Efficient Shot Change Detection on Compressed Video Data*

Tony C.T. Kuo, Y.B. Lin and Arbee L.P. Chen

Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
Email: alpchen@cs.nthu.edu.tw

Shu-Chin Chen and C. Y. Ni

Computer and Communication Research Labs
Industrial Technology Research Institute
Hsinchu, Taiwan 300, R.O.C.
Email: sschen@dcs.ccl.itri.org.tw

Abstract

Video segmentation is an elementary work for video index construction. A video sequence is usually decomposed into several basic meaningful segments. In this paper, we propose a new approach to detect shot changes for video segmentation, which is based on the processing of MPEG compressed video data. This approach takes advantage of the information implied in the compressed data. The reference ratios among video frames are analyzed to determine their similarities. A shot change is detected if the similarity degrees of a frame and its adjacent frames are low. A function is used to quantize the results into the shot change probabilities. Considering the motion variations of video contents between frames, a conversion function is designed to increase the correctness of the shot change detection. The experimental results showed the performance of our approach.

1 Introduction

Digital videos are widely used in many applications. Due to the rich information of video data, queries can be specified not only by video titles, video descriptions, and alpha-numeric attributes of video data, but also by the video contents. Therefore, video index construction for supporting powerful query capabilities is an important research issue for video database systems.

A video sequence consists of a sequence of continuous image frames. It can be organized as shot, scene, episode and video [2]. A shot is a basic unit for video browsing. Moreover, the spatial and temporal information of content objects can be extracted from the shot to support more powerful query capabilities. Many research works have been proposed for segmenting video sequences into video shots [1, 8, 4, 6, 7, 5].

A shot consists of a sequence of frames which represents a continuous action in time and space. Therefore, the contents of the frames belonging to the same shot are similar. A shot change is defined as the discontinuity between two shots. The similarity measurement of continuous frames is then used for shot change detection. It can be performed by analyzing the characteristics of the image contents of frames such as color difference, color histogram, and intensity. The most obvious approach is to compute the color distances of two continuous frames pixel by pixel. When the value is larger than a given threshold, the shot change is detected. However, it is motion sensitive. Object motions or camera operations often cause the misdetections. [6] compared color histogram distribution between frames to detect shot changes. [7] detected shot changes by computing the rate of color histogram changes. Some camera operations were also considered. [4] investigated a normalized X^2 test to compare the distance between the color histograms for detecting shot changes. In this approach, a frame is

*This work was partially supported by the Republic of China National Science Council under Contract No. NSC 84-2213-E-007-024.

0-8186-7469-5/96 $05.00 © 1996 IEEE

divided into subframes. For comparing subframe pairs of two continuous frames, the locality of color histogram distribution is then considered. To avoid the effects of camera flash and noise, the largest differences are omitted. [8] presented an approach to automatically compute the threshold of color histogram difference. Color histogram based approaches need to parse the whole video frame by frame and pixel by pixel. It makes the processing inefficient due to the large amount of raw video data. [5] analyzed different approaches for the measurement of frame similarity. A projection detecting filter was proposed to avoid detecting too many shot changes in a short period of time as well as to reduce some of the misdetection.

Based on compressed video data, [1] proposed an approach by computing the Discrete Cosine Transform (DCT) coefficients of two frames. The DCT coefficients of each frame are extracted as a set of vectors. The inner product of the vectors of two continuous frames is computed to measure their similarity. However, the similarity degree can lie in a range where a shot change cannot be determined. In this case, color histogram measurement has to be performed.

In this paper, we propose a mask matching approach to detect shot changes for MPEG [3] compressed video. This approach takes advantage of the information implied in the compressed data. Since a frame can be referenced by or reference to other frames, the reference ratios are computed for the similarity measurement among frames. This approach is more efficient since only the references of frames have to be evaluated. A function is used to quantize the results to shot change probabilities such that a shot change can be easily recognized. We also design a conversion function for the similarity measurement, which can eliminate the effects of fast camera moving and the fast moving of large content objects. The correctness of the shot change detection can thus be improved.

This paper is organized as follows: In section 2, the MPEG data format is introduced. The information implied in MPEG compressed video data, which can be applied for shot change detection is presented. Section 3 presents an efficient approach to detect shot changes for MPEG compressed video. The experimental results are shown in section 4. The final section presents the conclusions and future works.

