



Shot boundary detection based on orthogonal polynomial

Sadiq H. Abdulhussain¹  · Abd Rahman Ramli² · Basheera M. Mahmmod¹ · M. Iqbal Saripan² · S. A. R. Al-Haddad² · Wissam A. Jassim³

Received: 6 July 2018 / Revised: 6 February 2019 / Accepted: 11 February 2019 /
Published online: 28 February 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Shot boundary detection (SBD) is a substantial step in video content analysis, indexing, retrieval, and summarization. SBD is the process of automatically partitioning video into its basic units, known as shots, through detecting transitions between shots. The design of SBD algorithms developed from simple feature comparison to rigorous probabilistic and using of complex models. Nevertheless, accelerate the detection of transitions with higher accuracy need to be improved. Extensive research has employed orthogonal polynomial (OP) and their moments in computer vision and signal processing owing to their powerful performance in analyzing signals. A new SBD algorithm based on OP has been proposed in this paper. The Features are derived from orthogonal transform domain (moments) to detect the hard transitions in video sequences. Moments are used because of their ability to represent signal (video frame) without information redundancy. These features are the moments of smoothed and gradients of video frames. The moments are computed using a developed OP which is squared Krawtchouk-Tchebichef polynomial. These moments (smoothed and gradients) are fused to form a feature vector. Finally, the support vector machine is utilized to detect hard transitions. In addition, a comparison between the proposed algorithm and other state-of-the-art algorithms is performed to reinforce the capability of the proposed work. The proposed algorithm is examined using three well-known datasets which are TRECVID2005, TRECVID2006, and TRECVID2007. The outcomes of the comparative analysis show the superior performance of the proposed algorithm against other existing algorithms.

Keywords Shot boundary detection · Temporal video segmentation · Hard transition · Abrupt transition · Orthogonal polynomial · Orthogonal moments

1 Introduction

Escalation of multimedia information in the cyberspace through the past decades has led to swift increase in data transmission volume and repository size. This increase has motivated the exploration for more efficient techniques to process and store data content [4].

✉ Sadiq H. Abdulhussain
sadiqh76@yahoo.com

There are several multimedia data types available in the cyberspace such as audio, image, and video; however, video is considered the most attracting one. Furthermore, video amongst other multimedia types is the most storage consuming in the cyberspace and contains more valuable information than other types of multimedia data. The dominance of videos in the cyberspace is due to the tremendous development in computer performance, spread of recording devices, and affordable storage media. In addition, video sharing websites (VSW), such as YouTube which has been used by individuals and companies to extend their audience. Statistics show that videos are uploaded and viewed at VSW in an inconceivable rate [5].

The enormous increase of video data has activated a substantial need for effective tools that can manage, manipulate and store that sheer volume of data [33]. This can be achieved by attaching video content with their storage (indexing) and then analyzing them to their basic units by video structure analysis. Video structure analysis is fairly considered a challenging task because of video attributes which are: vast information compared to images, huge size of raw data, and no prior definition of video structure [13].

The aim of video structure analysis is to partition the video into its basic elements. Video structure levels are: 1) frames, 2) shots, 3) scenes, and 4) stories. Shot is considered the basic element of video and it is considered the most suitable level for content based video indexing and retrieval (CBVIR) [4].

Shot is defined as an incessant sequence of frames that are captured by single and non-stop operation of video capturing device. Shots are accumulated together to form a scene, scenes are linked to compose a story and so forth to produce a video. When attaching two shots directly together, a hard transition (HT) is produced. On the other hand, soft transition is generated using editing effect (indirect concatenation). HTs are the most predominant in videos than ST [19]. Shot boundary detection (SBD), also known as temporal video segmentation, is the process of analyzing the video to its basic elements, namely shots. SBD is the initial and substantial step of CBVIR and its accuracy largely affects the entire CBVIR efficiency [4].

SBD algorithms include the following three main steps: 1) representation of visual information (feature extraction), 2) similarity/dissimilarity measure (continuity signal), and 3) transition identification. Feature extraction is a significant module in SBD algorithms. Several types of features are utilized in SBD algorithms such as pixel-based, histogram-based, edge-based and others [4]. Similarity measure is performed by finding the distance between consecutive frames features. City-block, Euclidean, and correlation are examples of similarity measures. To find transition between shots, transition identification (detection) is performed using statistical machine learning (supervised and unsupervised) and threshold (fixed and adaptive).

SBD has attracted many researchers' attention in the past two decades. It is mainly divided into two categories: compressed and uncompressed domains. The latter has received more attention compared to compressed domain because of the tremendous and valuable visual information. However, uncompressed domain-based algorithms require additional processing time due to the decoding process for video frames [4].

