Modelling of Content-Aware Indicators for Effective Determination of Shot Boundaries in Compressed MPEG Videos

Juan Chen, Jinchang Ren and Jianmin Jiang

Digital Media and Systems Research Institute, University of Bradford, United Kingdom
{J.Chen12, J.Ren, J.Jiang1}@bradford.ac.uk

Abstract. In this paper, a content-aware approach is proposed to design multiple test conditions for shot cut detection, which are organized into a multiple phase decision tree for abrupt cut detection and a finite state machine for dissolve detection. In comparison with existing approaches, our algorithm is characterized with two categories of content difference indicators and testing. While the first category indicates the content changes that are directly used for shot cut detection, the second category indicates the contexts under which the content change occurs. As a result, indications of frame differences are tested with context awareness to make the detection of shot cuts adaptive to both content and context changes. Evaluations announced by TRECVID 2007 indicate that our proposed algorithm achieved comparable performance to those using machine learning approaches, yet using a simpler feature set and straightforward design strategies. This has validated the effectiveness of modelling of content-aware indicators for decision making, which also provides a good alternative to conventional approaches in this topic.

Keywords: Shot boundary detection, content-aware modelling, decision-tree, finite state machine (FSM), compressed domain video processing.

Corresponding author: Dr. Jinchang Ren

Contact details:

Tel. +44-(0)1274-235462 Fax. +44-(0)1274-233727 Email: j.ren@Bradford.ac.uk

Postal Address:

School of Computing, Informatics and Media (EIMC)
University of Bradford, Richmond Road
Bradford, BD7 1DP
United Kingdom

1 Introduction

Shot boundary detection is a major and important task for all content based video processing, analysis, and applications. For the past decades, numerous algorithms and techniques have been reported in the literature. Recent research trends in semantics based video content analysis [1,2] requires shot boundary detection as the first step to divide video sequences into sections maintaining a certain level of visual consistency so semantics can be extracted within content consistent sections [4,11,12,15]. To provide objective evaluations, the well-known TRECVID activity is introduced as an annual event to gather and measure relevant techniques using their collected massive video data with ground truth maps [3,5,7,13], and the work in this paper formed part of our submissions to the TRECVID 2007.

To detect shot cuts, most of the existing work reported in the literature measures the content difference between the current frame and its preceding one, and then apply a threshold to decide whether the difference measured is large enough to justify a cut detection [6,9,14]. Many algorithms have been developed and reported on how to decide such a threshold, which can be briefly summarized as: (i) statistics based approaches such as Bayesian rules, maximum likelihood based probabilistic modelling etc. [3,10]; (ii) empirical studies [9,14]; and (iii) machine learning approaches such as SVM, neural networks etc to bypass the threshold and make decisions based on the training and learning process [5,7,13]. The fundamental issue here, however, is that such a measured neighbourhood frame difference indicator alone would not provide sufficient information for shot cut detection.

Consequently, it is our aim to investigate content-aware approach for this task in order to make use of multiple indicators, which include our proposed four feature measurements and each of them is extracted with context information over a temporal window. With the assistance of our content-aware modelling, a top-to-down processing is employed for robustness in detecting shot transitions, using decision-tree and finite-state-machine (FSM) based techniques. The detection itself contains a coarse-to-fine procedure (initial detection followed by validations) for improved accuracy and robustness. Since the proposed approach is completely implemented in MPEG compressed domain, it benefits additionally from high efficiency achieved.

Decision trees and FSM are straightforward solutions for machine learning and data mining, which provide a great framework for both description of the relevant contexts and detection of shot boundaries within corresponding contexts. Actually, the decision tree in the paper was produced by a learning process, where sets of training videos are processed and analyzed. From the analysis, a decision tree structure is determined as such that contexts can be represented by a set of conditions. The learning process is similar to that of C4.5, yet Fisher's discriminant ratio rather than entropy is employed. In the initial detection stage, we only check the discriminative power of each single attribute in building the decision tree. While for validation, up to two attributes are checked together. If the classification error using a single attribute is comparable to that using two, this single attribute is employed for a simplified tree structure. The extended learning process will be applied to general pattern recognition tasks and reported separately.

There are several advantages for using decision trees and FSM, such as i) it uses a white-box model and is easy to understand and interpret, ii) it is able to deal with both numerical and categorical data, iii) it is robust and perform well with large data in a short time, iv) it can be validated using statistical tests. Moreover, FSM is particularly useful in modeling complex events which contain several states such as gradual transitions, such as good results reported in [27].

In comparison with our previous work [28], the novelty of this paper can be summarised as follows, though both the two papers focus on shot boundary detection in MPEG videos. Firstly, features used in this paper, including the luminance difference, edge ratio difference, motion feature, and distance-frame difference, are completely different from those in [28]. The latter utilizes statistical mean, standard derivation and the percentage of active blocks derived from difference of neighboring DC-images.

Regarding cut detection, in [28] cuts are categorized into five sub-classes and detected via a combined likelihood derived by neighboring DC-images, followed by global similarity based validation via phase-correlation on DC images to remove motion-caused false positives. In this paper, however, cuts are detected in a contextual window of 11 frames using a multiple phase decision tree, including one coarse decision stage and two validation stages. In the validation stages, both false positives and false negatives are further removed using decision trees considering motion effects. The straightforward design of the proposed algorithms in this paper is well motivated in comparison with the likelihoods defined in [28].

For gradual transitions, separate model-based approaches are used in [28] to detect fade, dissolve et al., such as fade detection via determining a V-shape in the corresponding energy curve. In this paper, however, finite state machine is used to detect both dissolve and fade effects, where fade is considered as a special case of dissolve.

The rest of the paper is organized as follows. In Section 2, the related work is summarised and analyzed. Section 3 reports our proposed content-aware approach in extracting features and constructing multiple indicators for shot detection. How the multiple content difference indicators are organized into a decision tree for abrupt cut detection and our FSM approach for dissolve detection are described in details in Section 4. Finally, we report experimental results and evaluations of the proposed algorithm in comparison with all the participating teams in TRECVID 2007 in Section 5, and provide concluding remarks to finish this paper in Section 6.

2. Related Work

Detection of shot boundary for video segmentation was originally introduced decades ago to detect abrupt cuts in videos [3, 9, 12-13]. Since then, many techniques have been developed in both the compressed and the uncompressed domains. In general, techniques in the uncompressed domain can be transferred to the compressed domain, though with lower resolution due to block-based representation of the data. On the other hand, compressed domain processing is highly desirable as it avoids the expensive inverse discrete cosine transforms (IDCT) used in video decoding. More importantly, it can make good use of many intrinsic pre-computed features in MPEG such as motion vectors and block averages for both accuracy and efficiency. Some representative techniques from both the uncompressed and the compressed domains are summarized and analyzed as follows.

