TRECVID 2003 Notebook:
- Shot Boundary Determination on MPEG Compressed Domain -

Masaru Sugano, Keiichiro Hoashi, Kazunori Matsumoto, and Yasuyuki Nakajima

KDDI R&D Laboratories Inc.
2-1-15 Ohara, Kamifukuoka, Saitama 356-8502, JAPAN

1. INTRODUCTION

KDDI R&D Laboratories has been participating in the past TREC conferences for text retrieval tasks. In this year we are newly participating Video Retrieval Evaluation at TREC 2003, shot boundary determination task and story segmentation task. The report on our story segmentation task will be presented on site. In shot boundary determination task, we applied our proprietary shot segmentation algorithm originally proposed in [1] and slightly upgraded for this task. In our methods, statistics such as histogram as well as motion vector information from MPEG coded bitstream are used to adaptively determine various types of shot boundaries.

In this notebook, our shot boundary determination methods are described in detail. In section 2, partial MPEG decoding is introduced to have coded data domain shot boundary determination. In section 3, shot boundary determination algorithm is described. In this section, detection of three types of scene changes, abrupt change, wipe and dissolve transitions is discussed. In section 4, experimental results are discussed.

2. PARTIAL MPEG DECODING

DCT DC coefficients give the lowest frequency component of image and at the same time they represent spatially scaled image since DC component is a block averaged value [2]. Furthermore, in I-pictures these coefficients can be directly obtained during VLD (Variable Length Decoding) process without time consuming process such as IDCT. In [2], more than 90% of abrupt scene change can be detected using DCT DC information on I-picture interval. Since I-picture is usually located every 12-15 frames, shot boundary is detected on the next I-picture when shot boundary exists on P- or B-picture. Therefore, GOP position of shot boundary can be detected but the exact position of shot boundary frame cannot be specified. Furthermore, this low temporal resolution may limit detection accuracy. For example, a scene with a very fast panning may change whole scene after one GOP period, which leads to false shot boundaries since current I-picture is completely different from previous one.

In order to enhance temporal resolution of shot boundary determination, coded frame information on P-picture and B-picture is required. DC components in these pictures can be obtained after some manipulation. In P- and B-pictures, although some of macroblocks may be intra coded, most of the coded blocks are inter coded where only prediction error after motion compensation is coded using DCT. In addition, there may be skip blocks and MC no Coded blocks where no DCT coefficient is coded.

DCT DC image can be regarded as reduced size image in 1/8 both horizontally and vertically. Therefore DC components of B- and P-pictures can be obtained using motion compensation in reduced size image domain. There are two ways to obtain DCT DC image for P- and B-pictures. One is to apply motion compensation (MC) using reduced size motion vectors in 1/8. The other is to apply weighted motion compensation reflecting contribution of all the blocks used for motion compensation [9][14]. Figure 1 shows a block diagram of the latter scheme. As for the SNR to the picture with 8×8 pixel average after full decoding, more than 8 dB gain has been obtained in the latter method in our simulation. Subjectively, it is also found that the latter has less visible noise due to motion compensation mismatch. Therefore we use the latter method to obtain DCT DC images for P- and B-pictures.

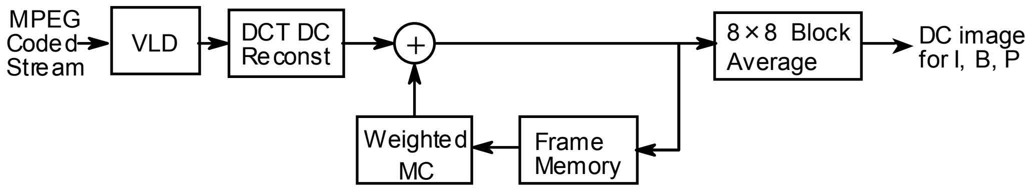


Figure 1. DC image with weighted MC

3. SHOT BOUNDARY DETERMINATION ALGORITHM

3.1 Abrupt shot boundary determination

By incorporating the MC operation mentioned above, P- and B-pictures can be roughly reconstructed so that temporal resolution can be greatly improved. Previously a good deal of research work has been reported on shot boundary determination [2-13]. The major technique includes pixel differences, histogram comparison, edge differences statistical differences, compressed data amount differences, and motion vectors. Although either one of the above techniques can achieve relatively high detection accuracy, each has its own disadvantage [1].

