

# Shot Boundary Detection and Low-Level Feature Extraction Experiments for TRECVID 2005

Kazunori Matsumoto, Masaru Sugano, Masaki Naito, Keiichiro Hoashi,  
Haruhisa Kato, Masami Shshibori\*, Kenji Kita\*, Fumiaki Sugaya, and Yasuyuki Nakajima

KDDI R&D Laboratories, Inc.  
2-1-15 Ohara, Fujimino,  
Saitama 356-8502, JAPAN  
{matsu, sugano, naito, hoashi, hkato, fsugaya, nakajima}@kddilabs.jp,  
{bori, kita}@is.tokushima-u.ac.jp

\* Tokushima University  
2-1 Mishimacho Nanjyo,  
Tokushima, 770-8506, JAPAN

## 0. STRUCTURED ABSTRACT

### Shot boundary detection

1. *Briefly, what approach or combination of approaches did you test in each of your submitted runs?*

- **bs-1:** Compressed domain approach, which corresponds to the best options in TRECVID 2004 with newly introduced luminance adaptive threshold.
- **bs-2:** Compressed domain approach with newly introduced luminance adaptive threshold and image cropping.
- **bs-3:** Compressed domain approach with parameter optimization in TRECVID 2004.
- **bs-4:** Compressed domain approach with newly introduced image cropping.
- **bs-5:** Compressed domain approach, which corresponds to the best options in TRECVID 2004 without any optimization and extension.
- **bs-6:** Uncompressed domain approach with an abrupt cut detector based on data fusion technique with SVM trained with TV2004 ref. But short gradual cuts are not trained. Novel feature derived from image synthesis parameters are introduced.
- **bs-7:** Uncompressed domain approach with the same technique of bs-6. Trained data is also TV2004 ref. But Short gradual cuts are trained.
- **bs-8:** Variant of bs-7. Abrupt cuts are trained from TV2004 ref. Short gradual cuts are trained from the subset of 2005 develop.
- **bs-9:** Result of bs-1's grad and that of bs-7's cut are merged.
- **bs-10:** Result of bs-1's grad and that of bs-8's cut are merged.

All these runs are conducted by KDDI Laboratories.

2. *What if any significant differences (in terms of what measures) did you find among the runs?*

Compared with our TRECVID 2004 approach, luminance adaptive threshold and image cropping which extend our original compressed domain approach gave reasonable

improvements for CUT and CUT+GRAD, respectively. Abrupt cut detector on uncompressed domain makes higher performance for CUT than our compressed domain approaches.

3. *Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?*

**Luminance adaptive threshold for compressed domain:** Enabled the detection in low luminance image including black and white image, thus improved recall.

**Image cropping:** Enabled the detection of changing a limited region in an image. However gradual effects or flashlights occurring only in the middle of images were over detected.

**SVM based abrupt detector on non-compressed domain:** Successful in preventing erroneous abrupt cut boundaries.

4. *Overall, what did you learn about runs/approaches and the research question(s) that motivated them?*

The novel feature is very useful to detect abrupt cuts and short cuts.

### Low-Level Feature Extraction

1. *Briefly, what approach or combination of approaches did you test in each of your submitted runs?*

- **Labs\_1:** A feature point-based method on uncompressed domain. Feature parameters are judged by SVM.
- **Labs\_2:** Global motion extraction on compressed domain
- **Labs\_3:** Another global motion extraction on compressed domain
- **Labs\_4:** A feature point-based method on uncompressed domain. The camera motion vector is generated with every shot, and these vectors are input to the SVM.
- **Labs\_5:** A feature point-based method similar to Labs\_4. The camera motion vector is generated every

window, which has a certain frame width and slides by a few frames.

- Labs\_6: Threshold-based detection using the feature parameters of Labs\_1.
- Labs\_7: Variant of Labs\_6. Precision-oriented threshold levels were used

Run Labs\_4 & 5 are based on the technique of Tokushima University. Other's are based on that of KDDI.

2. *What if any significant differences (in terms of what measures) did you find among the runs?*

The results of a motion vector approach on compressed domain and a feature point tracking approach does not yield a little difference.

3. *Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?*

No.

4. *Overall, what did you learn about runs/approaches and the research question(s) that motivated them?*

This year's MPEG files have a good motion vectors. In the case of non-well vectors, the performance of the motion vector approach is still open.

## 1. INTRODUCTION

This is the third TRECVID participation for KDDI R&D Laboratories. This year, we have participated in the shot boundary detection and low-level feature extraction (camera motion) tasks. For the shot boundary detection task, our main focus was to conduct both the threshold based detection on compressed domain and SVM-based detection on uncompressed domain. For the low-level feature extraction task, we also conduct interest point base approach on uncompressed domain and the motion vector approach based only on compressed domain.

## 2. SHOT BOUNDARY DETECTION

This section describes shot boundary detection methods and experiments. Threshold based method on compressed domain and SVM based method on uncompressed domain are discussed.

### 2.1 Compressed domain approach

In TV2005, we especially focused on improving recall of abrupt shot boundaries.

#### 2.1.1 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 [1]. Furthermore, in I-pictures these coefficients are directly obtained during VLD (Variable Length Decoding) process without time consuming process such as Inverse DCT. DC components in these pictures can be obtained after some manipulation. In P- and B-pictures, although some of macro blocks 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 is a reduced size image by 1/8 both horizontally and vertically. Therefore DC components of P- and B-pictures are obtained using motion compensation in reduced size image domain. There are two ways to obtain DCT DC image for P-/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 [2][3]. Figure 1 shows a block diagram of the latter scheme. Subjectively, it is 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.

