# Determination of Shot Boundary in MPEG Videos for TRECVID 2007

Jinchang Ren, Jianmin Jiang and Juan Chen

Digital Media and Systems Research Institute, School of Informatics,
University of Bradford, BD7 1DP, United Kingdom

{j.ren, j.jiang1, j.chen12}@bradford.ac.uk

0. STRUCTURED ABSTRACT

Detection of shot boundary plays important roles in many video applications. Herein, a novel method on shot boundary detection from compressed video is proposed. Firstly, we extract several local indicators from macroblocks, and these features are used in determining candidate cuts via rule-based decision making into five sub-spaces. Then, global indicators of frame similarity between start and end frames of candidate cuts are examined, using fast phase correlation on cropped DC images. Gradual transitions like fade and dissolve as well as combined shot cuts are also identified in compressed domain. Experimental results on the test data from TRECVID 2007 have demonstrated the effectiveness and robustness of our proposed methodology. Moreover, our submissions can achieve nearly 5 times faster than real-time video play (25 frames/s) due to the nature of its compressed-domain processing, achieving additional advantages in terms of processing speed and computing costs.

*1) Briefly, what approach or combination of approaches did you test in each of your submitted runs? (please use the run id from the overall results table NIST returns)*

We have three switches and by turning them on or off we have generated 8 different runs, i.e. UB_IAIS0, UB_IAIS1, UB_IAIS4, UB_IAIS5, UB_IAIS8, UB_IAIS9, UB_IAIS12, and UB_IAIS13. The other run with a system ID UB_IAIS15 is designed separately. These three switches are: 1) To consider 3-frame event as cut or gradual transition; 2) To merge combined shots as a whole or separate cuts; 3) To consider change of half frame as a cut or not.

- UB_IAIS0: One basic run with all three switches turned off;
- UB_IAIS1: One basic run with the first switch on and the other two off;
- UB_IAIS4: One basic run with the 2nd switch on and the other two off;
- UB_IAIS5: One basic run with the first two switches on and the third off;
- UB_IAIS8: One basic run with the third switch on but the other two off;
- UB_IAIS9: One basic run with the first and the third switches on and the 2nd off;
- UB_IAIS12: One basic run with the 2nd and 3rd switches on and the 1st off;
- UB_IAIS13: One basic run with all the three switches on;
- UB_IAIS15: A relative separate submission using traditional features without post-processing of phase-correlation to remove false alarms.

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

Except for UB_IAIS15, all other eight runs share same concepts in algorithm design. The only difference between them is turning on or off of three switches to deal with complex cases for robustness, and this has slightly changed the detected results in fusion stage.

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

First of all, it is our extracted features which demonstrate very well in such a context. Then, the classification of cuts into five categories has brought benefits in implementing effective detection. Moreover, phase-correlation for post-processing help to reduce about 3% false alarms. Detecting of combined shots has improved the effectiveness and accuracy in determination of gradual transitions. The overall improvement in comparison with traditional methods is about 5%.

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

Firstly, we prefer compressed domain processing as it is straightforward for MPEG videos, and this can help to extract motion features inside and skip expensive decoding for real-time applications. We find when shot transitions occur there are significant changes in luminance and/or chromatic intensity no matter how bright or dark the original frames are, and this has brought our attention to consider change ratios rather than absolute values of frame difference. Then, to reduce false alarms caused by motion, either from camera or objects, we propose phase-correlation methods in DCT images as it

provides overall measurement of frame correlation/similarity and can help to reduce such errors.

## 1. INTRODUCTION

Detection of shot boundary for video segmentation is not a new topic, which was originally introduced decades ago to detect abrupt cuts in videos [1-4]. From then on, many techniques have been developed in either compressed domain or uncompressed domain. Recently, a formal study of shot boundary detection is presented [5] in which the problem is basically divided into three parts, namely feature extraction (or visual contents representation), constructing continuity signal (via similarity measurement), and decision (or classification). With features extracted from compressed and/or uncompressed domain, a continuity signal is constructed which represents similarity between neighboring frames. As for decision, rule-based approaches and statistical machine learning like SVM and neural network become more popular than traditional threshold-based approaches, even the latter may cover some adaptability [7].

