Available online at www.sciencedirect.com
SCIENCE DIRECT
Pattern Recognition 39 (2006) 2092–2100
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

A fuzzy logic approach for detection of video shot boundaries

Hui Fang, Jianmin Jiang*, Yue Feng

School of Informatics, University of Bradford, Richmond Road, Bradford, West Yorkshire BD7 1DP, UK

Received 26 October 2005; received in revised form 3 March 2006; accepted 28 April 2006

Abstract

Video temporal segmentation is normally the first and important step for content-based video applications. Many features including the pixel difference, colour histogram, motion, and edge information etc. have been widely used and reported in the literature to detect shot cuts inside videos. Although existing research on shot cut detection is active and extensive, it still remains a challenge to achieve accurate detection of all types of shot boundaries with one single algorithm. In this paper, we propose a fuzzy logic approach to integrate hybrid features for detecting shot boundaries inside general videos. The fuzzy logic approach contains two processing modes, where one is dedicated to detection of abrupt shot cuts including those short dissolved shots, and the other for detection of gradual shot cuts. These two modes are unified by a mode-selector to decide which mode the scheme should work on in order to achieve the best possible detection performances. By using the publicly available test data set from Carleton University, extensive experiments were carried out and the test results illustrate that the proposed algorithm outperforms the representative existing algorithms in terms of the precision and recall rates.

Keywords: Temporal video segmentation; Shot cut detection; Shot-boundary detection; Content-based video processing

1. Introduction

Multimedia applications have been extremely expanded over the past decade, and a hierarchical structure is needed for managing the video content. With such a demand for a hierarchical model of videos, many researchers are moving into this area and developing algorithms for visual content interpretation, analysis, and management, where content-based video retrieval and content-based video coding [1] stand out as the most representative examples in this trend.

To build a system and manage the structure of videos, it is widely recognised that the first step would be the automatic detection of shot cuts or their boundaries to divide the video sequence into manageable sections, where the visual content remains consistent in terms of camera operations and other visual events. From the media production side, meaningful stories and sceneries are generated via sequences of video editing, which resulted in a range of different shot boundaries, including abrupt shot boundary, dissolved shot boundary, fade in, and fade out. These shot boundaries can be grouped into two categories according to the duration of the change. One is called abrupt shot cut if an instantaneous change happens from one shot to another. This type of shot boundaries is primarily used by editors to cut scenes into consistent sections. The other one is called gradual shot boundary including fade in, fade out and dissolve because the shot changes gradually. A fade is defined as a shot appear or disappear through several of frames. Dissolve is made when a shot fades in whilst another shot fades out. Samples of these shot boundaries are illustrated in Fig. 1.

Corresponding to these shot boundaries as illustrated in Fig. 1, many algorithms have been proposed and reported in the published literature [2–12]. Pixel-based difference analysis remains to be one of the most adopted approaches to find the dissimilarity between shot boundaries. Exploitation of temporal information, such as those based on motion estimation and compensation techniques [13], represents another major direction for shot cut detection, in which the large difference value caused by the global motion can be

* Corresponding author. Tel.: +44 1274 233695; fax: +44 1274 233727.
E-mail address: j.jiang1@bradford.ac.uk (J. Jiang).

0031-3203/$30.00
doi:10.1016/j.patcog.2006.04.044

Fig. 1. Real video illustration of shot boundaries: (a) example of abrupt shot cut; (b) example of fade-in; (c) example of fade-out; (d) example of dissolve.

easily detected and extracted by a simple block matching procedure. In Ref. [2], shot cut boundaries are detected by calculating the normalised correlation among blocks and locating the maximum correlation coefficient in the frequency domain. In line with the motion information, spatial information such as edges etc. is often regarded as another important feature for shot cut detection. The edge tracking algorithm reported in [8] stands out as a representative example for this direction of research, where the proposed techniques are mainly based on the principle that, when most of the edge information is lost in the consecutive frames, a shot cut can be declared. In Ref. [4], the feature of edge pixel count is proposed for shot cut detection, where Sobel edge detector is used. But the most widely used feature is colour histogram, examples of which includes histogram intersection, "twin-comparison", local histogram etc. [3–6]. These histogram features, however, have drawbacks in the sense that, when the global motion is fast or the contrast of the frames is low, these colour histogram based techniques will miss many shot boundaries, and thus their recalling performance becomes poor. In Ref. [7], a visual rhythm-based technique is reported, which produced good experimental results in shot cut detection. Due to the complexity of the algorithm design, however, this technique can only be used for off-line shot cut detection. In contrast to most existing approaches, we propose a fuzzy logic approach in this paper to exploit the feature-based techniques and detect shot boundaries. The advantages of our contribution can be highlighted as: (i) a range of features can be integrated by fuzzy logic operation to exploit their individual strength collectively; and (ii) while directly thresholding features remains sensitive to noises, selecting threshold in fuzzy domain provides a buffered operation and thus makes the detection more reliable. The rest of this paper is organised as follows. Section 2 describes detailed design of the proposed algorithm. Section 3 reports experimental results in comparison with three existing algorithms, and Section 4 provides conclusions.

