ELSEVIER

Information Sciences 141 (2002) 169–191

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

A mask matching approach for video segmentation on compressed data

Tony C.T. Kuo [a,*], Arbee L.P. Chen [b,1]

[a] *Department of Information Management, Yuan Pei Institute of Science and Technology, Hsinchu, Taiwan 300, ROC*

[b] *Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan 300, ROC*

Received 17 November 1999; received in revised form 23 December 2000; accepted 8 May 2001

Abstract

Video segmentation provides an easy and efficient way for video retrieval and browsing. A frame is detected as a shot change frame if its content is very different from its previous frames. The process of segmenting videos into shots is usually time consuming due to the large number of frames in the videos. In this paper, we propose a new approach for segmenting videos into shots on MPEG coded video data. This approach detects shot changes by computing the shot change probability for each frame. The MPEG coded video data are only partially decoded such that the time for decoding and processing video data frame by frame and pixel by pixel can be avoided. A set of masks for different types of MPEG coded frames (I, P, and B frames) is defined for the computation of the shot change probability.

Experiments based on various parameters are performed to show a 95% of detection rate in average. With further consideration on detecting the dissolve effect, the result is improved to reach an average 98% recall and 96% precision of the detection. A video indexing tool based on this approach was implemented. The results of detected shot changes are kept such that video retrieval and browsing can be provided. © 2002 Elsevier Science Inc. All rights reserved.

Keywords: Video segmentation; Shot change detection; MPEG compressed data; Content-based retrieval; Video browsing

* Corresponding author. Fax: +886-3-5385353.

E-mail addresses: tonykuo@pc.ymit.edu.tw (T.C.T. Kuo), alpchen@cs.nthu.edu.tw (A.L.P. Chen).

[1] Fax: 886-3-5-723694.

PII: S0020-0255(01)00196-7

1. Introduction

Digital videos are widely used in many applications. A simple and efficient mechanism for video retrieval is required. The video contents contain much useful information which is possible to be used for querying. Query-by-content-object in the video is a powerful mechanism for video retrieval [6,11,12,18] in which users specify the spatial and/or temporal relationships of the content objects to form a query. Since content objects are difficult to be recognized from the video, the indexes for content objects and their spatial and temporal relationships are difficult to be automatically constructed.

Retrievals based on the image features of video frames are more efficient and practical in many ways. By measuring the similarities among video frames, a hierarchical cinematic structure [5] such as *shots*, *scenes*, *episodes* of a video can be constructed as an index. Users can browse the hierarchical cinematic structure to retrieve the interesting cinematic units. Moreover, some representative *keyframes* can be extracted as an index. Users can query the videos by an example image, and the system compares the query image and the keyframes to find the possible results. Parsing the video into shots (i.e., shot change detection) is the first step for constructing such indexes for querying video data.

A shot is a sequence of frames which represents a continuous action in time and space. The contents of frames belonging to a shot are similar. Therefore, shot change detection can be performed by the similarity measurement of continuous frames. A number of methods have been proposed for shot change detection [2,9,10]. Previous works for shot change detection can be divided into two classes [9]: processing on the uncompressed domain and on the compressed domain. Template matching and color histogram comparison are two straightforward approaches [4,16,23,24] for the similarity measurement on the uncompressed domain. In template matching approach, the similarity between two frames is measured by comparing the value of each pixel in a frame to the pixel at the same location in the other frame. The sum of the differences of the values is computed. A shot change is detected when the sum exceeds a predefined threshold [16]. Template matching may easily cause misdetections since it is too motion sensitive. The color histogram approach summaries the color distribution of a frame and computes the differences with its adjacent frames [23]. When the difference exceeds a predefined threshold, a shot change is detected. Without considering the spatial distribution of colors, two different frames with the same color histogram will be treated as very similar. The X^2 histogram approach [16,31] reported better results than other histogram or template matching approaches. It uses the square of the difference between two histograms for the strong reflection of the differences. The block matching approach [20,23] can be used to increase the tolerance against motion and noise. In this approach, a frame is divided into some regions and the regions can be compared by the color histogram or template matching approaches.

To improve the quality of detection, misdetections and the loss of detection should be avoided. A misdetection may occur due to a quick variance of video contents, such as the effect caused by an electronic camera flash. A loss of detection may occur due to similar video contents in consecutive shots. Otsuji and Tonomura [17] and Ueda and Yoshizawa [25] considered the situation of large differences on the contents of continuous frames due to fast motion. By using a filter for detecting such a situation, the misdetection on fast motion video frames can be reduced. Shot change detection for special applications [19,21,22,29] such as news programs can achieve better quality. This is because the detecting algorithm can focus on the characteristics of the applications and the domain knowledge can be provided for assisting the detection.

Adjeroh and Lee [1] presented a method to dynamically adjust the threshold for detecting the shot changes. Moreover, a window size (in terms of the number of frames) was defined to avoid false detecting of more than one shot change in a short period of time.

Special camera effects make shot change detection more difficult. The twin-comparison method [31] was proposed to solve it. In this method, two thresholds are required. The higher threshold is for the detection of abrupt shot changes and the lower one for gradual shot changes. Zabih et al. [28] detected and classified production effects, including fades, dissolves, wipes and captions, in video sequences by analyzing the variation of the locations of the intensity edges between frames. Since edge detector is dependent on the relative contract of regions in the frames, rapid changes in overall scene brightness can cause a false positive. Oh et al. [8] computed background difference between frames, and used *background tracking* to handle various camera motions for shot change detection and classification. Gradual changes such as fades and dissolves were considered. All these detection methods on uncompressed video data suffer from the following drawbacks: (1) the processing is time consuming since the size of the uncompressed video data is large, and (2) since video data are often stored in compressed format such as MPEG [7], video data should be decompressed before the processing.

Based on compressed data, Arman et al. [3] proposed an approach by computing the Discrete Cosine Transform (DCT) coefficients for each frame. These DCT coefficients of each frame are stored as a set of vectors. The inner product of the vectors of two continuous frames is computed to measure their similarity. When the similarity degree falls in a range where a shot change cannot be determined, the color histogram comparison approach has to be performed.

Yeo and Liu [26,27] used the DC image sequence of the compressed video data for shot change detection. Their experimental results show over 99% of abrupt shot change detection and 89.5% of gradual shot change detection. Although it is efficient compared with the approaches on full image sequence, decompression for the DC image sequence is required. In MPEG coded video data, a frame can be referenced by or referenced to other frames. The reference ratios can be computed for the similarity measurement among frames. Zhang et al. [30] and Meng et al. [15] considered both the references and DCT coefficients for detecting shot changes. The situations of shot changes occurring on different MPEG coded frames, i.e., I, P, and B frames, were discussed. Shot changes with the dissolve effect were considered.