Pixel-based, histogram-based, edge-based, and motion-based are examples of techniques that are used for feature extraction. Extensive survey can be found in [4]. Recently, the analysis of SBD is performed in transform domain rather than in the time domain. The reason is that transform domain allows to view signals in different domain and gives a massive shift in terms of its powerful capability for analyzing the signals' components [25, 26]. To transform a signal/image from time/spatial domain into transform (moment) domain,

the orthogonal polynomials are employed. Orthogonal moments, transform coefficients, are scalar quantities used to represent the visual information. These moments represent the projection of signal on orthogonal polynomials. The ability of the orthogonal polynomials (OPs) is characterized by their energy compaction and localization properties [26].

Different OPs occupy a significant position in signal processing and computer vision fields [2] such as Discrete Krawtchouk-Tchebichef transform (DKTT) [26]. DKTT shows a high energy compaction and good localization properties compared to Tchebichef polynomial (TP) and Krawtchouk polynomial (KP).

Energy compaction is an important property where the tendency of DKTT to return back a large fraction of signal energy into relatively few coefficients of the transform components. In addition, the localization property improves the overall quality of the DKTT especially when the ROI is located priori in the image and adds value to feature extraction by reducing the computation time.

Hence, this paper introduces the use of DKTT as a feature extraction tool for representing video frames. In addition, new features, namely moments of smoothed and gradients frames, are extracted from video frames using OPs (DKTT). These features are then fused to form a single feature vector. Support vector machine (SVM) is used to detect transition and non-transition frames based on the fused features.

This paper is organized as follows: Section 2 describes a brief survey on the related work. Section 3 introduces OPs and the mathematical model of the implemented OP. Section 4 provides the proposed features extraction method based on OP to detect HT. Section 5 presents the experimental results to highlight the effectiveness of the extracted features. Finally, the Section 6 concludes the paper.

2 Related work

In the literature, various methods based on different schemes for SBD have been introduced. Generally, SBD workflow is divided into three stages: 1) representing the visual information for video frames (feature extraction), 2) constructing the continuity signal from the extracted features, and 3) detecting transitions. In this section, related works for each of the aforementioned stages are discussed.

2.1 Feature extraction

The extracted features from video frames are used to concisely represent the visual information of these frames. Different types of visual information representation are introduced such as: pixel-based histogram-based, edge-based, and transform-based. The simple and fast approach is the pixel-based technique (PBT) [14, 43, 53], where the pixel intensities are directly employed to represent the visual information. However, the PBTs are highly sensitive to object/camera motion and various types of camera operations as well as global motion. Thus, this sensitivity highly affects the accuracy of the SBD algorithm which in fact reduces the precision rate due to the high false alarm rate [4]. In addition to their sensitivity, missed detection results [18].

To alleviate the sensitivity of the PBT, histogram-based techniques (HBTs) were introduced [15, 16, 42]. The HBTs replace the dependency on the spatial information by considering the color distribution for each frame. This dependency makes the HBTs to be less sensitive to small global and local motions. Different types of color spaces are employed

to extract histogram from frame such as gray [41], RGB [40], HSV [11], and $L^*a^*b^*$ [20]. However, HBTs are less sensitive to object/camera motion compared to PBTs, false positives are reported due to large object/camera motions and flash lights. In addition, miss detection highly occurs between two shots that fit in the same scene because of the similarity in the color distribution between the neighboring shots' frames.

Algorithms based on edge features were also presented to reduce the influence of object/camera motion and flash light. In [49], Yoo et al. presented an edge object tracking-based algorithm to detect shots. On the other hand, the ratio of exiting and entering edge between consecutive frames is utilized to detect shot transitions [51, 52]. In spite of the fact that edge based techniques (EBTs) get the benefit of edges as a frame feature, the detection accuracy is still unsatisfactory due to: 1) EBTs are expensive due to the number of processes employed (edge detection, edge change ratio, motion compensation or edge tracking), 2) their performance is below the performance of HBT [4]. However, EBTs are able to remove the flash light because they are invariant to illumination change.