2. The proposed algorithm design

2.1. Feature extraction and mode selector design

To detect relatively abrupt scene changes between boundaries of shots, we propose to combine a range of representative features to construct a hybrid feature for shot cut detection. By representative, we mean those features that: (i) are widely used in existing shot cut detection algorithms; (ii) are relatively mature that substantial evaluations have been reported and supportive results obtained in the literature; and (iii) whose implementation does not incur intensive computing cost and their complexity level is limited to a manageable level. Details of our selected features are given below.

2.1.1. Color feature

Being non-sensitive to motion, color histogram is often regarded as a global feature in video sequences. Histogram intersection, which is part of the histogram feature, is widely

used for temporal video segmentation because of its global characteristic as described in Ref. [4]. Given the ith frame inside a video sequence, a normalised histogram intersection between the two neighbouring video frames can be defined as follows:

$$HI_i = 1 - \left(\frac{1}{3n}\right)\left[\sum_{j=1}^{256}\min(I_{rj}^{i}, I_{rj}^{i-1}) + \sum_{j=1}^{256}\min(I_{gj}^{i}, I_{gj}^{i-1}) + \sum_{j=1}^{256}\min(I_{bj}^{i}, I_{bj}^{i-1})\right], \quad (1)$$

where n stands for the total number of pixels inside the video frame, I for the number of pixels in j bin, and r, g, b for the red, green and blue color components.

Previous research [4] reveals that the value of HI_i varies within [0, 1], and the closer to 1, the larger the difference between the two adjacent frames.

2.1.2. Motion compensation feature

Extraction of motion information enables detection of visual discontinuity between consecutive frames, via which false detection caused by large global motion or local noise can be avoided. To estimate motion, we adopt the simple MPEG scheme to extract the motion feature. Given the ith frame as the current frame, i.e. the frame under examination, its previous frame is taken as a reference frame. When both frames are divided into blocks of 16×16 pixels which have the same size with the macro-blocks in MPEG, motion estimation can be carried out in terms of block matching, where each block inside the current frame is used to search for its best match inside the reference frame. During this process, the sum of absolute differences (SAD) is calculated to measure the level of match between the current block and those inside the reference frame. Specific SAD value is calculated via the following definition [13]:

$$SAD(u, v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}|I_i(x, y) - I_{i-1}(x+u, y+v)|, \quad (2)$$

where $I_i(x, y)$ represents the pixel intensity located at (x, y) inside the block of the ith frame, N is the size of macro-blocks which is 16 in this paper.

The block with the smallest SAD value is selected as the best matching block, based on which a motion compensation feature can be defined as follows:

$$MC_i = \frac{1}{N_B}\sum_{n=1}^{N_B}(|\overline{Y_n} - \overline{Y_n'}| + |\overline{U_n} - \overline{U_n'}| + |\overline{V_n} - \overline{V_n'}|), \quad (3)$$

where N_B is the number of blocks in the frame; $\overline{Y}, \overline{U}, \overline{V}$ are the average Y, U, V components of the blocks; and $\overline{Y'}, \overline{U'}, \overline{V'}$ are the corresponding components of the best matching block. If no motion estimation is done, this feature will become the same as the pixel difference metric, which is also the main feature used in Refs. [6,8,11].

2.1.3. Texture feature

Gray-level co-occurrence matrix is defined in Ref. [14] as a texture map to characterise the texture feature of an image. To exploit the texture feature inside video frames for shot cut detection, we propose to quantise the luminance component of the current frame into 16 intensity levels with equal quantization steps inside the intensity depth, and then construct four texture maps along four directions: horizontal, 45°, vertical and 135°. As each of these texture maps records the occurrence number of neighbouring pixels along its specific direction, a feature of energy can be extracted as follows:

$$Energy_d = \sum_{x=0}^{15}\sum_{y=0}^{15}\left(\frac{c(x, y)}{N}\right)^2, \quad d \in [0°, 45°, 90°, 135°], \quad (4)$$

where (x, y) indicates the location of texture map (row and column), $c(x, y)$ is the occurrence number of neighbouring pixels for the location of (x, y), and N is the total number of occurrences recorded inside the texture map.

As a result, the texture relation between adjacent video frames can be characterised by the difference of their total texture energy as given below:

$$TD_i = \frac{1}{4}\left|\sum_d Energy_{d,i} - \sum_d Energy_{d,i-1}\right|, \quad (5)$$

where TD stands for the texture difference between the ith frame and its neighbour, the $i-1$th frame, and d indicates the four directions used to construct the texture map.