In this paper, we extend a method we proposed in [13] to detect shot changes for MPEG coded video data. This approach analyzes the references among MPEG coded frames. For each MPEG coded frame (I, P, or B frame), a set of frames required for evaluating a candidate shot change frame is defined as a *mask*. This approach has the advantages over [15,30]: (1) no decompression into DCT coefficients is needed, and (2) the number of the required frames for detecting shot changes is smaller. A function is used to quantize the evaluation results to shot change probabilities such that a shot change can be easily detected. For detecting gradual shot changes such as dissolve, we analyze the variance of the frame differences and define a set of rules for it. Experiments on different types of videos and MPEG formats are performed. The results show this approach is outstanding. A video indexing tool is implemented based on this approach. Users are allowed to verify the detected shot changes using VCR-like functions. Based on the detected shots, a set of keyframes can be extracted for querying by image examples. These keyframes can also allow users to do video browsing.

This paper is organized as follows. In Section 2, the MPEG data format is introduced. The information contained in MPEG coded video data, which can be used for shot change detection is presented. Section 3 presents our approach to detect shot changes. The experimental results are shown and discussed in Section 4. Section 5 presents the advanced consideration for shot change detection. A video indexing tool is presented in Section 6. Section 7 presents the conclusions and future work.

2. Analysis of MPEG compressed data

MPEG is a standard for video compression, which achieves a high compression rate. For multimedia applications, the video data are often stored in MPEG format. Shot change detection algorithms on raw video data are not suitable to be applied on MPEG coded videos. It is more efficient to directly detect the shot changes on MPEG coded videos.

For improving the compression rate, MPEG uses the motion compensation technology to reduce the codes of similar image patterns between adjacent frames. In the following, we introduce the MPEG data format and discuss the information which can be used for shot change detection.

2.1. MPEG data format

In MPEG coding, a frame is divided into marcoblocks. Each macroblock is a 16 by 16 image and is a basic coding unit. A macroblock can be coded by DCT or references to its adjacent frames when it matches the similar image patterns of these frames. A macroblock coded by DCT is named *intra-coded* macroblock. A macroblock referencing to similar image patterns is named *forward-prediction coded*, *backward-prediction coded* or *bidirectional-prediction coded* macroblock when it refers to the image pattern of the preceding frame, subsequent frame, or both preceding and subsequent frames, respectively. A reference to the preceding frame is named *forward reference*, and to the subsequent frame *backward reference*.

By the referencing patterns of macroblocks, there are three types of frames, named *I* frame, *P* frame and *B* frame. All macroblocks in an I frame must be intra-coded macroblocks. That is, the I frame is independently coded. It can be decompressed without referencing to other frames. Macroblocks of the P frame may have forward references to its preceding I or P frame. That is, the macroblock is a forward-prediction coded macroblock when a similar image pattern is found in the preceding I or P frame. The macroblock is intra-coded when a similar image pattern cannot be found in the preceding I or P frame. A B frame may have references to its adjacent I or P frames. Bidirectional references are allowed. The macroblock in a B frame can be a bidirectional-prediction coded, forward-prediction coded, or backward-prediction coded macroblock.

In an MPEG coded video, the number and sequence of I, P, and B frames are pre-determined in the encoding phase. In general, there may have a number of P and B frames between two I frames, and a number of B frames between two P frames or an I and a P frame. An example is shown in Fig. 1 to illustrate the structure of MPEG coded frames. The ratio of the numbers of I, P, and B frames (named *IPB-ratio*) is 1:2:6. An I frame is followed by two P frames and six B frames in the sequence.

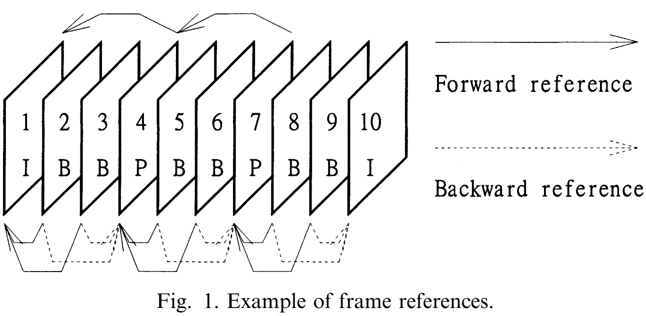

Fig. 1. Example of frame references.

2.2. References between video frames

For the P frame and B frame, macroblocks may reference to adjacent frames. We can compute the number of marcoblocks for each type of references to measure the similarity with the adjacent frames. We define two types of *reference ratios* (RRs) as follows:

1. *Forward reference ratio* (FRR) $= R_f/N$, where R_f is the number of the forward-prediction coded macroblocks of a frame, and N is the number of total macroblocks of the frame.
2. *Backward reference ratio* (BRR) $= R_b/N$, where R_b is the number of the backward-prediction coded macroblocks of a frame, and N is the number of total macroblocks of the frame.

The range of FRR and BRR is between 0 and 1. A P frame has an FRR. A B frame has both an FRR and a BRR. When the FRR is high, it indicates the frame is similar to its preceding frame. When the BRR is high, it indicates the frame is similar to its subsequent frame. An I frame has no FRR nor BRR. Therefore, to measure the similarity between an I frame and its adjacent frames, we have to evaluate the FRR or BRR of these adjacent frames which reference to this I frame.

In a video sequence, the contents of continuous frames are usually similar when the shot is not changed. Therefore, the reference ratios of these frames are high. When a shot change occurs, the contents of the frames are usually different from the preceding frames. The references ratios are then low.

In the following section, we propose an approach to detect shot changes by evaluating the reference ratios of MPEG coded frames. Since only the information of reference ratios of frames has to be computed, this approach is efficient. For example, assume a video sequence contains 10,000 continuous frames and each frame is a 256 by 256 image (i.e., a frame contains 256 macroblocks). To compute the reference ratio of a frame, it only needs to perform 256 add operations. It is therefore more efficient than color histogram-based approaches and the approach of computing the DCT coefficients of frames.

3. Shot change detection

3.1. Shot change occurrence analysis

A shot change often implies a shot with a different content from the previous shot. Therefore, frames of the previous shot may have low BRRs to the current shot. On the other hand, frames of the current shot may have low FRRs to the previous shot, as shown in Fig. 2.