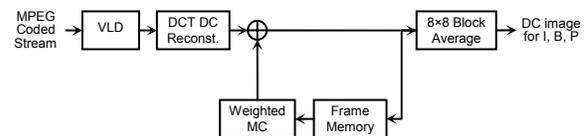


Figure 1. DC image with weighted MC

#### 2.1.2 Abrupt shot boundary determination

By incorporating the MC operation mentioned above, P- and B-pictures are 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], [4-13]. 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 achieves relatively high accuracy, each has its own disadvantage [14].

We proposed shot boundary determination from I-picture sequence of MPEG coded video in 1994 [1]. We use both pixel differences and histograms methods to overcome problems when either one of them is used. Here, we extend this approach to detect shot boundaries in one frame unit.

### Pre-processing

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 succeeding shot boundary determination. The inter-frame difference is obtained by:

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

, where  $M$  and  $N$  are total number of  $8 \times 8$  blocks in a frame for vertical and horizontal direction, respectively. For example, in MPEG-1 in SIF size ( $352 \times 240$ ),  $M=30$  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  $8 \times 8$  block is 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 conditions are further investigated in abrupt shot boundary detection in the following.

### Shot boundary determination using luminance and chrominance change

Both luminance and chrominance characteristics dramatically change at shot boundaries. Thus ordinary shot boundaries are 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 [1]. 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,  $\alpha$  is a weighting factor for the detection.  $\rho_n$  is chrominance histogram correlation obtained by:

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

, 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  $H_{n,k,l}$  ( $k, l = 0, 1, 2 \dots hc-1$ ) 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 do not satisfy Eq. (3).

$$\rho_n > Th_{ac} \quad (5)$$

, where  $Th_{ac}$  is a threshold value for determination of temporal peak in  $\rho_n$ .

Furthermore, when consecutive two shots are different only in camera angle, color histogram will be similar 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 are 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.

$$\beta D_n > D_{n-1}, D_{n+1} \quad (6)$$

$$D_n - Th_{ad} > D_{n-1}, D_{n+1} \quad (7)$$

, where  $\beta$  and  $Th_{ad}$  are a weighting factor and a threshold value for detecting a temporal peak in  $D_n$ , respectively. Basically, Eq. (6) will detect shot boundaries 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).

### Shot boundary determination for low luminance images

The majority of missed abrupt shot boundaries in the previous years are those between relatively dark shots and between black-and-white shots. In order to achieve higher recall of abrupt shot changes, we introduced the luminance adaptive evaluation. Before pre-processing an averaged luminance value  $aveY_n$  within an image is evaluated:

$$aveY_n = \frac{1}{MN} \sum_{i=0}^M \sum_{j=0}^N Y_n(i, j) \quad (8)$$

If  $aveY_n$  is lower than a certain threshold, the luminance difference  $D_n$  is multiplied by  $\gamma$  ( $\gamma > 1.0$ ). The subsequent processes are exactly the same as described above. This strategy is effective not only for shot boundaries between low luminance images, but also for shot boundaries between black-and-white images.

### Shot boundary determination for changes in only limited region

Another reason that causes lower recall rates for abrupt shot boundaries is that shot changes on a limited region

in an image are not correctly determined. For example, these misdetections occur when the upper and bottom parts of an image is unchanged such as closed captions and market information, while a shot changes in the middle of an image. Picture-in-picture can also be causes. We simply avoid these misdetections by image cropping; input images are vertically divided into four rows, and the middle two rows (i.e. the half of an image) are used for further analysis.

### ***Shot boundary determination between fields in a progressive sequence***

The detailed discussions on a method for determining between fields in a progressive sequence can be found in [15]. Some parameter optimization has been performed using TV2005 development data.

#### ***2.1.2 Dissolve shot boundary determination***

Dissolve shot boundary determination algorithm is almost the same as in TV2004, the only changes are applying image cropping described in the previous subsection and optimizing determination parameters. Thus detailed explanation is omitted here.

#### ***2.1.3 Wipe shot boundary determination***

Since our wipe shot boundary determination algorithm has not been changed since TV2003 with parameter optimization, the detailed descriptions are left out from this paper. Refer to [16] for details.

#### ***2.1.4 Flashlight and subliminal effect detection***

Since our flashlight and subliminal effect detection method has been unchanged since TV2004, the detailed descriptions are left out from this paper. Refer to [15] for details.

## **2.2 SVM-based approach on uncompressed domain**

In TV2004, SVM approach of KDDI seems to have over-adapted to the TRECVID 2003 data, which we consider as the main cause of the poor result. Therefore we improve our 2-stage data fusion technique with SVM, which can't be explained in TRECVID2003 because of presentation's limit.

The aim of this approach is to obtain better recall and precision. Processing time of this approach is considered less than that of the approach on compressed domain.

