An Efficient Video Segmentation Scheme for MPEG Video Stream using Macroblock Information*

Jongho Nang, Seungwook Hong, Youngin Ihm

Dept. of Computer Science, Sogang University
jhnang@ccs.sogang.ac.kr

ABSTRACT

In this paper we propose and implement an efficient scheme for automatically detecting the abrupt shot changes in a video stream compressed in MPEG video format. In the proposed scheme, the type of each macroblock in a B-frame is compared with the type of corresponding macroblock (*i.e.* the macroblock in the same position) of the previous B-frame. The results of comparisons are accumulated and compared to a threshold in order to decide if a shot change occurs. Since the proposed scheme uses not only information about the type of each macroblock but also its location, it can provide more robust detection capability. Moreover, since the proposed scheme can also detect shot changes in both I- and P-frames based on the information in B-frame, it can detect a changing point more precisely, that is, the granularity of detection is *the frame* in the proposed scheme.

Keywords

Video Segmentation, MPEG Video, Shot-Change Detection, Macroblock Information

1. INTRODUCTION

The first step to build a video database is to segment the video data into elementary units (shots), so that each shot could be indexed and annotated with respect to its contents. Since shots are obviously a fundamental unit of video manipulation such as production, indexing, and representation, there have been a lot of researches to detect the shot boundaries [3]. The researches to detect the shot boundaries of MPEG video data could be classified into two groups according to the information they used to detect the shot boundaries. One is the schemes [1, 7, 8] comparing the DCT coefficients of blocks of adjacent frames, and the other is the schemes [2, 4, 5, 8] using the results of motion compensations of macroblocks of adjacent frames. Since the former schemes compare the DCT coefficients of blocks, the detection schemes should decode the MPEG video data to the block level, whereas since the latter schemes use the type information of macroblocks, they decode the MPEG video data only to macroblock level. It leads the latter schemes have an advantage that the decoding procedure is quite simple and it makes the detecting procedure to be faster. However, since these schemes only count the number (or ratio) of macroblocks in adjacent P- or B-frames that have specific types, there is a possibility to miss the shot boundary between these frames if they have a similar number of macroblocks that have specific types but they belong to different shots.

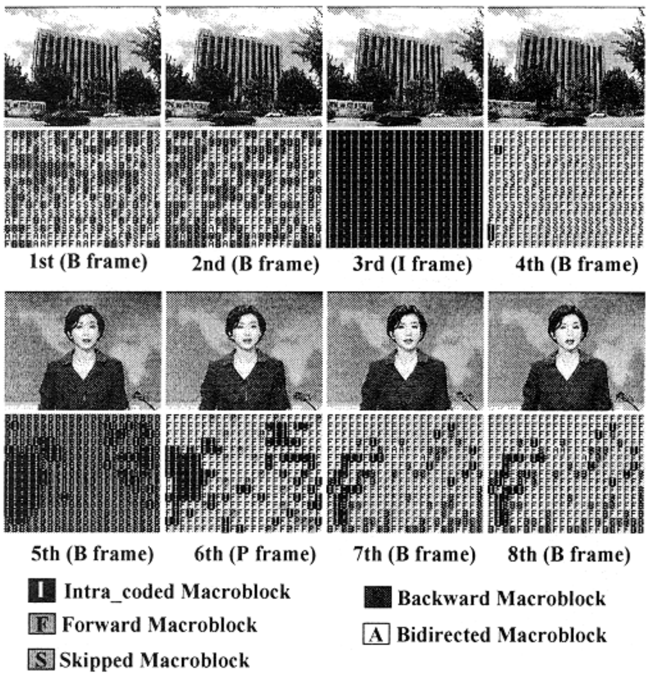
This paper proposes an efficient and robust shot boundary detection scheme for the MPEG video that uses the information at the macroblock level. In the proposed detection scheme, the type of each macroblock in B-frame is compared with the type of corresponding macroblock (*i.e.* the macroblock at the same location) in the adjacent B-frame. The dissimilarity of each macroblock in two adjacent B-frames is accumulated, and a shot change in the video sequence is declared if it exceeds a certain threshold. When computing the dissimilarity of two macroblocks in adjacent B-frames, the anchor frames that B-frame refers should be considered because they could be different. This paper analyzes the cases when two macroblocks are dissimilar, and proposes two dissimilarity tables for comparing two macroblocks while distinguishing the cases when they refer the same anchor frames or not. Several experiments using Korean news and movie video streams show that the accuracy of the proposed method could be over 99%. The proposed indexing scheme could be used to build a digital video library based on MPEG video streams.