In 1994, we proposed shot boundary determination from I-picture sequence of MPEG coded video [2]. We use both pixel differences and histograms methods to overcome problems when either one of them is used. Here, we extends this approach to detect shot boundaries in one frame unit rather than 10-15 frames unit used in [2].

(1) Pre-processing

In order to exclude undesired false detection mainly due to camera motion and object movement, only frames with high inter-frame difference are picked up for the following shot boundary determination. The inter-frame difference is obtained by:

$$D_n = \sum_{i=0}^{M} \sum_{j=0}^{N} \left| Y_n(i,j) - Y_{n-1}(i,j) \right| \quad (1)$$

, where M and N are total number of 8×8 blocks in a frame for vertical and horizontal direction, respectively. For example, in MPEG-1 with picture size 352×240, M=30 and N=44. $Y_n(i,j)$ is the luminance block average at block (i,j) in the n th frame. Since DCT DC component of each 8×8 block can be obtained from section 2, $Y_n(i,j)$ for each frame is directly given from this value. Then the following equation is used as pre-processing:

$$D_n > Th\_pre \quad (2)$$

Only those frames which satisfy the above condition are further investigated in abrupt shot boundary detection in the following section.

(2) Shot boundary determination using luminance and chrominance change

Both luminance and chrominance characteristics dramatically change at shot boundaries. Therefore ordinary scene change can be detected when both the luminance and chrominance information greatly change. We use temporal peak detection of both inter-frame luminance difference and chrominance histogram correlation [2]. A frame is declared as a shot boundary when:

$$\alpha D_n > D_{n-1}, D_{n+1} \text{ and } \rho_n > \rho_{n-1}, \rho_{n+1} \quad (3)$$

Here, α is a weighting factor for the detection. ρ_n is chrominance histogram correlation obtained by:

$$\rho_n = \frac{\sum_{k,l} H_{n,k,l} H_{n-1,k,l}}{\left(\sum_{k,l} {H_{n,k,l}}^2 \sum_{k,l} {H_{n-1,k,l}}^2 \right)^{1/2}} \qquad (4)$$

where $H_{n,k,l}$ is a chrominance histogram matrix. The histogram is obtained classifying DC chrominance Cb and Cr data in a frame into *hc* classes for each chrominance component. Then two dimensional $hc \times hc$ histogram matrix in the *n* th frame $H_{n,k,l}$ ($k, l = 0, 1, 2... hc-1$) is obtained.

When shot boundary exists on scenes with large motion, it is very difficult to find temporal peak using frame difference since frame difference may be very large all the way due to motion so that Eq. (3) may not detect such shot boundaries. Therefore, only chrominance correlation is used to detect such shot boundary for those frames which don't satisfy Eq. (3).

$$\rho_n > Th\_ac \qquad (5)$$

, where Th_ac is a threshold value for determination of temporal peak in ρ_n.

When consecutive two shots are different only in camera angle, color histogram will be about the same and thus it is difficult to detect shot boundary by the above conditions such as Eq. (3) and (5). However, since pixel difference usually has a very large peak at these shot boundaries, peak detection of luminance difference can be applied. When either of the following equation is satisfied for those frames which are not declared as scene change in the above process, the frame is declared as shot boundary.

$$\beta D_n > D_{n-1}, D_{n+1} \qquad (6)$$
$$D_n - Th\_ad > D_{n-1}, D_{n+1} \qquad (7)$$

, where β and Th_ad are a weighting factor and a threshold value for determination of temporal peak in D_n, respectively. Basically, Eq. (6) will detect shot boundary in similar scenes. However, Eq. (7) is also used for such cases when motion is involved since all of the inter-frame differences are kept relatively high and the ratio of D_n to D_{n-1} or D_{n+1} may not be significantly high enough to find the shot boundary using Eq. (6).