### ***2.2.1 Data fusion with multiple SVMs***

From a learning theory perspective, it is a natural approach to combine promising features in order to decide whether a boundary exists or not within a given video sequence. But naïve feature combination makes

an excessive feature space to handle. Therefore, we adopt a 2-stage data fusion approach with a Support Vector Machine (SVM) technique. The overview of our data fusion approach is as follows: At the first stage, every adopted feature is judged by a specific SVM. This means the number of feature types is equal to the number of SVMs at the 1<sup>st</sup> stage. And the SVM at the second stage synthesizes the judgments from the 1<sup>st</sup> stage.

Figure 2 shows our 2-stage discriminator in prediction mode. "F1" ~ "F6" represent the feature values extracted from a video sequence. A conventional and useful *multiple pair-wise* technique [17] is applied for all these features. Table 1 shows the brief description of these features. The decision is made frame by frame.

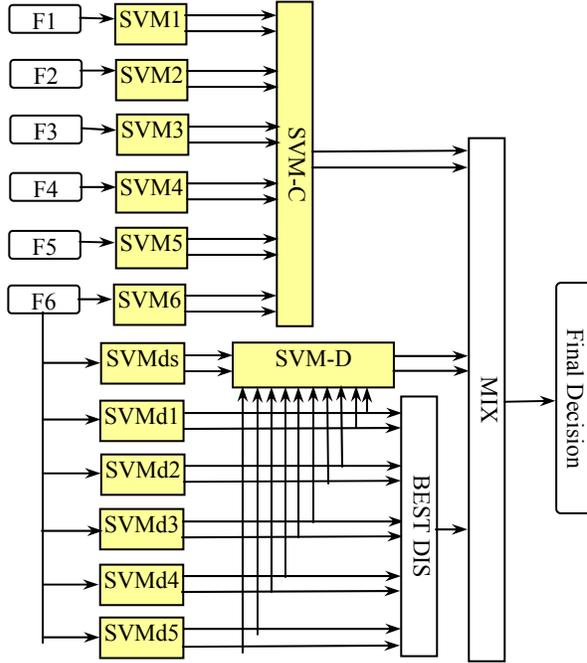
"SVM1" ~ "SVM6" are Support Vector Machines at the 1<sup>st</sup> stage. Each SVM is designed to detect an abrupt cut based on a specific feature. "SVMs" is designed to detect a dissolve cut with any transition span. "SVMd1" ~ "SVMd5" are designed to detect a dissolve transition with a specific length. For example, "SVMd1" discriminates the existence of a dissolve transition whose length is 1. Every SVM outputs two kinds of values: the probability that a specified type of cut is detected and the probability that the same is not detected. "SVM-C" and "SVM-D" are Support Vector Machines at the 2<sup>nd</sup> stage. "SVM-C" discriminates the existence of an abrupt cut based on the result of the 1<sup>st</sup> stage, while "SVM-D" also discriminates a dissolve cut.

The functionality of "MIX" on Figure 2 is an arbitration of "SVM-C" and "SVM-D", based on the four probabilistic values. When "SVM-D" detects a dissolve cut and "SVM-C" does not detect an abrupt cut, "BEST DIS" chooses the most probable length of the dissolve transition.

Please note that SVM training may be a resource consuming task, but the computation cost of SVM's prediction is less than that of feature extraction.

**Table 1: Explanation of adopted features.**

Feature ID	Description	# dimension
F1	the number of in-edges and out-edges in divided regions (4 by 4) based on [18]	224
F2	Standard deviation of pixel intensities in divided regions (4 by 4)	224
F3	TRECVID2005 approach by FXPAL[19] with Ohata's color space, with PAC	192
F4	TRECVID2005 approach by FX PAL[19] with RGB color space, with PAC	192
F5	Edge change ratio described in [21]	192
F6	Novel feature described in section 2.2.2	210



**Figure 2: Structure of 2-stages SVMs on a prediction mode.**

### 2.2.2 Novel feature derived from image synthesis parameters

Frame image in a dissolve transition are synthesized from two images, which come from different two video sequences respectively [19]. There are two scaling parameters for the synthesis. Values of the pixel in a

**Table 2: The matrix showing which type of cut affects the two types of difference.**

Type of difference \ Type of cut in a sequence	None	Abrupt cut	Dissolve cut
Image difference between the target and synthesized image	small	significant	small
Image difference between non-target images	small	significant	significant

synthesized image are proportional to the values of the corresponding pixel. Our basic idea is to use these synthesis parameters for cut detection is as follows:

Let there be three continuous frame images, of which the middle one is regarded as a target image. Assume a synthesized image from the two non-target images. In our approach we minimize the difference between the target and the synthesized image by adjusting synthesis parameters. If an abrupt shot boundary exists within the continuous three images, the difference will become significant, even after optimal minimization is conducted. A fast optimization algorithm is described later.

In addition, we consider the difference between non-target images simultaneously. Dealing with these two different types at the same time results in a better performance. There is little difference between the target image and a synthesized image, both in the case of a non-cut sequence and one including a dissolve cut. However, it must be noted that the difference between non-target images is considerable in the case of dissolve although minor in the case of a non-cut sequence. The matrix in Table 2 summarizes the qualitative trend of these two differential values.