2. A NEW SHOT CHANGE DETECTION ALGORITHM

One of the basic principles to code a B-frame is to find the best similar macroblock in past and future reference frames and coded it with a motion vector pointing that macroblock. It means that the pattern of encoded macroblock types of two adjacent B-frames would be similar if they belong to a same shot and refer the same anchor frames. Let us show a real encoding example showing this principle. <Figure 1> shows a real frame and its encoded macroblock type for successive eight frames, in which a shot change occurs between 4^{th} and 5^{th} frames. As shown in this example, the referencing pattern (*i.e.* the macroblock types) of 1^{st} and 2^{nd} frames (or 7^{th} and 8^{th} frames) are similar, whereas those of 4^{th} and 5^{th} frames are different although they refer the same anchor frames (3^{rd} and 6^{th} frames). It comes from the fact that since 4^{th} and 5^{th} frames are encoded by referencing 3^{rd} and 6^{th} frames, respectively, there are a lot of FMBs (Forward Macroblocks) in 4^{th} frame and there are a lot of BMBs (Backward Macroblocks) in 5^{th} frame. This property forces the previous shot

* This work is supported by University Research Program of Ministry of Information & Communication in South Korea.

boundary detection schemes based on the macroblock types to simply count and compare the number of FMBs and BMBs in adjacent B-frames to detect the shot boundary between these two B-frames. Although it is a simple enough detection scheme, it would cause a false detection or missing the boundary because there is a possibility that the number of FMBs and BMBs in the adjacent B-frames are similar although they belong to the different shot. Furthermore, these schemes could only be applied to detect the shot boundary between two adjacent B-frames that refer the same anchor frames. If the anchor frames are different, simply counting the number of FMBs and BMBs would be meaningless.



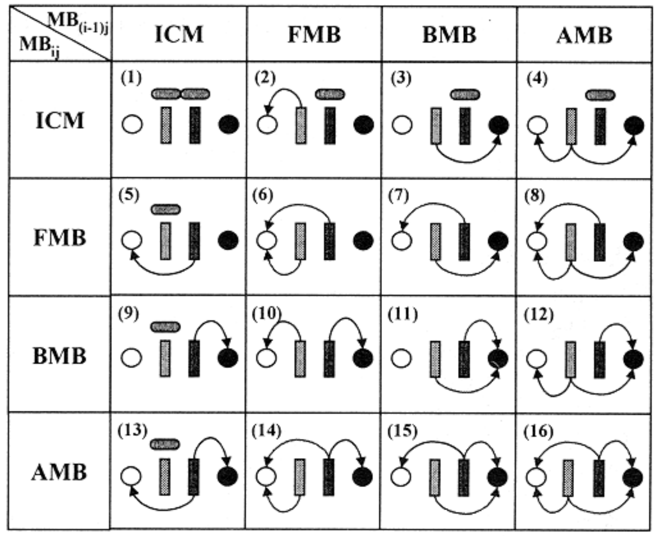
<Figure 1> An Example of B-Frame Encoding

2.1 Detecting Shot Boundary in B-Frames

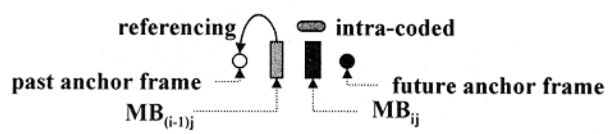
The main reason why previous detection schemes sometimes fail to detect the shot boundary is that they just count the numbers of FMBs and BMBs in B-frame regardless of their positions in the frame. If we consider the position of macroblock that have a specific type when comparing the adjacent B-frames, more robust detection is possible. Let us explain a new shot boundary detection scheme considering not only the type but also the position of specific macroblock in the frame. Note that when comparing the macroblock type of adjacent B-frames, we should consider two cases where they refer the same anchor frames or not.