3.2 Dissolve shot boundary determination

3.2.1 Basic detection algorithm of dissolve and fade in/out

In gradual transition caused by dissolve and fade in/out, two different shots are usually synthesized in the course of transition. For example, in dissolve transition, gradual change from one shot to another occurs with simultaneous decrease and increase of intensities of preceding and following shots. Since both shots are synthesized during transition, activity of the each frame shows U-shape curve surrounded by flat shoulders when dissolve occurs [13]. In the case of fade in/out, activity curve shows monotonous increase/decrease. The frame activity for *n* th frame FA_n can be described as:

$$FA_n = \sum_{i=0}^{M} \sum_{j=0}^{N} \left\langle Y_n(i,j)^2 - \left\langle Y_n(i,j) \right\rangle^2 \right\rangle \qquad (8)$$

In [13], positive peak before dissolve and negative peak during dissolve are used to detect U-shape variance curve. It assumes that only single pair of positive and negative peaks with large peak to peak difference exists during dissolve period. However, in the actual video sequences, it rarely shows these shapes due to motion and local fluctuations. Figure 2(a) shows one of variance curve using TV news sequence. It can be seen from the figure that many local peaks appear around dissolve. Therefore it is difficult to find real positive and negative peaks of dissolve region even if the variance shows U-shape curve. Furthermore, peak to peak difference may not always be large due to picture flatness or motion. For example, second dissolve near 6430 th frame in the figure shows very small variance valley.

In order to detect these shapes avoiding false detection, we have applied filtering process as noise reduction for the DCT DC activity data. Since dissolve and fade processes take long duration, temporal filtering with long tap is suitable to absorb spontaneous fluctuations and examine long duration variation. As a temporal filtering, we use moving average of activities MA_n for a period of frames *VF* which includes current and previous (*VF -1*) frames:

$$MA_n = \frac{1}{VF} \sum_{t=n}^{n-VF+1} FA_t \qquad (9)$$

After temporal filtering, temporal peak or monotonous increase/decrease can be detected. However, since duration of dissolve and fade depend on how the shots are edited, such a technique as simple peak detection may result in false detection. Furthermore, very flat U-shape curve will be expected when dissolve transition occurs in between relatively flat shots. Therefore it is necessary to contrast these curves with others. We use first order derivative of the filtered activity DA_n in order to detect these curves. It is obtained as:

$$DA_n = MA_n - MA_{n-1} \qquad (10)$$

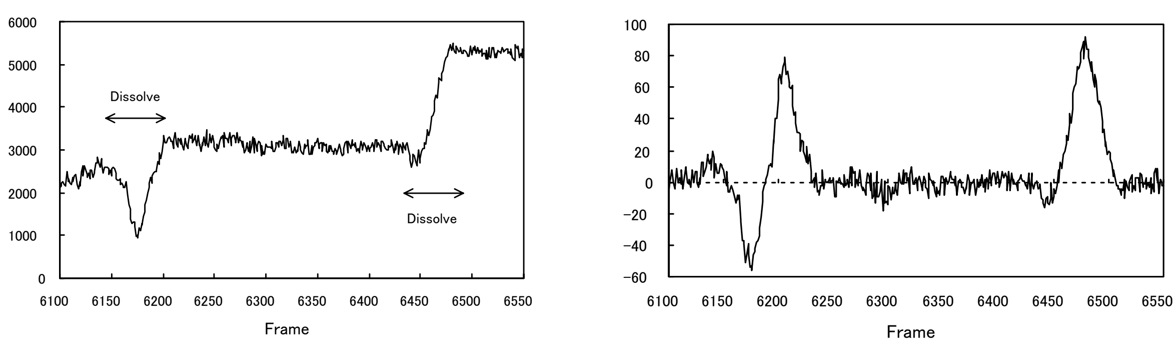
Figure 2(b) shows the derivative curve of the same test sequence as Figure 2(a). As can be seen from the figure, negative and positive region can be clearly seen in dissolve region, and the original method [1] utilizes this characteristics. However, in TRECVID data, the derivative curve tends to be negative in our preliminary experiment. Therefore dissolve period can be found when the derivative curve continuously takes negative values during a certain period. Fade in/out period can also be found when only positive/negative period is detected. In order to exclude undesired detection in such scenes as motion, we use chrominance correlation between *n* th and (*n-dd*) th frames to confirm that the region is a shot boundary candidate. Therefore dissolve sequence candidates can be detected using the following equations.

