic video indexing and full video search for object appearance, In: Proc IFIP'92, Visual Database Systems II, (1992) 113–127.
- [8] H.J. Zhang, A. Kankanhalli, S.W. Smoliar, Automatic partitioning of full-motion video, ACM Multimedia Systems 1 (1993) 10–28.
- [9] S. Shahraray, Scene change detection and content-based sampling of video sequence, In: Proc. SPIE'95, 2419 (1995) 2–13.
- [10] D. Swanberg, C.F. Shu, R. Jain, Knowledge guided parsing in video database, In: Proc. SPIE'93, (1993) 13–24.
- [11] A. Joshi, S. Auephanwiriyakul, R. Krishnapuram, On Fuzzy Clustering and Content Based Access to Networked Video Database, In: IEEE conference, Eighth International Workshop on Continuous-Media Databases and Applications (1998) 42–49.
- [12] B. Gunsel, A.M. Ferman, A.M. Tekalp, Temporal video segmentation using unsupervised clustering and semantic object tracking, Journal of Electronic Imaging 7 (1998) 592–603.
- [13] A. Hanjalic, H.J. Zhang, An Integrated Scheme for Automated Video Abstraction Based on Unsupervised Cluster-Validity Analysis, IEEE Transactions on Circuits and Systems for Video Technology 9 (1999) 1280–1289.
- [14] R.M. Ford, C. Robson, D. Temple, M. Gerlach, Metrics for shot boundary detection in digital video sequences, Multimedia Systems 8 (2000) 37–46.
- [15] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.

- [16] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, New Jersey, 1998.



**Chi-Chun Lo** was born in Taipei, Taiwan, Republic of China, on August 22, 1951. He received the BS degree in mathematics from the National Central University, Taiwan, in 1974, the MS degree in computer science from the Memphis State University, Memphis, TN, in 1978, and the PhD degree in computer science from the Polytechnic University, Brooklyn, NY, in 1987.

From 1981 to 1986, he was employed by the AT & T Bell Laboratories, Holmdel, NJ, as a Member of Technical Staff. From 1986 to 1990, he worked for the Bell Communications Research as a Member of Technical Staff. Since 1990, he has been with the Institute of Information Management, National Chaio-Tung University, Taiwan. At present, he is professor and director of the institute. His major current research interests include network design algorithm, network management, network security, network architecture, and multimedia query system.



**Shuenn-Jyi Wang** was born in Taoyuan, Taiwan, Republic of China, on September 30, 1963. He received the BS degree in Applied Mathematics, and MS degree in Electronic Engineering from the Chun Chang Institute of Technology, Taiwan, in 1987 and 1993, respectively. Currently, he is a PhD candidate in the Institute of Information Management, National Chaio-Tung University, Hsin-chu, Taiwan. His major current research interests include digital image processing, and multimedia database system.