2.1.4. Mode selector

Up to this stage, we have extracted three features, histogram intersection, motion compensation, and texture energy difference. All the three features share a common principle that they are always calculated between the current frame and its preceding frame, representing a difference between the two frames. From Eqs. (1), (3) and (5), it can be seen that all the three features are always positive. Therefore, we propose to set up a sliding window with seven frames to detect the boundary of a shot. Specifically, given the ith frame being the current frame under examination, the sliding window can be constructed to contain frame $i-3$, $i-2$, $i-1$, i, $i+1$, $i+2$ and $i+3$. As mentioned in Ref. [6], abrupt change in the temporal domain is a local obvious change. Considering the positive nature of all the three features, it can be inferred that the necessary condition for frame-i to be the point of an abrupt shot cut is that a peak of at least one feature value should be detected at frame-i. As a result, each frame can be screened by such peak detection and those frames where no peak is detected can be excluded for possible shot cuts.

To detect the peak, we define

$$\Delta_1^k = F_i^k - F_{i-1}^k, \tag{6}$$

$$\Delta_2^k = F_{i+1}^k - F_i^k, \tag{7}$$

where F^k stands for one of the three feature values extracted in Eqs. (1), (3) and (5), and thus $k \in [1, 3]$.

As a result, given the ith frame, the mode selector can be designed as follows:

For $(k = 1, k \leqslant 3, k++)\{$
  If $(\Delta_1^k > 0 \&\& \Delta_2^k < 0)$ Stop the for-loop and go to $abrupt - shot - cutmode$;
  Else go to gradual-shot-cut mode.
$\}$ (8)

2.2. Detection of shot boundaries

2.2.1. Abrupt shot cut detection

Whenever a feature peak is detected, the current frame will be examined to see if any abrupt shot cut exist between this frame and its next frame. Such abrupt shot cut detection is carried out by considering a ratio of respective feature values inside the sliding window. Considering the fact that some abrupt shot cut may contain a short dissolve, we adopt the protection slot suggested in Ref. [8] to exclude the frame $i - 1$ and frame $i + 1$ from the calculation of the feature ratios. Hence, feature ratios can now be defined as follows:

$$R^k = \frac{F_i^k}{F_{i-3}^k + F_{i-2}^k + F_{i+2}^k + F_{i+3}^k} \quad \text{if } F_{i-3}^k + F_{i-2}^k + F_{i+2}^k + F_{i+3}^k \neq 0, \tag{9}$$

where $k \in [1, 3]$ indicates that the ratios can be calculated for all three features defined in Eqs. (1), (3) and (5). If $F_{i-3}^k + F_{i-2}^k + F_{i+2}^k + F_{i+3}^k = 0$, it must be a noise as this indicates an impossible case that all the frames are kept the same throughout the sliding window.

The exclusion of frames $i - 1$ and $i + 1$ characterises the situation that some abrupt shots may have a short dissolve, where the shot boundaries could span over more than one frame. The window which we use to calculate the ratios is set to seven frames, a compromised selection under the principle that the number of false detection increases due to some noises if the window is set too small, while the processing time would be increased heavily if the window is set too large.

Existing research on detecting shot boundaries by direct feature processing approaches has a number of weakness, which include: (i) it is difficult to determine which features are more important than others and which features are more suitable for detecting a particular type of shot cuts; (ii) it is difficult to determine an optimal threshold for individual features or design a global threshold for all features. To this end, we propose to: (i) integrate all adopted features via their fuzzy memberships to improve their combination and interaction; and (ii) set thresholds in fuzzy domain and produce fuzzy rules to do the shot boundary detection. In this way, all the features and their ratios can be fuzzified to improve the robustness and reliability in shot cut detection, due to the nature of uncertainty and vagueness in practical shot boundaries of general videos. This is supported by many other applications in resolving problems under uncertain circumstances in computer vision [15]. Instead of estimating the probability by training the Bayesian Network or using machine learning algorithm, fuzzy logic calculates the possibility through the membership functions and the fuzzy relationship among different fuzzy sets.

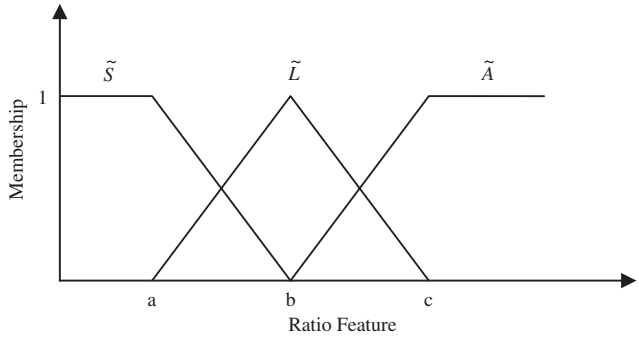

Fig. 2. Illustration of the fuzzy membership function.