$$DA_n < -Th\_dis1 \text{ and } \rho_{n,\, n-dd} < Th\_dis2 \qquad (11)$$

Between *n* th and (*n-dd*) th frame, if the number of frames satisfying Eq. (11) is larger than *Th_dis3*, dissolve transition is determined in this period. In order to avoid detecting motion scenes, the following equation should be considered:

$$k < Th\_dis4 \qquad (12)$$

, where *k* is number of non-intra blocks. Dissolve detection is carried out for those frames which are determined as non abrupt scene change in the previous section.



(a) Variance　　　(b) 1st order derivative of variance after moving average

Figure 2. Dissolve region characteristics

Although the above equations can detect most of the dissolving, there are two problems in terms of detection accuracy. One is that it is difficult to detect those dissolve transitions in similar color shots or in shots with large flat areas, since conditions in Eq. (11) assumes that two shots have different color distributions with non-flat regions. The other is that it may also detect panning or motion scenes as dissolving since these scenes may have similar activity curve in such cases when scenes with large flat object appear during panning. In the following two sections, countermeasures for these errors in the detection are described.

3.2.2 Dissolve determination in shots with flat areas

As for the first problem in the previous section, it is necessary to have more detailed observation of activity variation for those frames which are determined as non-dissolve in Eq. (11). Since negative period is continuing in dissolve as described earlier, closer investigation of these characteristics can be carried out by the following equation:

$$\forall i \in n\_p, \quad DA_i > -Th\_ea \quad \& \quad \sum_{i \in n\_p} DA_i > -Th\_sa \tag{13}$$

where $n\_p = \langle n-dm, n-dm-1, \cdots, n-dm-dh \rangle$ (14)

Here, detection of negative period is carried out by observing the derivative of activity DA_i and sum of DA_i in a period n_p are greater than threshold values $-Th\_ea$ and $-Th\_sa$, respectively as shown in Eq. (13).

Although the above equations can detect dissolving which has relatively small variations in activity during dissolving, they may also detect such scenes as very slow panning since both characteristics will show relatively flat activity curve. In order to distinguish dissolve from such non-dissolve scenes, we have also used prediction error information obtained from coded bitstream. In the scenes with very small motion, most of the blocks are successfully motion compensated and prediction error in the MC block is relatively small. On the other hand, in the case of dissolve, inter-frame difference may be as small as that of small motion case. However, prediction error is large in dissolve transition since motion compensation is usually ineffective in the course of synthesizing of two shots. The normalized prediction error NPE_n in n th frame is obtained as following equation.

$$NPE_n = \left(\sum_{i,j \in non\_intra}^{k} DC_n(i,j) \right) MN/k \tag{15}$$

Here, $DC_n(i,j)$ is DC component of prediction error which is directly obtained from (0,0) element of DCT coefficients. MN is total number of blocks in a frame and k is number of non-intra blocks. Since NPE_n in the dissolve period has a large value, the numbers of frames which have a large prediction error around dissolving are compared with threshold values which are shown in the following equations.

$$\sum_{l \in bd\_p} PE_{fl} > Th\_dbd \quad \& \quad \sum_{l \in dd\_p} PE_{fl} > Th\_ddd \tag{16}$$

where $dd\_p = \langle n, n-1, \cdots, n-df \rangle, \quad bd\_p = \langle n-df, n-df-1, \cdots, n-df-db \rangle$ (17)

Here PE_{fl} is "1" if normalized prediction error NPE_l in l th frame is larger than the threshold value Th_pe. Using predetermined values of df and db, dd_p and bd_p are periods during dissolve and before the dissolve, respectively. Therefore, dissolve in flat region is detected when Eq. (13) and Eq. (16) are satisfied.