In uncompressed-domain, frame difference is usually measured using pixel difference, histogram similarity, texture/edge and inter-frame correlation, etc., followed by decision making via thresholding, statistical analysis and machine learning [3, 5, 8-10, 13-14]. In Grana and Cucchiara [3], a linear transition model is developed to identify the shot transition centre and length. The proposed iterative algorithm measures the linear behaviour of shot transitions by minimizing an error function, though its performance may suffer from camera and object motions. In Fang et al. [8], colour histogram intersection, motion compensation, texture change and edge variances are integrated in a fuzzy logic framework for temporal segmentation of videos. However, it seems that relevant domain knowledge are excluded in the proposed fuzzy rules. In Yuan et al. [13], a graph partition model is employed to construct features for the SVM classification of shot boundaries, where the massive training and the complex fusion of SVM classification results are required. In Bescos et al. [14], inter-frame distance values are mapped onto a multidimensional space, and shot changes are then detected using a set of manually defined thresholds, 19 for cut detection for 10 features plus another 10 features for gradual transitions. In Cooper et al. [5], the local temporal structure of shot transitions is represented using the pair-wise inter-frame similarity derived from YUV colour histograms. A discriminative feature selection process is performed, offline, based on mutual information for the KNN (K-Nearest Neighbours) classification of video shots. In Urhan et al. [9], a hard-cut detection system is presented based on modified phase correlation with application to archived films. Video frames are spatially sub-sampled for phase correlation and the generated peaks are detected by double thresholding, i.e. the global and local thresholds. Though it benefits from phase correlation in terms of robust to illumination changes, it also suffers the drawbacks of phase correlation in dealing with non-overlapped regions, noise and motion-caused inconsistency between images. In Boccignone et al. [10], a video is partitioned into shots based on a foveated representation. The proposed method computes a consistency measure of the foveation sequences and a Bayesian inference is adopted to detect the change of consistency. Due to the requirement of computing visuomotor traces, the method is computing intensive.

Regarding shot detection in the compressed domain, most of the work using features like DCT coefficients, motion vectors, and macro-block type information to characterize shot boundaries [7, 20-21, 23-25]. However, the shot boundary detection is disturbed by special editing effects. In Cao and Cai [7] and Pei. et al [21], macro-block information is utilized for shot detection, followed respectively by a multi-class SVM (support vector machine) classifier and simple thresholding. Another compressed domain method, introduced in [20, 23, 24], extracts DC images from videos and an intensity variance sequence is generated to find "U" shape intervals for shot detection via an ART2 neural network. In practice, however, such "U" shape of intensity variance sequence often becomes indistinct due to motion, light changes, and error propagation caused by inaccurate feature extractions. For accurate shot detection, it is essential to combine such features and introduce motion adaptive measurements of the changes, such as using the magnitude of motion vectors to adjust the decision thresholds [25]. This will be considered in our content-aware modelling which will be described in details in the next sections.

3. Constructing content-aware indicators

Three frame difference measurements over a temporal window are extracted to form our content-aware indicators for shot cut detection, which include (i) neighbourhood frame difference indicators to measure the content difference between the current frame and its preceding one; (ii) inter-frame difference indicators to measure the content difference between the current frame to be examined and its preceding d^{th} frame where $d > 1$; and finally (iii) comparative frame difference indicator to measure the difference of all the indicators inside the shifting window. While the first frame difference indicator follows the same principle adopted by all existing work [6-21] that, if there exist a cut at the current frame, there should exists some content difference between the two neighbouring frames. The second frame difference indicator is mainly used to verify such a possible cut by examining the difference between the current frame and a frame some distance away [14,18] to overcome the false positives caused by factors other than cuts, such as motion, camera movement, or editing effect etc. Selection of the frame distance d is dependent on the degree of motion and other effects which may cause apparent content changes within frames. Too short distance may have inadequate content changes for shot detection, and too long distance may lead to more false alarms. In fact, such a distance has been used by others in [14] and [18] and $d = 9$ is adopted to represent 0.36s in videos at a frame rate of 25 frames per second (fps). The third frame difference indicator is to compare the indicators within a shifting window to test the consistency and remove the false detections for cases that some cuts may present small content differences.

3.1 Extracting frame difference measurements

To measure the three frame difference indicators, we propose to extract four features to construct the neighbourhood frame difference indicators and fully exploit MPEG compression techniques to enable shot cut detection to be carried out in compressed domain for efficiency. These features involve luminance, colour, edge, and motion, details of which are described as follows.

Given a MPEG compressed video input, firstly a DC image sequence Y_n is extracted. If the original video frame size is $W \times H$, the DC image will have the size of $W/8 \times H/8$. Around the current DC frame, Y_n, which is to be examined for shot cut detection, we define a shifting window with 11 neighbouring DC frames to test all the features extracted and determine whether there exists a cut or not between the frame Y_{n-1} and the frame Y_n . In other words, the proposed shot cut detection is essentially carried out inside the window of $\{Y_{n-k} \mid k \in [5,-5]\}$.

Based on the work described in [25], a normalized luminance feature is proposed to make it convenient for evaluation of all the features in a systematic and unified way. Such normalized luminance difference between the n^{th} frame and the $(n-1)^{th}$ frame is represented as D_n below:

$$D_n = (255MN)^{-1} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| y_n(i,j) - y_{n-1}(i,j) \right| . \tag{1}$$

where M and N denote respectively the number of 8×8 blocks inside video frames along the vertical direction and horizontal direction; $y_n(i,j)$ is the DC luminance value of the block positioned at (i,j) inside the n^{th} DC frame, and $D_n \in [0,1]$.

Considering our previous work on block-based edge detection directly carried out in compressed domain [26], the following edge ratio difference between the n^{th} frame and the $(n-1)^{th}$ frame is also proposed as defined below:

$$\gamma_n = (MN)^{-1} \max(| N_{ex}(n) - N_{ex}(n-1) |, | N_{ey}(n) - N_{ey}(n-1) |) . \tag{2}$$

where $N_{ex}(n)$ and $N_{ey}(n)$ denote respectively the number of vertical and horizontal block-edges in the n^{th} frame, and γ_n refers to the edge difference between the n^{th} frame and the $(n-1)^{th}$ frame.

As MPEG has the motion information available in the compressed domain, a normalized motion feature is extracted based on the MPEG motion vector $\left(V_x(i,j),\ V_y(i,j)\right)$ in the n^{th} frame as:

$$\overrightarrow{M_n} = \max(T_{vx}^{-1} \sum_{i,j} | V_x(i,j) |, T_{vy}^{-1} \sum_{i,j} | V_y(i,j) |) . \tag{3}$$

where $\left(T_{vx}, T_{vy}\right)$ is the maximum allowable motion vector designed by MPEG.

Since shot transitions may occur with limited intensity changes but apparent chromatic differences, especially for gradual transitions, another distance-frame difference Δ_n is defined to include all YUV components using three histograms $H_n^{(Y)}$, $H_n^{(U)}$ and $H_n^{(V)}$. With 32 bins being contained in each histogram, Δ_n is defined below where the parameter d denotes a frame distance as mentioned before.

$$\begin{aligned}\Delta_n &= 1-3^{-1}[N_y^{-1}\sum\nolimits_i D(H_n^{(Y)}(i)) + N_u^{-1}\sum\nolimits_i D(H_n^{(U)}(i)) + N_v^{-1}\sum\nolimits_i D(H_n^{(V)}(i))], \\ D(H_n^{(.)}(i)) &= \min(H_n^{(.)}(i) - H_{n-d}^{(.)}(i)); \quad i \in [1,32]\end{aligned} \quad . \tag{4}$$

where N_y, N_u, N_v are the corresponding number of 8×8 blocks for Y, U, V components.

Consequently, the features defined above can be employed as content features, which can be directly used to indicate the content difference and thus detect the shot cuts, or context features, which can be used to indicate the contexts of the content changes. For example, the luminance and colour can be readily used as content features since both of them are primarily used to represent the visual information in all image generation process (such as TV, cameras, printing etc.). Yet motion and edges can be used as context features since both of them mainly reflect the activities inside the captured visual scenes. In this way, shot cut detection can be made adaptive to the context changes as well as content changes. When motion is high, for example, it indicates that proportional content difference is caused by motion rather than by cuts, and thus the threshold should be moved higher. To this end, we have assembled a training video set drafted from the TRECVID test sequences in 2001 and 2005, and carried out empirical studies by extracting the neighbourhood frame difference indicators for all the four features for appropriate decision making.