Specifically, we define three fuzzy sets, S, L, A, as shown in Fig. 2 to represent slight change, large change and abrupt change in ratio features given in Eq. (9). While the slight change of ratio features indicates that the current frame under examination is definitely not a shot cut, the fuzzy set of abrupt change indicates that the current frame is definitely an abrupt shot cut. Correspondingly, the fuzzy set of large change in ratio features represents the uncertain situation that the current frame could be or not to be a shot cut. The corresponding membership functions are defined as follows:

$$M(R^K)_A = \begin{cases} 1 & \text{if } R^k > c, \\ \dfrac{R^k - b}{c - b} & \text{if } b < R^k < c, \end{cases}$$

$$M(R^k)_L = \begin{cases} \dfrac{c - R^k}{c - b} & \text{if } b < R^k < c, \\ \dfrac{R^k - a}{b - a} & \text{if } a < R^k < b, \end{cases}$$

$$M(R^k)_S = \begin{cases} 1 & \text{if } R^k < a, \\ \dfrac{b - R^k}{b - a} & \text{if } a < R^k < b, \end{cases} \tag{10}$$

where R^k represents the ratio features defined in Eq. (9), and the three parameters a, b, and c represent the boundaries of the three fuzzy sets as shown in Fig. 2.

From Fig. 2, it can be seen that: (i) the feature ratio smaller than a indicates a definite membership for the set S; and (ii) the ratio value larger than c indicates a definite membership for the set A. Therefore, the way to determine the specific

values of a and c is to see under what circumstance the ratio value can be regarded as small change and abrupt change. To this end, we carried out some preliminary test on the training sequence, and the observations reveal that: (i) a small change can be primarily established when the ratio value is less than 1; and (ii) an abrupt change can be primarily established when the ratio value is larger than 3. As a result, a can be determined as 1, and c can be determined as 3. Since the parameter b has equal distances to both a and c, it can be determined as 2. As a matter of fact, the three parameters are mainly used to determine the boundaries of three fuzzy sets, and thus there exists certain flexibility around their specific values. In other words, their values do not need to be unique and their variation within a small range would not affect the final shot cut detection results.

After the fuzzification, the crisp decision making process is changed into a soft one, where each ratio value is assessed with certain level of possibility that the current frame represents a shot cut or not. As a result, the features are mapped into fuzzy domain through the fuzzification process.

Following the fuzzification, a feature fusion is designed to process all the membership values and reach the final decision on abrupt shot cut detection. Given the feature ratio value R_k, the fuzzification process produces three membership values $R_{k,s}$, $R_{k,L}$, and $R_{k,A}$ corresponding to the three fuzzy sets as illustrated in Fig. 2. By using the fuzzy inference method described in Ref. [15], all the membership values are subject to AND and OR operations as defined below:

$$\{x_1, x_2, x_3\} \quad \text{AND} \quad \{y_1, y_2, y_3\} = \{Max(x_1, y_1), Max(x_2, y_2), Max(x_3, y_3)\},$$

$$\{x_1, x_2, x_3\} \quad \text{OR} \quad \{y_1, y_2, y_3\} = \{Min(x_1, y_1), Min(x_2, y_2), Min(x_3, y_3)\}. \tag{11}$$

A complete fuzzification and feature fusion system is illustrated in Fig. 3, where the final output values, P_S, P_L, P_A, represent the possibility that the change detected upon the current frame belongs to slight change, large change or

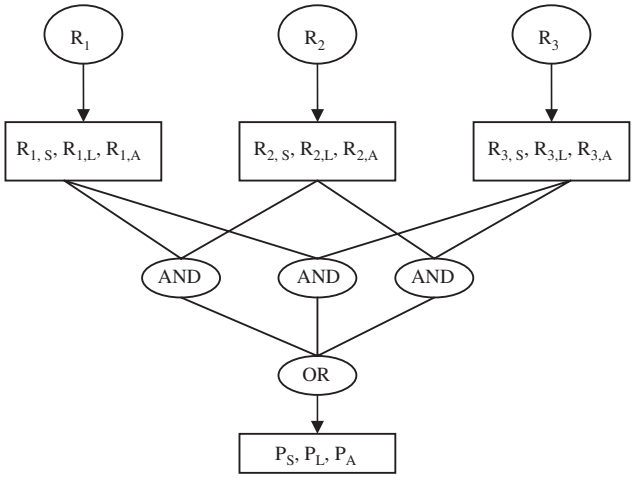

Fig. 3. Fuzzy fusion of the ratio features.