3.2.3 Exclusion of panning/motion scenes in dissolve determination

Although panning/motion scenes when flat object is appeared have very similar activity curves to normal dissolve case as described earlier, panning/motion scenes have different characteristics concerning motion. In panning/motion scenes, most of all the macroblocks will be motion compensated and inter-frame difference will be very large, whereas in the case of dissolving the number of motion compensated block is usually small and inter-frame difference is also small. Therefore, we use the number of motion compensated blocks and inter-frame difference in order to exclude these false scenes from detected dissolve frames. As for number of motion compensated blocks, the following condition is applied since it has a large value in the case of panning and motion scenes.

$$MVC, PMVC > Th\_mvc \qquad (18)$$

where *MVC* and *PMVC* are numbers of motion compensated blocks in the most recent P-picture and its previous P-picture, respectively. In order to exclude motion compensated blocks which are not real motion involved, only blocks which have motion vector size larger than threshold value *Th_mv* are counted in Eq. (18). We have also applied several conditions described in the following since above equation may also exclude dissolve in the panning/ motion scenes.

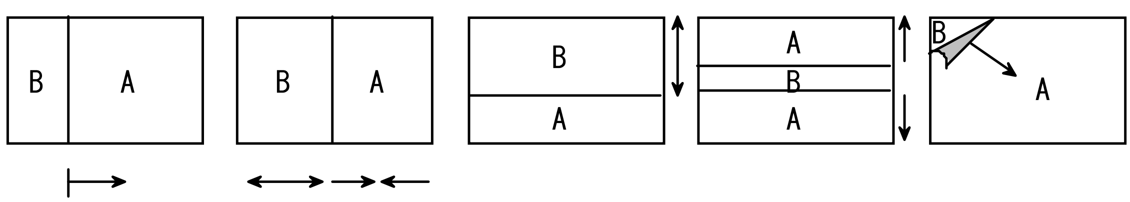
Motion scenes are characterized as large inter-frame difference whereas panning scenes are characterized that most of all the motion vectors are in the same direction. Therefore the following conditions are used.

$$D_n, D_{n-1} > Th\_bm \qquad (19)$$
$$D_n > Th\_mm \;,\; |<mvx>| \text{ or } |<mvy>| > Th\_am \qquad (20)$$

Eq. (19) corresponds to motion scenes where consecutive motion is detected using inter-frame difference. Eq. (20) corresponds to panning where frame average horizontal/vertical motion vectors are compared with threshold value. Therefore if either Eq. (19) or Eq. (20) along with Eq. (18) are satisfied, the frame is declared as panning/motion scenes.

3.3 Wipe shot boundary determination

A wipe is scene transition where a new shot appears and at the same time current shot disappears changing their spatial positions. Although wipe in TV program can be found mostly in TV news and may not be found in other programs like commercials and film, wipe tends to be recognized more easily than other scene changes due to its rather long transition duration and therefore it usually plays an important semantic role in the program.



(a) (b) (c) (d) (e)