Recently, transform based techniques (TBTs) have been utilized for transition detection. The TBT uses linear transform to compute the transform coefficients. These coefficients are considered as features in the SBD algorithm, for example Fourier transform coefficients [34, 45]. Porter et al. [34] used the correlation between transform coefficients of video frame blocks as features to detect transitions. Vlachos [45] utilized the phase correlation between transform coefficients as a feature. Priya and Domnic [35, 36] proposed an SBD using Walsh-Hadamard transform (WHT). The algorithm extracted features for small block size (4×4) using different OP basis functions. The feature extraction is performed after resizing video frame to 256×256 . Non-Subsampled Contourlet Transform (NSCT) is utilized for SBD in [27]. This algorithm extracted features from the low-frequency sub-band and seven high-frequency subbands for each CIE $L^*a^*b^*$ color space. The reported accuracy of this method is acceptable, and it is comparable to the SBD algorithm presented in [36]. Although the algorithms proposed in [27, 36] reported high accuracy, their computational cost was high. The computational cost in [36] stems from extracting features for small block size and different OP basis functions. On the other hand, the computational cost of the algorithm in [27] was due to employing multi-level of decomposition, block processing, and considering all CIE $L^*a^*b^*$ channels in the algorithm. Zernike and Fourier-Mellin transforms were used to extract features in [8] to detect shot transitions. These features were considered as shape descriptors in addition to other features, namely the color histogram, and phase correlation. These three features were combined to form a features vector. Then, the feature vector was used to detect HT. This algorithm provided an acceptable HT detection accuracy; however, the transforms, Zernike and Fourier-Mellin, require coordinate transformation and suitable approximation [28] which increase the computational cost.

2.2 Continuity signal

The continuity signal is constructed to find the temporal variation between features. Basically, the continuity signal is constructed by finding the difference (dissimilarity) between frames features or by computing the correlation coefficients between features. It should be noted that, the continuity signal is found either between consecutive frames or two frames within a temporal distance. The similarity signal between frames features are measured using cosine similarity [10], normalized correlation [34], and correlation [44]. On the other hand, frames features dissimilarity is computed using city-block distance [36], edge change ratio [52], histogram intersection [15], and accumulating histogram difference [16].

2.3 Classification of continuity signal

The constructed continuity signal (similarity or dissimilarity measure) is utilized to detect the shot transitions, i.e. to differentiate between the transition and non-transition frames. The threshold-based techniques are considered the simplest methods for classification [4]. In threshold-based technique, the continuity signal is compared with a threshold to declare the state of transition or non-transition. Algorithms based on threshold to detect transitions are considered unreliable since the threshold can not be generalized to different types of video genre [6]. In addition, algorithms based on single threshold show inability to distinguish between transition, non-transition, object/camera motion, and flash light. Machine learning, which is based on categorization problem, is used to overcome the limitation of utilizing threshold and employing multiple features. Neural network and SVM are examples of machine learning algorithms. In addition, machine learning algorithms are very popular in image processing [9, 21] and can be classified into supervised and unsupervised methods. The difficulty of employing machine learning algorithms is resulted from the selection of suitable feature combination [32] because good features significantly increase the classifier performance [22].

It is obvious from the literature that SBD algorithms have good performance in transition detection. Moreover, there is an evident progress in SBD algorithms from simple feature comparison to rigorous probabilistic and using of complex models of the SBD. Nevertheless, accelerating the process of transitions detection with higher accuracy needs to be improved. Essentially, the accuracy of transition detection depends on the ability of the algorithm to distinguish between transitions and objects/camera motion. This is because features cannot model the difference between images clearly. Another important drawback that is obvious in most existing SBD methods is the high computational cost which becomes a bottleneck for real-time applications. Among the various techniques presented, TBTs, which are based on OP, show an improvement in the detection accuracy for their ability in reducing the impact of disturbance factors on transition detection. However, the existing TBT suffers from different drawbacks. First, employing single similarity signal leads to inadequate performance. Besides, each transform has its own limitation such as: Zernike and Fourier-Mellin require coordinate transformation and approximation, Walsh-Hadamard requires frame resizing, and discrete wavelet transform requires predefined multiple decomposition levels. It is clear that there is a growing need to design an algorithm that has a joint evaluation of the following aspects: minimizing computational cost, tackling the problem of detection accuracy by increasing recall and precision for various video genres. Moreover, the algorithm must be based on a robust backbone transform that provides: good representation of the visual video content, good localization property, high energy compaction capability, and effective extraction of the appropriate features. All points above are critical in SBD algorithm and need to be considered.