2.1.1 When Referencing the Same Anchor Frames

Basically, the macroblocks in current B-frame are compared with the macroblocks in previous B-frame. Let F_i be the i-th frame, $MB_{i,j}$ be the j-th macroblock of F_i, and $T(MB_{i,j})$ be the macroblock type of $MB_{i,j}$. $T(MB_{i,j})$ would be ICM (Intra_Coded Macroblock), FMB (Forward Macroblock), BMB (Backward Macroblock), or AMB (Bidirected Macroblock). There are sixteen cases when compares the macroblock $MB_{i,j}$ and $MB_{(i-1),j}$ in adjacent B-frames F_i and F_{i-1}. <Figure 2>-(a) shows these sixteen cases in more detail when two adjacent B-frames refer the same anchor frames using the notations defined in <Figure 2>-(b).



(a) Sixteen Cases when Comparing Two Macroblocks



(b) Notation

<Figure 2> Relationship of Two Macroblocks when Referencing Same Anchor Frames

Let us explain the meaning of <Figure 2> in more detail. If $T(MB_{i,j})$ (or $T(MB_{(i-1),j})$) is an ICM (case of (1), (2), (3), (4), (5), (9), and (13)), it means that there was no similar macroblock in the anchor frames so there might be a change in the two adjacent B-frames at least in the viewpoint of $MB_{i,j}$ or $MB_{(i-1),j}$. Furthermore, if $T(MB_{i,j})$ is a BMB and $T(MB_{(i-1),j})$ is FMB (case of (10)), there might be also a change in the two adjacent B-frames since they refer different anchor frames. Except these eight cases, it would be a little change (or no change) in the adjacent two B-frames since they refer the same anchor frames although the degree of reference would be different. The degree of dissimilarity would be different with respect to the referencing pattern of macroblocks. From this analysis we can obtain the dissimilarity table ($DecisionTbl_1$) when comparing two macroblocks in adjacent two B-frames referencing the same anchor frames as shown in <Table 1>. In this table, a higher value implies that there would be a higher possibility that the adjacent two B-frames are different.

<Table 1> Dissimilarity Table ($DecisionTbl_1$) when Referencing Same Anchor Frames

$T(MB_{i,j})$ \ $T(MB_{(i-1),j})$	ICM	FMB	BMB	AMB
ICM	1.0	1.0	0.5	0.5
FMB	0.7	0.0	0.2	0.1
BMB	1.0	1.0	0.0	0.1
AMB	0.7	0.1	0.1	0.1

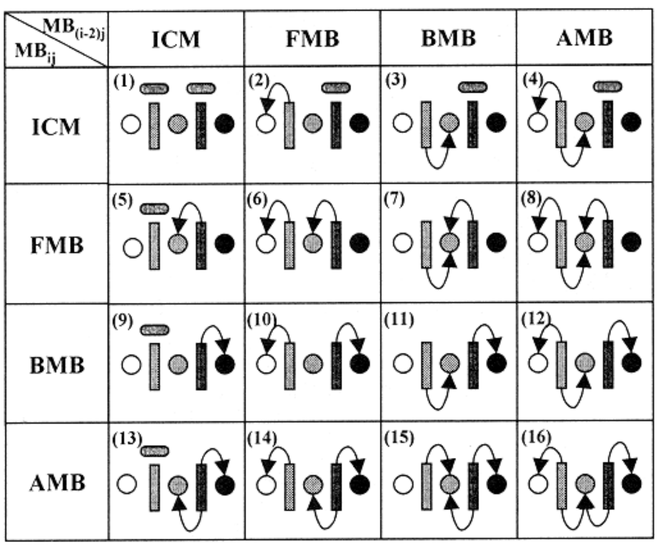
The dissimilarity value is accumulated for all macroblocks $MB_{i,j}$ in frame F_i while comparing with the corresponding macroblocks $MB_{(i-1),j}$ as follows, where TD_i represents the accumulated dissimilarity value of F_i which consists of *TOTAL_MBS* macroblocks.