Figure 3. Wipe models

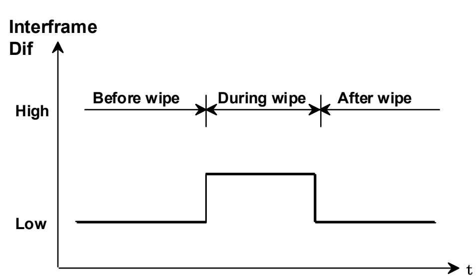


Figure 4. Wipe model using inter-frame difference

Several examples of wipe transitions are depicted in the Figure 3. Figure 3(a) shows most typical wipe where a new shot ***B*** translated to the right direction over the current shot ***A***. Figure 3(b) is modified version of Figure 3(a) where shot ***B*** expands horizontally whereas shot ***A*** shrinks accordingly. Figure 3(c) and (d) are modifications of wipe model in Figure 3(d) where a new shot ***B*** expands over shot ***A*** in vertical direction. Figure 3(e) is a page-turn type wipe where a new shot ***B*** appears as if a current *page **A*** is turned. As can be seen from these patterns in the figure, it is difficult to use motion information for wipe determination since various kinds of motion models are required for corresponding wipe patterns. Although moving patterns are completely different depending on wipe models, spatial positions of two shots are always moving during wipe periods in any types of wipe and each shot before/after wipe period is usually still and stable unless large motion is involved in shots. Furthermore, moving speed of shots in wipe is slow and steady during wipe period. Therefore, when inter-frame difference is used as determination measure, each wipe can be represented by the simple model as shown in Figure 4. Then a wipe is declared when the following equations are satisfied for those frames which are not designated as abrupt nor dissolve scene change.

$$BW > Th\_bw \,,\ DW > Th\_dw,\ AW > Th\_aw \tag{21}$$

Here BW, DW, and AW are number of frames which are recognized as periods before wipe, during wipe and after wipe, respectively. These variables are obtained by:

$$BW = \sum_{k=prw+1}^{ws} DL(k), \quad DW = \sum_{k=ws+1}^{we} DH(k), \quad AW = \sum_{k=we+1}^{pow} DL(k) \tag{22}$$

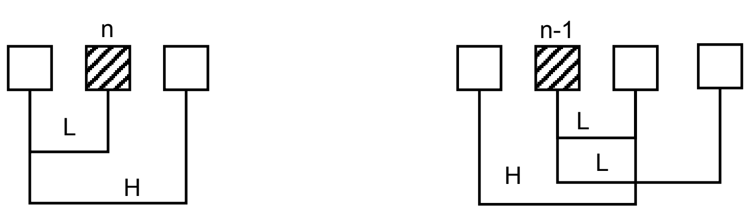
where $DL(k)$ and $DH(k)$ are flags which show that k th frame has low and high interframe difference D_k, respectively. These flags are determined by the following conditions.

$$\text{if } D_k > Th\_wp \text{ then } DL(k)=0,\ DH(k)=1 \text{ else } DL(k)=1,\ DH(k)=0 \tag{23}$$

3.4 Flashlight and subliminal effect detection

A flashlight scene is spontaneous frame change due to flashlight in a shot. For example, in TV news sequence, a flashlight scene appears while an important person gives a speech in a press conference. Also a subliminal effect (simply subliminal, hereafter) may be inserted into TV programs or films with a certain intention. Since a flashlight frame and a subliminal frame are quite different from preceding and following frames, frames with flashlight/subliminal and after flashlight/subliminal are often falsely detected as scene change. Luminance and chrominance distributions in flashlight/subliminal frame are completely different from those in the previous frames. However, unlike shot boundary, these distributions in flashlight return to the previous states after one or a few frames. Therefore by investigating frames before and after flashlight scene, flashlight scene can be excluded from scene change points. Single flashlight model is depicted in Figure 5. For example, when n th frame is flashlight scene, correlation between n th and $n-1$ th is low whereas correlation between $n+1$ and $n-1$ is high as shown Figure 5(a). In the same way, especially consecutive flashlight scenes can be easily modeled by extending single flashlight model. We use chrominance histogram correlation as correlation measure in order to distinguish flashlight from other shot boundary. Therefore flashlight/ subliminal effect at n th frame is detected when:

$$\rho(n, n-1) < Th\_fl \,,\ \rho(n+1, n-1) > Th\_fh \tag{24}$$



(a) Flashlight/subliminal at *n* th frame (b) Flashlight/subliminal at *n-1* th frame

Figure 5. Single flashlight/subliminal effect models

4. EVALUATION RESULTS

We applied the above mentioned shot boundary determination to TRECVID 2003 test data (totally 12 sequences). The overall flowchart is shown in Figure 6. All the parameters used in the above equations are determined through a 20 minutes TV sequence encoded by MPEG-1, not in TREC test data.