3 Discrete Krawtchouk-Tchebichef polynomials and moments

The objective of this section is to introduce the mathematical model of the utilized OPs, namely Krawtchouk-Tchebichef polynomial (KTP) [26], and their moments. The KTP of the n th order, $P_n(x; p)$, can be defined as follows:

$$P_n(x; p, N) = \sum_{j=0}^{N-1} k_j(n; p) t_j(x) \quad (1)$$

where $k_j(n)$ is the KP of the j th order which is given as follows [3]:

$$k_j(n; p) = \sqrt{\frac{A}{B}} {}_2F_1\left(-j, -n; 1 + N; \frac{1}{p}\right) \tag{2}$$

$$j, n = 0, 1, \dots, N - 1; N > 0, p \in (0, 1)$$

where A and B are defined as follows:

$$A = \binom{N-1}{n} p^n (1-p)^{N-1-n} \tag{3}$$

$$B = (-1)^j \left(\frac{1-p}{p}\right)^j \frac{j!}{(1-N)_j} \tag{4}$$

where $\binom{a}{b}$ represents the binomial coefficients, and $(a)_j$ stands for Pochhammer symbol [2]. $t_j(x)$ symbolizes TP [1] of the j th order which is given as follows:

$$t_j(x) = \frac{(-N+1)_j {}_3F_2(-j, -x, j+1; 1, 1-N; 1)}{\sqrt{(2j)! \binom{N+j}{2j+1}}} \tag{5}$$

$$j, x = 0, 1, \dots, N - 1; N > 0$$

Lastly, ${}_2F_1$ and ${}_3F_2$ in (2) and (5) demonstrate the hypergeometric functions [2].

Signal functions projection on OPs basis functions are orthogonal moments. Orthogonal moments of a 2D signal $I(x, y)$ (image) with a size of $N_1 \times N_2$ can be computed as follows:

$$\phi_{nm} = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} P_n(x) P_m(y) I(x, y) \tag{6}$$

$$n = \frac{N_1 - 2}{2}, \frac{N_1}{2}, \dots, \frac{N_1 - MO_n}{2}, \frac{N_1 + MO_n - 2}{2}$$

$$m = \frac{N_2 - 2}{2}, \frac{N_2}{2}, \dots, \frac{N_2 - MO_m}{2}, \frac{N_2 + MO_m - 2}{2}$$

where MO_n and MO_m are the maximum moments order used for signal representation. The computation of moments can be performed in matrix form as follows:

$$\Phi = P_1 I P_2^T \tag{7}$$

where P_1 and P_2 are the matrix form for $P_m(y)$ and $P_n(x)$, respectively. Φ is the matrix form of the moments ϕ_{nm} . For moment order selection, Fig. 1 shows the construction of matrix P with the localization parameter p , and moment order MO .

To reconstruct I from the moment domain, the inverse moment transformation is applied as follows:

$$\hat{I} = P_1^T \Phi P_2 \tag{8}$$

where \hat{I} is the reconstructed 2D signal from the moment domain.

4 SBD algorithm based on orthogonal moments

This section introduces the proposed SBD algorithm based on orthogonal moments. The proposed SBD algorithm includes three steps: A) OP-based feature extraction, B) dissimilarity measure, and C) Detection of transitions (shot boundaries). The proposed methodology is shown in Fig. 2, and the detailed description of each step for the proposed algorithm is explained in the following subsections.

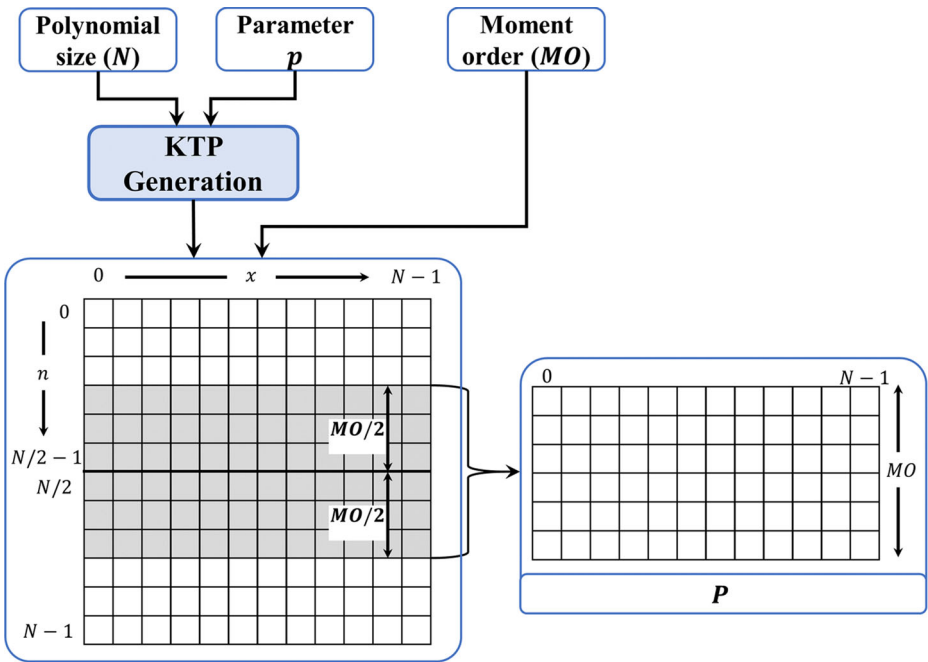


Fig. 1 KTP generation with moment order (MO)