Let  $f, f_A, f_B$  be a synthesized image and images to be used to synthesize  $f$ . Let  $A_R, A_G, A_B$  be scaling parameters of  $f_A$ . Let  $B_R, B_G, B_B$  be scaling parameters of  $f_B$ . Let  $X_i^R, Y_i^R, Z_i^R, X_i^G, Y_i^G, Z_i^G, X_i^B, Y_i^B, Z_i^B$  be luminance of pixel  $i$  in  $f_A, f_B, f$ . Then

$$A_R X_1^R + B_R Y_1^R - Z_1^R = \varepsilon_1^R$$

$$A_R X_2^R + B_R Y_2^R - Z_2^R = \varepsilon_2^R$$

$$A_R X_n^R + B_R Y_n^R - Z_n^R = \varepsilon_n^R$$

Let  $F_R(A_R, B_R) = \sum(\varepsilon_i^R)^2$ , the estimation problem is to find  $A_R$  and  $B_R$  which minimize  $F_R(A_R, B_R)$ .

This estimation can be solved. As  $\frac{\partial}{\partial A} F_R(A_R, B_R) = 0$ , following equation should be solved.

$$\begin{bmatrix} \sum(X_i)^2 & \sum X_i Y_i \\ \sum X_i Y & \sum(Y_i)^2 \end{bmatrix} \begin{bmatrix} A_n \\ B_R \end{bmatrix} = \begin{bmatrix} \sum Y_i Z_i \\ \sum X_i Z_i \end{bmatrix}$$

Our features are the values of  $F_R(A_R, B_R)$ ,  $F_G(A_G, B_G)$ ,  $F_B(A_B, B_B)$ .

Once optimal synthesis parameters  $A_R, B_R, A_G, \dots$  are obtained, we can easily calculate the feature values by the definitions.

### 2.3 Evaluation results

We applied the above mentioned methods to TRECVID 2005 test data (totally 12 sequences). All the parameters used in 2.1 are determined through TRECVID 2004 test data. Training of SVM described in 2.2 is conducted with 2004 test data and 2005 development data. Tables 2 to 4 show the results of shot boundary determination; recall, precision, and F-measure for ALL, CUT, and GRAD results for submitted 5 runs of compressed domain approach and 5 runs of uncompressed domain approach. The relation between each RunID and their details are presented in the structured abstract.

#### Analysis I: approach on compressed domain (From runID=1 to 5)

As shown in Table 3, most of abrupt shot boundaries are successfully detected, although only slight improvements has been obtained between the best result of TRECVID 2005 (RunID=1) and TRECVID 2004 (RunID=5). The major misdetections are as follows:

- i) Shot boundaries between extremely dark shots.
- ii) Shot boundaries between shots with high motion activities.
- iii) Shot boundaries which successively appear in a very short duration.
- iv) Shot boundaries between shots with a large logo or characters appear during shot transitions.

As affects of cropping images, abrupt shot boundaries where gradual effects or flashlights occur only in the middle of images were over detected.

On the contrary, as for gradual transitions, while recall gains are about 10% at maximum compared with the last year's approach, the total improvements are only a few percents. These gains are obtained mainly from image cropping. However, over 30% of gradual transitions are still undetected. The causes of these misdetections are almost the same as those described for abrupt shot boundaries.

Our method achieves very fast operation. The internally measured processing times are; total run time = 799.6 [sec], total decode time = 533.1 [sec], and total segmentation time = 266.5 [sec] on the normal Windows PC with Pentium 4 1.8GHz CPU and 512MB RAM. This corresponds to about 929 frames/sec of total run time.

#### Analysis II: approach on uncompressed domain (From runID=6 to 10)

From runID=6 to 10, abrupt cut detectors based on multiple SVMs with novel feature make high performance for CUT. The F1 of runID=7,8 of CUT(Table 4) is best among TV2005 participants. Please note that the performance of GRAD of runID=9, 10 is the same as runID=1, because all the estimated dissolve transition come from dissolve detector used in runID=1.

**Table 3. Recall, precision and F-measure of ALL**

#	RunID	Recall	Precision	F1
1	kddi_labs_sb_run_1	0.875	0.823	0.848
2	kddi_labs_sb_run_2	0.888	0.797	0.840
3	kddi_labs_sb_run_3	0.857	0.838	0.847
4	kddi_labs_sb_run_4	0.873	0.809	0.840
5	kddi_labs_sb_run_5	0.835	0.859	0.847
6	kddi_labs_sb_run_6	0.613	0.976	0.753
7	kddi_labs_sb_run_7	0.693	0.955	0.803
8	kddi_labs_sb_run_8	0.697	0.949	0.804
9	kddi_labs_sb_run_9	0.867	0.862	0.864
10	kddi_labs_sb_run_10	0.871	0.860	0.865

**Table 4. Recall, precision and F-measure of CUT**

#	RunID	Recall	Precision	F1
1	kddi_labs_sb_run_1	0.932	0.896	0.914
2	kddi_labs_sb_run_2	0.955	0.847	0.898
3	kddi_labs_sb_run_3	0.929	0.898	0.913
4	kddi_labs_sb_run_4	0.955	0.849	0.899
5	kddi_labs_sb_run_5	0.914	0.910	0.912
6	kddi_labs_sb_run_6	0.822	0.976	0.892
7	kddi_labs_sb_run_7	<b>0.930</b>	<b>0.955</b>	<b>0.942</b>
8	kddi_labs_sb_run_8	<b>0.936</b>	<b>0.949</b>	<b>0.942</b>
9	kddi_labs_sb_run_9	0.920	0.956	0.938
10	kddi_labs_sb_run_10	0.926	0.952	0.939