2 MPEG Compressed Data Analysis

MPEG is a standard for video compression, which achieves a high compression rate. It is popular in many applications. Video data are often stored in MPEG format. Shot change detection algorithms which perform the image processing on raw video data are not suitable for MPEG coded video. Additional processing for decompressing the compressed video into a raw video has to be performed first. Therefore, it will be more efficient to directly detect the shot changes on MPEG compressed video. In order to improve the compression rate, MPEG uses the motion compensation technology to reduce the codes of similar image patterns among adjacent frames. Therefore, the similarity matching is performed in encoding. 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 this section, we introduce the information needed for shot change detection in MPEG coded data. The MPEG coding algorithm uses DCT to compress raw video data. Additionally, it uses block-based motion compensation to reduce temporal redundancy. By motion compensation, codes of similar blocks can be reduced by referencing to the image contents of adjacent frames. The more blocks a frame references, the more similar these two frames are. Therefore, by analyzing the references among coded frames, the similarity can be determined.

In MPEG coding, a frame is divided into marcoblocks. Each macroblock is a 16 by 16 image as 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* macroblocks when it references to the image patterns 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 are defined. 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 can not 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 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 Figure 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.

2.2 References among video frames

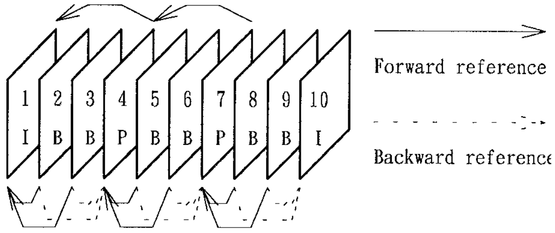

Figure 1: Example of frame references.

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. The RR is regarded as high when it exceeds a given threshold. 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.

In a video sequence, the contents of continuous frames are 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 not similar to the preceding frames anymore. The references ratios are then low.

In the next 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, there is no need to decompressing each coded frames. A large amount of time can thus be saved. For example, a video sequence contains 10,000 continuous frames. Each frame is a 256 by 256 image. That is, a frame contains 256 macroblocks. To compute the reference ratio of a frame, it needs to perform 256 add operations. It is 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 causes the contents different from the previous shot. Therefore, frames of the previous shot may have low BRRs to the next shot. On the other hand, frames of the next shot may have low FRRs to the previous shot, as shown in Figure 2.

A shot change may occur in any type of frames. In the following, we consider the situations 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

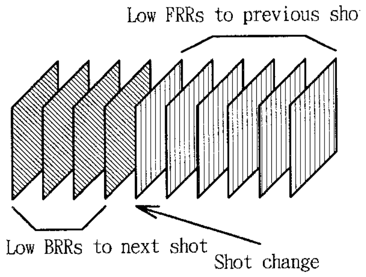

Figure 2: The varieties of reference ratios when the shot change occurs.

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 the shot change detection.

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.

Consider the MPEG video sequence in Figure 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. The situation is different when a shot change occurs at B frame 5 or B frame 6. 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 5 is low, and so are the FRRs of P frame 7 and B frame 6.

3.2 The mask matching approach

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...

Figure 3: An example video sequence.

From the above analysis, to detect 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. This approach examines the MPEG coded video frame by frame. For each video, a set of masks is defined. The RRs of the frames specified in the masks are evaluated. Different types of frames have to be matched with different masks. When a frame is matched with the mask, it is detected as a shot change frame. Since there are I, P, and B frames, the types of masks are I_frame mask, P_frame mask, and B_frame mask, respectively.

A *mask* denotes the qualification for detecting shot changes for frames. It 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:

$$M_i = FR, \text{ where } F \in \{I, P, B\}, R \in \{f, b\}.$$

F denotes the frame type of this mask, 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:

$$M = \{mask\_type; (M_1, M_2, .., M_n)\},$$
where $mask\_type \in \{I, P, B\}$, M_i are mask frames.

To denote the sequence of the frames, the mask frame beginning with an '@' indicates the current frame. For example, in Figure 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, the B frame may have two different situations: 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 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 skipped.

