Adaptive Methods for Motion Characterization and Segmentation of MPEG Compressed Frame Sequences

C. Dulaverakis[1], S. Vagionitis[2], M. Zervakis[3], E. Petrakis[4,*]

Department of Electronic and Computer Engineering
Technical University of Crete
Chania, Crete, Greece
{[1]dulaver, [2]bagsta, [3]michalis}@systems.tuc.gr, [4]petrakis@ced.tuc.gr

Topic Area: Image and Video Processing and Analysis, Image and Video Coding

Keywords: MPEG video, video segmentation, motion characterization, video content, adaptive thresholding

[*] Corresponding author.

Abstract. A fast and accurate method for scene change detection and classification of camera motion effects in MPEG compressed video is proposed. The method relies on adaptive threshold estimation and on the analysis and combination of various types of video features derived from motion and intensity information. This analysis is also applied for cleaning-up false shot boundaries due to camera motion effects. Two techniques for adaptive threshold estimation are also proposed and evaluated.

1 Introduction

Temporal video segmentation is intended to partition a video into consecutive shots [1-3]. The transitions between consecutive shots can be abrupt or gradual. Video effects caused by camera panning, titling or zooming result in interframe signal or histogram transitions of the same order of magnitude as gradual transitions. Moreover, transitions caused by sharp changes in camera motion direction are of the same order of magnitude as camera breaks. Both types of effects introduce false shot boundaries (false positives). Detecting the exact type of video effects and cleaning-up the false positives is a difficult task [4-6]. This is exactly the focus of this work.

This work also focuses on gray-scale video processing and analysis directly on the compressed domain. MPEG-2 compressed information is computed for 8x8 pixel regions called *blocks,* whereas motion vectors are computed for 16x16 pixel regions called *macroblocks*. In this work, intensity and motion information is encoded in the same coherent way regardless of frame type (i.e., DC intensity and forward predicted motion vector for each macroblock for any *I*, *P* or *B* frame) [7].

The following summarizes the contributions of this work:

- An approach is proposed for the segmentation of MPEG compressed video. It allows for more reliable video segmentation than intensity histogram thresholding by correctly identifying camera motion effects and by utilizing adaptive threshold estimation in detecting such effects.
- The novelty of the method relies on the analysis and combination of various types of video features derived from motion information directly in the compressed domain. This analysis is also applied for cleaning-up false shot boundaries due to camera motion effects.
- Following the example of [8-10] two adaptive threshold estimation methods are proposed and evaluated. Compared to existing adaptive methods they are more theoretically principled and are easily integrated within a proposed simple, intuitive and fast video segmentation algorithm.

In the rest of this paper, threshold selection methods are discussed in Sect. 3; the proposed video analysis method is presented in Sect. 4; experimental results are presented and discussed in Sect. 5 followed by conclusions in Sect. 6.

2 Threshold selection

The state-of-the-art approach for automatic threshold selection is referred to as "Twin-Comparison" (TC) approach [11]. TC requires that the whole video must be scanned

once prior to segmentation and for detection of gradual transitions computes two thresholds. As globally optimum, these thresholds do not adapt to local properties of the histogram differences. The proposed threshold estimation techniques do not require preprocessing, compute only one threshold and adapt the threshold to local properties of the input signal.

2.1 Sliding Window (SW) threshold

A threshold T_a is computed over a small range of W frames (W=15 in this work). The SW method starts by taking the first W histogram differences D and for video partitioning works as follows:

Repeat until end of video stream:

1. Compute the mean $\mu(i)$ and variance $\sigma(i)$ within a range of W frames (window W), where i is the rightmost difference $D(i)$ value of W.
2. Compute threshold as $T_a(i) = \mu(i) + \alpha\sigma(i)$, where α is a user defined parameter as in TC.
3. Compare $T_a(i)$ with the next histogram difference $D(i+1)$ outside the window.
4. If $T_a(i) > D(i)$ advance W one position to the right ($i = i+1$) to include $D(i+1)$. Go to *step 1*.
5. if $T_a(i) < D(i)$ a transition is found. Increase i (move W to the right) until $D(i) < T_a(i)$. Count the number n of skipped positions. If $n < w$ ($w = 5$ in this work) a *camera break* is declared, if $n > w$ a *gradual transition* is declared.