Table 1 shows the results of shot boundary determination; recall and precision for total, recall and precision for abrupt shot boundaries, and recall, precision, frame-recall and frame-precision for gradual transition boundaries. These scores are calculated using TREC shot boundary evaluation software provided by NIST.

As shown in Table 1, most of abrupt shot boundaries are successfully detected. However, in spite of incorporating flashlight exclusion algorithm, most of the false detections for abrupt shot boundaries are flashlights. In addition, sudden changes of brightness such as shining are falsely determined as abrupt shot boundaries. As for un-detection, the abrupt shot boundaries between fields are not detected since the test data is encoded in frame structures. Also the shot boundaries where the frame is only partly changed are not detected.

As for gradual transitions, about half of the shot boundaries are detected in our algorithm. The cause of false detection is roughly categorized in two cases; one is that a scene is falsely determined as wipe or dissolve when a large object slowly comes into a frame, and the other case is when an object suddenly starts to move very fast from still mode. These false detections require more detailed observation of motion of the object. As for un-detection, many of wipe transitions cannot be determined. In addition, dissolve transitions between very similar shots in terms of color, texture, etc. are not detected. Therefore more detailed analysis is needed for enhancing the gradual transition detection accuracy, which corresponds to future challenge.

As for detection speed, our method achieves very fast operation, about 24 times faster than real-time playback on the normal Windows PC with Pentium 4 1.8GHz CPU, since all the processes are performed on compressed data domain.

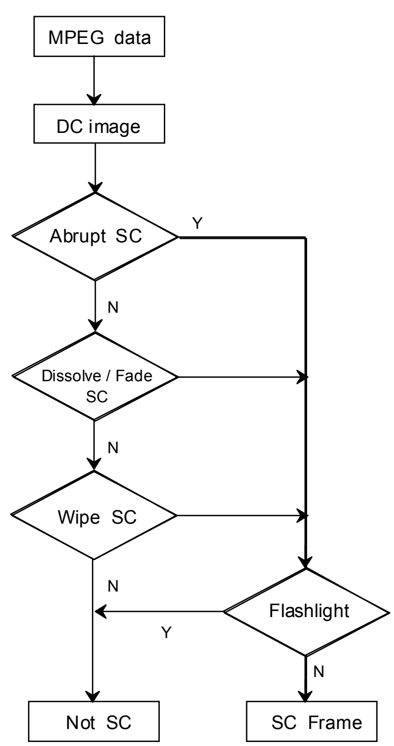


Figure 6. Overall shot boundary determination flowchart

Table 1. Experimental results for TRECVID 2003 test sequences

Sequence	All		Abrupt		Gradual			
	Recall	Precision	Recall	Precision	Recall	Precision	F-Recall	F-Precision
19980203	0.758	0.786	0.928	0.830	0.479	0.672	0.590	0.524
19980222	0.846	0.812	0.961	0.855	0.495	0.625	0.404	0.624
19980224	0.800	0.844	0.952	0.867	0.458	0.750	0.440	0.508
19980412	0.815	0.810	0.973	0.850	0.416	0.633	0.514	0.626
19980425	0.776	0.785	0.942	0.810	0.505	0.716	0.593	0.533
19980515	0.826	0.832	0.957	0.879	0.541	0.689	0.491	0.576
19980531	0.873	0.842	0.974	0.868	0.537	0.716	0.542	0.595
19980619	0.839	0.868	0.984	0.915	0.472	0.681	0.538	0.540
19990303	1.000	0.684	1.000	1.000	0.000	0.000	-	-
19990308	0.960	0.857	0.960	1.000	0.000	0.000	-	-
20010614	1.000	0.470	1.000	1.000	0.000	0.000	-	-
20010702	1.000	0.937	1.000	1.000	0.000	0.000	-	-