Regarding feature extraction, pixel-based methods are perhaps the simplest ones for shot detection, but they are sensitive to either motions or lighting changes, and motion compensation is usually utilized to obtain a more reasonable measurement of frame differences [8]. Then, histogram-based approaches, including luminance and chromatic histograms, are introduced, as they are invariant or insensitive to motions [5]. Although texture and edge (including edge change ratio) information is useful in image segmentation, they are less effective in shot detection possibly due to the fact that these features are not dominant in general video sources. On the other hand, motion feature is widely adopted in shot cut detection or other video processing applications in both compressed domain and uncompressed domain [3,4].

Shot boundary detection (SBD) is the fundamental task in content-based analysis, indexing and retrieval of videos, which is also one of the three tasks from the well-known TRECVID competition. In this paper, techniques are discussed in details for our submission to TRECVID'07 on shot boundary detections, including abrupt cuts and several types of gradual transitions. Additional advantages are also achieved by our techniques due to the fact that all shot boundary detection is performed in compressed domain, achieving almost 5 times faster than real-time video play. Therefore, our contributions can be highlighted as: (i) From a very small set of features as local content indicators, abrupt shot changes are effectively detected; (ii) Phase correlation on DC images of candidate shot boundaries is utilized to remove false alarms caused by motions; (iii) Full implementation in compressed domain for efficiency.

## 2. FEATURE EXTRACTION FROM MPEG VIDEOS

In our implementation, a macroblock of $16 \times 16$ pixels in MPEG videos is taken as a basic element for motion analysis and feature extractions. Without losing generality, we take the 4:2:0 video format to describe the technique we propose for extracting local content features, where four luminance blocks and two chrominance blocks for Cb and Cr components are included.

For the $i^{th}$ input frame $f_i$, let $N_h$ and $N_v$ denote the number of macroblocks in horizontal and vertical directions, and $Y_{dc}(i), U_{dc}(i)$ and $V_{dc}(i)$ represent the corresponding DC images of Y, Cb and Cr components. If we denote $N_y, N_{cb}$ and $N_{cr}$ as the numbers of elements in $Y_{dc}(i), U_{dc}(i)$ and $V_{dc}(i)$, respectively, we can easily have

$$N_y = 4N_hN_v, \; and \; N_{cb} = N_{cr} = N_hN_v \tag{1}$$

Since all the macroblocks are intra-coded in I-frames, the corresponding DC images can be directly extracted for I-frames. As for P-frames and B-frames, weighted motion compensation is applied as the current macroblock may contain contributions from its four original neighboring blocks in the reference frame. As each DC value corresponds to the average pixel value inside the related block, the DC image provides a low-resolution version of the original frame, which presents a scaled-down visual content platform for further analysis.

Based on the extracted DC image, a motion prediction error can be defined for the *i*th frame:

$$err(i) = \frac{1}{C_i}\sum_{j=1}^{C_i}\left|Y_{dc}{}^{i}(j)\right| \tag{2}$$

where $C_i$ is the number of non-intra coded $8 \times 8$ blocks indexed by $j$.

From the luminance component of the DC image, a normalized energy of Y component in the $i^{th}$ frame can be extracted as follows:

$$E_y(i) = \frac{1}{E_{0\_y}}\sum_{j=0}^{N_y-1} Y_{dc}^2(j) \tag{3}$$

where $E_{0\_y} = (L-1)^2 N_y$ represent the maximum value of energy in Y components, and $j = 256$ is the number of intensity levels in the frame.

Since $N_y = 4N_{cb}$, in each macroblock, we have six DC values, four from Y and two from U and V components, respectively. For the convenience of description, we use the mean of the four Y values as an overall measurement of the macroblock, and thus from each of Y, U and V components, we have only one DC value corresponding to each macroblock, which can be denoted as $\overline{Y}_{dc}$.