Algorithm 1: Sliding window approach for video partitioning

Parameter a, is user defined and depends on video properties. In Fig. 1 a camera break is declared at frame 150 and a gradual transition between frames 300 and 325.

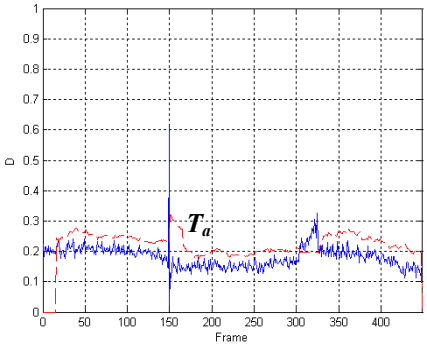

Fig. 1: Sliding window approach. The dashed line represents local values of threshold T_a

2.2 Adaptive Window (AW) threshold

This method works the same way as Algorithm 1, but T_a is defined as in an autoregressive form [12]

$$T_a(i) = \mu(i) + \alpha\sigma(i) \quad \text{with}$$
$$\mu(i) = \mu(i\text{-}1) - c\,(\mu(i\text{-}1) - D(i)), \quad \sigma(i) = |\mu(i)^2 - \lambda(i)|^{1/2} \text{ and}$$

$\lambda(i) = \lambda(i-1) - c\,(\lambda(i-1) - D(i))^2$. Moreover, $\mu(1) = D(1)$, $\lambda(1) = D(1)^2$, $\sigma(1) = 0$. Parameter c controls the sensitivity of the threshold to signal changes ($c = 0.05$).

3 Video Shot Characterization by Motion Information

A forward predicted vector is computed to each macroblock [7] except for *intracoded* macroblocks for which no matching macroblock can be detected in the next frame. In the present consideration, macroblocks with magnitude of motion vector less than 1 are called *static* (they include *skipped* macroblocks with 0 motion). In the following, comparisons involving motion information are normalized with respect to the number of macroblocks with identified motion (no intracoded or static).

3.1 Direction Histogram

A motion vector is described by a pair (u,v) representing horizontal and vertical displacement of the macroblock respectively. From this pair, the angle (direction) of motion is computed as $\theta = \tan^{-1}(v/u)$, taking values in $[0,2\pi]$. All angles are quantized into *8* directions, multiples of $\pi/4$. The notion or *direction histogram* is introduced as a tool for global motion analysis. Each bin in this histogram counts number of macroblocks in each angle range. An additional bin (bin 1) is also added in the direction histogram representing number of static macroblocks (with $|(u,v)| < 1$) in the frame. Fig. 2 illustrates the direction histogram of the scene on its left.

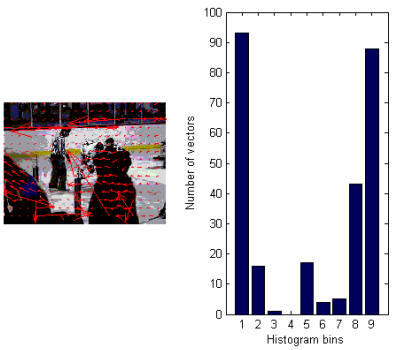

Fig. 2: Scene and its corresponding direction histogram

3.2 Camera Zooming, Panning and Tilting

The analysis is based on the following observations:

- **Zooming**: The motion vectors are equally spread on the direction histogram.
- **Panning or Tilting**: The motion vectors tend to concentrate at a single histogram bin. The position of this bin denotes the direction of camera motion.

- **Static Camera**: Most motion vectors are concentrated at bin 0; the distribution of motion vectors in the remaining bins is irrelevant.

