Computer
& Robotics

Compressed Domain Scene Change Detection Based on Transform Units Distribution in High Efficiency Video Coding Standard

Navid Dorfeshan [a], Mohammad Reza Ramezanpour [b,*]

[a] *Department of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran*
[b] *Department of Computer Engineering, Mobarakeh Branch, Islamic Azad University, Mobarakeh, Isfahan, Iran*

Received 25 May 2018; Revised 31 July 2018; Accepted 02 September 2018; Available online 18 September 2018

Abstract

Scene change detection plays an important role in a number of video applications, including video indexing, searching, browsing, semantic features extraction, and, in general, pre-processing and post-processing operations. Several scene change detection methods have been proposed in different coding standards. Most of them use fixed thresholds for the similarity metrics to determine if there was a change or not. These thresholds are obtained empirically or they must be calculated before the scene change detection after the whole sequence is obtained. Efficiency of scene change detectors decreases considerably for videos with high scene complexity and variation. In this paper, we propose a novel scene change detection algorithm in the HEVC compressed domain. In the proposed method, we have developed an efficient method based on the analysis of the Transform Units distribution in HEVC standard. In order to enhance the accuracy of detecting the scene changes, we have also defined an automated, dynamic threshold model which can efficiently trace scene changes. The experimental results on UHD videos indicate a higher performance with significantly improved accuracy combined with minimum complexity.

Keywords: *High Efficiency Video Coding, Scene Change Detection, Dynamic Threshold, Transform Unit.*

1. Introduction

With the rapid growth of video database, video analysis, indexing, browsing, summarization and retrieval systems become more and more important [1]. The video scene change detection is the first step of these applications. There are two basic types of scene changes. One is the abrupt change where two adjacent frames are completely uncorrelated, and the other is the gradual change where the difference between frames corresponding to two scenes is substantially reduced [2]. Current works for the scene change detection are mainly using two approaches, which are based on the spatial domain and the compressed domain respectively. In the spatial domain, the histogram comparison is usually used. The main drawback of spatial domain approaches is time consuming because of the uncompressing process and the complexity of the algorithm. In the compressed domain, a lot of features can be extracted from bit streams without fully decoding it. Many works [3-6] have been done in H.264/AVC compressed domain. They have used different features such as DC images, Discrete Cosine Transform (DCT) coefficients, macro block coding

* Corresponding author. Email: ramezanpour@mau.ac.ir

modes and motion vectors, and shown the good performance.

In the field of video processing, scene change detection can be applied either in pre-processing and post-processing operations, according to the purposes that the detection phase has to achieve, and with different features and performance. As an example, in H.264/AVC video coding applications, scene change detection can be used in pre-processing as a decisional algorithm, in order to force Intra-frame encoding (I) instead of temporal prediction (P), when a scene change occurs, and to confirm predicted or bi-predicted (B) coding for the remaining frames. As discussed in [6], a dynamic threshold model for real time scene change detection among consecutive frames may serve as a criterion for the selection of the compression method, as well as for the temporal prediction; it may also help to optimize rate control mechanisms at the encoder.

In this paper, we propose a novel scene change detection algorithm in the High Efficiency Video Coding (HEVC) compressed domain. In the proposed method, we have developed an efficient method based on the analysis of the Transform Units (TUs) distribution in HEVC standard. To increase the accuracy of detecting the scene changes, we have also defining an automated, dynamic threshold model which can efficiently trace scene changes. Experimental results indicate that the proposed algorithm achieves the good performance with a low computational complexity.

The rest of this paper is organized as follows. A review of some previous works on scene change detection is provided in Section 2. An overview on HEVC standard to the extent which is necessary for understanding the other parts in the paper is presented in Section 3. The proposed method is presented in Section 4 and its performance is evaluated in Section 5. Finally, conclusions are given in Section 6.

2. Related Works

Many researchers have proposed various methods for scene change detection such as pixel based, histogram based, edge based, statistics based, compression based, and hybrid methods. Pixel based methods used the difference of pixel values between successive frames [7]. Gray or color histogram-based schemes compared histograms for adjacent frames, respectively [8]. Edge based methods employed either object segmentation or edge detection scheme and estimated the degree of change compared with outlines of consecutive frames [9]. Statistics based methods employed many statistical inferences used in signal processing [10]. Compression based methods utilized a concept which describes more information is needed to encode a frame when there is a scene change [11]. Hybrid methods apply more than two methods to scene change detection in order to obtain better detecting performance [12].