For the $i^{th}$ frame, we define its DC-differencing image between the $i^{th}$ frame and the $(i+1)^{th}$ frame as follows:

$$\begin{aligned} D(i) = 3^{-1}\{&|\,\overline{Y}_{dc}(i) - \overline{Y}_{dc}(i+1)\,| + \\ &|\,U_{dc}(i) - U_{dc}(i+1)\,| + |\,V_{dc}(i) - V_{dc}(i+1)\,|\} \end{aligned} \tag{4}$$

Figure 1 illustrates one example of cut in four consecutive frames and their corresponding DC-images. For each two consecutive DC-images, their differencing image is extracted and shown in Fig. 2(a). For each difference image $D(i)$, we further obtain its mean and standard derivation, $\mu(i)$ and $\sigma(i)$, as follows:

$$\mu(i) = \frac{1}{N_h N_v} \sum_{x=0}^{N_h-1} \sum_{y=0}^{N_v-1} D(i)(x,y) \tag{5}$$

$$\sigma(i) = \sqrt{\frac{1}{N_h N_v} \sum_{x=0}^{N_h-1} \sum_{y=0}^{N_v-1} [D(i)(x,y) - \mu(i)]^2} \tag{6}$$

Furthermore, we define $p_1(i)$ and $p_2(i)$ as two proportions which represent the percentage of pixels in $D(i)$ that are larger than the two given thresholds $\lambda_1(i)$ and $\lambda_2(i)$ determined by $\mu(i)$.

In fact, this thresholding will help to yield two binary masks in which white and black pixels refer to those whose $\mu(i)$ are larger or smaller than $\lambda_1(i)$ and $\lambda_2(i)$, respectively. According to differencing images in Fig. 2(a), three binary masks obtained by thresholding using $\lambda_2(i)$ are given in Fig. 2(b) in which majority white pixels can be found in the middle image when a cut occurs. In comparison, the other two images in Fig. 2(b) have a small proportion of white pixels. However, this small proportion may become large when motion exists in frames as it will inevitably lead inconsistency between frames and high values in $D(i)$. To overcome this drawback, median filtering (in 3*3 window) of $D(i)$ is employed before calculating its mean and standard derivation for further adaptive thresholding. The results after median filtering of $D(i)$ and new obtained binary masks are illustrated in Fig. 2(c) and 2(d). Please note that before and after median filtering, the corresponding $p_2(i)$ in three differencing images are found as (3.8%, 96.2%, 14.6%) and (0.3%, 97.1%, 2.3%), respectively. This has clearly demonstrated that median filtering of differencing image can help to reduce the effect of local motions and produce more accurate measurement of changed blocks for more robust detection of shot changes.

## 3. DETECTING ABRUPT SHOT CHANGES

Based on the local features extracted in DCT domain as described in previous section, a feature vector $V_i$ can be further constructed:

$$V_i = (err(i), E_y(i), \mu(i), \sigma(i), p_1(i), p_2(i)) \tag{7}$$

As most of the cuts are found appearing as a peak in the sequence of $\mu(i)$ and also a small peak of $\sigma(i)$, their relative heights are defined as a change ratio in comparison with its neighboring frames as follows.

$$\mu_{left}(i) = \mu(i) / \mu(i-1) \tag{8}$$

$$\mu_{right}(i) = \mu(i) / \mu(i+1) \tag{9}$$

$$\mu_{\min}(i) = \min(\mu_{left}(i), \mu_{right}(i)) \tag{10}$$

$$\mu_{\max}(i) = \max(\mu_{left}(i), \mu_{right}(i)) \tag{11}$$

$$\sigma_{\min}(i) = \min(\frac{\sigma(i)}{\sigma(i-1)}, \frac{\sigma(i)}{\sigma(i+1)}) \tag{12}$$

$$\sigma_{\max}(i) = \max(\frac{\sigma(i)}{\sigma(i-1)}, \frac{\sigma(i)}{\sigma(i+1)}) \tag{13}$$

*1) Categories of Different Cuts*

In our system, we have category cuts into 5 sub-spaces, denoting as $\Omega_c(k) \mid k \in [1,5]$, for effective detection. In $\Omega_c(1)$ and $\Omega_c(2)$, two boundary frames of a cut almost share nothing in background and foreground. In $\Omega_c(1)$ we can find very highly change of intensity in frame images while in $\Omega_c(2)$ the intensity change seems limited though may be apparent in colors. In $\Omega_c(3)$, there is a relative large part of common background or foreground can be found. For the three subspaces of cuts above, they all satisfy $\mu_{\min}(i) > 1$ and $\sigma_{\min}(i) > 1$, i.e. a peak of $\mu(i)$ and $\sigma(i)$.