The analysis of various video types indicates that for reliable prediction of camera motion effects, at least 40% of the total number of macroblocks in a frame must be motion predicted (no intracoded or static). The variance of the histogram provides the means for analyzing the structure of the direction histogram:

$$\sigma_h^2 = \sum_{i=1}^{9} p_i (h_i - \bar{h})^2 \text{ with } p_i = \frac{h_i}{\sum_{i=1}^{9} h_i} \text{ and } \bar{h} = \sum_{i=1}^{9} p_i i \text{ where}$$

h_i involves the histogram values at bin i and p_i represents the probability of occurrence of motion vectors with angle i (bin 1 indicates static regions). The maximum variance is encountered when the motion vectors are equally spread along the histogram while, the minimum variance is encountered when all motion vectors are concentrated at a single histogram bin. The variance is further normalized with respect to the actual number of motion vectors in each frame:

$$\sigma_{h,norm}^2 = \gamma \sigma_h^2 \text{ where } \gamma = 1 - \frac{IntracodedVectors + StaticVectors}{TotalMacroblocks\ in\ Frame}.$$

Detection of camera zooming, panning and tilting relies on the application of appropriate threshold values on the plot of normalized variance (computed from bin 2 through 9 that is, over the part of the histogram representing angles of motion). The SW method or the AW method can be applied.

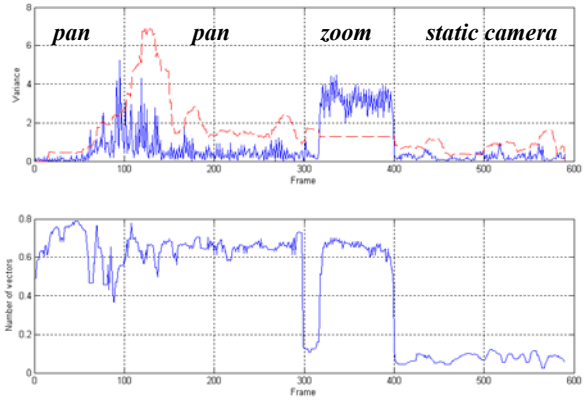

Fig. 3: Normalized variance of angles of motion vectors with two panning and one zoom areas. The diagram below illustrates number of actual motion vectors per frame

Fig. 3 illustrates the behavior of this measure on pan sequences with small variance, zoom sequences with large variance and static camera regions with small variance but with small number of motion vectors. The lower figure illustrates the number of actual motion vectors (no static and intracoded) in each frame. The dashed

line plots the SW threshold T_a. It correctly detects the zoom section, while it falsely detects isolated camera changes before frame 100, which are due to camera instability.

Algorithm 2 summarizes the above scheme (γ is the percentage of motion predicted vectors and T_v is set to 40% of the frame size measured in macroblocks).

- If $\gamma > T_v$ and $\sigma_{h,norm} < T_a$, the frame belongs to a panning or tilting sequence.
- If $\gamma > T_v$ and $\sigma_{h,norm} > T_a$, the frame belongs to a zoom sequence.
- If $\gamma < T_v$ none from the above holds (no camera motion is detected).

Algorithm 2: Detection of zooming, panning and tilting frame sequences

3.3 Camera direction changes

This kind of camera effects is characterized by changes in the distribution of motion vectors. They are detected by direct comparison of direction histograms which for two histograms f_{n-1} and f_n is defined as

$$D_H(f_n, f_{n-1}) = \sum_{i=1}^{9} \left| p_i^{f_n} - p_i^{f_{n-1}} \right| \text{ where } p_i = \frac{h_i}{\sum_{i=1}^{9} h_i}.$$

The static vectors are included in the difference, since they reflect a motion property that needs to be measured (e.g., in the transition between static camera and pan). Camera motion changes are detected by thresholding on the plot D_H differences.

3.4 Video segmentation using motion information

The above approach can also be applied to enhance existing video segmentation methods by cleaning-up false shot boundaries due to camera motion effects.

3.4.1 Camera breaks

Traditionally, camera breaks are detected when the number of intracoded vectors exceeds a threshold [13]. The SW or the AW approach is applied on the plot of the number of intracoded macroblocks of a frame sequence. The method of Sect. 3.3 is applied to clean-up the false shot boundaries. Algorithm 3 summarizes this approach

1. If the condition for camera motion changes is not satisfied (Sect. 3.3) and
2. If the number of intracoded vectors exceeds the threshold then
3. A camera break is detected.