$$TD_i = \sum_{j=1}^{TOTAL\_MBS} DecisionTbl_1(T(MB_{i,j}), T(MB_{(i-1),j}))$$

If TD_i exceeds a certain threshold TS_1 (it means that there are a lot of dissimilar macroblocks in current B-frame), a shot change between F_{i-1} and F_i could be declared.

2.1.2 When Referencing Different Anchor Frames

When two adjacent B-frames refer different anchor frames (it means that there is an anchor frame (I- or P-frames) between adjacent two B-frames), a different dissimilarity computing scheme is required. Let us explain this case in more detail. If current B-frame is F_i, the previous B-frame would be F_{i-2}, since there is an anchor frame (I- or P-frame) between these adjacent two B-frames. Of course, there are also sixteen cases when comparing the macroblocks in the adjacent two B-frames as shown in <Figure 3>. In this case, if $T(MB_{i,j})$ is FMB or AMB and $T(MB_{(i-2),j})$ is BMB or AMB (case of (7), (8), (15), and (16)), the dissimilarity value for that macroblock would be small since there are references to the common anchor frame. Otherwise, the dissimilarity value would be high since it may be very hard to say these two macroblocks are similar. From this analysis, we can obtain another dissimilarity table ($DecisionTbl_2$) as shown in <Table 2>.



<Figure 3> Relationship of Two Macroblocks when Referencing Different Anchor Frames

<Table 2> Dissimilarity Table ($DecisionTbl_2$) when Referencing Different Anchor Frames

$T(MB_{(i-2),j})$ / $T(MB_{i,j})$	ICM	FMB	BMB	AMB
ICM	1.0	1.0	1.0	1.0
FMB	1.0	0.9	0.0	0.0
BMB	1.0	1.0	0.9	0.9
AMB	1.0	1.0	0.1	0.1

A shot change is declared if the accumulated dissimilarity value for all macroblocks exceeds a certain threshold as in previous case, except that $DecisionTbl_2$ and different threshold value (TS_2) are used.

2.2 Detecting Shot Boundary in Anchor Frame

Since the algorithm proposed in this paper adopts a pair-wise comparison of macroblocks in adjacent two B-frames, basically it can detect the shot boundary between these two B-frames. However, using this technique, a shot boundary between B-frame and adjacent anchor frame (I- or P-frame) could be also detected. As explained in the previous section, if there is an anchor frame (F_{i-1}) between two adjacent B-frames (F_{i-2} and F_i) and the accumulated dissimilarity value of frame F_i, TD_i, exceeds a threshold TS_2, a shot boundary is declared. In this case, a new shot may start either at the B-frame F_i or at the anchor frame F_{i-1}. If there is a large number of FMBs in F_i, it implies that the anchor frame F_{i-1} and current B-frame F_i are similar and a new shot starts at the anchor frame F_{i-1}. On the other hand, if there is a large number BMBs in F_i, it implies that the anchor frame F_{i-1} and current B-frame F_i are dissimilar and a new shot starts at the current B-frame F_i.