**Table 5. Recall, precision and F-measure of GRAD**

#	RunID	Recall	Precision	F1
1	kddi_labs_sb_run_1	0.709	0.627	0.665
2	kddi_labs_sb_run_2	0.694	0.642	0.667
3	kddi_labs_sb_run_3	0.648	0.653	0.650
4	kddi_labs_sb_run_4	0.634	0.671	0.652
5	kddi_labs_sb_run_5	0.604	0.686	0.642
6	kddi_labs_sb_run_6	0.000	0.000	---
7	kddi_labs_sb_run_7	0.000	0.000	---
8	kddi_labs_sb_run_8	0.000	0.000	---
9	kddi_labs_sb_run_9	0.709	0.627	0.665
10	kddi_labs_sb_run_10	0.709	0.627	0.665

## 2.4 Conclusion

In the first half of this Section, shot boundary determination algorithm based on MPEG compressed domain, one of our contributions to TRECVID 2005 shot boundary determination task, 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 is proposed. In our methods, low-level features such as luminance, chrominance and edge extracted from DC images are used to detect various types of shot boundaries. In addition, adaptive threshold according to luminance of images as well as image cropping which achieves higher recall rate of abrupt shot boundaries have been introduced.

In the latter half of the section, SVM based approach on uncompressed domain is described. Remarkable result comes from data fusion approach. By the other experiments, we confirmed the good contribution of our novel feature derived from image synthesis parameters. The detailed analysis of our new feature will be showed in another future presentation.

## 3. LOW-LEVEL FEATURE EXTRACTION

### 3.1 Camera motion detection using video mosaicing

In this paper, we apply an image mosaicing method based on the correlation between feature points, to detect camera motion from TV programs such as news. TV images have an adverse effect on estimating the correlation between feature points, since they include outliers, such as telops and moving objects. Therefore, we introduce a scheme to detect the telop region and iterative foreground and background image separation to remove these regions from extracting feature points to estimate the correlation between two frames. Furthermore, since the video image has many frames, we select the appropriate frame pair from those possible and generate an accurate background image.

Based on the position of frames on the background image, camera motion is detected using a threshold-based method or a SVM-based method.

#### 3.1.1 Background image generation using video mosaicing

Firstly, a background image is generated from a consecutive video image; based on the image mosaicing method and progressively estimating the rotation, scale change and the projective distortion between feature points on two images using stratified matching [21].

The outline of background image generation is as follows:

- (1) Telop regions in video image are detected by using a moving text detection method. They are then removed from areas where the feature points are extracted in the following steps:
- (2) For every frame pair on consecutive video frames, feature points are extracted from both frames using a harris operator [22] and the correlations between the two frames are estimated by using [21].
- (3) A temporal background image is generated by using correlations between the frame pairs obtained in step 2
- (4) The foreground image is detected based on the method [23] and they are removed from the area to extract feature points on the next iteration.
- (5) Steps 2 to 4 are repeated a predetermined number of times.
- (6) The position of each video frame is calculated, on the generated background image.

Figure 3 shows separated foreground and background image resulting in step 4. Figure 4 shows the positions of frames on the background image.



Figure 3. Separated foreground and background images; original image (top left), background image (bottom left), foreground image (bottom right) and generated background image (top right).

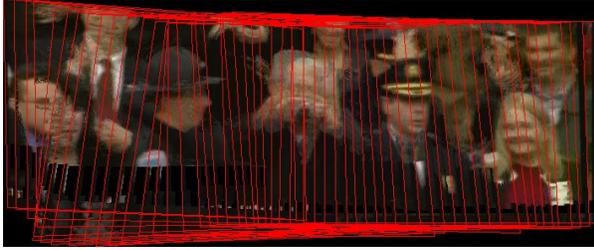


Figure 4. Position of frames on generated background image.

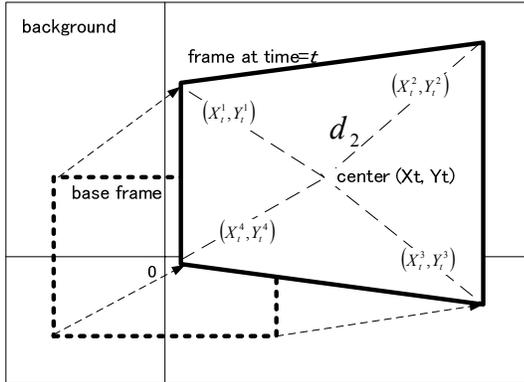


Figure 5. Relation between camera frame position parameters and feature parameters for detection.

### 3.1.2 Threshold-based camera motion endpoint detection

#### Feature parameters

Firstly, the camera frame position parameters are converted to feature parameters for camera motion detection as follows. In this method, (1) the coordinate

of the center of each quadrangle camera frame and (2) the distance between the center and vertex of the quadrangle camera frame are used as feature parameters for detection. Figure 5 illustrates the relation between the frame position and feature parameters for camera motion detection. The center of the quadrangle at the base frame, which is the start frame of each shot, is used as the origin. Then the coordinates of the center of each quadrangle frame  $(X_t, Y_t)$  and the normalized log distance between the center and vertex of the quadrangle frame  $z_t$  were calculated using the following formula [24]:

$$X_t = \sum_{n=1}^4 X_t^n / 4$$

$$Y_t = \sum_{n=1}^4 Y_t^n / 4$$

$$d_t = \sum_{n=1}^4 \left( (X_t^n - X_t)^2 + (Y_t^n - Y_t)^2 \right)$$

$$z_t = \log(d_t / d_{base})$$

, where  $(X_t^n, Y_t^n)$  are the coordinates of the vertex of the quadrangle frame at time  $t$ ,  $d_{base}$  and  $d_t$  are the distances between the center and vertex of the quadrangular frame at frame  $base$  and  $t$ , respectively.