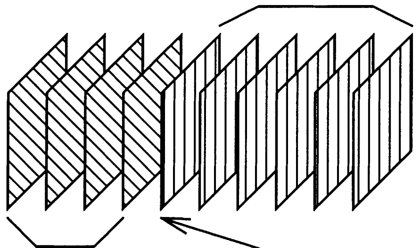

Fig. 2. The varieties of reference ratios when the shot change occurs.

A shot change may occur in any type of frames. In the following, we consider the conditions when shot changes occur at I frames, P frames and B frames, respectively.

(1) *A shot change occurs at an I frame*. Because I frames are encoded independently of other frames, they do not have forward and backward references. What we need to take into account is the B frames between this I frame and the preceding I or P frames. These B frames use this I frame as a backward reference to encode. They cannot easily find similar image patterns from this I frame, so their BRRs must be low. We do not consider the FRRs of these B frames since they are not relevant to this I frame. The B frames between this I frame and the subsequent P frame need not be considered since they are not relevant to whether this I frame is a shot change frame.

(2) *A shot change occurs at a P frame*. The B frames between this P frame and the preceding I or P frame behave the same as in the previous discussion. The difference of this case is that P frames have forward references. Since this P frame is the shot change frame, it cannot easily find similar patterns from the preceding I or P frame. The forward reference will be low.

(3) *A shot change occurs at a B frame*. This B frame itself will have a low FRR. If there exist B frames between this B frame and the preceding I or P frame, their BRRs must be low. If there exist B frames between this B frame and the next I or P frame, their FRRs must be low, too. Besides, if the first non-B frame in the following sequence is a P frame, the FRR of this P frame must be low since it forward references the preceding I or P frames in a different shot.

Consider the MPEG video sequence in Fig. 3. If a shot change occurs at I frame 13, the B frames 11 and 12 will have low BRRs. If a shot change occurs at P frame 10, the BRRs of B frames 8 and 9 are low, and so is the FRR of P frame 10. If B frame 5 is the shot change frame, P frame 7 and B frames 5 and 6 will have low FRRs. If a shot change occurs at B frame 6, the BRR of B frame

I	B	B	P	B	B	P	B	B
1	2	3	4	5	6	7	8	9

P	B	B	I	B	B	P...
10	11	12	13	14	15	16...

Fig. 3. An example of video sequence.

5 is low, and so are the FRRs of P frame 7 and B frame 6. Different positions of a B frame incur different conditions.

3.2. The mask matching approach

From the above analysis, to determine whether a frame has a shot change, the FRRs and/or BRRs of this frame and its adjacent frames have to be examined. In this section, we present a *mask matching* approach to detect possible shot changes.

A *mask* denotes the qualification of a shot in a frame. Different types of frames require different types of masks. A mask consists of two parts. One is the type of this mask. The other is a sequence of *mask frames* which have to be examined. A mask frame M_i can be denoted as follows:

$$M_i = FR,$$

where $F \in \{\mathrm{I},\ \mathrm{P},\ \mathrm{B}\}$, $R \in \{f, b\}$. F denotes the type of this mask frame, and R denotes the RR which should be low (f for FRR and b for BRR). High RRs are not used to detect the occurrences of shot changes.

A mask M can be denoted as:

$$M = \{mask\_type;\ (M_1, M_2, \ldots, M_n)\},$$

where $mask\_type \in \{\mathrm{I},\ \mathrm{P},\ \mathrm{B}\}$, and M_i's are mask frames. The mask frame beginning with an '@' indicates the frame to be tested for an occurrence of shot change. For example, in Fig. 4, there are four masks for the video with the IPB-ratio 1:2:6. Mask M1 is for the I frame and M2 for the P frame. Because of the IPB-ratio 1:2:6, there are two B frames with different conditions: one is preceded by an I or a P frame and followed by a B frame, and the other is preceded by a B frame and followed by an I or a P frame. Therefore, there are two masks, M3 and M4, for the B frame. For M3, it indicates: (1) the current B

M1={I; (Bb, Bb, @I)};

M2={P; (Bb, Bb, @Pf)};

M3={B; (@Bf, Bf , Pf) or (@Bf, Bf, I)};

M4={B; (Bb, @Bf, Pf) or (Bb, @Bf, I)};

Fig. 4. Masks of the video with IPB-ratio 1:2:6.

frame should have a low FRR, (2) its subsequent B frame should have a low BRR, and (3) its subsequent P frame should have a low FRR. If the subsequent frame is an I frame, it can be ignored.

MPEG coded video data usually have a fixed IPB-ratio. The masks of a video can be defined by examining the IPB-ratio of the video. It follows two properties.

Property 1. *The number of masks for a video is equal to* $2 + \beta$, *where* β *is the number of consecutive* B *frames. There are one mask for* I *frames, one for* P *frames and* β *for* B *frames.*

I and P frames, each requires one mask. For B frames, different positions of a B frame cause different conditions on detecting shot changes. There are different positions of a B frame. Therefore, β masks are required for B frames, and $2 + \beta$ masks in total are required for a video.

Property 2. *The number of mask frames of a mask is equal to* $\beta + 1$. *The sequence of the mask frames is the consecutive* B *frames followed by an* I *or* P *frame.*

Consider a sequence of frames, starting and ending with an I or P frame, with in-between B frames, and a shot change occurs in one of these frames. If the shot change occurs in the B frames or the ending I or P frame, all the B frames must be the mask frames since either the forward or backward references of these B frames must be low. If the ending frame is a P frame, then its forward reference must be low, and it must be a mask frame too. If the shot change occurs in the starting I or P frame, the subsequent B frames will have high FRRs, which cannot be used as mask frames. The preceding B frames must have low BRRs and be used as mask frames. If the starting frame is a P frame, it must have low FRR and be used as a mask frame.

The mask matching approach examines the MPEG coded video frame by frame. We use the previous example of Fig. 3 to demonstrate the examination.

To test whether I frame 13 has a shot change, the M1 mask is applied. If I frame 13 has a shot change, the mask frames of M1, i.e., the preceding two B frames, should have low BRRs.

3.3. Shot change probability

In the following, we introduce a function for transforming the result of the mask matching into a *shot change probability*. The probability will be low when the considered frame does not have a shot change. The shot change probability function P is defined as follows:

$$P = 1 - \frac{RR_{M_1}^2 + RR_{M_2}^2 + \cdots + RR_{M_n}^2}{RR_{M_1} + RR_{M_2} + \cdots + RR_{M_n}}, \tag{1}$$

where $M_1, M_2, \ldots, M_n$ are the mask frames, and RR_{M_i} is the RR of mask frame M_i. If $\forall RR_{M_i} = 0, 1 \leqslant i \leqslant n$, P is set to 1.