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)};

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

We use the previous example of Figure 3 to demonstrate the examination. To check I frame 13, the M1 mask is applied. By checking the mask frames of M1, the preceding two B frames should have low BRRs when I frame 13 has a shot change. That is, B frame 11 and 12 have low BRRs.

In the mask matching, to determine whether a frame has a low reference ratio, the reference ratio has to be compared with a predefined threshold. Different types of videos may have different thresholds.

4 Implementation

Some experiments are made to verify the mask matching approach. In the experiments, we design a function to transform the results of the mask matching into the *shot change probabilities.* The probability will be low when a frame is similar to its adjacent frames. The function is introduced in the following section.

4.1 Shot change probability

In session 3, our approach takes advantage of the concept of mask matching to detect whether a frame has a shot change or not. To implement this concept, the results of mask matching are quantized to a value which indicates the shot change probability. The shot change probability function P is as follows:

$$P = 1 - \frac{RR_{f_1}^2 + RR_{f_2}^2 + \ldots + RR_{f_n}^2}{RR_{f_1} + RR_{f_2} + \ldots + RR_{f_n}} \qquad (4.1)$$

where $f_1, f_2, \ldots, f_n \in$ the mask frames of the current frame, RR_{f_i} is the corresponding RR of mask frame f_i. If $\forall RR_{f_i} = 0$, $1 \leq i \leq$ 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 (4.1) is the weighted sum of the corresponding RRs of 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 Figure 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 a shot change frame.

We use Figure 5 to illustrate another example. 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 (4.1). The probability that a shot change occurs at P frame 6 is low in this case.

... I	B	B	P	B	B	P	B
... 0	1	2	3	4	5	6	7

B	P	B	B	I	...
8	9	10	11	12	...

Figure 5: An example for computing shot change probability.

After all the shot change probabilities are computed, a threshold is defined to get the final result. As long as the shot change probability of a frame is larger than the threshold, it is regarded as a *shot change frame.*

4.2 Experiments and analysis

In this section, we present our experiment results and discuss the correctness.

By (4.1), the shot change probability of each frame can be computed. The threshold is defined as (F+F')/2, where F is the average probability of the 97.5% of all frames, which have the lowest probabilities, and F' is the average probability of the 2.5% of all frames, which have the highest probabilities. The reason to choose 97.5% is to suppose there are one shot change for every 40 frames in average. These two parameters can be adjusted for different types of videos. If both F and F' are lower than 0.5, the threshold is set to 0.5 for the case without shot changes. In the following, some experimental results are illustrated. The sample video sources can be retrieved from the FTP sites, as shown below.

1. ftp://ftp.ccu.edu.tw/pub3/havefun.stanford.edu/tennis.mpg.
2. ftp://ftp.ccu.edu.tw/pub3/havefun.stanford.edu/us.mpg.
3. ftp://ftp.ccu.edu.tw/pub2/anime/anim/laputa.mpg.gz.
4. ftp://ftp.ccu.edu.tw/pub3/havefun.stanford.edu/2001.mpg

First, we show the shot change probabilities of frames for all the sample video sources and explain them.

- The result of *tennis.mpg* is illustrated in Figure 6: In this video, the image background is very stable. The motion occurs only in a small area. Therefore, the RR of each frame is high, except the shot change frames. There are two shot changes in this MPEG video. They were correctly detected.

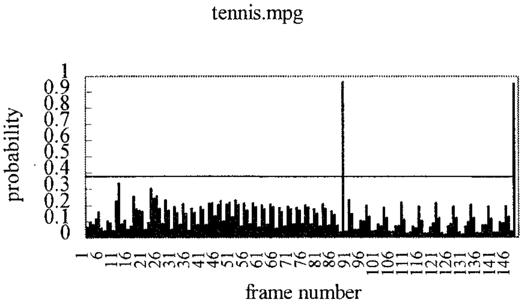

Figure 6: Results of tennis.mpg, threshold=0.382

- Figure 7 is the result of *2001.mpg*. It is an animation. There is no shot change in this video. When the content of the frames moves quickly, the probabilities are high, and the probabilities are low when the contents of the frames are static. The results are correct.