3. EXPERIMENTS AND ANALYSIS

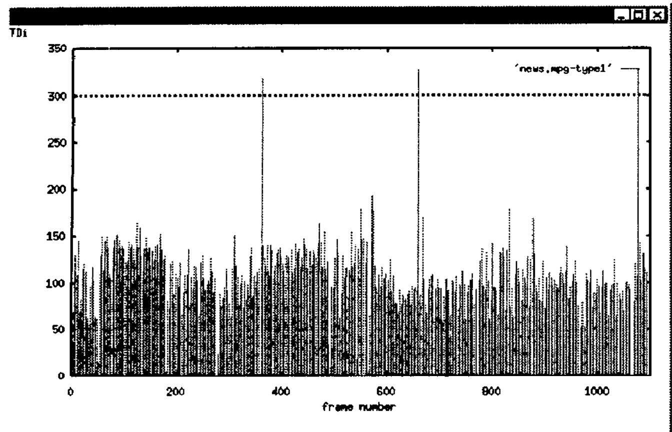
The shot change detection algorithm has been implemented on Windows95 to show the usefulness[1]. The proposed scheme has been experimented with the MPEG video streams[2] (320x240, 30frames/sec.), in which a frame consists of 330 macroblocks. <Figure 4>-(a) and (b) show the accumulated dissimilarity value, TD_i, of B-frame F_i for a sample MPEG Video stream as a function of the frame number i, when two adjacent B-frames refer the same anchor frames and different anchor frames, respectively[3]. As shown in <Figure 4>-(a), when two B-frames refer the same anchor frames, the TD_i value of B-frame is extremely larger when a shot change occurs. On the other hand, as shown in <Figure 4>-(b), the TD_i value of B-frame when adjacent two B-frames refer the different anchor frames is also larger than others, but the difference is not so large. This result comes from the fact that when adjacent two B-frames refer the different anchor frames, the correlation of macroblock types between these two B-frames would be relatively small. It leads the pair-wise type comparison of macroblocks is less effective than the case when two B-frames refer the same anchor frames. This fact forces us to use two different thresholds (TS_1 and TS_2) for a shot change detection. <Table 3> shows the experimental results of proposed shot change detection scheme for several MPEG video streams. The thresholds for the experiments are fixed as TS_1=*297* and TS_2=*310* for all experiments.

---

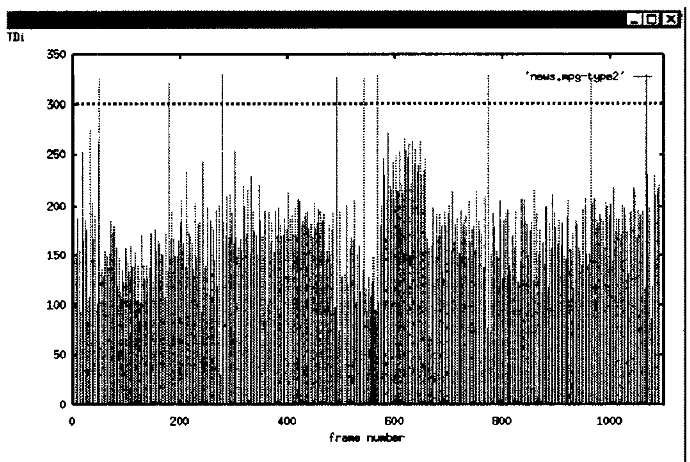
[1] It can be downloaded from http://pllibra.sogang.ac.kr/Scene/SceneChange.exe that is implemented with a MPEG decoder by MPEG Software Simulation Group [6].

[2] The video streams used in the experiments have been generated using a real-time MPEG-1 encoder, RT-5 by VITEC Multimedia Inc..

[3] For an easy reference, we depicted two cases separately.