The shot change probability is between 0 and 1. The larger the value is, the more possible a shot change occurs at the frame. The second term in Eq. (1) is the weighted sum of the RRs of the mask frames. By the weighted sum, if one of the RR is much larger than others, the result of the weighted sum will approach the larger RRs. It makes the effect of the larger RRs outstanding. Therefore, the shot change probability will be low if there exists a mask frame with a high RR. For example, consider the video stream as shown in Fig. 5. The mask used to detect P frame 6 is {P; (Bb, Bb, @Pf)}.

Suppose BRR of B frame 4, BRR of B frame 5 and FRR of P frame 6 are all 0.2. The probability that a shot change occurs at P frame 6 is computed as $(1 - 0.2) = 0.8$. This indicates P frame 6 is highly probable to be a shot change frame. Suppose the BRR of B frame 4 is 0.8, the BRR of B frame 5 is 0.2 and the FRR of P frame 6 is 0.2. The shot change probability can be computed as $(1 - 0.6) = 0.4$ by applying Eq. (1). The probability that a shot change occurs at P frame 6 is low in this case.

Once the shot change probability of a frame is computed, it can be compared with a predefined threshold. If it is larger than the threshold, it is regarded as a *shot change frame*.

...I	B	B	P	B	B	P	B	B	P	B	B	I	...
...0	1	2	3	4	5	6	7	8	9	10	11	12	...

Fig. 5. An example for computing shot change probability.

4. Experiments and analysis

In this section, we examine our approach on a number of MPEG video sources. Since more than one shot change appearing in a short period of time is unlikely, we use a sliding window with size $W = 3$ frames to ensure that at most one shot change in three consecutive frames is possible in the experiments to reduce misdetections due to the effects of camera flash. There are five types of videos captured from TV or VCR. Their contents are described as follows.

1. movie1.mpg: a clip of a movie in which no shot changes have a special effect.
2. movie2.mpg: a clip of a movie in which some shot changes have the special effects of dissolve.
3. news.mpg: a clip of a news program.
4. tutorial.mpg: a clip of a tutorial program.
5. animation.mpg: a clip of an animation program.

All of these video clips are in the format of 352×240, 30 frames/s, and about 60 s of length. The IPB-ratio is 1:4:10.

4.1. Observation of the results

First, we show the result of our experiment on the news program in Fig. 6.

In Fig. 6, the threshold is computed by the average of the *low probability* and the *high probability* where the *high probability* is the average of probabilities higher than or equal to 0.7, and the *low probability* is the average of probabilities lower than 0.7. There are totally 14 shot changes in news.mpg. The result shows that the 14 shot changes can be correctly detected. The frames can be classified into two groups. One is the group with probabilities higher than or equal to 0.71501. This group has 14 frames which are exactly the shot change frames. The other group with probabilities lower than or equal to 0.40532 contains frames which are not shot change frames. We can observe that there is a large gap from 0.40532 to 0.71501. Table 1 shows the top 20 probabilities of the news.mpg.

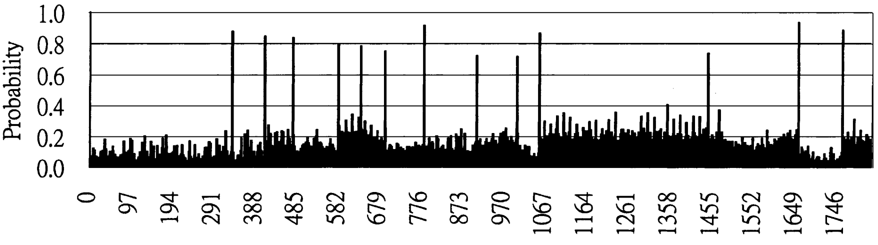

Fig. 6. Result of news.mpg, threshold = 0.46925.

Table 1
Top 20 probabilities of news.mpg

Rank	F. No.	Prob.	Rank	F. No.	Prob.	Rank	F. No.	Prob.	Rank	F. No.	Prob.
1	0	1.000	6	1053	0.867	11	691	0.750	16	1470	0.369
2	1654	0.932	7	409	0.846	12	1445	0.735	17	1230	0.354
3	783	0.915	8	475	0.835	13	907	0.721	18	1305	0.352
4	1757	0.886	9	582	0.793	14	1001	0.715	19	1110	0.348
5	333	0.877	10	635	0.783	15	1350	0.405	20	614	0.340

Therefore, the shot changes can be correctly detected when the threshold is set to be in between 0.715 and 0.405. Because of the large gap generated by our approach, the threshold can be easily defined for correct shot change detection.

Fig. 6 also shows the degree of content changes in different scenes. For example, the contents of frames 0–332, and frames 1680–1722 are the anchor person presenting the news. Since the contents are static (the content objects and the camera are both static), the corresponding probabilities are relatively low. Frames 1050–1133 show a protest activity in the parliament with camera panning. The corresponding probabilities are thus relatively higher.

Fig. 7 presents the result for the tutorial program. Most of the frame contents are the instructor speaking. It is very similar to the condition of the anchor person presenting news in the news program. Therefore, the probabilities of the frames are relatively low. From frames 889 to 991 (see Fig. 7(a)), the contents are switched into a description of a certain topic with words being scrolled up. It causes higher probabilities of the corresponding frames. There are also single frames such as frames 1269 and 1296 (see Fig. 7(b)) which have higher probabilities. The frame contents are a demonstration for the use of a software. Windows are popped up and down in the monitor, which causes the relatively high probabilities.

Fig. 8 shows the result of movie2.mpg. We can observe that from frame 705 to 751, there are a sequence of relatively high probabilities. It is because of the fast moving of the camera. Because of the effect of dissolve, we can also find that there are two to three frames (such as frames 650–652 and 749–751) with the mid-range probabilities (0.3–0.4). Such shot changes are difficult to detect. We will discuss this problem and provide a solution for it in the following section.

4.2. Analysis of the experimental results

4.2.1. Results on the five types of videos

The results of applying the mask matching approach to the five types of video clips are presented in Table 2.