4.1 Feature extraction

The proposed SBD algorithm design is based on moments computed (features) from DKTT domain. The considered features are: the smoothed moment, moments of image gradients (MOGs) in x -direction, and MOGs in y -direction, i.e. three groups of moments are used. The averaging (smoothing) is applied to video frames to decrease the effect of noise and camera/object motion [4]. To extract moments of a smoothed video frame, the smoothed image kernel is applied to the video frame prior to feature extraction process. Assume f_k is the k th video frame with a size of $N_1 \times N_2$, S_x and S_y are the smoothing image kernels in x - and y -directions, respectively. Then, the smoothed image in the x -direction can be obtained by convolving the video frame with S_x as follows:

$$I_{SX} = S_x * f_k \tag{9}$$

When the obtained smoothed image (I_{SX}) is convolved with the image kernel in the y -direction (S_y), the resulted image becomes the smoothed image in x and y -directions. This operation is performed as follows:

$$I_{SXY} = S_y * I_{SX} \tag{10}$$

where I_{SXY} is the smoothed image in both directions. To reduce the number of convolution operations, the associative convolution property is applied so that both kernels are combined to obtain a single kernel. From this point of view, we can achieve a smoothed video frame in both directions (I_{SXY}) as follows:

$$I_{SXY}(k) = S_{xy} * f_k \tag{11}$$

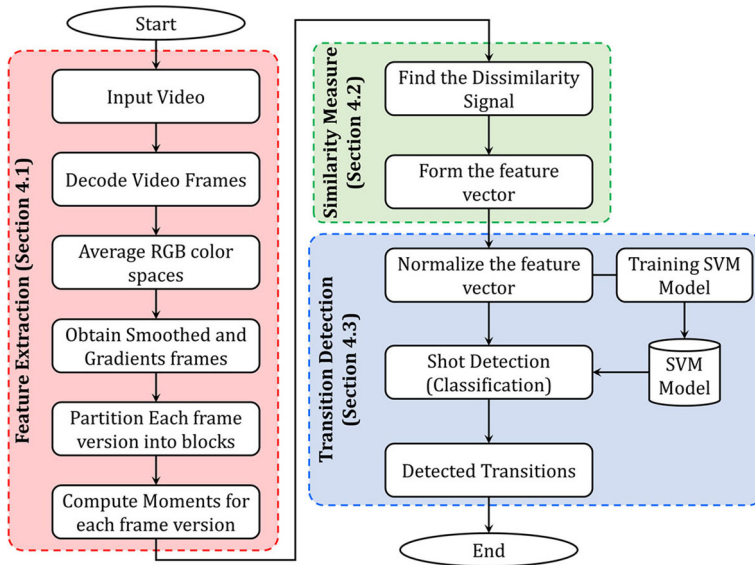


Fig. 2 The proposed methodology

where * is the convolution operation, $S_x = \frac{1}{2\pi\sigma_x^2} e^{-\frac{(x-\mu_x^2)}{2\sigma_x^2}}$, and $S_y = \frac{1}{2\pi\sigma_y^2} e^{-\frac{(y-\mu_y^2)}{2\sigma_y^2}}$. To compute the moments of the smoothed video frame, (7) is applied as follows:

$$\Phi_S(k) = P_1 I_{SXY}(k) P_2^T \tag{12}$$

where P_1, P_2 are the KTP matrices. To enhance the detection accuracy, gradient image kernels such as difference, Sobel, and Prewitt can be utilized to decompose an image into its gradients in x - and y -directions. The image gradient is used to find the intensity change in the horizontal and vertical directions. Image gradients have been proved to be useful and reasonable tools for image representation [4] and there were good attempts to employ image gradient in computer vision applications [24]. Accordingly, MOGs are considered as features in the design of the proposed SBD algorithm because of their ability to reduce the flash light effect [17].

To compute MOGs (Φ_{GX} and Φ_{GY}) in the x - and y - directions, the gradient image kernels operators G_X and G_Y are utilized to find the gradient of video frames as follows:

$$\Phi_{GX}(k) = P_1 I_{GX}(k) P_2^T \tag{13}$$

$$\Phi_{GY}(k) = P_1 I_{GY}(k) P_2^T \tag{14}$$

where I_{GX} and I_{GY} are the image gradients of the frame f_k in the x - and y -directions, respectively, which are computed as follows:

$$I_{GX}(k) = G_X * f_k \tag{15}$$

$$I_{GY}(k) = G_Y * f_k \tag{16}$$

where $G_x = [-1 \ 1]$, and $G_y = [-1 \ 1]^T$.