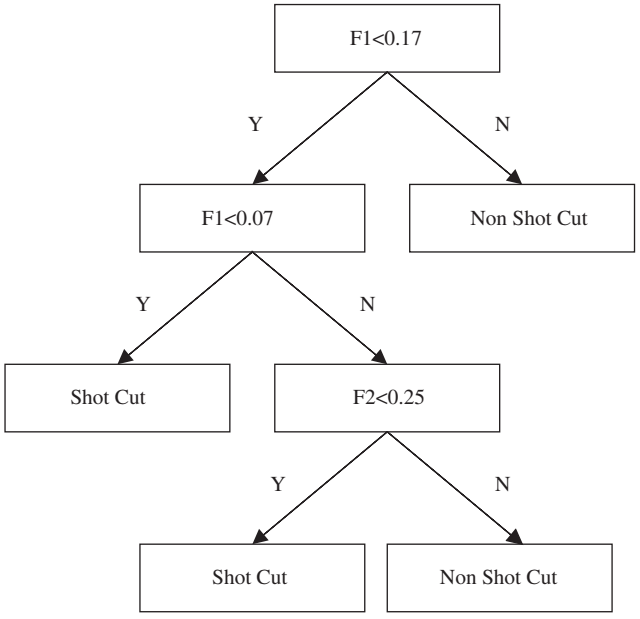

Fig. 4. Summary of decision-making process.

abrupt change. To make a final decision, the basic principle is that the smaller the value of P_S, representing the possibility of slight change, the more likely that the change in current frame represents an abrupt shot cut. Similarly, the larger the value of P_A, representing the possibility of abrupt change, the more likely that the current frame represents an abrupt shot boundary. To ensure that all these factors are taken into consideration, we further adopted a so-called C4.5 decision tree [16] to formulate optimal rules for final decision making. C4.5 is a popular data mining algorithm due to its capability of handling numeric attributes and post-prune the rules to a simplest decision tree. Specifically, the P_s, P_l and P_A, which are the output of the fuzzy fusion part, are regarded as the input for a C4.5 decision tree. A sequence which includes 90 abrupt shot boundaries is taken as a training sequence and the output is the decision rules we use to find the shot boundary. The decision trees and the rules are shown in Fig. 4.

2.2.2. Gradual change detection

When the mode selector given in Eq. (8) decides to go to the gradual shot-cut-mode, variance changes can be monitored to detect the boundary of gradual shot cuts [2,17]. If the variance decreases over a certain number of video frames continuously, in principle, this could indicate a possible fade-out. Similarly, a possible fade-in can be identified if the variance increases over a certain number of consecutive frames. However, the change of the variance for some of the dissolves could be very small when the fade-in and the fade-out happen closely. Therefore, based on the work described in Ref. [2], we propose to add an edge detector and thus the nature of gradual changes can be better characterised by the variance change of edge pixels rather than the

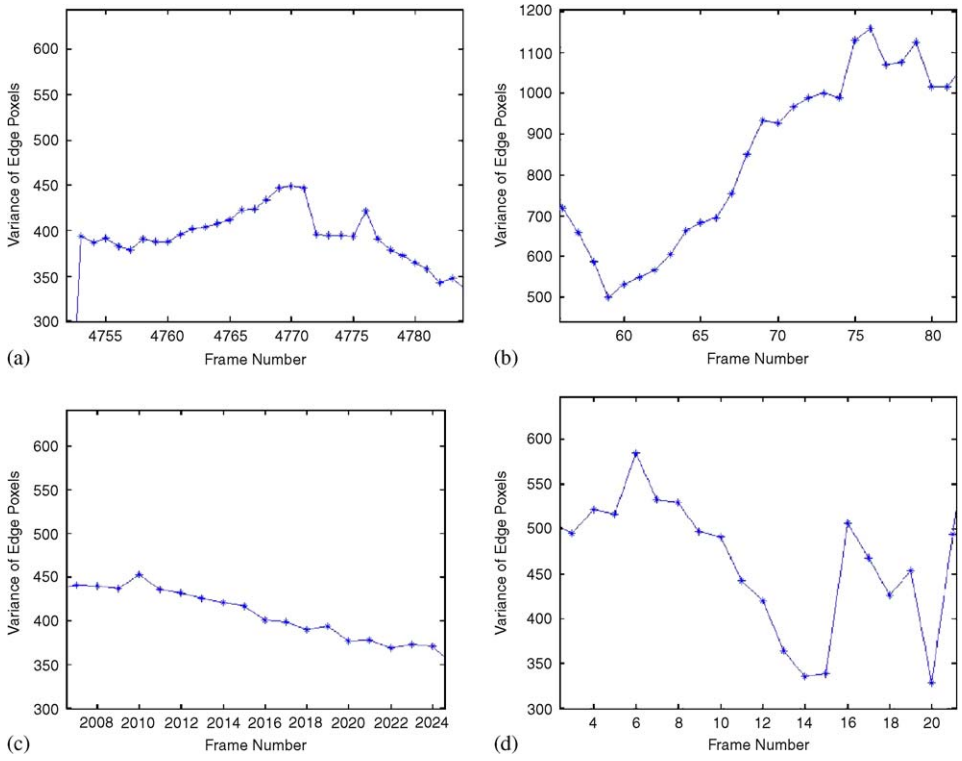

Fig. 5. The variance changing rate of: (a) fake fade-in; (b) real fade-in; (c) fake fade-out; (d) real fade-out.

variance change of all pixels. Specifically, we firstly applied a Canny edge detector [18] to the frame under examination and turn the image into an edge image, where all edge points are identified. For each edge point, we set up a window with 3×3 points to calculate its variance, details of which are given below

$$\sigma_{ij}^2 = \frac{1}{3 \times 3} \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} (Y_{k,l} - \mu_{ij})^2, \tag{12}$$

where σ_{ij}^2 stands for the variance of edge point located at (i, j), Y_{kl} stands for the luminance value at position (k, l) and μ_{ij} stands for its mean value calculated within the window.