3.2 Extracting context-ware indicators

With the extracted feature measurements in terms of the luminance difference D_n, edge ratio difference γ_n, motion feature $\overrightarrow{M_n}$, and distance-frame difference Δ_n, shot cuts can be determined via examining if these measurements are sufficient enough against one or more given thresholds to indicate corresponding shot transitions [17, 21]. Since the content changes have different appearances and may lead to a wide range in measuring these features, such simple thresholding seems lack of robustness. As a result, we propose to model content-aware indicators from these measurements for decision making as follows.

Firstly, for each content feature the dominant range of its values can be determined within an interval [T_L, T_H] via statistical analysis in correspondence with the ground truth inside a training set. In fact, T_L is a value where no more than 10% of all corresponding shot transitions have their values below it; and T_H is a value where no more than 10% of shot transitions have their values above it. Details of determined ranges for these features are summarised in Table-I. If the corresponding feature value is smaller than T_L, it is unlikely to be a shot event. On the contrary, if the value is larger than T_H, it is almost certainly to be a shot transition. In more common cases, however, most of the values will be found lying within the range (T_L, T_H), and more contexts need to be identified to fine tune the shot detection by determining some relative change ratios as described below, rather than using the absolute values for robustness.

Secondly, several change ratios are determined and denoted as $\hat{D}_n$, $\hat{\gamma}_n$, and $\hat{\Delta}_n$ by

$$\hat{D}_n = D_n / D_{n2}, \qquad D_{n2} = \max_{m \in [-5,5], m \neq 0} (D_{n-m}) \tag{5}$$

$$\hat{\gamma}_n = \gamma_n / \gamma_{n2}, \qquad \gamma_{n2} = \max_{m \in [-5,5], m \neq 0} (\gamma_{n-m}) \tag{6}$$

$$\hat{\Delta}_n = \Delta_n / \Delta_{n2}, \qquad \Delta_{n2} = \max_{m \in [-5,5], m \neq 0} (\Delta_{n-m}) \tag{7}$$

where D_{n2}, γ_{n2} and Δ_{n2} respectively denote the peak value in the temporal window excluding the current value of D_n, γ_n and Δ_n.

Apparently, large change ratios above indicate significant content changes at the frame n, and this is measured by comparing the change ratios against a pre-defined threshold α. In our system, this parameter is empirically set as $\alpha = 2.2$. Larger the values of α selected, the higher the comparative peaks are required to confirm the shot transitions. How to use these extracted measurements for shot boundary detection is discussed in the next section.

4. Content-aware enabled decision making for shot boundary detection

In this section, we apply modelling of content-aware indicators for shot boundary detection. Among several shot transitions, abrupt cuts and dissolve effects are utilized to validate our techniques as they are the most commonly encountered shots in videos [3, 12]. Please note that our dissolve detector can also detect fade effects as the latter can be considered as a special case of dissolve [3, 14, 19]. Relevant technical details are presented as follows.

4.1 Detecting abrupt cut transitions

With the proposed multiple frame difference and content-aware indicators, we design the shot cut detection in a coarse-to-fine manner with three phases. The first phase is an initial shot cut detection, in which we aim at filtering through all suspicious candidates that could be abrupt cuts. Following that, two further phases are used to validate cut and non-cut candidates, respectively.

Given the current DC frame Y_n, its luminance content feature D_n is primarily used for the first phase cut detection and the whole process is illustrated in Fig. 2. As mentioned earlier, if D_n is smaller than T_L, it refers to a non-cut candidate. On the other hand, $D_n > T_H$ refers to a cut. When D_n is found within the interval of $[T_L, T_H]$, it is hard to make a decision as cut or not. As a result, additional information using our defined content-aware indicators is employed to determine cut candidates. Since the major aim in this first phase is to detect as many cuts as possible, a cut candidate is identified if at least one of its associated change ratios, including $\hat{D}_n$, $\hat{\gamma}_n$ and $\hat{\Delta}_n$, is larger than the threshold α. For all the cut candidates, they need to be further validated in the second stage. For non-cut candidates, they are verified in the third stage to recover cuts which have very limited inter-frame difference in D_n measurements but large relative change ratios.

In the second phase, the primary aim is to remove false positives by applying the principle that, if a peak value detected at Y_n in the initial phase is accompanied by another inter-frame difference peak at Y_{n-1}, this peak difference at Y_n is likely caused by factors other than a cut. As a result, the inter-frame difference indicators play leading roles in the second phase detection, and the entire process is structured into another decision tree as illustrated in Figure-3.

As seen, satisfaction of the first condition $\Delta_{n-1} \geq T_H \wedge \hat{\Delta}_n < \alpha$ establishes that the peak frame difference detected in the first phase is not caused by a true cut, since it is accompanied by a high inter-frame difference in other frame(s) within the temporal window, yet there exist no comparative peak at Y_n. As a result, the input cut candidate is detected as a false positive.

Non-satisfaction of the first condition leads to further examination of Δ_{n-1} across all the remaining regions, where $T_{M1} = T_L + (T_H - T_L)/3$ and $T_{M2} = T_L + (T_H - T_L)*2/3$ are two parameters to equally divide the whole interval of $[T_L, T_H]$ into three parts. Accordingly, Figure-3 illustrates that the remaining tests of Δ_{n-1} is arranged in terms of $\Delta_{n-1} \in C_{\Delta 3} = [T_{M2}, T_H)$, $\Delta_{n-1} \in C_{\Delta 2} = [T_{M1}, T_{M2})$, and others where $\Delta_{n-1} < T_{M1}$, respectively. Since all these regions have different strength in indicating the inter-frame difference at Y_{n-1}, we need to use other features to indicate its contexts and complete the false positive detection.

Furthermore, in the test $\Delta_{n-1} \in C_{\Delta 3} \wedge \left|\overrightarrow{M_n}\right| \geq T_H$, $\left|\overrightarrow{M_n}\right| \geq T_H$ is a context condition to improve the strength of $\Delta_{n-1} \in C_{\Delta 3}$. In other words, if $\Delta_{n-1} \in C_{\Delta 3}$ is true and meanwhile the motion feature is more than the higher threshold, indicating that the peak value between Y_n and Y_{n-1} is probably caused by motion. As a result, the initially detected cut could still be a false positive.

A positive test on $\Delta_{n-1} \in C_{\Delta 2}$ indicates that a relative weak peak is still detected at Y_{n-1}, and the effect of motion is examined using a relative lower threshold as $\left|\overrightarrow{M_n}\right| \geq T_L$. If both the two conditions hold, it refers again to false positives. Otherwise, the high frame difference is likely introduced by abrupt shot transitions, which is further verified by the relative change ratio $\hat{\Delta}_n$. If $\hat{\Delta}_n < \alpha$, it indicates a false positive; if $\hat{\Delta}_n \geq \alpha$, it is determined as a true positive of cut.

Regarding validation of non-cut candidates which are detected from the initial stage, the third phase is employed as shown in Fig. 4. Since the overall inter-frame difference is low, i.e. $D_n < T_L$, for robustness we require false negatives satisfying that their associated three relative change ratios including $\hat{D}_n$, $\hat{\gamma}_n$, and $\hat{\Delta}_n$ are all above the threshold α. Furthermore, neighbouring peak Δ_{n-1} and motion magnitude $\left|\overrightarrow{M_n}\right|$ are examined to ensure such relative content changes are not caused by motion and other effects, and this is constrained respectively by $\Delta_{n-1} < T_L$ and $\left|\overrightarrow{M_n}\right| < (T_L + T_H)/2$ to remove false negatives and recover missing cuts.