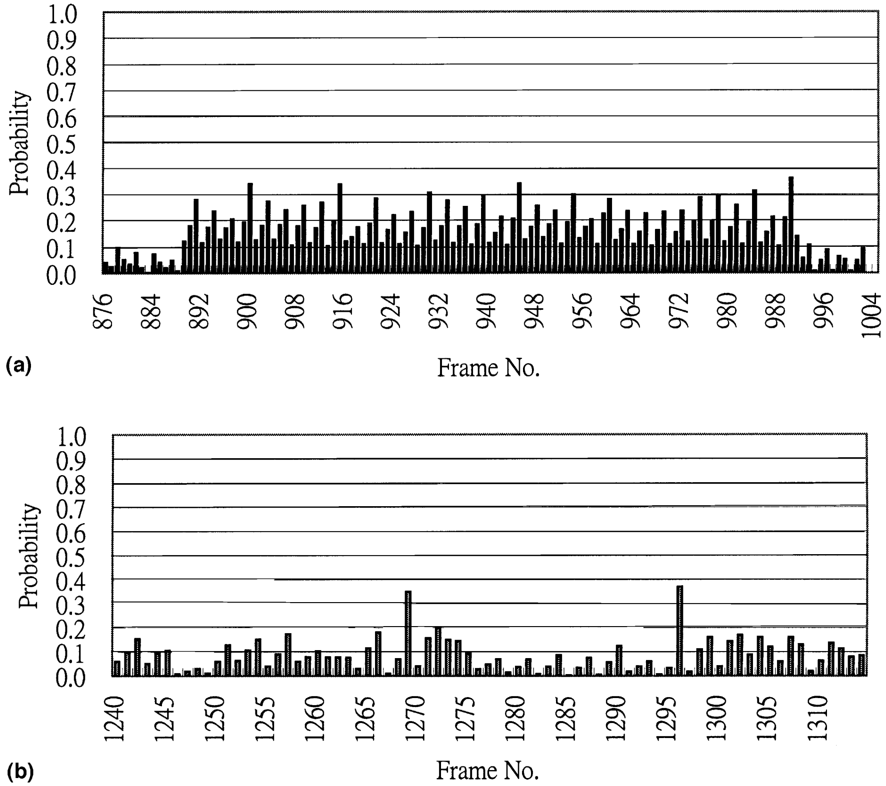

Fig. 7. Results of tutorial.mpg.

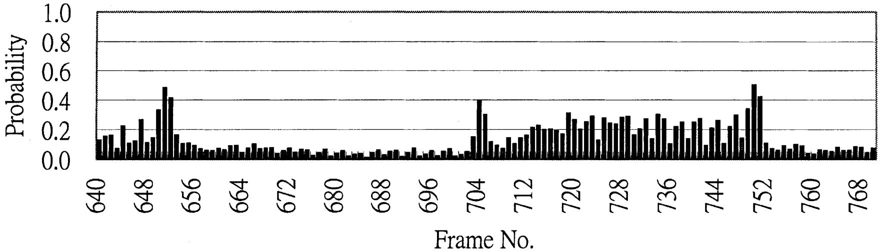

Fig. 8. Result of movie2.mpg.

From Table 2, all the shot change in the news program and tutorial program can be correctly detected. It is because the contents of these two videos are relatively static and the shot change do not have special effects. For movie2, two shot changes are lost due to the effect of dissolve. For movie1, three

Table 2
Detection results of five types of video sources

Video	Shot change	Detected	Correct	Loss	Misdetection
Movie1	6	9	6	0	3
Movie2	19	17	17	2	0
News	14	14	14	0	0
Tutorial	5	5	5	0	0
Animation	17	19	16	1	3

misdetections occur due to fast camera moving. For the animation video, there is a loss of detection due to a long duration of dissolve from frame 64 to frame 105. There are also three misdetections due to fast variant of a large object in the frames.

According to Table 2, we compute the detection rates for the five video clips as shown in Table 3.

In general, the approach achieves high detection rates for various kinds of video sources.

4.2.2. Results on various MPEG formats

Since our approach is based on the analysis of MPEG coded video data, the variation of IPB-ratios and frame sizes also affects the shot change detection.

Table 4 shows the results on different IPB-ratios for news.mpg. The frame size is 320×240. There are four misdetections for IPB-ratio 1:2:3. All these misdetections occur on the I frame. For this IPB-ratio, a shot change on an I frame can only be determined by its preceding B frame (since there is only one preceding B frame for this IPB-ratios). It is insufficient to detect the shot

Table 3
Detection rates of the five video clips

	Correct	Loss	Misdetection
Number	58	3	6
Rates	0.9508	0.0492	0.09836

Table 4
Detection results of various formats of news.mpg

IPB-ratio	Shot change	Detected	Correct	Loss	Misdetection
1:2:3	14	18	14	0	4
1:2:6	14	15	14	0	1
1:4:5	14	14	14	0	0
1:4:10	14	14	14	0	0

Table 5
Detection results of various formats and frame sizes of movie2.mpg

Frame size	IPB-ratio	Shot change	Detected	Correct	Loss	Misdetection
320×240	1:2:3	19	16	16	3	0
	1:2:6	19	17	17	2	0
	1:4:5	19	17	17	2	0
	1:4:10	19	17	17	2	0
160×120	1:2:3	19	14	14	5	0
	1:2:6	19	14	13	6	1
	1:4:5	19	13	13	6	0
	1:4:10	19	14	13	6	1

change. Therefore, the more the I frames are, the more the misdetections may occur.

Table 5 shows the results for various formats with two different frame sizes for movie2.mpg. When the frame size is smaller, more detection errors are observed. Since a smaller frame size implies a lower resolution of the frame, the information for shot change detection becomes less precise.

In the following, we summarize the reasons for the misdetection or loss of detection.

- *The appearance of strong intensity such as flashlight or explosion*. Strong intensity causes the similarities between adjacent frames to be low and a misdetection may occur.
- *Fast moving large objects and fast camera operations*. It causes the content quickly changed. A misdetection may occur.
- *The contents between successive shots are very similar*. Shot changes with special effects such as dissolve and fade are in this condition and it may cause a loss of detection.
- *Special format of IPB-ratio*. For the condition where there exist two or more consecutive I frames and the shot change occurs at the second one, our approach will have a loss of detection.

5. Further consideration for shot change detection

5.1. Detection for shot changes with special effects

In Section 4, we performed experiments on various types of video sources with different IPB-ratios and frame sizes. Our approach reaches a 95% detection rate in average. However, it does not handle special effects such as dissolve well. In this section, we present a further consideration for our approach to handle this problem.

In our approach, we define a function to compute the shot change probability for each frame. If the probability is larger than the predefined threshold, the corresponding frame is regarded as a shot change frame. There are two key issues associated with this approach. One is to form an *ideal distribution* of the probabilities such that the probability of the shot change frames are not overlapped with those of the non-shot change frames. The other issue is to choose a suitable threshold. For an ideal distribution, there exists a threshold by which all the shot changes can be correctly detected. On the other hand, it is impossible to find a threshold for correctly detecting all the shot changes given a non-ideal distribution.