In [7], a pixel based-algorithm for abrupt scene change detection is presented. The algorithm requires a two-stage processing of the frames, before passing them to the H.264/AVC encoder. In the first stage, subsequent frames are tested against a dissimilarity metric, and in the second stage, the set of frames not previously discarded are normalized via a histogram equalization process, through a progressive refinement. Kim et al. [10] have proposed a prediction mode based scene change detection algorithm in H.264/AVC compressed domain. It only detected the change on I frames and took no account of P and B frames. The weighted prediction tool in H.264/AVC has been used to detect fade in and out by Damghanian et al. [11]. Zeng et al. [12] used a weighted city block distance of intra mode histograms between two adjacent I frames and the ratio of intra macro blocks for P and B frames. All of the above algorithms are mainly based on the fixed threshold to detect the scene changes. However, a fixed threshold algorithm cannot have good performances for all video sequences due to the diversity of the video characteristics.

3. A Review on HEVC Standard

Joint Collaborative Team on Video Coding (JCT-VC) which consisted of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) started working on the development of HEVC standard [13] in April 2010. The main purpose of this group was to prepare a standard to be able to provide a significant improvement in coding efficiency compared with the H.264/AVC, while maintaining the same video quality mainly in high resolution videos [13], [14]. The standard approved by both ISO/IEC and ITU/T in January 2013. The approved HEVC standard could achieve 50% lower bitrate than H.264/AVC in the similar coding qualities. The high coding efficiency in HEVC is achieved at the expense of increasing the computational complexity.

In HEVC standard, each picture is divided to equal size blocks named Coding Tree Unit (CTU). CTUs are the root of quad-tree partitioning which employed in HEVC quad-tree partitioning divides each CTU to CUs. Each CU consists of one luma and two chroma blocks and its size is 2n×2n where n is an integer number ranging from three to six. Fig. 1 indicates partitioning of a 64×64 CTU into 32×32 to 8×8 CUs and the arrows in Fig. 1 indicates the coding order of CUs in a CTU. Each CU includes one or more Prediction Units (PUs). CU is the unit that is coded either by intra frame or inter frame and for each PU the same prediction mode is employed. PU sizes are in the form of 2m×2m where m is an integer number ranging from two to six. There are various DCT sizes in HEVC and TU indicates transform size. As a result, TU size correspond to DCT sizes in HEVC which is in the form of 2k×2k where k is an integer number ranging from two to five. Hence the residues for each PU can be transformed by various sizes of DCT which means each PU consists of one or more TUs. Partitioning a PU to one or more TUs is indicated by residual quad-tree structure [15].

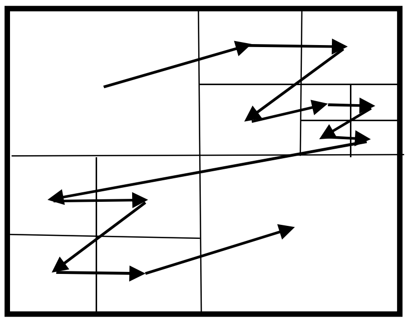


Fig. 1. An example of the partitioning of a 64×64 CTU into CUs.

4. Proposed Method

The proposed method consists of three stages. In the first stage, TU sizes are obtained. In the second stage, the number of each TU size are calculated and the corresponding histogram is plotted. In the last stage, the two related histograms are compared.

4.1. TU Size Feature Vector

TU size is selected according to the details in the coded images and it can be used as a common feature for the image texture. As a result, a TU size histogram can be used as a feature vector for the similarity between two images coded in HEVC standard. In order to compute the feature vector, the normalized TU histogram is defined as follows:

$$\mathbf{H_{TU}[i]} = \left(\sum_{\mathbf{j=1}}^{\mathbf{Wl}} \boldsymbol{\delta}(\mathbf{i} - \mathbf{TUSizeIndex(j)})\right)/\mathbf{w} \qquad (1)$$

Where $H_{TU}[i]$ is the bin I of histogram, *i* ranges from 1 to 4, *TUSizeIndex* is the TU size for the jth block, and $\delta(i - TUSizeIndex(j))$ is the unit impulse function that is 1 if i-*TUSizeIndex* (j)=0 and is zero otherwise. *W* is equal to the number of 4×4 blocks in the luma component of the picture. This histogram has 4 bins and the relationship between TU size and *TUSizeIndex* is shown in Table 1. The value of each bin specifies the number of 4×4 blocks which are according to the transform unit size. Then, after determining the type and size of the TUs, the number of each block is specified and subsequently they are used to compare the images in the database.