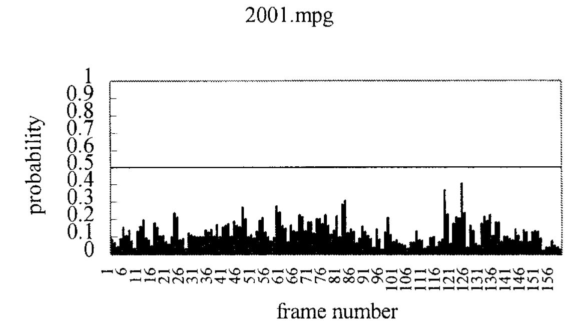

Figure 7: Results of 2001.mpg, threshold=0.5

- Figure 8 is the result of *us.mpg*. Our method loses two shot changes occurring in the beginning 200 frames. The reason is that all the frames in this period are very dark. Therefore, even shot changes occur, the RRs of frames are still high. This situation results in the low shot change probabilities, which fails our approach. Moreover, three misdetections occurs due to the scene of explosion.

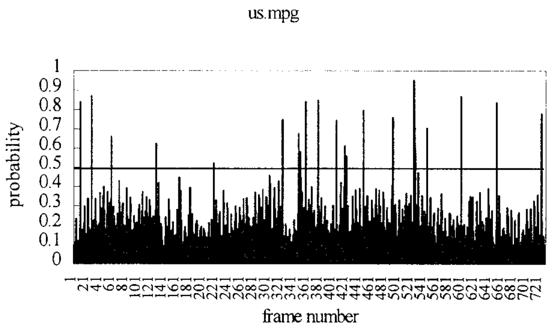

Figure 8: Results of us.mpg, threshold=0.489

- The result of *laputa.mpg* is shown in Figure 9. Frame 22 is a misdetected frame. This is because the objects that occupy much of the image are fast moving.

We have chosen various kinds of videos to demonstrate the performance of our approach. The approach performs well. Some misdetections and loss of detections occur because of the reasons discussed below:

1. The appearance of flashlight and explosion: It causes the intensity to change suddenly. The similarities are then low. These problems are

encountered also in other approaches.

2. The contents between successive shots are very similar: Similarity measurement based approaches compare image contents to detect the shot change. It may cause loss of detections. For example, in our experiment, some of the frames of the us.mpg are very dark. One way to reduce such an effect is to dynamically adjust the threshold.
3. Special format of IPB-ratio: For the situation 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. According to the IPB-ratio, when the number of B frames is much larger than I and P frames, it causes a large number of consecutive B frames in the coded video sequence. The reference ratios will be low for most of the B frames. However, such a case seldom occurs since it will decrease the performance of the MPEG compression.
4. Large object motion and camera operations: It causes the content quickly changed. A misdetection may occur.

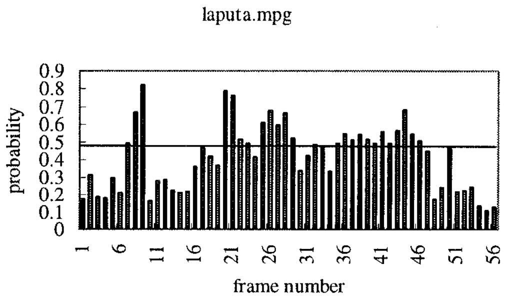

Figure 9: Results of laputa.mpg, threshold=0.612

We further design a conversion function to adjust the shot change probabilities to reduce the effects described above. This function is as follows:

$$F(P_i)=\begin{cases} P_i-MIN(P_k),\ \text{if}\ P_i=MAX(P_k),\ \text{where}\ i-j\leq K\leq i+j. \\ 0,\ \text{if}\ P_i\neq MAX(P_k),\ \text{where}\ i-j\leq K\leq i+j. \end{cases} \tag{4.2}$$

F(P $F(P_i)$ reduces the effects of fast motions. Moreover, by adjusting the value of *j*, the situations of two or more shot changes in a short period of time can be avoid. Figure 10 (a), (b), (c), (d) shows the results of *F(P)*, where *j* is set to 1.

The results show that the number of misdetection of us.mpg due to the scene of explosion is reduced. Moreover, the effects of fast motions in laputa.mpg are eliminated.