To speed up the gradual change detection, we use an average variance to describe the frame under examination

$$\overline{\sigma^2} = \frac{1}{N} \sum_j \sum_i \sigma_{ij}^2, \tag{13}$$

where N is the total number of edge points inside the frame.

As a result, the gradual shot cut detection becomes a problem of examining a sequence of average variance values. Whenever the average variance decreases or increases consecutively over several frames, these frames are set as the fade-out candidate or fade-in candidate. If a fade-out candidate is followed by a fade-in candidate, it is regarded a dissolve candidate. In practice, not all such candidates represent gradual shot cuts. In general, the duration of most gradual shot boundaries is more than 1 s, which means that the duration of such changes is around 30 frames. To ensure the best possible accuracy, we propose two stages of operations and comparisons for such shot cut detection. The first stage is to threshold the length of candidates, or the total number of frames inside the candidates. The second stage is to threshold the changing rate achieved by each candidate over its length. Some examples are illustrated in Fig. 5, where part (a) and (c) show the average changing rate of a fake gradual shot boundary in sequence H of the test data set [1], which is much smaller than the rate of the real gradual shot cut in sequence F and A inside the test set. This is also shown in part (b) and (d).

In summary, the proposed gradual shot cut detection counts the number of consecutive frames, for which their average variances are increasing or decreasing. This process is similar to run-length coding widely adopted in JPEG and MPEG, where a sequence of the same pixels or the same DCT values is counted. Whenever a different value is encountered, the run is broken. Starting from the current frame, gradual shot cuts can be detected by the following operations:

$$\begin{aligned}
&\text{If } (M < 10) \text{ no gradual shot cut;}\\
&\text{Else if } \left(M > 20 \text{ or } \left(\frac{|\overline{\sigma_{last}^2} - \overline{\sigma_{start}^2}|}{M} > T_G \right) \right)\\
&\quad \text{declare a gradual shot cut;}\\
&\text{Else no gradual shot cut;}
\end{aligned} \tag{14}$$

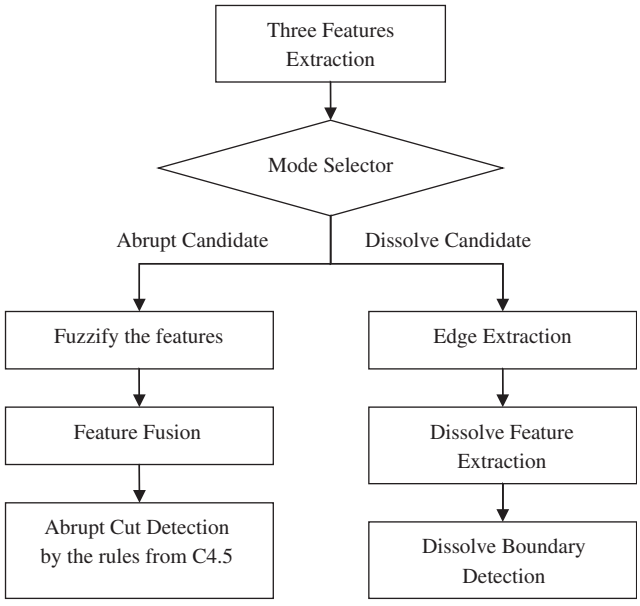

Fig. 6. A flow-chart illustration of the proposed algorithm.

where M defines the number of consecutive count as described above, $\overline{\sigma^2_{last}}$ is the average variance of the last frame inside the sequence of consecutive increase or decrease, $\overline{\sigma^2_{start}}$ is the average variance of the first frame inside the sequence, and TG is a threshold designed for detecting shot cut for those sequences, where the total number of frames is less than 20 but larger than 10. For the TG selection, the criterion is that a gradual shot boundary exists if the average change value is large. Therefore, TG is used to reflect the fact that large number of gradual shot boundaries incurs significant changes in average variance values. To select the value of TG, we adopted a minimum entropy algorithm [15] to optimise the value of TG via the training sequences.

Essentially, the proposed scheme highlights the fact that: (i) when the consecutive increase or decrease of average variance extends over 20 frames, either fade in or fade out exist inside the sequence; (ii) any consecutive increase or decrease less than 10 frames is unlikely to be the gradual shot cut; and (iii) the average variance changing rate over the total number of frames as defined in Eq. (14) represents the key for gradual shot cut detection. Since the variance value in our proposed scheme is only calculated for those edge points, the feature extracted is able to characterise those active pixels and thus provide more accurate information for shot cut detection. In summary, the overall algorithm can be illustrated in Fig. 6.