5. CONCLUSION

In this notebook, firstly a preprocessing for shot boundary determination is described. By using motion vectors and DCT DC information, DC image in 1/64 of original coded sized has been obtained directly from MPEG bitstream for P- and B-pictures as well as I-pictures. Shot boundary determination algorithm not only for abrupt scene change but also for gradual transitions such as dissolve and wipe have been proposed. In our methods, statistics like histogram as well as motion vector information from coded bitstream are used to adaptively detect various types of shot boundaries. In addition, exclusion algorithms for panning and flashlight/subliminal scenes have also been proposed. In the experiment around 95% of abrupt shot boundaries have been successfully detected for the TRECVID test data. As for gradual transitions, about half of shot boundaries have been detected. Since its process is very fast and only less than 5% of normal playback time is required, the proposed method well realizes efficient shot boundary determination used for higher level processing such as content base video analysis.

6. ACKNOWLEDGMENT

The authors would like to thank Dr. T.Asami, Dr. S.Matsumoto, and Dr. M.Wada for their continuous support and also wish to thank to the original proponents, Ms. K.Watanabe and Mr. A.Yoneyama.

7. REFERENCES

[1] Y. Nakajima, K. Ujihara, and A. Yoneyama, "Universal scene change detection on MPEG-coded data domain," in Proceeding SPIE Visual Communications and Image Processing, vol. 3024, pp. 992-1003, 1997.
[2] Y.Nakajima, "A Video Browsing Using fast scene change detection for an efficient networked video database access," IEICE Transactions on Information & Systems, vol.E-77-D, No.12, pp.1355-1364, Dec.1994.
[3] Y.Tonomura, A.Akutsu, Y.Taniguchi, and G.Suzuki, "Structured video computing," IEEE Multimedia, pp.34-43, Fall 1994.
[4] K.Otsuji and Y.Tonomura, "Projection detecting filter for video cut detection", Proceedings of First ACM International Conference on Multimedia, pp.251-257, Aug.1993
[5] H.J.Zhang, A.Kankanhalli, and S.W.Smoliar, "Automatic parsing of news video," Proc.IEEE Int'l Conf. Multimedia Computing and Systems, May. 1994.
[6] S.W.Smoliar and H.J.Zhang, "Content-based video indexing and retrieval," IEEE Multimedia, pp.62-72, 1994.

[7] A.Nagasaka and Y.Tanaka, "Automatic video indexing and full-motion search for object appearances", Visual database systems, vol.II, E.Knuth and L.M.Wegner, eds, Elsevier, Amsterdam, pp.113-127, 1992.
[8] F.Arman, R.Depommier, A.Hsu, and M.Y.Chiu, "Image processing on compressed data for large video databases", Proceedings of First ACM International Conference on Multimedia, pp.267-272, Aug.1993.
[9] B.L.Yeo and B.Liu, "Rapid scene analysis on compressed video", IEEE Transactions on Circuits and Systems for Video Technology, Dec.1995.
[10] B.Shahrarary, "Scene change detection and content-based sampling of video sequences", Digital Video Compression: Algorithms and Technologies, SPIE, Vol.2419, pp.2-13, 1995.
[11] A.Hampapur, R.Jain and T.Weymouth, "Digital Video Segmentation", Proc. ACM Multimedia 94, pp.357-364, 1994.
[12] K.Shen and E.J.Delp, "A Fast Algorithm for Video Parsing Using MPEG Compressed Sequences", Proceeding of IEEE ICIP '95, pp.252-255, 1995.
[13] J.Meng, Y,Juan and S-F Chang, "Scene Change Detection in a MPEG Compressed Video Sequence", Digital Video Compression: Algorithms and Technologies, SPIE, Vol.2419, pp.14-25, 1995.
[14] Y.Nakajima, K.Ujihara, and T.Kanoh, "Video structure analysis and its application to creation of video summary", IEICE 2nd Joint Workshop on Multimedia Communications, pp.3-2, Oct.1995.