In $\Omega_c(4)$, shots satisfy $\sigma_{\min}(i) > 1$ and $\mu_{left}(i) > 1$, but $\mu_{right}(i)$ is near or smaller than 1. This corresponds to a shot followed by sudden intensity changes like the effect of flash lighting. Although this kind of shot change is not a strict cut as it is usually a three-frame event, they are considered as cut in TRECVID on evaluation. Finally, $\Omega_c(5)$ contains shot changes followed by strong motions, which leads to $\mu_{\min}(i) > 1$ but $\sigma(i)$ a very small peak or non-peak.

In our system, these five categories of cuts are classified in a sequential way, i.e. to check and see if the conditions are satisfied category by category. If conditions in the prior category are matched, a cut is detected as in the corresponding category. Otherwise, the following conditions are examined. If a change in the video sequence is found failed satisfying either category of cuts above, it is defined as "non-cut".

*2) Validating of Detected Cuts*

Although the above decision rules can successfully detect real cuts, there are still quite a few false alarms owing to camera motion like panning or object motion such as moving people. Here the well-known phase correlation method is utilized to measure the similarity of two frames as follows:

1) For two frames $f_i$ and $f_{i'}$, their DC image in Y component, $Y_{dc}(i)$ and $Y_{dc}(i')$, are taken as two coarse versions of the original images;
2) Let $\Gamma_{dc}(i)$ and $\Gamma_{dc}(i')$ be Fourier transform (represented by $\Im(\cdot)$) of $Y_{dc}(i)$ and $Y_{dc}(i')$, respectively, their phase correlation $\Pi_{dc}(i,i')$ is then obtained as follows:

$$\begin{aligned} \Gamma_{dc}(i) &= \Im[Y_{dc}(i)] \\ \Gamma_{dc}(i') &= \Im[Y_{dc}(i')] \\ \Pi_{dc}(i,i') &= \Im^{-1}\left[\frac{\Gamma_{dc}(i)\Gamma^*_{dc}(i')}{|\Gamma_{dc}(i)\Gamma^*_{dc}(i')|}\right] \end{aligned} \quad (14)$$

3) Find the global peak of maximum amplitude over $\Pi_{dc}(i,i')$ surface, and the corresponding amplitude which belongs $[0,1]$ is then taken as similarity between frames $f_i$ and $f_{i'}$.

Let $f_i$ to $f_{i+1}$ be one candidate cut detected above, and then we extract four phase correlation results on DC images below:

$$\begin{cases} c_0(i) = \Pi_{dc}(i-1,i) \\ c_1(i) = \Pi_{dc}(i,i+1) \\ c_2(i) = \Pi_{dc}(i+1,i+2) \\ c_3(i) = \Pi_{dc}(i,i+2) \end{cases} \quad (15)$$

If a real cut exists between frame $f_i$ and $f_{i+1}$, both $(f_{i-1}, f_i)$ and $(f_{i+1}, f_{i+2})$ should be pair of similar images which lead to large value of $c_0(i)$ and $c_2(i)$. On the contrary, $c_1(i)$ and $c_3(i)$ should be small due to dissimilar pair of images involved. These facts are utilized in verifying the candidate in our system. .

## 4. Detecting Gradual Transitions

After detection of abrupt changes, we need to identify boundaries of gradual transitions within each pair of neighboring cuts. Except for fade, dissolve and wipe, combined shot of cuts is also considered and the techniques are discussed below.

### *1) Detecting Combined Shots*

Combined cuts containing cuts and a series of black or white frames can be classified into two parts namely normal cuts in the boundary and monochrome frames (black or white) in the middle. To detect such patterns, frame energy is taken as priority owing to the fact that these black or white frames are normally of nearly constant energy. Most importantly, this constant energy appears greatly smaller or larger than the energy in the start or end frames of relevant normal cuts. Based on the above analysis, such combined shots can be successfully identified.