Table. 1. The relationship between *TUSizeIndex* and transform unit size

TU Size	**4×4**	**8×8**	**16×16**	**32×32**
TUSizeIndex	1	2	3	4

4.2. Obtaining TU Histogram

TU histogram can be used as an indicator to compare the images. The horizontal axis in the histogram shows the TU size indexes and the vertical axis shows the value of TU in each image. Fig. 2 shows the histogram of TU sizes in two sequence frames. Fig. 3 shows two images with different backgrounds. One of the images has a uniform background and the other has a cluttered background. After computing TU, their histograms are obtained. By comparing the two histograms, it can be concluded that the image with more cluttered background has the larger number of 4× 4 blocks, and if the image has a uniform background, the number of the small blocks is reduced and the image is divided into larger blocks.

4.3. The Scene Change Detection

The computational schemes describe a similarity measure between two sequence frames. When this measure shows a big enough change, a scene change defined. These schemes define a threshold, usually a fixed one. If the value of the measure exceeds the threshold, a scene change is detected, and conversely, if it is less than the threshold, the scene change does not appear. If sum of histogram differences between two sequence frames exceeds threshold, the scene change occurs. Obtaining the optimal value for the threshold is fixed. If it is settled too high, some cuts probably remain undetected. If it is too low, the detection scheme makes false detection. For the proposed thresholding function, we use local statistical properties of the sequences. Let Xi be a random variable in frame i. We use a sliding window of length N. If the actual frame number is n, then the window lies between [n-1, n-(n+1)]. Then, we calculate the empirical mean value mn and the standard deviation of Xi is defined as follows:

$$\mathbf{m_n} = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=n-N-1}}^{\mathbf{n-1}} \mathbf{X_i} \qquad (2)$$

$$\boldsymbol{\sigma}_{\mathbf{n}} = \sqrt{\frac{\mathbf{1}}{\mathbf{N-1}} \sum_{\mathbf{i=n=N-1}}^{\mathbf{n-1}}} \ (\mathbf{X_i} - \mathbf{M_n}) \qquad (3)$$

We use (2), (3) together with Xn-1 to define the threshold *T(n)* as follows:

$$\mathbf{T(n)} = \boldsymbol{w}_{\mathbf{1}}.\mathbf{X_{n-1}} + \boldsymbol{w}_{\mathbf{2}}.\mathbf{m_n} + \boldsymbol{w}_{\mathbf{3}}.\boldsymbol{\sigma}_{\mathbf{n}} \qquad (4)$$

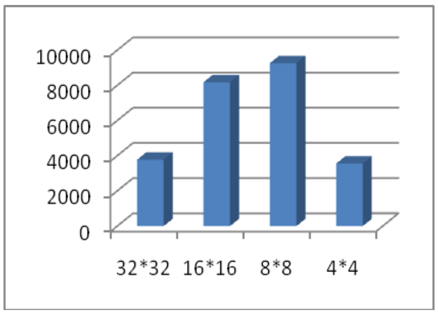
Where *n* shows the current frame and w_1, w_2, w_3 are constants. In other way, the equation (4) is as follows:

$$\mathbf{T(n)} = \boldsymbol{w}_{\mathbf{1}}.\mathbf{X_{n-1}} + \boldsymbol{w}_{\mathbf{2}}.\mathbf{m_n} + \boldsymbol{w}_{\mathbf{3}}.\boldsymbol{\sigma}_{\mathbf{n}} \qquad (5)$$

The following discussion presents a greater understanding about appropriate choice of the constants w_1, w_2, *and* w_3. The constants w_1, w_2, *and* w_3 are very important because they define the character of the thresholding function. If w_2 takes high values, the threshold will also get high values and this avoids wrong change detection like in intense motion scenes, but from the other point of view, the detector can miss some low value scene changes and this may cause scene change detection becomes difficult. If w_2 takes low values, it allows us to identify the regions with high activity, which can provide valuable information for other applications including error concealment, bit rate control and etc. Like the standard deviation, high values of w_3 prevent detecting intense motion in a scene change. On the contrary, w_1 must have a small value. The whole procedures of selecting w_1, w_2, *and* w_3 is a trade-off and should be done according to the intended application. Furthermore, the dynamic threshold model explained above, a function based lowering the threshold is defined to avoid false detection instantly after a scene change. When a scene change is detected in frame *p*, the TU value of this frame is allocated to the threshold. In this case, for the next *k* frames, we use the $T_e(n)$ threshold which is an exponentially decaying function, in other words, to avoid false scene change detection in relation to the previous change, we use the equation (6) in which *s* controls the negative speed:

$$\mathbf{T_e(n)} = \mathbf{X_{n-1}}\,.\,\mathbf{exp(-s(n-p))} \qquad (6)$$

TU Size	32×32	16×16	8×8	4×4	TU Size	32×32	16×16	8×8	4×4
N	3776	8176	9260	3556	N	4544	7824	9348	3193