In addition to the static feature parameters, e.g.  $X_t$ , we introduce a time derivative parameter, which is called a delta coefficient, to take the speed of motion into account. The delta coefficients are computed using the following regression formula [24]:

$$\Delta C_t = \sum_{\theta=1}^{\Theta} \theta \times (C_{t+\theta} - C_{t-\theta}) / 2 \sum_{\theta=1}^{\Theta} \theta^2$$

, where  $\Delta C_t$  is a delta coefficient at frame  $t$  computed in terms of the corresponding static feature parameters  $C_{t-\Theta}$  to  $C_{t+\Theta}$ . In the following experiments,  $\Theta$  was set to 6.

#### Endpoint detection

In this method, camera motions are detected whether the feature parameters exceed or go below a given threshold level. With this method, camera motion start point detection begins from the start frame of each shot. Then the start point of camera motion is detected when the feature parameters  $X_t$  or  $\Delta X_t$  exceed given threshold levels  $Th_x$  or  $Th_{\Delta X}$  for successive  $L_x^B$  frames (in the case of pan right). Once a start point is found, end point detection begins. Subsequently, the end point of camera motion is detected when the feature parameter  $\Delta X_t$  goes below a given threshold level  $Th_{\Delta X}$  for successive  $L_x^E$  frames. These

procedures are repeated until the end of the shot, but only delta coefficients are used for detecting the 2nd and subsequent beginning points in a single shot.

The same procedure is used with a different threshold level to detect pan left. Furthermore, camera motion parameters  $Y_t$  and  $\Delta Y_t$  are used to detect tilt up/down and  $Z_t$  and  $\Delta Z_t$  are used for detecting zoom in/out.

### 3.1.3 SVM-based camera motion detection

For the sake of comparison, a simple SVM technique is applied to this camera motion detection using video mosaicing. The same method in 3.1.2 is used to obtain background images for every shot. Distances of corresponding corners between two adjacent frames are inputted to SVM. Training is conducted with 12 files in TV2005 development data set.

## 3.2 Camera motion detection based on movement of feature points between images

This algorithm estimates the camera motion by using the movement information of feature points extracted from each frame image in the video. The outline of this algorithm is shown in Fig. 6.

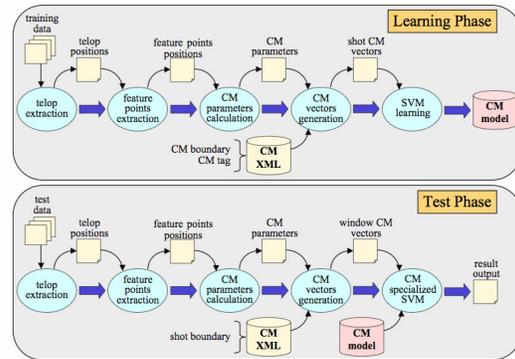
### 3.2.1 Feature points extraction

In the case that the telop regions appear in the video, many feature points are often detected around characters of the telop. In order to solve this problem, initially, the position of the telop area is detected from the video data. Next, some feature points are extracted from the frame image, excepting the telop area by using the Harris interest operator [22], [25]. The time complexity of the Harris interest operator is  $O(N^3)$ , where  $N$  is the total number of pixels in a frame. If feature points are extracted from all frames, the time cost becomes high, so this system extracts every 3 frames. The minimum distance between neighboring feature points is set to 30 pixels, in order to avoid the detection of many gathered points.

And then all correspondences of each feature point between two neighboring frames are decided. Because the moving distance of the feature point seems very small for 3 frames, we associate each feature point in the preceding frame with not all points in the following frame but only points in the circle area, where the center is the original point and the radius is 30 pixels, in the following frame. For these points, the distance ( $dist$ ) between two points ( $a$  and  $b$ ), where one is the point in the preceding frame and another is the following frame, is calculated by the following equation. The closest point in the following frame is associated with the original point in the preceding.

$$dist(a, b) = \alpha \times col\_dist(a, b) + \beta \times rgb\_dist(a, b) + \gamma \times weight\_dist(a, b)$$

where  $col\_dist(a, b)$ ,  $rgb\_dist(a, b)$  and  $weight\_dist(a, b)$  indicate the Euclid distance of coordinate, RGB color value and Harris interest weight between  $a$  and  $b$ , respectively.  $\alpha$ ,  $\beta$  and  $\gamma$  are the weight parameters of each attribute.



**Figure 6 Outline of the CM estimation system using the movement information of feature points.**

### 3.2.2 Camera motion parameters calculation

Three parameters (zoom, pan, tilt) of the camera motion (CM) are calculated, based on the movement information of feature points by the conventional method, which has been proposed by Tan [26]. They estimate camera motion parameters based on the corresponding motion vectors of the intercoded motion blocks in MPEG video data. On the other hand, we apply the following closed-form expressions to all correspondences of each feature point.