Fig. 9 shows the probability distribution of the experimental result shown in Fig. 6. In the figure, the point with a probability, say 0.3, represents the number of frames with probabilities between 0.2 and 0.3. The probabilities of the non-shot change frames are located in the low probability area. Probabilities of the shot change frames are all higher than 0.7. It is an ideal distribution and we correctly detect all the shot changes.

Fig. 10 presents the probability distribution of movie2.mpg where the frame size is 160×120 and IPB-ratio is 1:4:5. Table 5 shows that there are six

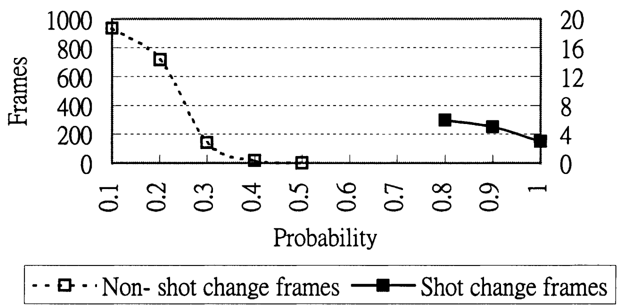

Fig. 9. The probability distribution of news.mpg.

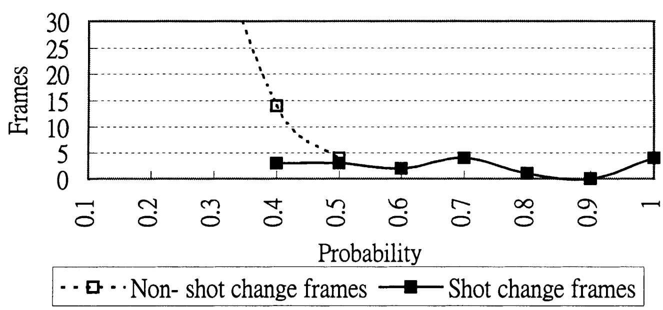

Fig. 10. The probability distribution of movie2.mpg with frame size 160×120, IPB-ratio 1:4:5.

undetected shot changes (when the threshold is 0.503). If we decrease the threshold, the loss of detection can be eliminated. However, it will cause a number of misdetections. The reason for the non-ideal distribution of Fig. 10 is the dissolve effect of movie2.mpg. In the following, we propose a method to deal with this problem.

A dissolve usually incurs two or three mid-range probabilities on the corresponding frames, with at least one relatively high probability. Therefore, we analyze the probabilities of the continuous frames too see weather the following conditions are met: (1) there are n ($n = 2$ or 3) continuous frames with probabilities larger than a given *dissolve threshold D*, and (2) the summation of the probabilities of these n frames is larger than a given *dissolve summation threshold DS*.

We applied the method to detect the shot changes with the dissolve effect for movie2.mpg. The dissolve threshold D is set to 0.3 and the dissolve summation threshold DS is set to $D * n + 0.1$. That is all 19 shot changes are correctly detected. When we remove the corresponding probabilities for these shot changes, an ideal probability distribution can be obtained as shown in Fig. 11. The six previously undetected shot changes were then detected as shot changes with dissolve.

5.2. Experimental results

From the previous experiments and analyzes, we show the effects of our approach and observe the suitable IPB-ratio for the shot change detection. Moreover, the dissolve threshold and the dissolve summation threshold are defined for detecting the shot changes with dissolve. We further performed the detection with the dissolve consideration on another set of MPEG videos. The duration of the videos is between 20 and 30 min (30 frames/s). There are also five types of videos: (1) a soap opera, (2) a news, (3) an animation, (4) a movie,

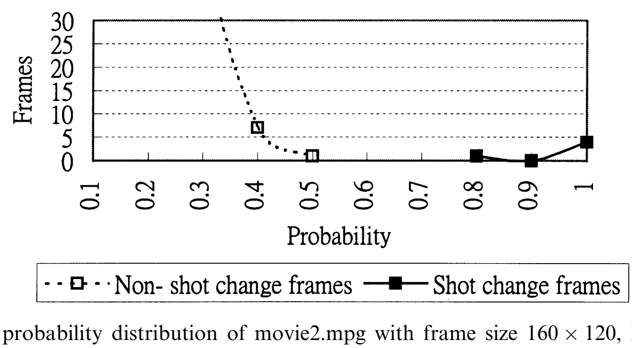

Fig. 11. The probability distribution of movie2.mpg with frame size 160×120, IPB-ratio 1:4:5 after removing the probabilities of the six shot changes with dissolve.

Table 6
The information of the five videos

Video	No. of frames	IPB-ratio	Size	No. of shot changes
Soap opera	49893	1:4:10	352×240	208
News	46524	1:4:10	352×240	380
Animation	33308	1:4:10	352×240	316
Movie	59509	1:4:10	352×240	457
Tutorial	64952	1:4:10	352×240	40

and (5) a tutorial program. The information of these videos is listed in Table 6. These videos are captured by an MPEG online capture card and are encoded into the format with frame size 352×240 and IPB-ratio 1:4:10. The previous experiments show a better result can be obtained in this format. The tutorial program has fewer shot changes than other programs since most of the frame contents are the instructor speaking. The number of shot changes of the animation program is relatively higher than the other programs since the contents of the animation change more often.

The experimental results are shown in Table 7.

$$\text{The precision value of the detection } P = \frac{\mathit{correct\ shot\ changes\ detected\ by\ this\ approach}}{\mathit{number\ of\ detected\ shot\ changes}}.$$

$$\text{The recall value } R = \frac{\mathit{correct\ shot\ changes\ detected\ by\ this\ approach}}{\mathit{number\ of\ shot\ change\ soft\ his\ video}}.$$

The recall and precision values in the experiments are high. The average recall value is 0.98 and the average precision value is 0.96. All 40 shot changes of the tutorial program are correctly detected. The number of misdetection of the animation program is relatively higher due to the fast motion of the large content objects. Fig. 12 shows an example of the fast motion of a content object which may cause a misdetection. A loss of detection may occur due to a long duration of dissolve. The changes of frame contents are gradual such that

Table 7
Precision and recall values of the experimental results

Video	Detected	Correct	Loss	Misdetection	Precision	Recall
Soap opera	210	205	3	5	0.98	0.99
News	386	372	8	14	0.96	0.98
Animation	330	309	7	21	0.94	0.98
Movie	469	452	5	17	0.96	0.99
Tutorial	40	40	0	0	1.00	1.00
Total	1435	1378	23	57	0.96	0.98

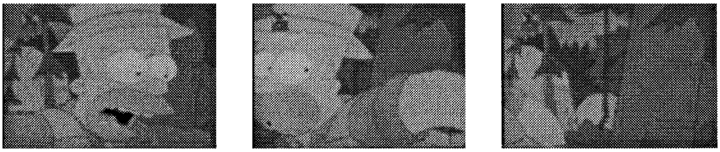

Fig. 12. A fast motion of a large content object.