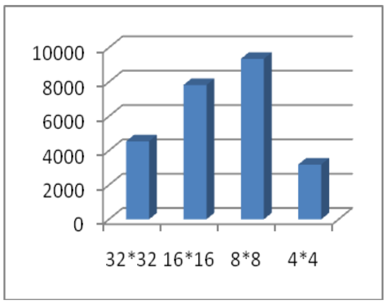
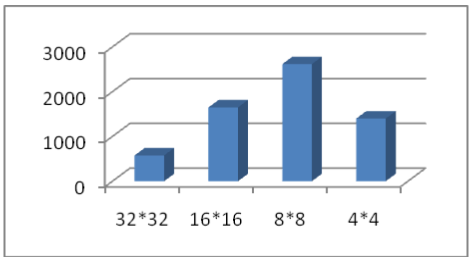
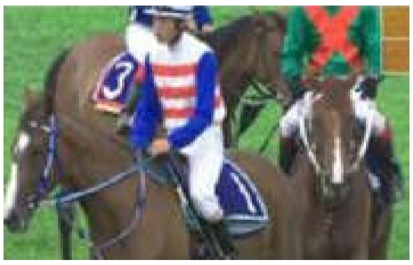
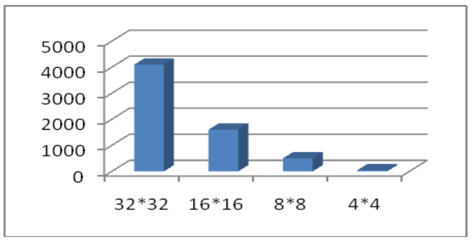


Fig. 2. The sum of histogram differences between two frames





TU Size	*32×32*	*16×16*	*8×8*	*4×4*
N	*576*	*1648*	*2616*	*1400*



TU Size	32×32	16×16	8×8	4×4
N	4096	1600	500	44

Fig. 3. Comparison of two TU size histogram

5. Simulation Results

All the simulations are defined under the common test conditions (CTC) [16] defined by JCT-VC. We have tested the proposed algorithms on six sequences defined in the CTC with different resolutions. The sequences are encoded with HEVC reference software HM-16.09. The encoder configuration is as follows:

All frames are coded as intra. Quantization parameters (QPs) are set to 22, 27, 32, and 37. The number of frames is considered 100. The minimum size of the prediction blocks is 4×4 and their maximum size is 64×64. We used video sequences of JCT-VC. They were chosen because they have scenes with intense motion, change of light conditions, high complexity and different types of scene changes. There are changes between the crowd and the field, which are easy to detect but there are also scene changes with the same background which are more complicated.

In Basketball Drill sequence, many effects like zooming in/out and transition effects like dissolving, fading in/out can be found. All of these features make Basketball Drill sequence challenging for a scene change detector. The Basketball Drill sequence has the resolution of 832×480 and its length is 150 frames and the videos are stored in YUV format, then the video as an input file is given to the high efficiency video encoder. Two conventional performance measures are used for scene change detection algorithm; recall and precision which are defined as follows:

Recall: it is defined as the number of correctly detected scenes to the number of actual scenes:

$$\boldsymbol{recall} = \frac{\boldsymbol{N_c}}{\boldsymbol{N_c + N_m}} \tag{7}$$

Where N_c is the number of correct detections, and N_m is the number of miss detections.

Precision: it is defined as the correctly detected scenes to the total number of detected scenes:

$$\boldsymbol{Precision} = \frac{\boldsymbol{N_c}}{\boldsymbol{N_c + N_f}} \tag{8}$$

Where N_f is the number of false detections. Table 2 shows the results of the various scene change detection with the fixed and dynamic threshold. The threshold is used for comparing the best case after obtaining TU for the whole sequences. We take the value that minimizes the number of missed and false detections.

The Dynamic threshold parameters are w_1, w_2, *and* w_3 that we set to the following values w_1=-2, w_2=4, w_3=-3. These values are selected for all of sequences however they have different resolution to test their sensitivity.

We use a window size as 100 frames and speed of exponential decaying s=0.06. Consider that scene change detection is more difficult in small resolutions like class A and therefore the results are better for sequence in class B. Optimal setting of the dynamic threshold model parameters may also be slightly different for sequences with another character. The experiments show that exponentially decaying improves the results slightly, but the added value is rather low in comparison to the contribution of the dynamic threshold model itself against the fixed threshold.