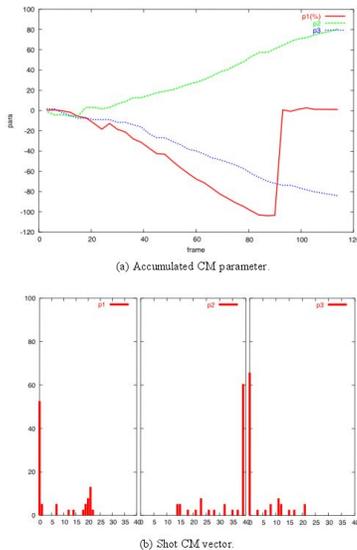
$$b = \frac{1}{N} \sum_{k=1}^N \left\| \frac{w'_k}{s} - w_k \right\|^2 = \frac{\overline{w'_k} - \overline{w}}{s}$$

$$s = \frac{\sum_{k=1}^N (w'_k - \overline{w}')^T (w_k - \overline{w})}{\sum_{k=1}^N \|w_k - \overline{w}\|^2}$$

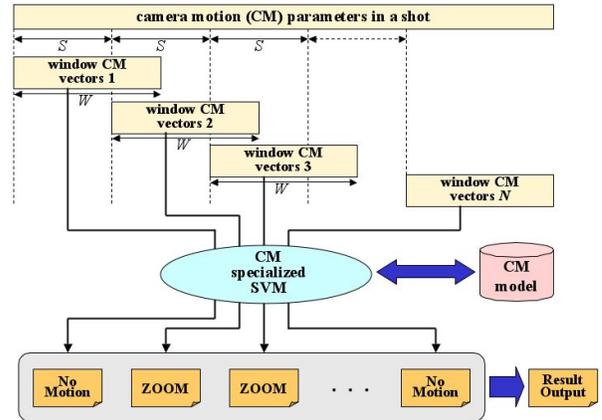
Here  $w_k$  and  $w'_k$  are the image coordinates of corresponding feature points in two neighboring frames. And  $s = p_1$ ,  $b = (p_2/p_1, p_3/p_1)^T$ , where  $p_1$  indicates the camera zoom parameter, and  $p_2/p_1$  and  $p_3/p_1$  are the change in the camera pan and tilt parameters between the two frames.

### 3.2.3 Camera motion vectors generation

For the development video data, we prepared the camera motion (CM) XML data, where CM boundaries and corresponding CM tags were described. By using the CM XML data, a CM vector expression for a CM in the shot is generated, based on the time series data of three accumulated CM parameters. On this CM vectors generation algorithm, firstly the range of CM parameters is evenly sliced into 40 parts. Next, the number of frames which have the CM parameter value within each sliced range is counted. As a result, we can obtain 40 frame frequencies per single CM parameter. Finally, they are regularized so that the total number of frame frequency may become 100. This vector is called the “shot CM vector”. Fig. 7-(b) shows the shot CM vector computed from the accumulated CM parameters in Fig. 7-(a). The vertical axis of Fig. 7-(b) indicated the frame frequency, and the horizontal axis is the sliced CM parameter range.



**Figure 7. Example of the shot CM vector; Accumulated camera motion parameters (top) and generated shot CM vector (bottom).**



**Figure 8. Window-based CM estimation method.**

For the test video data, we prepared the CM XML data, where only shot boundaries were described. In general, some camera motions of various lengths appear in a shot, and the motion scene starts from arbitrary time in a shot. Then, we adopted the window-based method shown in Fig. 8. This method divides a shot into some small windows. The width of the window is  $W$  frames and it slides by  $S$  frames in the shot. On this system, we set  $W$  and  $S$  to 50 and 10 respectively. The CM vectors generation algorithm makes a CM vector expression from accumulated CM parameters within a window by referring to the CM XML data. This vector is called the “window CM vector”.

### 3.2.4 Camera motion estimation based on SVM model

On the learning phase, this system classifies all shot CM vectors of the training data into CM classes, including the non-motion class, by referring to the CM XML data. As there was a large amount of data of the non-motion class, we selected 2,000 data by k-means algorithm. Subsequently, they are input into the Multi-Class Support Vector Machine [27]. On the test phase, one system generates a shot CM vector for each shot and estimates a corresponding CM class to the shot. On the other hand, another system generates a window CM vector for each window and estimates a corresponding CM class to each window CM.

### 3.3 Global motion extraction on compressed domain

A conventional method to extract global motion on compressed domain is applied to obtain the baseline of this task.

**Table 6. Recall, precision and F-measure of camera motion detection with TRECVID2005 LLFE test data**

RunID	PAN			TILT			ZOOM			MEAN		
	prec.	recall	F1									
Labs_1	0.990	0.508	0.671	1.000	0.310	0.473	0.987	0.452	0.620	0.992	0.423	0.593
Labs_2	0.921	0.734	<b>0.817</b>	0.863	0.419	0.564	0.881	0.640	0.741	0.888	0.598	0.715
Labs_3	0.970	0.663	0.788	0.939	0.219	0.355	0.964	0.732	0.832	0.958	0.538	0.689
Labs_4	0.907	0.583	0.710	0.871	0.290	0.435	0.699	0.464	0.558	0.826	0.446	0.579
Labs_5	0.840	0.779	0.808	0.737	0.548	0.629	0.598	0.787	0.680	0.725	0.704	0.714
Labs_6	0.978	0.685	0.806	1.000	0.557	<b>0.715</b>	0.914	0.728	0.810	0.964	0.657	<b>0.781</b>
Labs_7	0.984	0.622	0.762	1.000	0.495	0.662	0.921	0.726	<b>0.812</b>	0.968	0.614	0.751

### 3.4 Evaluation results with TRECVID2005 LLFE test data.

The submitted RunID **Labs\_1** is the result obtained by SVM technique applied to video mosaicing described in 3.1.3.