The efficiency of SBD algorithm can be increased by employing local features which are more robust than global features. Generally, local features are utilized to minimize the disturbance effect of object/camera motion and to draw more consistent temporal variation within shots [23]. From this point, video frames are divided into blocks to extract local features to reduce the effect of flash light, object/camera motion, and camera operation. To obtain local moments (features), the moments are computed for each block in the smoothed frame (I_{SXY}) and gradient frames (I_{GX} and I_{GY}).

To summarize the feature extraction process of the proposed SBD algorithm, the procedure of local moments (features) computation is as follows:

1. The video frames are decoded.
2. For each video frame, average all RGB color planes as follow:

$$f_k = \sum_{c \in R, G, B} f_k(x, y, c) \tag{17}$$

3. (a) For each video frame, compute smoothed and gradient frames using (11), (15), and (16).
 - (b) Acquire the video frame information (N_1 and N_2).
4. (a) Set the number of blocks for local features (v_1 and v_2) and moment order (MO_n and MO_m). Then compute B_1 and B_2 as follows:

$$B_1 = \frac{N_1}{v_1} \tag{18}$$

$$B_2 = \frac{N_2}{v_2} \tag{19}$$

- (b) Partition smoothed and gradient images into non-overlapped blocks of size B_1 and B_2 .
5. (a) Generate KTPs (P_1 and P_2) using (1).
 - (b) Compute Φ_S for each video frame block using (12).
 - (c) Compute Φ_{GX} and Φ_{GY} for each video frame block using (13) and (14).

The aforementioned steps are shown in Fig. 3 for more elucidation.

4.2 Construction of dissimilarity signal

In this section, the construction of the dissimilarity signal is performed using the extracted smoothed moments (Φ_S) and MOGs (Φ_{GX} and Φ_{GY}). The accumulation of the dissimilarity signal between the blocks of k and $k + 1$ frames is considered. The city-block distance is used to find the dissimilarity between two consecutive frames moments as follows:

$$D_S(k) = \sum_{v=1}^{v_1 \times v_2} \sum_{n, m \in \Phi_S} |\Phi_S(k) - \Phi_S(k + 1)| \tag{20}$$

$$D_X(k) = \sum_{v=1}^{v_1 \times v_2} \sum_{n, m \in \Phi_{GX}} |\Phi_{GX}(k) - \Phi_{GX}(k + 1)| \tag{21}$$

$$D_Y(k) = \sum_{v=1}^{v_1 \times v_2} \sum_{n, m \in \Phi_{GY}} |\Phi_{GY}(k) - \Phi_{GY}(k + 1)| \tag{22}$$