Table. 2. The results of the scene change detection

Class A (832×480)	N_C	N_F	N_M	RECALL	PRECISION
fixed threshold	24	7	4	0.85	0.77
dynamic threshold without an exponential decaying	30	2	1	0.96	0.94
dynamic threshold with and exponential decaying	30	1	1	0.96	0.96
Class B (1920×1080)	NC	NF	NM	RECALL	PRECISION
fixed threshold	26	9	14	0.65	0.78
dynamic threshold without an exponential decaying	41	3	2	0.93	0.93
dynamic threshold with an exponential decaying	41	3	2	0.93	0.93

6. Conclusion

In this paper, a method for scene change detection on the coded images in HEVC standard is presented. In this method, the internal frame feature vector is extracted and can be applied to the high resolution images. In the proposed method, TU feature vector is computed and with respect to the result, the histogram is plotted. Then according to the local statistical properties of the video sequences, an automated dynamic threshold model is presented for scene change detection. This method is designed and implemented in HEVC standard but it can also be used for any other codec or even for raw video sequences. The benefit of the proposed method is lower complexity. Moreover, it is suitable for real time applications. This method is implemented considerably better than an optimum fixed threshold setting and presents the good results. The results of experiment have shown that the proposed algorithm is robust for scene change detections in compressed video by HEVC standard. Future work will focus on utilizing a robust algorithm for the scene change detection in HEVC bit streams with P and B frames.

References

[1] Nakamura, J., "Image sensors and signal processing for digital still cameras". CRC Press; 1 edition (April 19, 2016).

[2] Wu, C.; Zhang, L.; Du, B., "Kernel slow feature analysis for scene change detection". IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 4, pp. 2367-2384 (2017).

[3] Rascioni, G.; Spinsante, S.; Gambi, E., "An optimized dynamic scene change detection algorithm for H. 264/AVC encoded video sequences". International Journal of Digital Multimedia Broadcasting. (2010).

[4] Laumer, M.; Amon, P.; Hutter, A.; Kaup, A., "Moving object detection in the H. 264/AVC compressed domain". APSIPA Transactions on Signal and Information Processing (2016).

[5] Hong, B.; Eom, M.; Choe, Y., "Scene change detection using edge direction based on intra prediction mode in H. 264/AVC compression domain". TENCON 2006 - 2006 IEEE Region 10 Conference (2006).

[6] Bulut, F.; Osmani, S., "Scene Change Detection using Different Color Pallets and Performance Comparison". Balkan Journal of Electrical and Computer Engineering, vol. 5, no.2, pp. 66-72 (2017).

[7] Yoon, Y.S.; Yoo, W.Y.; Suh. Y.H., "A Robust Scene Change Detection Using Mode Distribution in H. 264/AVC". SEMAPRO 2012: The Sixth International Conference on Advances in Semantic Processing, (2012).

[8] Eom, Y.; Park, S.; Yoo, S.; Choi, J.S.; Cho, S., "An analysis of scene change detection in HEVC bitstream". Semantic Computing (ICSC), 2015 IEEE International Conference, pp. 470-474 (2015).

[9] Xiaona, Z.; Guoqing, Q.; Qiang, W.; Tao, Z., "An improved approach of scene change detection in archived films". Signal Processing (ICSP), 2010 IEEE 10th International Conference, pp. 825-828 (2010).

[10] Kim, S.M.; Byun JW, Won CS. A scene change detection in H. 264/AVC compression domain. Pacific-Rim Conference on Multimedia 2005 Nov 13 (pp. 1072-1082). Springer, Berlin, Heidelberg.

[11] Damghanian, B.; Hashemi, M.R.; Akbari, M.K., "A novel fade detection algorithm on H. 264/AVC compressed domain". Pacific-Rim Symposium on Image and Video Technology 2006 Dec 10 (pp. 1159-1167). Springer, Berlin, Heidelberg.

[12] Zeng, W.; Gao, W., "Shot change detection on H. 264/AVC compressed video". Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium, pp. 3459-3462 (2005).

[13] Heidari, B.; Ramezanpour, M., "Reduction of intra-coding time for HEVC based on temporary direction map". Journal of Real-Time Image Processing, pp. 1-13 (2018).

[14] Zhu, W.; Yi, Y.; Zhang, H.; Chen, P.; Zhang, H., "Fast mode decision algorithm for HEVC intra coding based on texture partition and direction". Journal of Real-Time Image Processing, pp. 1-8 (2018).

[15] Ramezanpour, M.; Zargari, F., "Fast CU size and prediction mode decision method for HEVC encoder based on spatial features". Signal, image and video processing. Vol. 10, no. 7, pp. 1233-1240 (2016).

[16] Bossen, F., "Common test conditions and software reference configurations". 9th Meeting of the JCT-VC in Geneva (2012).