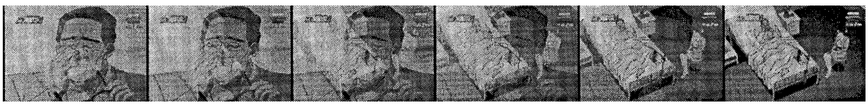

Fig. 13. A frame sequence of the long duration dissolve.

it is difficult to be detected. Fig. 13 illustrates an example of the long duration dissolve effect. The detection algorithm successfully detects most of the shot changes with dissolve by the dissolve threshold and dissolve summation threshold. Some shot changes with dissolve cannot be detected since the dissolves last for more than three frames.

6. A video indexing tool

A video index tool is implemented to illustrate the use of our approach. In this tool, the following functions are provided:

• *Automatic shot change detection*. By performing our algorithm, an MPEG format video program can be automatically parsed. The results are shown in the *shot change window* which locates at the left of the window as shown in Fig. 14.

• *Shot change editing*. A set of VCR-like functions is provided for verifying and editing the results of the detection. The contents of the processed video are shown in the *video play window* which locates at the center of the window as shown in Fig. 14. The *play* button allows users to play the video in the normal speed. The *pause* button can be used to pause the playing and then perform the step by step checking via the *continue* (go to the next frame), *next* (go to the next shot change frame), and *rewind* (go to the first shot change frame). Users can also navigate to a shot change frame by clicking the frame number listed in the shot change frame window and the *go* button below the video play window. A shot change can be deleted or added via the *add* and *delete* buttons.

• *Fast video browsing*. The detected shot change frames are saved as the keyframes of the shots, which can be used as the index for querying by example images and fast video browsing. For fast video browsing, a video is presented by the list of its keyframes. Users can browse the keyframes to select the interesting video shots. Fig. 15 shows the fast browsing of the news video.

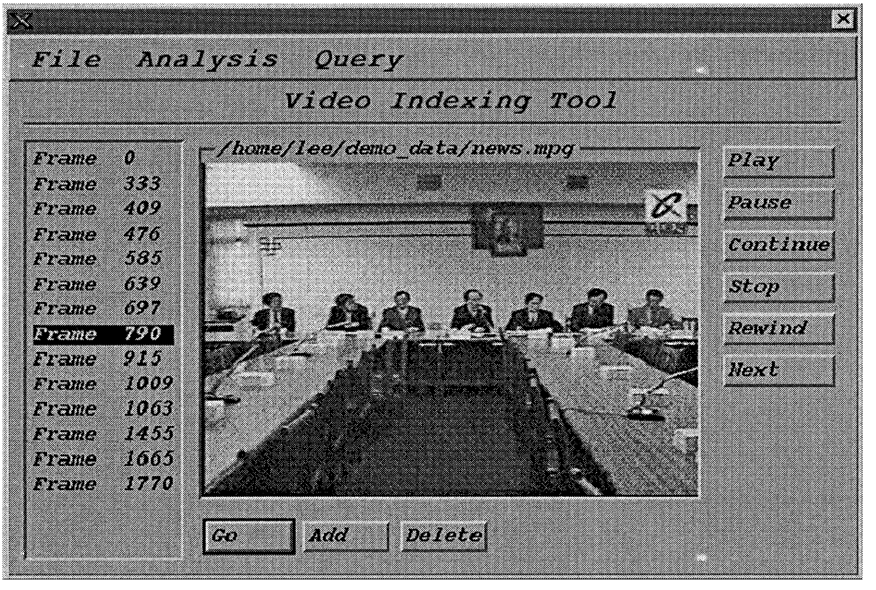

Fig. 14. The video indexing tool.

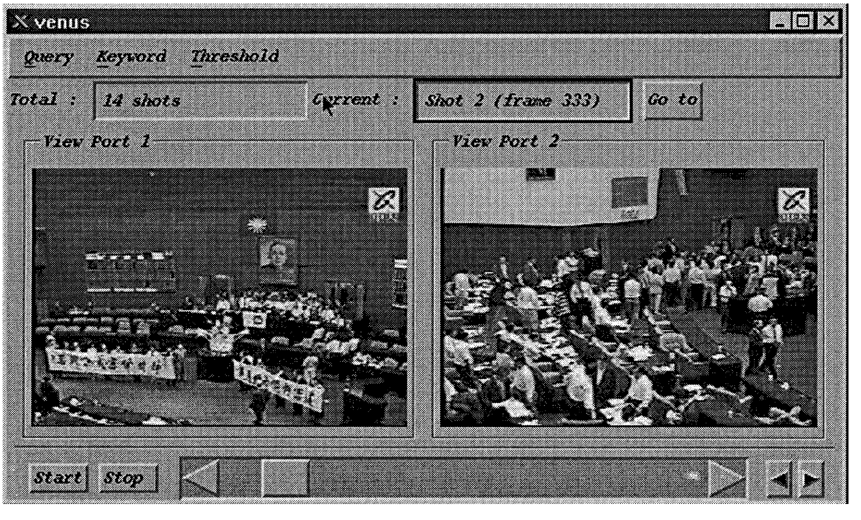

Fig. 15. Fast video browsing.

There are two view ports for showing the video shots. View port 1 shows the *current* video shot while view port 2 shows the next video shot for reference use. The number of video shots is shown in the text box above the view port 1. The shot number of the video shot currently being shown in the view port 1 is listed in the text box above the view port 2. Users may specify the number of video shot for navigating to it by clicking the *goto* button. The slider bar allows users to shift the view ports to the next or previous shots. The current shot can be played by clicking the *start* button.

• *Video query*. The video query provides automatic searching for interesting videos and shots. To retrieve a video, video by video and frame by frame browsing are very inefficient. The interface provides keyword-based searching and keyframe matching for the video retrieval. For keyword-based searching, the keyword can be input for retrieving videos which have the same keywords. For keyframe matching, an example image has to be input. The example image is then matched with the keyframes of all videos. The videos or shots which contain the matched keyframes are shown as results. A DCT based image similarity matching [14] is performed between the example image and the keyframes.