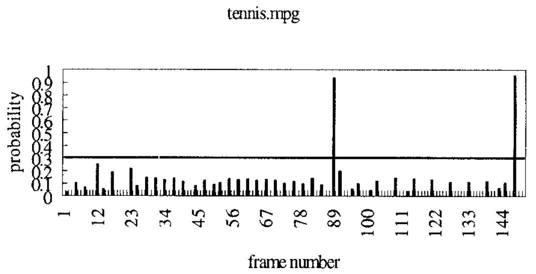

Figure 10 (a): F(P) results of tennis.mpg

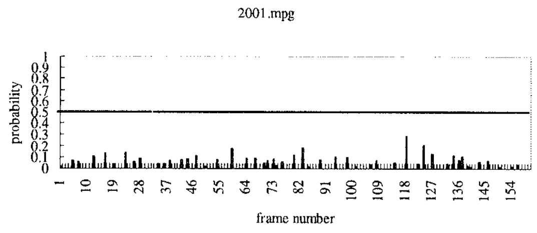

Figure 10 (b): F(P) results of 2001.mpg

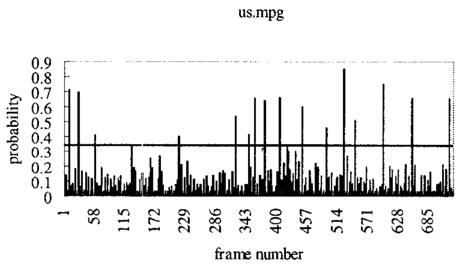

Figure 10 (c): F(P) results of us.mpg

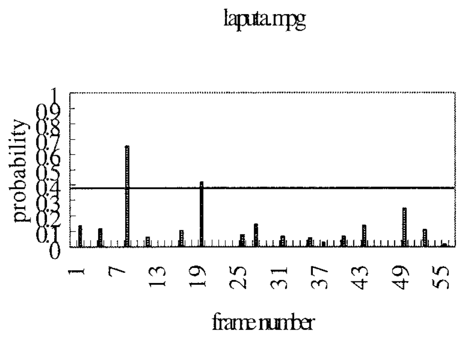

Figure 10 (d): F(P) results of laputa.mpg

5 Conclusion

In this paper, we present a mask matching approach to detect shot changes on MPEG coded video stream. It takes advantage of reference ratio variances of macroblocks between MPEG coded frames to detect shot changes. In this approach, the processing time is saved by directly evaluating MPEG coded data. In the implementation, a function is designed to quantized the results into shot change probability values. The results are illustrated and discussed. Moreover, a conversion function is presented for the elimination of the misdetection and loss of detection of shot changes.

Based on this approach, an extension for content object detection is currently in progress. By content object detection, content objects as well as their spatial and temporal information can be detected. Video indexes for supporting content-based queries can therefore be automatically and efficiently constructed.

References

[1] Farshid Arman, Arding Hsu, and Ming-Yee Chiu. "Image Processing on Compressed Data for Large Video Databases," Proc. of First ACM Int'l Conf. on Multimedia, 1993.

[2] G. Davenport, T.A. Smith, and N. Pincever. "Cinmatic Primitives for Multimedia," IEEE Computer Graphics & Applications, pp. 67-74, July 1991.

[3] D. Le Gall. "MPEG: A video compression standard for multimedia applications," Communications of ACM, 34(4):46-58, April 1991.

[4] A. Nagasaka and Y. Tanaka. "Automatic Video Indexing and Full-video Search for Object Appearances," in 2nd Working Conference on Visual Database Systems, pp. 119-133, Budapest, Hungary, October 1991, IFIP WG 2.6.

[5] Kiyotaka Otsuji and Yoshinobu Tonomura. "Projection Detecting Filter for Video Cut Detection", Proc. of ACM Multimedia, pp. 251-257, 1993.

[6] Y. Tonomura and S. Abe. "Content Oriented Visual Interface Using Video Icons for Visual Database Systems," Journal of Visual Languages and Computing, 1:183-198, 1990.

[7] H. Ueda, T. Miyatake, and S. Yoshizawa. "Impact: An Interactive Natural-motion-picture Dedicated Multimedia Authoring System," In Proc. of Human Factors in Computing Systems (CHI91), pp. 343-354, New Orleans, Louisiana, 1991.

[8] H. Zhang, A. Kankanhalli, and S.W. Smoliar. "Automatic Partitioning of Video," in IEEE Multimedia System Vol. 1, No. 1, pp. 10-28, 1993.