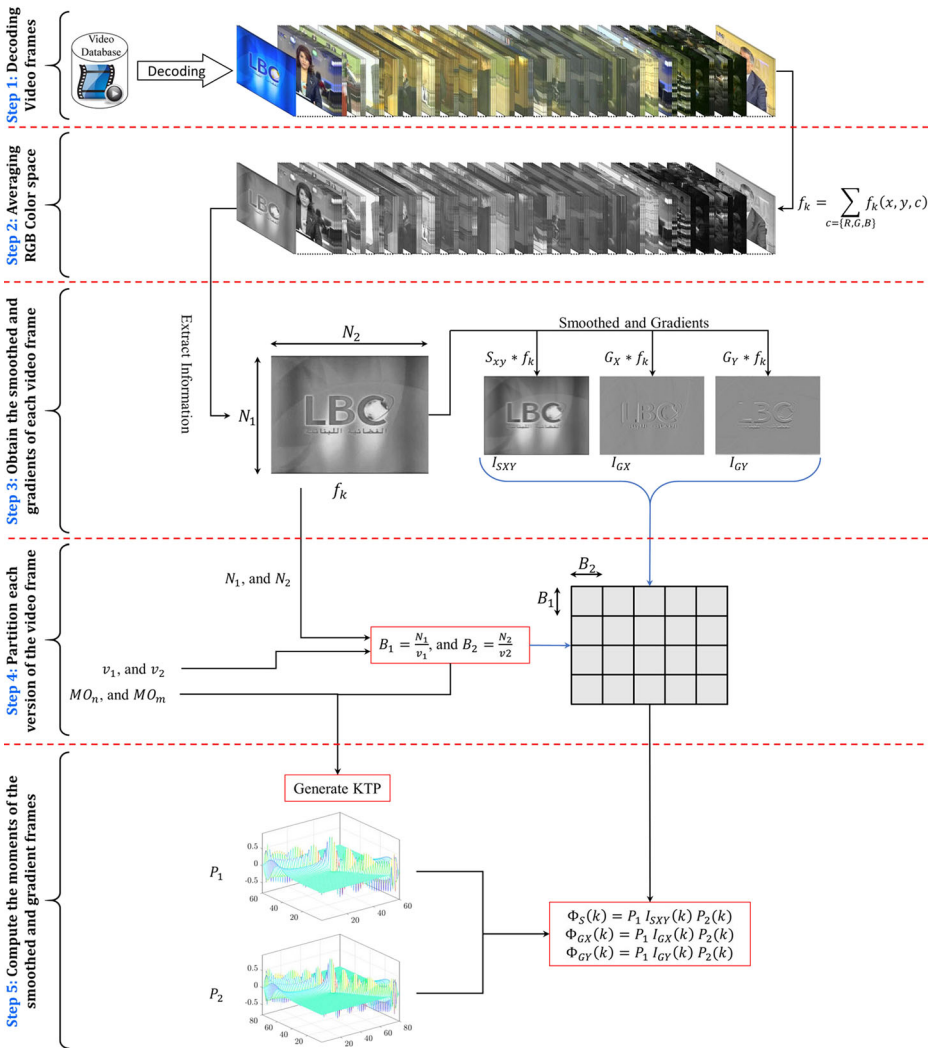


Fig. 3 Flow diagram of the moment extraction stage

The three dissimilarity signals (D_S , D_X , and D_Y) are concatenated to form the similarity signal, feature vector (FV), as follows:

$$FV(k) = \begin{bmatrix} D_S(k) \\ D_X(k) \\ D_Y(k) \end{bmatrix} \implies FV = \begin{bmatrix} D_S(1) & D_S(2) & \dots & D_S(N_f - 1) \\ D_X(1) & D_X(2) & \dots & D_X(N_f - 1) \\ D_Y(1) & D_Y(2) & \dots & D_Y(N_f - 1) \end{bmatrix} \quad (23)$$

Obviously the size of feature vector (FV) is $3 \times (N_f - 1)$, where N_f is the number of frames in video.

Temporal (contextual) information is an important factor for detecting transitions in which previous and next frames features should be considered to improve the detection accuracy [23]. Accordingly, the temporal information of previous two frames, next two

frames, and the current frame features are combined to form the following feature vector (FV_Z) as follows:

$$FV_Z(k) = \begin{bmatrix} FV(k-2) \\ FV(k-1) \\ FV(k) \\ FV(k+1) \\ FV(k+2) \end{bmatrix} \quad (24)$$

It is worth mentioning that the feature vectors (FV_Z) size is $15 \times (N_f - 1)$, which results from 3 dissimilarity signals ($D_S(k)$, $D_X(k)$, and $D_Y(K)$) by 5 temporal information ($k-2$, $k-1$, k , $k+1$, $k+2$). The obtained feature vector FV_Z is considered as input to the SVM classifier for training and testing.

4.3 Transition detection

In this stage, SVM [7] is used to identify transition and non-transition frames. SVM is utilized in this work to detect HTs, owing to its powerful ability in classification [29, 46]. In this regard, a training phase is required to obtain SVM model. Then the trained model is used to classify the input feature vector. Temporal information has been considered in this work such that the previous and next frames features are taken into consideration to improve the detection accuracy.

Features normalization is considered as an important process and it should be applied for both training and testing feature vectors [12]. In practice, features lie within diverse dynamic ranges, hence the impact of large features values is greater than small features values in the cost function [38]. The normalization process is performed to tackle the problem of great numeric ranges dominating this in smaller ranges, i.e. to ensure similar dynamic range [3]. In other words, the features normalization process is performed such that the values of features remain within the similar ranges. This can be achieved by transforming the k th feature FV_Z of mean μ_{FV_Z} and standard deviation σ_{FV_Z} into the desired mean μ_{des} and standard deviation σ_{des} as follows:

$$FV_{ZN} = (FV_Z(k) - \mu_{FV_Z}(k)) \left(\frac{\sigma_{des}}{\sigma_{FV_Z}} \right) + \mu_{des} \quad (25)$$

where FV_{ZN} is the normalized feature vector. In this work, the feature vector FV_Z is normalized with $\mu_{des} = 0$ and $\sigma_{des} = 1$ to achieve the final feature vector (FV_{ZN}) which is used in the training and detection phases.

The cross-validation and grid-search methods are used to tune the SVM parameters, optimal kernel parameter (γ) and penalty parameter (C). Where, cross-validation procedure can prevent the overfitting problem. Several different pairs of (C , γ) values were tested in the SVM model and the one with the highest cross-validation accuracy was adopted. Based on 5-fold cross validation results, the grid-search successfully finds the optimal pair of both parameters.