7. Conclusions and future work

In this paper, we present a mask matching approach to detect shot changes on MPEG coded video data. It takes the advantage of the reference ratios between MPEG coded frames to detect shot changes. This approach is efficient because it directly evaluates MPEG coded video data. The experimental results are illustrated for various video sources, IPB-ratios, and frame sizes. It shows that our approach reaches a 95% detection rate in average. Moreover, a method is presented for the elimination of the loss of detection due to the dissolve effect. An experiment with the further consideration on detecting the dissolve effect is performed on longer duration videos (between 20 and 30 min). The result shows an average 98% recall and 96% precision of the detection.

A video indexing tool is implemented to illustrate the use of our approach. In this tool, automatic shot change detection with a friendly video playback interface is provided. Users can perform shot change detection and use the playback functions to verify and adjust the results of the detection. It also provides fast video browsing and query-by-image capabilities.

Based on this approach, an extension for detecting content objects and extracting their motion tracks is currently in progress. By this extension, content objects and their motion tracks can be organized as a video index for supporting content-based queries.

Acknowledgements

This work was partially supported by the Ministry of Education of the Republic of China under Contract No. 89-E-FA04-1-4.

References

[1] D.A. Adjeroh, M.C. Lee, A principled approach to fast partitioning of uncompressed video, in: Proceedings of IEEE Multimedia Data Base Management Systems Workshop, August, 1996, pp. 115–122.

[2] G. Ahanger, T.D.C. Little, A survey of technologies for parsing and indexing digital video, Journal of Visual Communication and Image Representation 7 (1) (1996) 28–34.

[3] F. Arman, A. Hsu, M.-Y. Chiu, Image processing on compressed data for large video databases, in: Proceedings of ACM Multimedia Conference 1993.

[4] T.-S. Chua, L.-Q. Ruan, A video retrieval and sequencing system, ACM Transactions on Information Systems 13 (4) (1995) 373–407.

[5] G. Davenport, T.A. Smith, N. Pincever, Cinmatic primitives for multimedia, IEEE Computer Graphics and Applications (July) (1991) 67–74.

[6] Y.F. Day, S. Pagtas, M. Iino, A. Khokhar, A. Ghafoor, Object-oriented conceptual modeling of video data, in: Proceedings of IEEE Data Engineering Conference, 1995, pp. 401–408.

[7] D.L. Gall, MPEG: A video compression standard for multimedia applications, Communications of ACM 34 (4) (1991) 46–58.

[8] J.-H. Oh, K.A. Hua, N. Liang, A content-based scene change detection and classification technique using background tracking, in: Proceedings of IS&T/SPIE Conference on Multimedia Computing and Networking, January, 2000, pp. 254–265.

[9] F. Idris, S. Panchanathan, Review of image and video indexing techniques, Journal of Visual Communication and Image Representation 8 (2) (1997) 146–166.

[10] H. Jiang et al., Scene change detection techniques for video database systems, in: Multimedia Systems, 1998, pp. 186–195.

[11] T.C.T. Kuo, A.L.P. Chen, Indexing, query interface and query processing for venus: a video database system, in: Proceedings of Cooperative Databases for Advance Applications 1996.

[12] T.C.T. Kuo, A.L.P. Chen, A content-based query language for video databases, in: Proceedings of IEEE Multimedia Computing and Systems, June 1996.

[13] T.C.T. Kuo, Y.B. Lin, A.L.P. Chen, Efficient shot change detection on compressed video data, in: Proceedings of IEEE Multimedia Data Base Management Systems Workshop August, 1996, pp. 101–108.

[14] C.-H. Lin, A.L.P. Chen, T.C.T. Kuo, C.-Y. Tsay, An efficient approach on content-based image retrieval with DCT coefficient indexes, in: Proceedings Symposium on Advanced Database Systems for Integration of Media and User Environments 1999.

[15] J. Meng, Y. Juan, S.-F. Chang, Scene change detection in a MPEG compressed video sequence, in: Proceedings of IS&T/SPIE Symposium, February, vol. 2419, 1995, pp. 14–25.

[16] A. Nagasaka, Y. Tanaka, Automatic video indexing and full-video search for object appearances, IFIP: Visual Database Systems II (1992) 113–127.

[17] K. Otsuji, Y. Tonomura, Projection detecting filter for video cut detection, in: Proceedings of ACM Multimedia Conference, 1993, pp. 251–257.

[18] E. Oomoto, K. Tanaka, OVID: design and implementation of a video-object database system, IEEE Transactions on Knowledge and Data Engineering (August) (1993) 629–643.

[19] M. Philips, W. Wolf, Video segmentation techniques for news, in: Proceedings of SPIE Photonics East Conference 1996.

[20] B. Shahraray, Scene change detection and content-based sampling of video sequences, in: Proceedings of IS&T/SPIE, February 1995.

[21] S.W. Smoliar, H.J. Zhang, Content-based video indexing and retrieval, IEEE Multimedia (1994) 62–72.

[22] D. Swanberg, C.-F. Shu, R. Jain, Knowledge guided parsing in video databases in: Proceedings of SPIE Conference, vol. 1908, pp. 25–36.

[23] Y. Tonomura, Video handling based on structured information for hypermedia systems, in: Proceedings of ACM International Conference on Multimedia Information systems, 1991, pp. 333–344.
[24] Y. Tonomura, S. Abe, Content oriented visual interface using video icons for visual database systems, Journal of Visual Languages and Computing 1 (1990) 183–198.
[25] H. Ueda, T. Miyatake, S. Yoshizawa, Impact: an interactive natural-motion-picture dedicated multimedia authoring system, in: Proceedings of Human Factors in Computing Systems (CHI91) Conference, New Orleans, LA, 1991, pp. 343–354.
[26] B.-L. Yeo, Efficient processing of compressed images and video, Ph.D. Thesis, Princeton University, 1996.
[27] B.-L. Yeo, B. Liu, A unified approach to temporal segmentation of motion JPEG and MPEG compressed videos, in: Proceedings of the International Conference on Multimedia Computing and Systems, May, 1995, pp. 81–88.
[28] R. Zabih, J. Miller, K. Mai, A feature-based algorithm for detecting and classifying production effects, Multimedia Systems (1999) 119–128.
[29] H.J. Zhang et al., Automatic parsing and indexing of news video, Multimedia Systems 2 (6) (1995) 256–266.
[30] H.J. Zhang et al., Video parsing using compressed Data, in: Proceedings of IS&T/SPIE Conference on Image and Video Processing II, 1994, pp. 142–149.
[31] H.J. Zhang et al., Automatic parsing of full-motion video, Multimedia Systems (1993) 10–28.