4.2 Detecting gradual transitions of dissolve effects

Dissolve effect is the most commonly used video transitions in post-production, which cross-fades from one shot to another and results from gradually scaling the intensity values of the two shots [20]. If such intensity change is modelled as a strict linear manner, a parabolic ('U' type) shape in terms of the intensity variance curve (IVC) is expected for its detection [19, 20]. However, the U shape inside IVC is often corrupted in reality due to motion, camera flash, and many other factors. Consequently, it is difficult in practice to capture such transition process, or in other words, the transition process is not sufficiently clear to be captured by the intensity variance curve. As a result, the detected shot boundaries based on such an ambiguous parabolic shape of IVC become inaccurate. In addition, misdetection of such parabolic shapes could cause error propagation, producing negative impact upon detection of other dissolves. Figure-5 illustrates an example of IVC, from which it can be seen that the parabolic shape is not sufficiently clear and thus making it difficult to detect dissolves accurately in many practical cases.

To look for an alternative feature and present stronger indication of dissolves, we have tested a range of possibilities and propose a new feature, MPEG motion compensation error indicator err_n, which is defined as follows:

$$err_n = (C_n \sigma)^{-1} \sum_{i=1}^{C_n} |DC(i)|. \quad (8)$$

where C_n is the number of inter-coded blocks, and σ is the threshold applied by MPEG to decide whether a block should be inter-coded or not.

In comparison with IVC, the MPEG motion compensation error indicator presents two advantages: (i) it can be readily extracted from MPEG compressed domain; (ii) it presents a sequence of peaks during dissolve transitions and thus can be exploited to detect dissolves. Figure-6 presents a graph for the MPEG motion compensation error for the same example video illustrated in Figure-5, from which it is seen that a sequence of peaks is present in every dissolves. While such peaks may not indicate the increase and the decrease transition for dissolves, their starting and ending locations would certainly be helpful for detection of boundaries inside the dissolves.

Unlike abrupt cuts which contain only single-frame transitions, a simple decision tree cannot be used for the detection of the dissolve effects due to its nature of complex gradual transitions. As a result, a multi-state finite state machine (FSM) is employed for dissolve detection, where several states are defined and each state has its decision rules to determine the transition boundary and verify the results. As summarized in Table II, actually four states are used from S1 to S4 which correspond to an initial state, detecting start of dissolve, detecting end of dissolve, and validation, respectively. In comparison with the work in [27], our FSM scheme features in: (i) we use only four states rather than five and two of them are defined differently; (ii) dissolve candidates are detected by monitoring the MPEG motion compensation errors in DC values.

Before applying the FSM for dissolve detection, a pre-processing is introduced to filter those frames that are not likely to be a dissolve candidate for efficiency. This is achieved using the following conditions:

$$\begin{array}{l} if \quad \Delta_n < T_L \vee err_n < T \qquad then \;\; non\_dissolve\_frames \\ else\,if \;\; \left|\overrightarrow{M_n}\right| \geq T_H \wedge \Delta_n < T_H \;\; then \;\; non\_dissolve\_frames \\ else\; possible\_dissolve\_frames. \end{array}$$

where $\Delta_n < T_L$ and $err_n < T$ respectively correspond to small inter-frame difference and low MPEG motion compensation errors, which is inconsistent with the requirements of dissolve effects. Even the inter-frame difference is high, we abandon those frames whose motion magnitude $|\overrightarrow{M_n}|$ is too high so that motion-caused false positives can be constrained. In addition, T is a new threshold introduced for err_n , which is determined as 0.015 via empirical studies as discussed earlier.

For all non-dissolve frames, we set their err_n as zero for simplicity. Accordingly, the remaining operation is focused on those frames with non-zero err_n values to detect candidates for dissolves under the principle that dissolves present a sequence of peak values in err_n. The structure of our designed FSM is illustrated in Fig. 5, and the conditions in controlling the state transitions are summarised in Table III. As can be seen, six conditions namely C1 to C6 are used to constrain the process of dissolve detection, and the motion compensation error err_n is the primary clue for this purpose. The whole detection process and details on state transition are explained as follows.

At the initial state S1, the transition from S1 to the start of dissolve state S2 is controlled by C1 when one candidate dissolve frame is detected, i.e. $err_{n-1} = 0$ and $err_n \neq 0$. At state S2, the condition C2 helps it to merge continuous candidate dissolve frames into the dissolve candidate. If a non-dissolve frame is found via C3, this indicates a potential ending of the candidate dissolve effects, thus the FSM is moved to S3 for further detection. In a special case if the length of the candidate dissolve L is too long and more than a given threshold $L_{\max}$, we directly terminate S2 and transition the FSM to S4 for verification. At the same time, condition C5 is used to enable state transition from S3 to S4 which indicates the termination of the candidate dissolve as consecutive non-dissolve frames are found more than a given gap L_{gap}. Otherwise, the FSM goes back to S2 as the gap is too small and it is not a real termination of detection. The parameter L_{gap} is useful to allow short frame gaps of small motion compensation error between those of higher errors for robustness. Finally, at state S4 the condition C6 is utilized to validate the detected candidate dissolve by examining if its length is long enough, i.e. more than $L_{\min}$, and also the average motion compensation error is more than the threshold T . If the validation is successful, a dissolve is detected. Otherwise, the FSM returns to the initial state S1 to examine newly inputted frames. In accordance with the definitions and requirements in TRECVID 2007, we set $L_{\min} = 5$, $L_{\max} = 100$ and $L_{gap} = 10$ in our algorithm and competitive results are reported in the next section.

5 Experimental Results

To evaluate the proposed algorithm, we carried out extensive experiments on a number of test video clips from the TRECVID activity, which is organized by NIST (National Institute of Standards and Technology) annually [22]. We also submitted our detection results as one of the 9 runs to participate in the shot boundary detection task in TRECVID 2007, and therefore, the experimental results reported here are mainly for TRECVID 2007 test data set, in which the resolution of the test sequences is 352×240 pixels. Table-IV provides a summary description of all test video sequences, including the total number of frames, abrupt cuts and gradual transitions within each video clip.

The computing environment used for software implementation of the proposed algorithm includes: (i) a PC with 1.73GHz CPU, 512MB memory and windows XP operating system; (ii) Microsoft VC++ 6.0 programming platform. The performances of the proposed algorithm are measured by recall rate, precision rate, and F1 rate as defined by TRECVID [17, 23]. Table-V presents the experimental results of the proposed algorithm in terms of an overall average performance over the 17 test sequences in comparison with the results of other 14 participating teams. All the results are evaluated and announced by TRECVID 2007, where the recall and precision rate figures are listed in four groups, including overall, cuts, gradual transitions and gradual transition boundary frame accuracy, in accordance with the submission requirement specified by TRECVID 2007 organizers. While the two groups, cuts and gradual transitions, directly relate to the performances on cut detection and gradual transition detection respectively, the overall recall and precision rates are worked out according to the proportion of cuts and gradual transitions inside each video sequence. The group, gradual transition boundary frame accuracy, is used by TRECVID 2007 to measure the accuracy of boundary detection for gradual transitions, and the figure specifies the percentage of detected frames that overlap with the ground truth.

To obtain a combined measurement of both precision and recall, F_1 measurement is also utilized whose definition is given below:

$$F_1(recall, precision) = \frac{2recall \cdot precision}{recall + precision} . \tag{9}$$

For different algorithms, their F_1 measurements are determined and used to rank their performances. In Table V, the F_1 measurements and the associated ranks are also given in terms of the four groups of evaluations, respectively.