### *2) Detecting Other Gradual Transitions*

When gradual transitions of dissolve occur, usually we can find that the corresponding prediction error in several frames becomes large. At the same time, $\mu(i)$ also turns big. This can help to design our method in detecting dissolve below.

Firstly, a candidate frame of dissolve is obtained if we have $err(i) \geq t_e$, where $t_e = 15$ is a threshold. Secondly, each single candidate frame is extended to a small clip by iteratively merging its neighboring frames if the neighboring frame has its prediction error larger than one third of the average prediction error in the formed clip. Thirdly, we combine all the short clips into a whole if their distance less than 3 frames.

Then, if the whole segment found contains less than 3 frames, it is abandoned. Otherwise, this candidate is further verified by checking if the difference between its two boundary frames is large enough as a cut.

As for the event of fade, it is detected only if a fade out event followed by fade in (FOI) exists. During such a FOI process, one apparent appearance is the change of luminance intensity which shows a clear V-shape. The left and right sides of this V-shape are corresponding to fade out and fade in, respectively. To determine such changes, we firstly locate the valley of lowest energy in a temporal window of $N_w$ frames, then we locate two peaks in the left and right of this valley. If we denote $i_v$, $i_{left}$ and $i_{right}$ as frame indexes of the valley and two peaks, a FOI is detected if $i_v - i_{left} > 2$, $i_{right} - i_v > 2$, and $\rho_e E_y(i_v) < \min(E_y(i_{left}), E_y(i_{right}))$, i.e. there are intermediate frames in both sides and the intensity, measured by energy, has significant changes ($\rho_e > 5$).

### *3) Fusion of Detected Results*

As we have different detectors for abrupt and gradual transitions of shot changes, a fusion procedure is necessary for a final decision and further improves both the accuracy and robustness. First of all, if two cuts are found too close to each other, say less than 5 frames, they are considered as false alarms. Then, a detected gradual transition is removed if it is found containing cut changes as cut results are taken of first priority. Next, overlapped gradual transitions are merged. Final output is generated in XML format in accordance with the requirements from TRECVID.

## 5. Evaluation Results and Discussions

We apply the proposed methodology to TRECVID 2007 data and present the results in this Section. In TRECVID 2007, in total we have 2320 shots in 17 test sequences of 637,805 frames, in which about 90% of the shots are cuts. For quantitative evaluation, ground truth data is manually extracted. It is worth noting that even after hard working there

are still some errors in these GT data such as missing or false definitions, and this is mainly due to unclear boundary of some special editing effects and massive labors involved.

Unlike news video used in previous years, test data in 2007 cover a wide range of sources from news reports, documentaries, and educational programmes to archived videos in black and white. For SBD, about 6 hours of test data in MPEG-1 is selected from 400 hours video sources provided by the Netherlands Institute for Sound and Vision (http://portal.beeldengeluid.nl/).

According to the data in TRECVID 2007, percentages of cuts in different categories are listed in Table 3 below, in which we can find these categories cover 99.63% of all cuts among the test data.

**Table 1**. Percentage of cuts in different categories, which covers 99.63% of cuts in the data from TRECVID 2007.

| $\Omega_c(1)$ | $\Omega_c(2)$ | $\Omega_c(3)$ | $\Omega_c(4)$ | $\Omega_c(5)$ |
|---|---|---|---|---|
| 11.38% | 82.00% | 3.52% | 1.26% | 1.47% |

*1) Overall Performance and Evaluation*

There are three measurements used in evaluating the results, i.e. recall and precision rate of cut detection, gradual transition, and overall performance. For gradual transitions, one additional measure is frame accuracy in locating shot boundaries. A combined measurement of both precision and recall, $F_1$ , is defined below to rank performance of different algorithms.

$$F_1(recall, precision) = \frac{2recall \cdot precision}{recall + precision} . \quad (16)$$

The overall performance of our suns in TRECVID 2007 is summarized in Table 2. According to the results in Table 2, we can easily find some facts below:

- For cut detection, the best results of our submission can achieve the recall and precision rates as 97.3% and 98.2%, respectively;
- For gradual transition detection, our submission can achieve detection rate and frame accuracy as (58.7%, 42.5%) and (82.3%, 76.1%), which need to be improved in further investigation;
- As for overall performance, our submission has a recall rate of 94.1% and precision rate of 91.9%, respectively.