(a) When adjacent B-frames refer the same anchor frames



(b) When adjacent B-frames refer different anchor frames

<Figure 4> Experimental TD_i for "News.mpg"

<Table 3> Experimental Results

Video Title (# of frames)	Type of Previous Frame	No. of Shot Change	Experimental Results		
			No. of Detecting	Recall	Precision
News (317)	B	10	10	1.0	1.0
	I or P	27	27	1.0	1.0
Volcano (450)	B	6	6	1.0	1.0
	I or P	9	10	1.0	1.0
Die Hard 3 (988)	B	3	3	1.0	1.0
	I or P	5	5	1.0	1.0
Soccer (172)	B	2	2	1.0	1.0
	I or P	8	7	0.86	1.0
Indep. Day (2053)	B	12	12	1.0	1.0
	I or P	24	28	1.0	0.86
Total		106	110	0.99	0.95

As shown in <Table 3>, the average recall value of proposed detection scheme is about 0.99 which implies that almost all shot boundaries could be detected using proposed scheme. The reason why there were some false positives in the experiments is as follows. When an encoder compresses the frame as a B-frame, it searches the anchor frames to find the most similar macroblock to reduce the bit-rate of MPEG stream. If a relatively large number of macroblocks in previous and current B-frames happens to be encoded as referencing the macroblocks in the past and future anchor frames to reduce the bit-rate, respectively, the proposed scheme declares a false positive shot change although there is no shot change at all. However, since this case is rare as shown in our experiments, the proposed detection scheme would work very well in most cases. Furthermore, since this detection scheme adopts a pair-wise comparisons of macroblocks in adjacent B-frames, it works very robustly even in the case where there are a lot of fast frame changes such as a building explosion and a car racing shots as well as some camera operations such as zoom-in and zoom-out upon our experiments.

4. CONCLUDING REMARKS

Indexing the video data compressed in MPEG format is the first step to build a video database. This paper proposes a robust and efficient shot change detection scheme that adopts a pair-wise comparison of macroblocks in the adjacent B-frames. If the accumulated dissimilarity value for all macroblocks exceeds a threshold, it declares a shot change. This algorithm basically compares the adjacent B-frames to detect a shot boundary so that it detects the boundaries between adjacent B-frames, however, it could also detect the boundaries in the anchor frames (I- or P-frames) by comparing the number of FMBs and BMBs in the B-frames. It means that the granularity of detection would be *the frame* (not GOP or anchor frames) in the proposed scheme. Upon the experimental results, we argue that the proposed scheme works very well for almost all kinds of video streams robustly. The proposed shot change detection scheme can be used to build a digital library using MPEG video streams.

5. REFERENCES

[1] F. Arman, A. Hsu and M. Chiu, "Image Processing on Compressed Data for Large Video Databases," *Proceeding of First ACM International Conference on Multimedia*, pp. 267-273, 1993.

[2] N. Gamaz, X. Huang, and S. Panchanathan, "Scene Change Detection in MPEG Domain," *Proceeding of 1998 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 12-17,1998.

[3] H. Jiang, A. Helal, A.K. Elmagarmid, and A. Joshi, "Scene Change Detection Techniques for Video Database Systems," *Multimedia Systems*, Vol. 6, pp. 186-195, 1998.

[4] V. Kobla, D. Doermann and K. I. Lin, "Archiving, Indexing, and Retrieval of Video in the Compressed Domain," *Proceeding of SPIE Conference on Storage and Retrieval for Still Image and Video Databases*, SPIE Vol. 2916, pp. 78-89, 1996.

[5] J. Meng, Y. Juan and S. F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence," *Proceeding of SPIE Conference on Digital Video Compression: Algorithms and Technologies*, SPIE Vol. 2419, pp. 14-25, 1995.

[6] MPEG Software Simulation Group, *MPEG-2 Video Codec*, Available from http://www.mpeg.org/MPEG/ MSSG.

[7] B. L. Yeo and B. Liu, "On the Extraction of DC Sequences from MPEG Compressed Video," *Proceedings of International Conference on Image Processing*, Vol. II, pp. 260-263, 1995.

[8] H. J. Zhang, C. Y. Low and S. W. Smoliar, "Video Parsing and Browsing using Compressed Data," *Multimedia Tools and Applications*, Vol. 1, pp. 89-111, 1995.