Algorithm 3: Detection of camera breaks by combining motion information

3.4.2 Gradual transitions

When the transition between frames is extended over time (gradual transition), the majority of motion vectors behave randomly. Such gradual transitions are detected based on the variance of the magnitude of motion vectors. The variance σ_l of the magnitude l of motion vectors in each frame is computed as

$$\sigma_l^2 = \frac{1}{n_v}\sum_{i=1}^{n_v}(l_i - \bar{l})^2 \text{ where } \bar{l} = \frac{1}{n_v}\sum_{i=1}^{n_v} l_i \text{ and}$$

l_i is the magnitude of the i-th vector and n_v is the number of motion vectors with $|(u,v)| > 1$. The variance is normalized with respect to the number of actual motion vectors.

$$\sigma_{l,norm}^2 = \gamma\sigma_l^2 \text{ where } \gamma = 1 - \frac{IntracodedVectors + StaticVectors}{TotalMacroblocks\ in\ Frame}$$

Fig. 4 plots the normalized variance for all frames. The SW or the AW method can detect peaks (potential gradual transitions). The dissolves after frame 100 and 500 are not detected because the threshold adapts very fast to their gradually rising form. This is an inherent weakness of methods using adaptive windows.

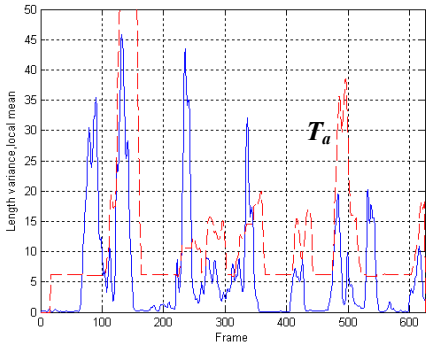

Fig. 4: Normalized variance of magnitude of motion vectors for a video stream and detected gradual transitions (peaks) by thresholds (dashed line) computed by the SW method

Notice that $\sigma_{l,norm}$ exhibits the same behavior in gradual transitions as in zoom sequences (the vectors have small magnitude at the center of the frame and high magnitude near the edges). Zooming can be distinguished from gradual transitions by Algorithm 2. For cleaning-up false positives due to camera motion, it is required that both the intensity histogram difference and the motion magnitude variance are above their thresholds. Algorithm 4 summarizes the above approach.

1. If the condition for zoom sequence (Algorithm 2) is not satisfied and
2. If the intensity condition (Algorithm 1) along with the condition on the normalized variance of the magnitude of motion vectors: $\sigma_{l,norm} > T_a$ are both satisfied then
3. A gradual transition is detected.

Algorithm 4: Detection of gradual transitions by combining motion and intensity information

4 Experimental Results

The effectiveness of each method is measured by the average (over 17 videos) precision and *recall*. Each method is represented by its precision and recall as a function of the threshold parameter a.

The plot on the left of Fig. 5 demonstrates that the AW method is particularly effective for the detection of panning or tilting achieving precision close to 1 ($a=2$) or recall close to 1 ($a=4.5$). The plot on the right demonstrates that AW is also effective for zoom detection. Higher precision is achieved for higher α and higher recall is achieved for lower α. The opposite was observed in panning and tilting since it is detected as $\sigma_{h,norm} > T_\alpha$ while zooming is detected as $\sigma_{h,norm} > T_\alpha$ (Sect. 3.2).

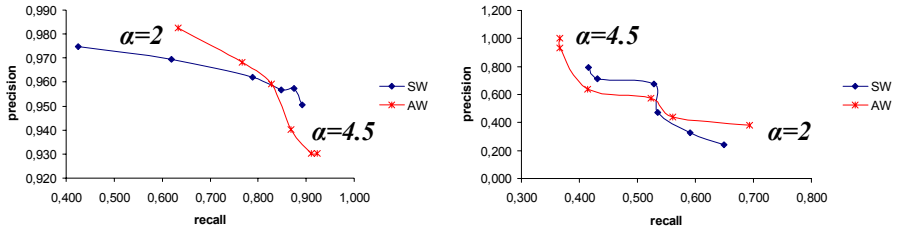

Fig. 5: Average precision and recall for panning or tilting (left) and zooming detection (right)

The plot of Fig. 6 demonstrates that the SW method achieves up to 10% better recall and almost always better precision in detecting camera motion changes.

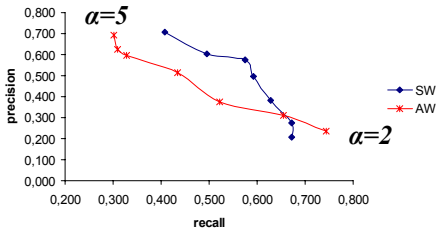

Fig. 6: Average precision and recall for camera direction change detection.