*2) Performance Analysis of Post-processing*

Here, we'd like to analyze the reasons which lead to the best results of our submission on cut detection. In this experiment, phase-correlation for post-processing is removed and the performance is compared in Table 3. From Table 3 we can see that this post-processing can help to reduce about 3% false alarms of improved precision rate while degraded recall rate of 0.2%.

**Table 3**. Percentage of cuts in different categories, which covers 99.63% of cuts in the data from TRECVID 2007.

| | Recall | Precision | F1 |
|---|---|---|---|
| No post-processing | 0.986 | 0.958 | 0.972 |
| With post-processing | 0.984 | 0.987 | 0.986 |

*3) Speed Analysis*

Owing to compressed-domain processing, our system achieves a speed of about 123 frames per second on our machine (PentiumD 2.8G/1G memory), in which nearly 84% of time is used for partially decoding of MPEG stream and 16% for video segmentation. This speed is about 5 times faster than real-time playing of the video.

## 6. Conclusions

A novel method is presented for shot boundary detection in compressed MPEG videos. Taking this fundamental task as a process of decision making, we have introduced a series of rules in mapping extracted local features into five sub-spaces. Post-processing using phase correlation is found essential in eliminating false alarms caused by camera or object motion. The effectiveness and robustness of the method has been fully validated by the results on a wide range of test data from TRECVID 2007.

## Acknowledgment

This work is supported under EU IST FP-6 Research Programme with the integrated project: LIVE (Contract No. IST-4-027312).

## Bibliographies

1. C. Cotsaces, N. Nikolaidis, and I. Pitas, "Video shot detection and condensed representation: a review," IEEE Signal Proc. Mag., 23(2): 28-37, 2006.
2. Z. Rasheed and M. Shah, "Detection and representation of scenes in videos," IEEE T-Multimedia, 7(6): 1097–1105, 2005.
3. H. Fang, J. Jiang, Y. Feng, "A fuzzy logic approach for detection of video shot boundaries," Pattern Recognition, 39(11): 2092-2100, 2006.
4. Y. P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, "Rapid estimation of camera motion from compressed video with application to video annotation," IEEE T-CSVT, 10(1): 133-146, 2000.
5. J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, "A formal study of shot boundary detection," IEEE T-CSVT, 17(2):168-186, 2007.
6..
7. G. Boccignone, A. Chianese, V. Moscato, and A. Picariello, A.: "Foveated shot detection for video segmentation," IEEE T-CSVT, 15(3):365-377, 2005.
8. H. J. Zhang, S. Y. Low, and S. W. Smoliar, "Video parsing and browsing using compressed data," Multimedia Tools and Applications, 1(1): 89-111, 1995.
9. J. Bescos, G. Cisneros, J. M. Martinez, J. M. Menendez, and J. Cabrera, "A unified model for techniques on video shot