In Table V, the results from us are listed under the team 'M', which can be further highlighted as follows. Among all 15 teams in TRECVID 2007, our proposed algorithm achieved the 5th best overall performance, the 5th best performance for gradual transition detection, the 4th best performance for gradual transition boundary frame accuracy, and the 6th best performance for abrupt cut detection. In addition, the runtime of the proposed algorithm and TRECVID 2007 participants is listed in Table VI for comparisons. As indicated by the ratio of runtime to real-time video playing, our algorithm is four times faster than real-time video playing and there is only one system named "J" has its processing speed and overall performance superior to ours. However, our system generates better results than that of team "J" in at least three sequences including "BG_11362", "BG_35050" and "BG_36628" in terms of overall performance, cut detection as well as frame-based gradual transition detection, despite of those with better results over one single evaluation item. In addition, it is worth noting that the runtime of ours is based on an un-optimized MPEG decoder from Berkeley [29]. When a better decoder such as MDC [30] is utilized, the runtime becomes less than 1000 seconds, i.e. the fastest system in the group of results. This on one hand has proved that both the efficiency and effectiveness of decision trees and FSM in determination of shot boundaries. On the other hand, it shows great potential to accurate locating the boundaries of gradual transitions using our straightforward decision rules.

To further evaluate the performance of the proposed algorithm, we have tested it using the data from TRECVID 2006 and the results are reported in Table VII. Also, results from the best four teams are shown for comparisons. Due to complex editing effects, the overall performance is worse than that from TRECVID 2007. However, our approach has achieved the 2nd best on cut detection, the 6th best on gradual transition detection and the 4th best on frame-based gradual transition detection. The overall performance of our system is the 5th best in terms of F1 measurement, and the 2nd fastest in the whole group of evaluations.

Although the performance on cut detection is overwhelmingly better than gradual transitions, the error in cut detection still matters due to the fact that cuts occupy more than 90% of shots in TRECVID 2007. As a result, small error rate in cut detection has inevitably led to degradation of the overall performance. Depending on the nature of the video sequence content, better results are achieved for some sequences than from others, such as poor results for the two sequences BG_36182 and BG_36628. Further analysis reveals that the relative poor performance of the proposed algorithm is largely due to the missed type defined by TRECVID 2007 as “others”, which are neither standard dissolve gradual transition nor abrupt cuts. Yet in our proposed algorithm, no techniques have been designed to target such special type of shot boundaries. Figure-6 illustrates two examples of such “others”, from which it is seen that part-(a) is very close to a cut since only a small part of the picture goes through the dissolve transition inside the second frame and then the third frame is entirely different. In part-(b) of Figure-6, the transition part is again very small, only involving a few white letters inside the middle of the picture. As the proposed algorithm is designed primarily for those standard cuts and dissolves, the performance on these kinds of “others” are poor and more dedicated detection techniques are required in the future work.

According to the report of TRECVID 2007, the majority of other systems adopt machine learning approaches for shot boundary detection, where support vector machine (SVM) is one typically used technique [13]. The results here show that modelling of content-aware indicators for decision making can achieve comparative performance to those machine learning techniques. However, the specific process and the feature set in our algorithm are much simpler than those of machine learning approaches, such as the graph partition model used in [13]. As illustrated in Table-I, such empirical study only needs to determine the ranges of feature values, where the principle is very clear that above T_L , the corresponding frame difference indicator should enable majority cuts to be detected, and above T_H , few false positives should exist. Following that, content-aware indicators such as relative change ratios are calculated to provide fine-tuned scopes for accurate decision making under complicated contexts. Similar to machine learning approaches, the proposed algorithm is also sensitive to training video sequences in special cases, such as a cut with very small content difference not occurred inside the training sequences. However, such special cases are rare in practice, which often require dedicated attention as illustrated in Figure-6. Under this circumstance, machine learning approaches will have no exception simply because learning from what happened inside the training video sequences is primarily required for all machine learning approaches to detect shot transitions. To this end, the robustness of the proposed algorithm remains the same as those machine learning approaches.

6 Conclusions

In this paper, we described a content-aware algorithm with multiple content difference indicators for shot cut detection. While this area is well researched for the past decades, our proposed algorithm has made three contributions. Firstly, we proposed a content-aware approach with multiple content difference indicators to deal with cuts and dissolves in practical cases, which are much more complicated than theoretically described or expected. The application of each individual threshold is controlled by multiple context indicators extracted in compressed domain. Secondly, the entire detection process is organized by decision trees as well as FSM to achieve operation efficiency and effectiveness in its performances. Thirdly, a coarse-to-fine procedure is introduced with pre-processing and post-processing modules to reduce the computation cost and increasing its detection reliability. Extensive experiments demonstrate that the proposed algorithm achieves promising results and performances for a well-known but complicated test set (TRECVID 2007), where video sequences present a wide range of cuts and gradual transitions under various circumstances and mixed scene changes. This has demonstrated that rule based decision making with modelling of content-aware indicators can generate comparative results to those using machine learning approaches, using a simpler feature set and straightforward design strategies, which provides a good alternative for this topic especially when sufficient representative training samples are hard to attain.

Finally, the proposed algorithm, however, also presents certain level of weakness in dealing with special types of shot cuts defined as "others" in TRECVID 2007 event. These include part of the picture incurs gradual transition, picture-in-picture transition, etc, which is the major reason why our submission was ranked lower than expected. How to accurate modelling such transitions will be the focus for our further investigation.

Acknowledgments. The authors wish to acknowledge the financial support from EU IST FP-7 Research Programme under the STREP project HERMES (Contract No. IST-216709).

References

1. J. Hoey and J. J. Little, "Value-directed human behavior analysis from video using partially observable Markov decision processes," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 7, pp. 1118-1132 , July 2007.
2. K.-C. Yang, C. C. Guest, K. El-Maleh, and P. K. Das, "Perceptual temporal quality metric for compressed video," IEEE Trans. Multimedia, vol. 9, no. 7, pp. 1528-1535, Nov. 2007.
3. C. Grana and R. Cucchiara, "Linear transition detection as a unified shot detection approach," IEEE Trans. Circuits and Systems for Video Technology, vol. 17, no. 4, pp. 483-489, April 2007.
4. S. Li and M.-C. Lee, "An efficient spatiotemporal attention model and its application to shot matching," IEEE Trans. Circuits and Systems for Video Technology, vol. 17, no.10, pp. 1383-1387, Oct. 2007.
5. M. Cooper, T. Liu, and E. Rieffel, "Video segmentation via temporal pattern classification," IEEE Trans. Multimedia, vol. 9, no.3, pp. 610-618, April 2007.