The left plot of Fig. 7 illustrates that video segmentation by the SW method on motion information (Sect. 4.3) is at least as accurate as the TC method (the test data did not contain enough sequences with camera direction changes). The plot on the right demonstrates the superiority of the proposed approach combining motion and intensity information for detecting gradual transitions.

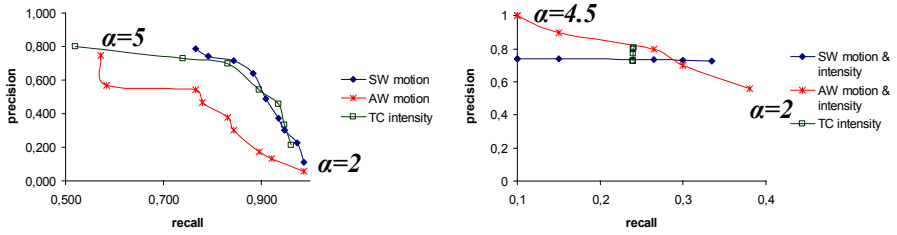

Fig. 7: Average precision and recall for detection of camera breaks (left) and gradual transitions (right).

The above experiments revealed the independence of the performance of all methods on the parameter *a*. Maximum precision and recall are achieved for the same values of *a*.

5 Conclusions

Two methods for automatic and adaptive threshold estimation on video information have been tested and proven particularly effective for video segmentation and for eliminating false positives caused by various camera effects. The novelty of the proposed approach relies on the analysis and combination of motion information in the compressed domain and adaptive threshold estimation on various signal patterns. The experimental results provide good support to the claims of efficiency.

References

1. Lefevre, S., J. Holler, and N. Vincent, A Review of Real Time Segmentation of Uncompressed Video Sequences for Content-Based Search and Retrieval. Real Time Imaging, 2003(9): p. 73-98.
2. Gargi, U., R. Kasturi, and S.H. Strayer, Performance Characterization of Video Shot Change Detection Methods. IEEE Trans. on Circuits and Systems for Video Technology, 2000. 10(1): p. 1-13.
3. Lienhart, R. Comparison of Automatic Shot Boundary Detection Algorithms. in Image and Video Processing. 1999.
4. Zabih, R., J. Miller, and K. Mai, A Feature-Based Algorithm for Detecting and Classifying Production Effects. Multimedia Systems, 1999(7): p. 119-128.
5. Truong, B.T., C. Dorai, and S. Venkatesh. New Enhancements to Cut, Fade, and Dissolve Detection Processes in Video Segmentation. in ACM Multimedia Conference. 2000. California, USA.
6. Milanese, R., F. Deguillaume, and A. Jacot-Descombes. Video Segmentation and Camera Motion Characterization Using Compressed Data. in Multimedia Storage and Archiving Systems II. 1997. Dallas, Texas: SPIE.
7. Kobla, V., et al. Compressed Domain Video Indexing Techniques Using DCT amd Motion Vector Information in MPEG Video. in Storage and Retrieval for Image and Video Databases. 1997.
8. Dugad, R., K. Ratakonda, and N. Ahuja. Robust Video Shot Change Detection. in IEEE Workshop on Multimedia Signal Processing. 1998.
9. Yusoff, Y., W. Christmas, and J. Kittler. Video Shot Cut Detection Using Adaptive Thresholding. in British Machine Vision Conference. 2000. Bristol, U.K.
10. Yeo, B.L. and B. Liu, Rapid Scene Analysis on Compressed Video. IEEE Trans. on Circuits and Systems for Video Technology, 1995(6): p. 533-544.
11. Zhang, H.J., A. Kankanhali, and S.W. Smoliar, Automatic Partitioning of Full Motion Video. Multimedia Systems, 1993. 1(1): p. 10-28.
12. Wessel, N., et al., Nonlinear Analysis of Complex Phenomena in Cardiological Data. Herzchrittmachertherapie und Electrophysiologie, 2000. 11(3): p. 159-173.
13. Zhang, H.-J., L.C. Yong, and S.W. Smoliar, Video Partitioning and Browsing Using Compressed Data. Multimedia Tools and Applications, 1995(1): p. 91-113.