3. Experimental result

To evaluate the proposed temporal segmentation algorithms, we need to address two important issues to prepare for experiments design. The first issue is the test data set and the second issue is the selection of a benchmark out existing work in relevant areas. In order to ensure that the proposed algorithm can be evaluated by any researcher in comparison with any other algorithms to be developed in the future, we select two open test data sets, one is publicly available via Internet download from the project of video segmentation & annotation in Carleton University. Such selection would enable the proposed algorithm to be assessed on an open platform without repeating the work described in this paper.

Table 1
Description of the video sequences in the test set

Sequences	Frm. num.	Dur. (s)	Transition
A	650	21	8
B	959	38	8
C	1619	53	52
D	2632	105	34
E	536	17	28
F	236	7	3
G	500	16	20
H	5132	205	40
I	479	15	4
J	873	36	88
Total	13,616	513	284

To benchmark the evaluations, we implemented three representative algorithms to compare with our proposed algorithm. These three algorithms include the edge feature tracking technique [9], the Histogram Method 'CutDet' in MOCA project [19], and Pixel-based method with localization [9]. Experiments for all the algorithms have been carried out on the first test data set, including a total of 284 transitions as shown in Table 1.

Among the three benchmarking algorithms, the first benchmark primarily relies on tracking edge features among difference frames to detect gradual shot boundaries. Such technique is similar to motion estimation and compensation used in standard MPEG videos, where inter-frame differences are quantified via edge feature tracking and pruning. Therefore, gradual shot boundaries are detected by monitoring these quantified inter-frame differences. With the second benchmarking algorithm, histograms are primarily used to monitor the differences among adjacent frames and thus shot boundaries are detected via empirically selected thresholds. However, to ensure a fair comparison, we adopted all 13 different thresholds provided in the benchmark [9] to produce the detected shot boundaries and only the best results are compared with our proposed algorithm. For the third benchmark, the algorithm exploits the inter-frame differences among pixel localizations to detect the shot boundaries, where the statistical characteristics that the positions of pixel values remain the same over a shot boundary are used to quantify the inter-frame differences. In our experiments, this algorithm is also run with 13 different thresholds and only the best results are included to benchmark the proposed algorithm.

Table 2
The precision and recall results of four algorithms

Sequences	Proposed algorithm		Edge tracking		Histogram (MOCA)		Pixel localization	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
A	100	100	100	100	100	100	100	100
B	100	100	100	100	100	37.5	82.5	82.5
C	88.1	100	59.5	87	93.6	53.6	76.4	77.8
D	100	100	100	100	100	94.1	100	100
E	93.8	100	93.8	100	95.5	70	86.7	86.7
F	100	100	100	100	100	100	0	0
G	95	100	81	94.4	100	66.7	70.8	99.4
H	90.9	100	89.5	89.5	97.1	89.5	92.7	100
I	100	100	100	100	100	50	100	100
J	92.9	89.7	49.7	89.7	85	39.5	62.3	54

Table 3
The $F1$ values of all the algorithms

Sequences	Proposed algorithm	Edge tracking	Histogram (MOCA)	Pixel localization
A	100	100	100	100
B	100	100	54.5	82.5
C	93.7	70.7	68.2	77.1
D	100	100	97	100
E	96.8	96.8	80.8	86.7
F	100	100	100	0
G	97.4	87.2	80	82.7
H	95.2	89.5	93	96.2
I	100	100	75	100
J	91.3	64	54	57.9
Avg. $F1$	97.4	90.8	80.2	78.3

To compare the performances of all the tested algorithms, we adopt the following measurements [2,8]:

$$Recall = \frac{N_c}{N_c + N_M}, \tag{15}$$

$$Precision = \frac{N_c}{N_c + N_F}, \tag{16}$$

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}, \tag{17}$$

where N_c is the number of correctly detected shot cuts, N_M is the number of missed shot boundary, N_F is the number of falsely detected shot cuts. The higher these ratios are, the better the performance.

Following our implementation and simulation of the proposed algorithm by Matlab 6.5, we present the experimental results with precision and recall figures in Table 2, where all the four algorithms are covered.

From the listed results given in Table 2, it can be seen that the proposed algorithm maintains the same performance for sequence-A and sequence-F, but outperforms MOCA for all other sequences. Compared with edge tracking technique, the proposed algorithm maintains the same performance for sequence A, B, F and I, but outperforms it for all other sequences. To provide an overall comparison across all the video sequences inside the test data set, we list the $F1$ metric, which combine the recall and precision and recall scores, in Table 3, from which it can be seen that the proposed algorithm achieves the best score in comparison with all benchmarking algorithms.