transition detection", IEEE T-Multimedia, 7(2):293-307, 2005.
10. A. Ekin, M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," IEEE T-IP, 12(7): 796-807, 2003.
11. L.-Y. Duan, M. Xu, Q. Tian, C.-S. Xu, J. S. Jin, "A unified framework for semantic shot classification in sports video," IEEE T-Multimedia, 7(6): 1066-1083, 2005.
12. NIST (National Institute of Standards and Technology), TRECVID Homepage, www-nlpir.nist.gov/projects/trecvid.
13. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," J. Computer and System Sciences, 55(1): 119-139, 1997.
14. J. Chen, J. Ren, and J. Jiang, "Compressed-domain shot boundary detection using finite state machine and content-based rules," to appear in Asia-Pacific Workshop on Visual Information Processing, Tainan, Republic of China, Dec. 2007
15. U. Gargi, R. Kasturi, and S. H. Strayer, "Performance Characterization of Video-shot-change Detection Methods", IEEE T-CSVT, 10(1): 1-13, 2000.
16. R. M. Ford, C. Robson, D. Temple, and M. Gerlach, "Metrics for shot boundary detection in digital video sequences", Multimedia System, 8(1): 37-46, 2000.
17. R. Lienhart, "Reliable transition detection in videos: a survey and practitioner's guide," Int. J. Image and Graphics (IJIG), 1(3): 469-486, 2001.
18. S. Porter, M. Mirmehdi, and B. Thosmas, "Temporal video segmentation and classification of edit effects," Image Vision Comput., 21(13-14): 1097-1106, 2003.
19. V. Kobla, D. Doermann, and K.-I. Lin, "Archiving, indexing, and retrieval of video in the compressed domain," In: Proc. SPIE. 2916, pp. 78-89, 1996.
20. J. Meng, Y. Yuan, and S. -F. Chang, "Scene change detection in a MPEG compressed video sequence," In: Proc. SPIE, 2417, pp. 14-25, Feb. 1995.
21. S. -C. Pei, and Y. -Z. Chou, "Efficient MPEG compressed video analysis using macroblock type information," IEEE T-Multimedia, 1(4): 321-333, 1999.
22. T. Vlachos, "Cut detection in video sequences using phase correlation," IEEE Signal Processing Letters, 7(7): 173-175, 2000.
23. B. L. Yeo, and B. Liu, "Rapid scene analysis on compressed video," IEEE T-CSVT, 5(6): 533-544, 1995.
24. 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", http://www-nlpir.nist.gov/projects/tvpubs/ tv.pubs.org.html#2003.
25. T. Y. Liu, K. T. Lo, X. D. Zhang, and J. Feng, "A new cut detection algorithm with constant false-alarm ratio for video segmentation," J. Vis. Commun. Image R., 15(2): 132-144, 2004.
26. P. Bouthemy, M. Gelgon, and F. Ganansia, "A unified approach to shot change detection and camera motion characterization," IEEE T-CSVT, 9(7): 1030-1044, 1999

**Figure 1**. Example of one cut in four consecutive frames with the original frame images (top) and their corresponding DC images (bottom) (enlarged for better visualization).