6. S. Lefèvre and N. Vincent, “Efficient and robust shot change detection,” Journal of Real-Time Image Processing, vol. 2, no. 1, pp. 23-34, October, 2007.
7. J. Cao and A. Cai, “A robust shot transition detection method based on support vector machine in compressed domain,” Pattern Recog. Letters, vol. 28, no. 12, pp. 1534-1540, Sept. 2007.
8. H. Fang, J. Jiang, and Y. Feng, “A fuzzy logic approach for detection of video shot boundaries,” Pattern Recognition, vol. 39, no. 11, pp. 2092-2100, November 2006.
9. O. Urhan, M. K. Gullu, and S. Erturk, “Modified phase-correlation based robust hard-cut detection with application to archive film,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 6, pp. 753- 770, June 2006.
10. G. Boccignone, A. Chianese, V. Moscato, and A. Picariello, “Foveated shot detection for video segmentation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 3, pp. 365- 377, March 2005.
11. Z. Rasheed and M. Shah, “Detection and representation of scenes in videos,” IEEE Transactions on Multimedia, vol. 7, no. 6, pp. 1097–1105, Dec. 2005
12. C. Cotsaces, N. Nikolaidis, and I. Pitas, “Video shot detection and condensed representation. a review,” IEEE Signal Processing Magazine, vol. 23, no. 2, pp. 28-37, March 2006.
13. J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, “A formal study of shot boundary detection,” IEEE Trans. Circuits and Systems for Video Technology, vol. 17, no. 2, pp. 168-186, Feb. 2007.
14. J. Bescos, G. Cisneros, J. M. Martinez, etc., “A unified model for techniques on video-shot transition detection,” , IEEE Trans. Multimedia, vol. 7, no. 2, pp. 293- 307, April 2005.
15. U. Gargi, R. Kasturi, and S. H. Strayer, “Performance characterization of video-shot-change detection methods,” IEEE Trans. Circuits and Systems for Video Technology, vol. 10, no. 1, pp. 1-13, Feb 2000.
16. R. M. Ford, C. Robson, D. Temple, and M. Gerlach, “Metrics for shot boundary detection in digital video sequences,” Multimedia Systems, vol. 8, no. 1, pp. 37 - 46, January 2000.
17. 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. Visual Communication and Image Representation, vol. 15, no. 2, pp. 132-144, June 2004.
18. A. Hanjalic, “Shot Boundary Detection: Unraveled and resolved?” IEEE Trans. Circuits and Systems for Video Technology, vol. 12, no. 2, pp. 90-105, Feb 2002.
19. R. Lienhart, “Reliable transition detection in videos: a survey and practitioner’s guide,” Int. J. Image and Graphics, vol. 1, no. 3, pp. 469 – 486, July 2001.
20. J. Meng, Y. Yuan, and S.-F. Chang, “Scene change detection in a MPEG compressed video sequence,” Proc. SPIE. vol. 2419, pp. 14-25, 1995.
21. S.-C. Pei and Y.-Z. Chou, “Efficient MPEG compressed video analysis using macroblock type information,” IEEE Trans. Multimedia, vol. 1, no. 4, pp. 321-333, Dec 1999.
22. Available online, http://www-nlpir.nist.gov/projects/trecvid/trecvid.data.html.
23. S. Porter, M. Mirmehdi, and B. Thosmas, “Temporal video segmentation and classification of edit effects,” Image and Vision Computing, vol. 21, no. 13-14, pp. 1097-1106, Dec. 2003.
24. B. L. Yeo, and B. Liu, “Rapid scene analysis on compressed video,” IEEE Trans. Circuits and Systems for Video Technology, vol. 5, no. 6, pp. 533-544, Dec 1995.
25. K. Matsumoto, M. Sugano, etc., “Shot boundary detection and low-Level feature extraction experiments for TRECVID 2005,” Available online, http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org. html#2005
26. K. Qiu, J. Jiang and G. Xiao, “An edge based content descriptor for content based image and video indexing,” Lecture Notes in Computer Science, vol. 4141, no. 1, pp. 673-684, 2006.
27. Z. Liu, D. Gibbon, E. Zavesky, B. Shahraray, and P. Haffner, “AT&T research at TRECVID 2006,” Available online, http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html#2006

28. J. Ren, J. Jiang and J. Chen, “Shot boundary detection in MPEG videos using local and global indicators,” IEEE Trans. Circuits and Systems for Video Technology, vol. 19, no. 8, pp.1234-1238, 2009.

29. K. Patel, B. C. Smith and L. A. Rowe, “Performance of a software MPEG video decoder,” Prof. 1st ACM Int. Conf. Multimedia, pp. 75-82, August 1993.

30. D. Li and Ishwar K. Sethi, “MDC: A Software Tool for Developing MPEG Applications,” Proc. IEEE International Conference on Multimedia Computing and Systems (ICMCS'99), vol. 1, pp. 445-450, 1999.

Table-I Summary of the value ranges in terms of several inter-frame difference indicators

Content/Values	D_n **(luminance)**	γ_n **(edge)**	$\lvert\overrightarrow{M_n}\rvert$ **(motion)**	Δ_n **(inter-frame difference of luminance and colour)**
Lower bound (T_L)	0.027	0.02	0.06	0.14
Higher bound (T_H)	0.09	-	0.2	0.3

Table-II FSM states description

FSM States	**Descriptions**
S1	Initial state
S2	Detection state for the beginning frame of a possible dissolve candidate
S3	Detection state for the ending frame of a possible dissolve candidate
S4	Verification state

Table-III Definition of conditions for inter-state transition in FSM

Conditions	**Definitions**	**Comments**
C1	$err_n \neq 0 \wedge err_{n-1} = 0$	A candidate dissolve frame is found at the n^{th} frame, and set $B = n$;
C2	$err_n \neq 0 \wedge err_{n+1} \neq 0$	The coming dissolve frame can be expanded, update $E = n$.
C3	$err_n \neq 0 \wedge err_{n+1} = 0$	Find a non-dissolve frame as potential ending of transition, $E = n$;
C4	$L = E - B + 1 \geq L_{\max}$	The detected candidate is too long to absorb the coming dissolve frame
C5	$\forall(err_{E+k} = 0), k \in [1, L_{gap}]$	The number of non-dissolve frames is larger than the defined gap;
C6	$L > L_{\min} \wedge L^{-1}\sum_{i=B}^{E} err_i > T$	Validate the candidate via comparing its length and average motion prediction error respectively against two thresholds $L_{\min}$ and T.

Table-IV Description of the video sequence in the test set

Video name	**Number of frames**	**Abrupt cut count**	**Gradual transition count**	**Sum of all shots**
BG_2408	35892	103	18	121
BG_9401	50049	89	3	92
BG_11362	16416	104	4	108
BG_14213	83115	107	60	167
BG_34901	34389	225	15	240
BG_35050	36999	100	2	102
BG_35187	29025	135	23	158
BG_36028	44991	87	0	87
BG_36182	29610	96	13	109
BG_36506	15210	77	6	83
BG_36537	50004	259	30	289
BG_36628	56564	196	6	202
BG_37359	28908	165	5	170
BG_37417	23004	84	4	88
BG_37822	21960	120	9	129
BG_37879	29019	95	4	99
BG_38150	52650	215	4	219
In total	637805	2257	206	2463

Table-V. Comparison of average performance for all teams in TRECVID 2007, and the team 'M' refers to our results.