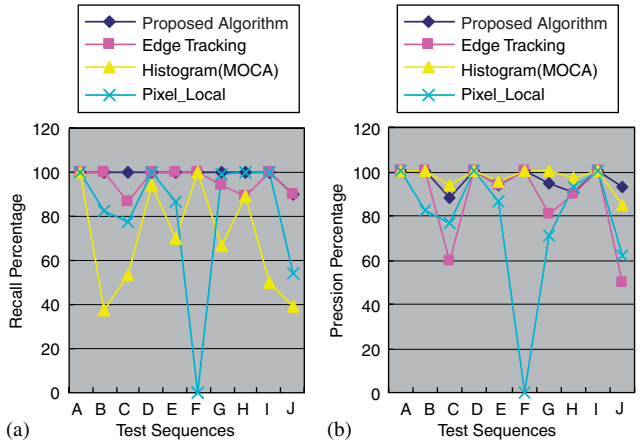

Fig. 7. (a) Comparison of the Recall criteria of the dataset and (b) comparison of the Precision criteria of the dataset.

Further, we summarise the experimental results in terms of precision and recall percentage as graphs in Fig. 7. As shown in part (a), the precision rate of the proposal algorithm achieves better performances than all the three benchmarking algorithms. Similarly, part (b) illustrates that the proposed algorithm achieves similar performances in terms of recall rate in comparison with the three benchmarks.

4. Conclusions

In this paper, we proposed a fuzzy logic approach for temporal segmentation of videos, where a number of features are integrated towards better performances of shot cut detection. These features include color histogram intersection, motion compensation, texture change, and edge variances. Experimental results support that the proposed algorithm is effective in video segmentation benchmarked by three existing algorithms and measured by precision and recall rates.

Acknowledgment

The authors wish to acknowledge the financial support from the EU IST Framework-6 programme under the IP project: Live staging of media events (IST-04-027312).

References

[1] J. Jiang, Y. Weng, Video extraction for fast content access to MPEG compressed videos, IEEE Trans. Circuits, Systems Video Technol. 14 (5) (2004) 595–605.
[2] S. Porter, M. Mirmehdi, B. Thosmas, Temporal video segmentation and classification of edit effects, Image Vision Comput. 21 (2003) 1098–1106.
[3] C. Lo, S. Wang, Video segmentation using a histogram-based fuzzy c-means clustering algorithm, Comput. Standards & Interfaces 23 (2001) 429–438.
[4] R.S. Jadon, S. Chaudhury, K.K. Biswas, A fuzzy theoretic approach for video segmentation using syntactic features, Pattern Recognition Lett. 22 (2001) 1359–1369.
[5] A. Hanjalic, Shot-boundary detection: unraveled and resolved?, IEEE Trans. Circuits Syst. Video Technol. 12 (2) (2002).
[6] C. Huang, B. Liao, A robust scene-change detection method for video segmentation, IEEE Trans. Circuits Syst. Video Technol. 11 (12) (2001).
[7] S.J.F. Guimaraes, M. Couprie, A.D.A. Araujo, N.J. Leite, Video segmentation based on 2D image analysis, Pattern Recognition Lett. 24 (2003) 947–957.
[8] T.Y. Liu, K.T. Lo, X.D. Zhang, J. Feng, A new cut detection algorithm with constant false-alarm ratio for video segmentation, J. Vis. Commun. Image R. 15 (2004) 132–144.
[9] A. Whitehead, P. Bose, R. Laganiere, Feature based cut detection with automatic threshold selection, Proceedings, Content Based Image and Video Retrieval CIVR, 2004.
[10] B. Gunsel, A.M. Ferman, A.M. Tekalp, Temporal video segmentation using unsupervised clustering and semantic object tracking, J. Electron. Imaging 7 (1998) 592–603.
[11] J. Bescos, Real-time shot change detection over online MPEG-2 video, IEEE Trans. Circuits Syst. Video Technol. 14 (4) (2004).
[12] A. Hanjalic, H.J. Zhang, Optimal shot boundary detection based on robust statistical models, Proceedings of the IEEE International Conference on Multimedia Computing and System, vol. 2, 1999, pp. 710–714.
[13] H. Fang, J. Jiang, Predictive-based cross line for fast motion estimation in MPEG-4 videos, Proceedings of IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology, San Jose, USA, January 2004, pp. 175–183.
[14] R. Jain, R. Kasturi, B.G. Schunck, Machine Vision, McGraw-Hill, New York, ISBN:0-07-032018-7.
[15] Timothy J. Ross, Fuzzy Logic with Engineering Applications, Wiley, New York, 2004, 0-470-86074-X.
[16] J.R. Quinlan, C45: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
[17] A. Hampapur, R. Jain, T. Weymouth, Digital video segmentation, Proceedings of the Second ACM International Conference on Multimedia, San Francisco, CA, USA, 1994, pp. 357–364.
[18] J. Canny, A variational approach to edge detection, Proceedings of the AAAI, Washington, 1983, pp. 54–58.
[19] S. Pfeiffer, R. Lienhart, G. Kuhne, W. Effelsberg, The MoCA Project-Movie Content Analysis Research at the University of Mannheim, Informatik'98, 1998, pp. 329–338.