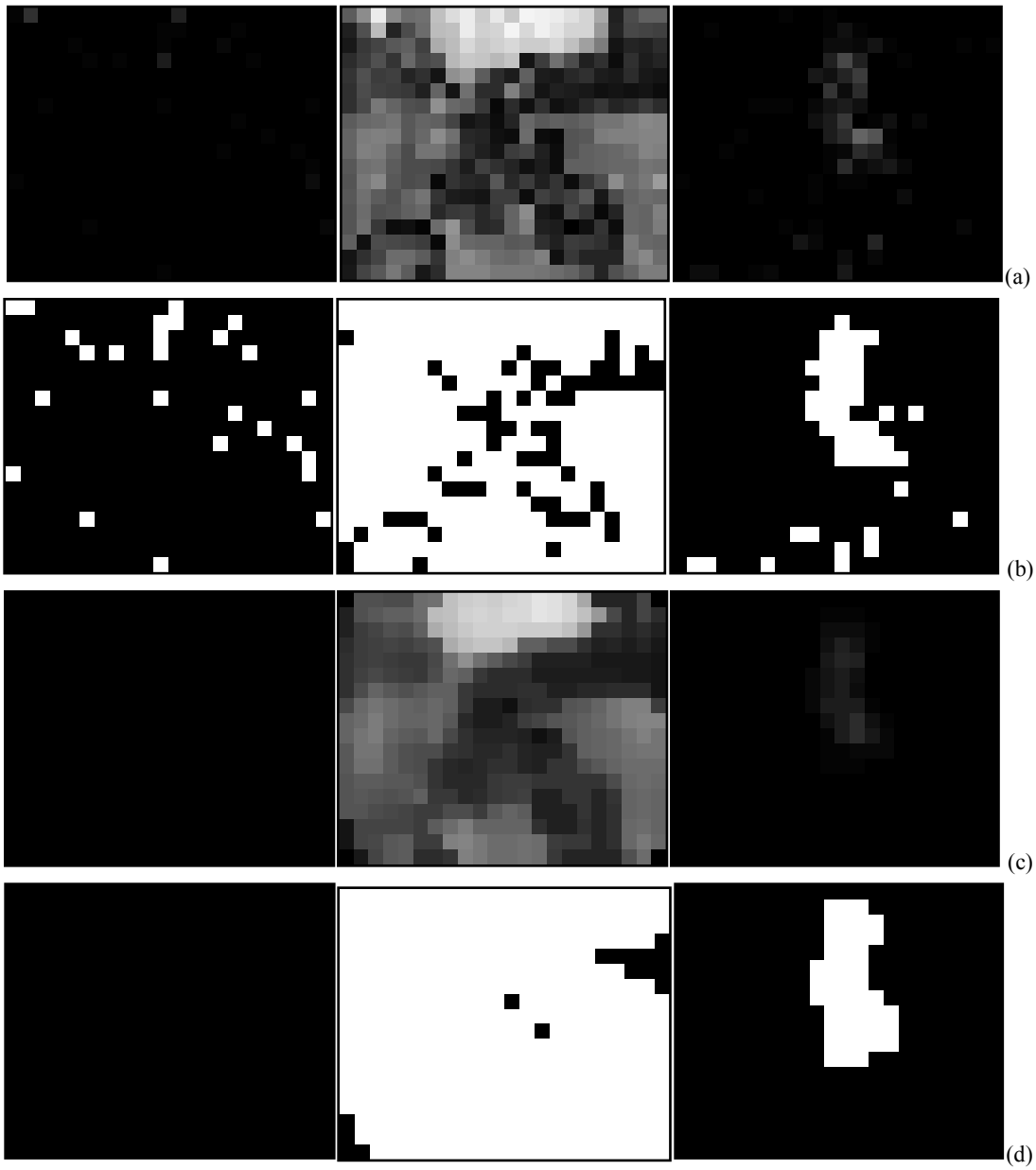

**Figure 2**. Three differencing images of each two DC images in Fig. 1 (a) and their corresponding binary masks after adaptive thresholding (b). Results in (c) are those by median filtering of (a) and their binary masks are given in (d).

**Table 2**. Results of our submitted runs, and the best results are highlighted in grey cells.

| Run ID | All Transitions | | | Cuts | | | Gradual Transitions (GT) | | | Frame Accuracy of GT | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Recall | Precision | F1 | Recall | Precision | F1 | Recall | Precision | F1 | Recall | Precision | F1 |
| AIS0 | 0.938 | 0.913 | 0.925 | 0.973 | 0.975 | 0.974 | 0.558 | 0.412 | 0.474 | 0.796 | 0.727 | 0.760 |
| AIS1 | 0.939 | 0.913 | 0.926 | 0.973 | 0.976 | 0.974 | 0.558 | 0.412 | 0.474 | 0.796 | 0.727 | 0.760 |
| AIS4 | 0.939 | 0.911 | 0.925 | 0.973 | 0.973 | 0.973 | 0.558 | 0.412 | 0.474 | 0.796 | 0.727 | 0.760 |
| AIS5 | 0.939 | 0.912 | 0.925 | 0.974 | 0.973 | 0.974 | 0.558 | 0.412 | 0.474 | 0.796 | 0.727 | 0.760 |
| AIS8 | 0.940 | 0.919 | 0.929 | 0.973 | 0.982 | 0.977 | 0.587 | 0.425 | 0.493 | 0.823 | 0.761 | 0.791 |
| AIS9 | 0.941 | 0.919 | 0.930 | 0.973 | 0.982 | 0.977 | 0.587 | 0.425 | 0.493 | 0.823 | 0.761 | 0.791 |
| AIS12 | 0.941 | 0.917 | 0.929 | 0.973 | 0.979 | 0.976 | 0.587 | 0.425 | 0.493 | 0.823 | 0.761 | 0.791 |
| AIS13 | 0.941 | 0.917 | 0.929 | 0.973 | 0.980 | 0.977 | 0.587 | 0.425 | 0.493 | 0.823 | 0.761 | 0.791 |
| AIS15 | 0.889 | 0.887 | 0.888 | 0.920 | 0.952 | 0.936 | 0.553 | 0.394 | 0.46 | 0.792 | 0.718 | 0.753 |