Team	Overall			Cut			Gradual transition			Gradual transition Frame-based		
	F1 (rank)	**Recall**	**Precision**	**F1 (rank)**	**Recall**	**Precision**	**F1 (rank)**	**Recall**	**Precision**	**F1 (rank)**	**Recall**	**Precision**
A	0.834 (9)	0.7435	0.9505	0.872 (10)	0.8035	0.9540	0.156 (12)	0.0900	0.6785	0.672 (8)	0.5365	0.9315
B	0.946 (1)	0.9419	0.9506	0.968 (2)	0.9718	0.9639	0.660 (1)	0.6142	0.7579	0.818 (1)	0.7312	0.9279
C	0.865 (7)	0.9220	0.8210	0.960 (3)	0.9614	0.9591	0.272 (8)	0.4909	0.2555	0.718 (7)	0.6628	0.7927
D	0.897 (4)	0.8879	0.9120	0.937 (5)	0.9689	0.9120	0.000 (14)	0.0000	0.0000	0.000 (14)	0.0000	0.0000
E	0.781 (10)	0.8360	0.7392	0.843 (11)	0.8840	0.8094	0.229 (10)	0.3124	0.2455	0.640 (10)	0.5811	0.8105
F	0.628 (15)	0.8801	0.4995	0.816 (13)	0.9064	0.7429	0.146 (13)	0.5923	0.0937	0.597 (11)	0.6540	0.6056
G	0.845 (8)	0.8797	0.8157	0.927 (8)	0.9201	0.9346	0.282 (7)	0.4378	0.2233	0.659 (9)	0.5390	0.8510
H	0.750 (11)	0.7514	0.8646	0.767 (14)	0.7663	0.8885	0.577 (4)	0.5873	0.6458	0.748 (5)	0.6584	0.8687
I	0.885 (6)	0.9018	0.8726	0.931 (7)	0.9288	0.9337	0.453 (6)	0.5922	0.4008	0.534 (12)	0.4010	0.8013
J	0.915 (3)	0.9036	0.9284	0.946 (4)	0.9304	0.9620	0.604 (3)	0.6118	0.6298	0.772 (3)	0.6824	0.8894
K	0.941 (2)	0.9509	0.9328	0.973 (1)	0.9692	0.9763	0.658 (2)	0.7504	0.6036	0.806 (2)	0.7755	0.8382
L	0.723 (13)	0.9005	0.6171	0.919 (9)	0.9276	0.9154	0.177 (11)	0.6014	0.1332	0.738 (6)	0.7490	0.7292
M	**0.888 (5)**	**0.8890**	**0.8870**	**0.936 (6)**	**0.9200**	**0.9520**	**0.460 (5)**	**0.5530**	**0.3940**	**0.753 (4)**	**0.7920**	**0.7180**
N	0.726 (12)	0.9525	0.6080	0.824 (12)	0.9778	0.7248	0.270 (9)	0.6747	0.1972	0.410 (13)	0.2649	0.9096
O	0.677 (14)	0.7108	0.7514	0.707 (15)	0.7758	0.7514	0.000 (14)	0.0000	0.0000	0.000 (14)	0.0000	0.0000

Table-VI. Average runtime of participants in TRECVID 2007, and the total time for video play is 25512.2s.

Team	Runtime (seconds)	Ratio to video play time	Rank
A	10586.3	0.4150	9
B	7325.5	0.2871	8
C	4157.9	0.1630	3
D	17540	0.6875	12
E	611200.2	23.9572	15
F	15637.9	0.6130	11
G	3615.1	0.1417	2
H	96948.8	3.8001	14
I	5517.9	0.2163	6
J	1686.5	0.0661	1
K	12974.7	0.5086	10
L	7319.7	0.2869	7
M	**5357.6**	**0.2100**	**5**
N	4223.1	0.1656	4
O	42397.3	1.6618	13

Table-VII. Comparison of average performance of our system and results from the best four teams in TRECVID 2006.

Team	Overall			Cut			Gradual transition			Gradual transition Frame-based		
	F1 (rank)	**Recall**	**Precision**	**F1 (rank)**	**Recall**	**Precision**	**F1(rank)**	**Recall**	**Precision**	**F1(rank)**	**Recall**	**Precision**
Ref-1	0.835 (1)	0.851	0.821	0.905 (1)	0.942	0.871	0.768 (2)	0.764	0.773	0.855 (2)	0.813	0.902
Ref-2	0.830 (2)	0.880	0.786	0.871 (3)	0.925	0.823	0.792 (1)	0.837	0.751	0.899 (1)	0.915	0.883
Ref-3	0.702 (3)	0.831	0.607	0.743 (5)	0.857	0.656	0.662 (3)	0.807	0.561	0.814 (3)	0.797	0.831
Ref-4	0.690 (4)	0.718	0.664	0.848 (4)	0.911	0.793	0.538 (4)	0.535	0.541	0.551 (5)	0.414	0.823
Our	0.672 (5)	0.733	0.620	0.886 (2)	0.913	0.861	0.462 (6)	0.562	0.392	0.754 (4)	0.788	0.722

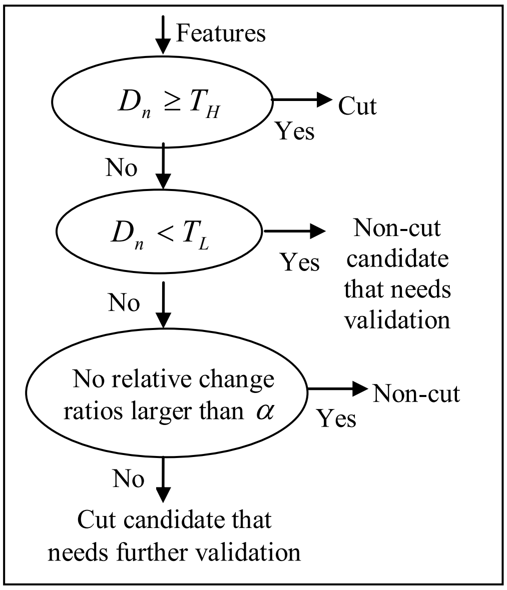
Features
$D_n \geq T_H$
Yes
Cut
No
$D_n < T_L$
Yes
Non-cut candidate that needs validation
No
No relative change ratios larger than α
Yes
Non-cut
No
Cut candidate that needs further validation

Figure-1 *Decision tree for initial shot detection*

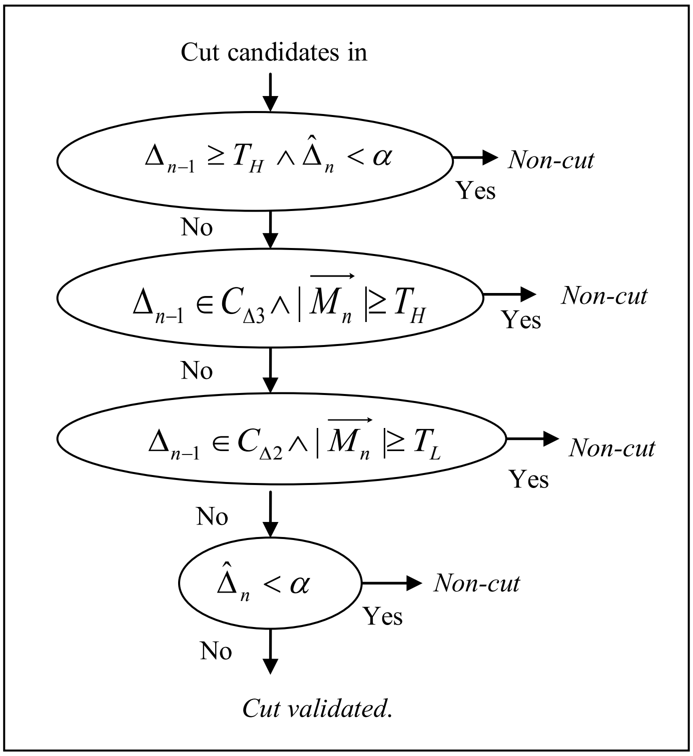
Cut candidates in
$\Delta_{n-1} \geq T_H \wedge \hat{\Delta}_n < \alpha$
Yes
Non-cut
No
$\Delta_{n-1} \in C_{\Delta 3} \wedge |\overrightarrow{M_n}| \geq T_H$
Yes
Non-cut
No
$\Delta_{n-1} \in C_{\Delta 2} \wedge |\overrightarrow{M_n}| \geq T_L$
Yes
Non-cut
No
$\hat{\Delta}_n < \alpha$
Yes
Non-cut
No
Cut validated.