RunID **Labs\_2** and **Labs\_3** are that of conventional global motion detection technique. These are conducted to obtain our baseline. However, **Labs\_2** & **3** made good mark in TV2005 of our submitted runs.

RunID **Labs\_4** and **Labs\_5** are results of SVM technique applied to feature point motion described in 3.2. RunID **Labs\_4** is that obtained by using the shot-based vector, on the other hand, **Labs\_5** is the window-based vector. **Labs\_5** achieved a higher accuracy compared to **Labs\_4**, because in particular, short camera motions often appear during the shot.

RunID **Labs\_6** and **Labs\_7** come from the threshold based technique described in 3.1.2. Experiments with various threshold levels were conducted with development data and the threshold levels achieving optimal performance were used in **Labs\_6**. RunID **Labs\_7** is that obtained by using a higher threshold level to improve precision. The reason why **Labs\_6** & **7** obtained better mark seems to be the nature of training data for **Labs\_1**.

It is difficult to conclude these results. We believe that the criteria of camera motion should be discussed. Table 6 shows the whole results of our submission. Bold print shows the best mark in our runs.

## 5. REFERENCES

[1] 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.  
 [2] B. L. Yeo and B. Liu, "Rapid scene analysis on compressed video", IEEE Transactions on Circuits and Systems for Video Technology, Dec.1995.

[3] 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.  
 [4] Y. Tonomura, A. Akutsu, Y. Taniguchi, and G. Suzuki, "Structured video computing," IEEE Multimedia, pp.34-43, Fall 1994.  
 [5] 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  
 [6] 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.  
 [7] S. W. Smoliar and H. J. Zhang, "Content-based video indexing and retrieval," IEEE Multimedia, pp.62-72, 1994.  
 [8] 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.  
 [9] 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.  
 [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 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.  
 [15] K. Hoashi, M. Sugano, K. Matsumoto, F. Sugaya, and Y. Nakajima: Shot boundary determination on MPEG

- compressed domain and story segmentation experiments for TRECVID 2004, Proc of TRECVID 2004, <http://www-nlpir.nist.gov/projects/tvpubs/tvpapers04/kddi.pdf>, 2004.
- [16] M. Sugano, K. Hoashi, K. Matsumoto, F. Sugaya, and Y. Nakajima: Shot boundary determination on MPEG compressed domain and story segmentation experiments for TRECVID 2003, Proc of TRECVID 2003, <http://www-nlpir.nist.gov/projects/tvpubs/tvpapers03/kddi.final2.paper.pdf>, 2003.
- [17] Amir, A., Berg, M., Chang, S.-F., Hsu, W., Iyengar, G., Lin, C.-Y., Naphade, M., Natsev, A. P., Neti, C., Nock, H., Smith, J. R., Tseng, B., Wu, Y., and Zhang, D. "IBM research TREC-2003 video retrieval system," *TREC Video Retrieval Evaluation (TRECVID 2003)*, Gaithersburg, MD, NIST, 2003
- [18] Rainer Lienhart. "Comparison of Automatic Shot Boundary Detection Algorithms," *Storage and Retrieval for Still Image and Video Databases VII 1999*, Proc. SPIE 3656-29, Jan. 1999.
- [19] J. Adcock, A. Girgensohn, M. Cooper, T. Liu, L. Wilcox, E. Rieffel, "FXPAL Experiments for TRECVID 2004," *TREC Video Retrieval Evaluation (TRECVID 2004)*, Gaithersburg, MD, NIST, 2004
- [20] Lienhart, R. "Reliable Transition Detection In Videos: A Survey and Practitioners Guide," *International Journal of Image and Graphics (IJIG)*, 1(3):469-486, 2001
- [21] Y. Kanazawa and K. Kanatani, "Image mosaicing by stratified matching," *Image and Vision Computing*, Vol. 22, No. 2 (2004-2), pp. 93-103.
- [22] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the 4<sup>th</sup> Alvey Vision Conference*, pp. 147-151, 1988.
- [23] K. JINZENJI et al.: "Global Motion Estimation for Sprite Production and Application to Video Coding," *IEICE D-II Vol. J83-D-II No. 2* pp. 535-544(in Japanese)
- [24] S. young. Et al.: *The HTK Book (for HTK Version 3.3)* <http://htk.eng.cam.ac.uk/>.
- [25] P. Montesinos, V. Gouet, and R. Deriche, "Differential Invariants for Color Images," *In Proceedings of 14<sup>th</sup> International Conference on Pattern Recognition*, Brisbane, Australia, 1998.
- [26] Y. P. Tan, D. D. Saur, S. R. Kulkarni, P. J. Ramadge, "Rapid Estimation of Camera Motion from Compressed Video with Application to Video Annotation," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 10, No. 1, pp. 133-146, 2000.
- [27] K. Crammer and Y. Singer, "On the Algorithmic Implementation of Multi-class SVMs," *Journal of Machine Learning Research*, 2001.