Figure-2 *Decision tree for removal of false positives in cut detection.*

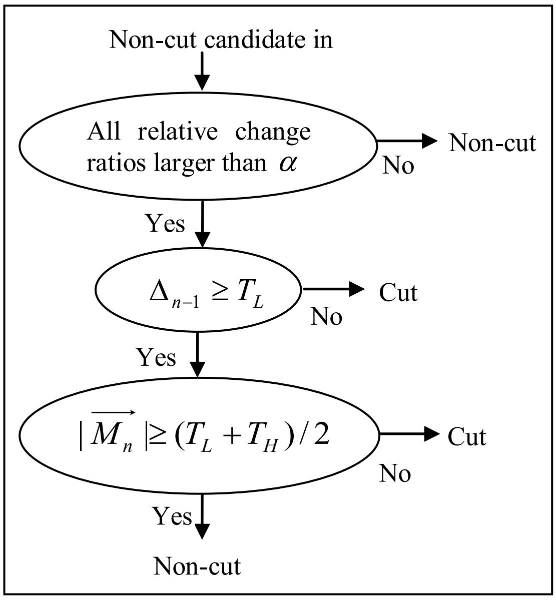
Non-cut candidate in
All relative change ratios larger than α
No
Non-cut
Yes
$\Delta_{n-1} \geq T_L$
No
Cut
Yes
$|\overrightarrow{M_n}| \geq (T_L + T_H)/2$
No
Cut
Yes
Non-cut

Figure-3 *Decision tree for removal of false negatives*

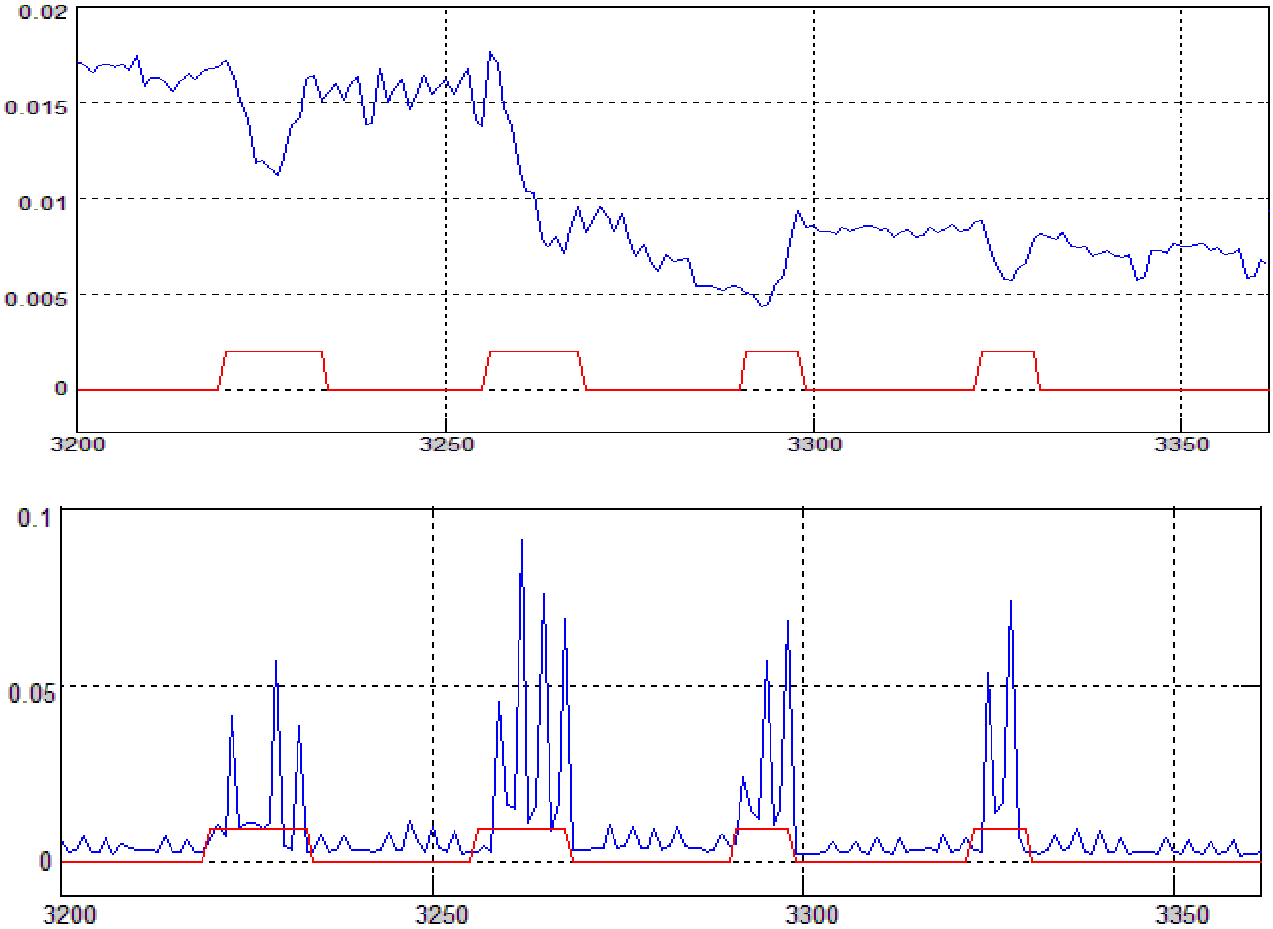

Figure-4. *Plots of feature measurements vs. frames in terms of IVC (top) and motion compensation error (bottom), where red curves indicates ground truth of dissolve effects and blue ones the corresponding features.*

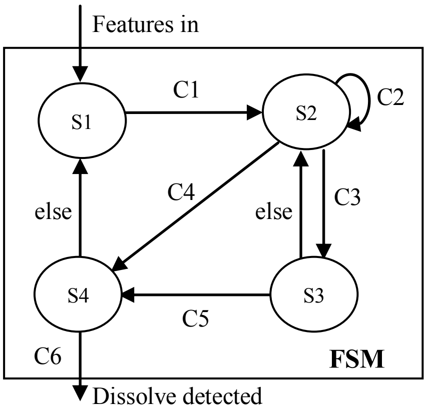

Figure-5 *Structure of FSM for dissolve detection*

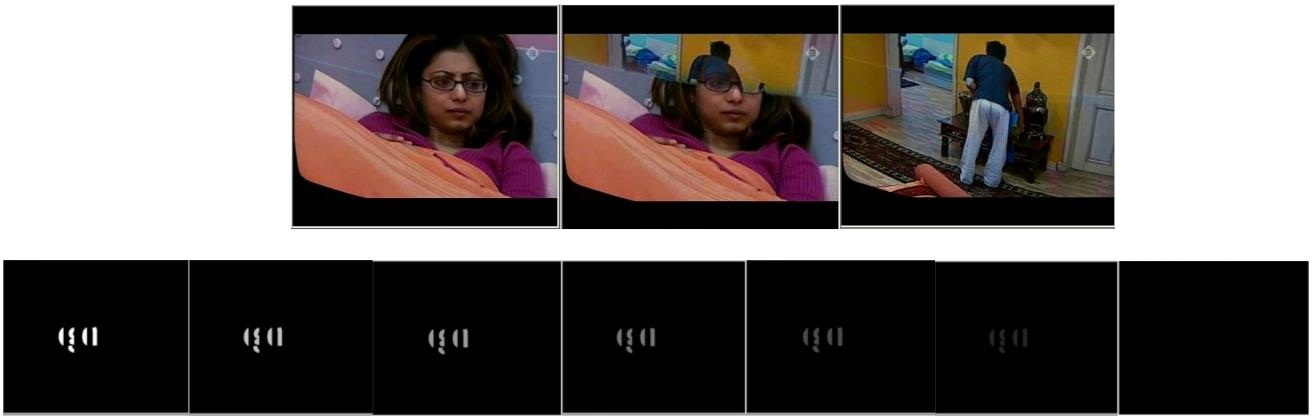

Figure-6 *Illustration two examples of "others" which are not modelled in our system.*