5 Experimental results

In this section, the performance of the proposed SBD algorithm is evaluated in terms of computational cost and ability to detect HTs. The proposed algorithm is tested on different datasets provided by TRECVID which is co-sponsored by the National Institute of Standards and Technology (NIST) [39]. The datasets used in this work are TRECVID2005,

TRECVID2006, and TRECVID2007. These datasets include 42 videos which contain more than 19 video hours.

To train the SVM model, 7 videos are utilized from the datasets. The remaining 35 videos are used for testing. The evaluation criteria is based on three measures which are: Precision (\mathcal{P}), Recall (\mathcal{R}), and harmonic mean ($F1$). These measures are defined as follows [4]:

$$\mathcal{P}\% = \frac{N_C}{N_C + N_F} \times 100 \quad (26)$$

$$\mathcal{R}\% = \frac{N_C}{N_C + N_M} \times 100 \quad (27)$$

$$F1 = \frac{2 P R}{P + R} \quad (28)$$

where N_C represents the correctly detected transitions, N_F is the falsely detected transitions, and N_M symbolize the missed transitions. The experiment was carried out using MATLAB on hp laptop.

Table 1 summarizes the performance of the proposed algorithm in terms of precision, recall, and F1 score. The performance evaluation is carried out using different types of moments selection order. In addition, an evaluation of the features with and without MOGs is performed. In the experiment, the number of blocks ($v_1 \times v_2$) is set to 4×4 . The moment selection ratios considered in the experiment are 10% and 20% out of the total number of moments (please see Fig. 1). Note that, the actual number of blocks are $2v_1 \times 2v_2$. This is because the localization property of the KTP can represent each block by 4 sub-blocks; therefore, the computation cost is reduced by 4.

From Table 1, it is observed that the moments selection ratio of 10% shows better performance accuracy than that with 20%. This is because the more moments included, the more fluctuation in the similarity signal is achieved. In addition, the proposed algorithm accuracy measures with MOGs shows better results when compared to the accuracy measures without MOGs. In other words, the MOGs reduce the effect of disturbance factors.

In addition to the accuracy results, the computational time is also computed for each dataset and reported in Table 2. The computational time, the number of processed frames, and the processing time per frame are presented in Table 2. It can be noted that the computational cost is increased when moment selection ratio of 20% is selected compared to 10%. For instance, the computational cost is reduced by 89 sec when considering 10% of moments and smoothed moments for feature extraction. Likewise, the total processing time is decreased by 162 sec when MOGs are considered. Furthermore, it is observed that the computational cost of the SBD algorithm is increased by ~ 2.5 times when both smoothed

Table 1 The accuracy measures of the proposed SBD

Dataset	Moment ratio	Without MOGs			With MOGs		
		$\mathcal{P}\%$	$\mathcal{R}\%$	$F1\%$	$\mathcal{P}\%$	$\mathcal{R}\%$	$F1\%$
2005	10	95.18	96.99	96.08	95.35	98.14	96.72
	20	94.93	96.69	95.80	95.22	97.97	96.57
2006	10	93.06	96.40	94.70	93.09	96.98	95.00
	20	92.79	95.83	94.28	92.96	96.66	94.77
2007	10	97.15	96.63	96.89	97.13	97.56	97.34
	20	96.81	96.38	96.59	96.98	97.46	97.22

Table 2 The processing time of the proposed SBD algorithm

Dataset	Moments ratio	Processed frames	Without MOGs processing time		With MOGs processing time	
			All frames (sec)	Per frame (msec)	All frames (sec)	Per frame (msec)
2005	10	588,439	1547	2.63	3298	5.60
	20	588,439	1567	2.66	3386	5.75
2006	10	499,590	1318	2.64	2827	5.66
	20	499,590	1353	2.71	2866	5.74
2007	10	576,398	1581	2.74	3286	5.70
	20	576,398	1615	2.80	3321	5.76
Total	10	1,664,427	4446	2.67	9411	5.65
	20	1,664,427	4535	2.72	9573	5.69



Fig. 4 Sample of HTs correctly detected using the proposed algorithm with the presence of disturbance factors

and MOGs are used. Although the computational time increases, the accuracy of the proposed SBD algorithm is increased (please see Table 1). Moreover, the computational time of the proposed algorithm is ~ 16% of real time video, which is adequate and can be used for real-time processing.

Particularly, TRECVID datasets are challenging owing to the diversity of their content from static to highly disturbed shots. Figure 4 shows the visual results of the correctly detected HT by the proposed algorithm. Different types of disturbance factors are illustrated



Fig. 5 Samples of different disturbance factors in the video datasets

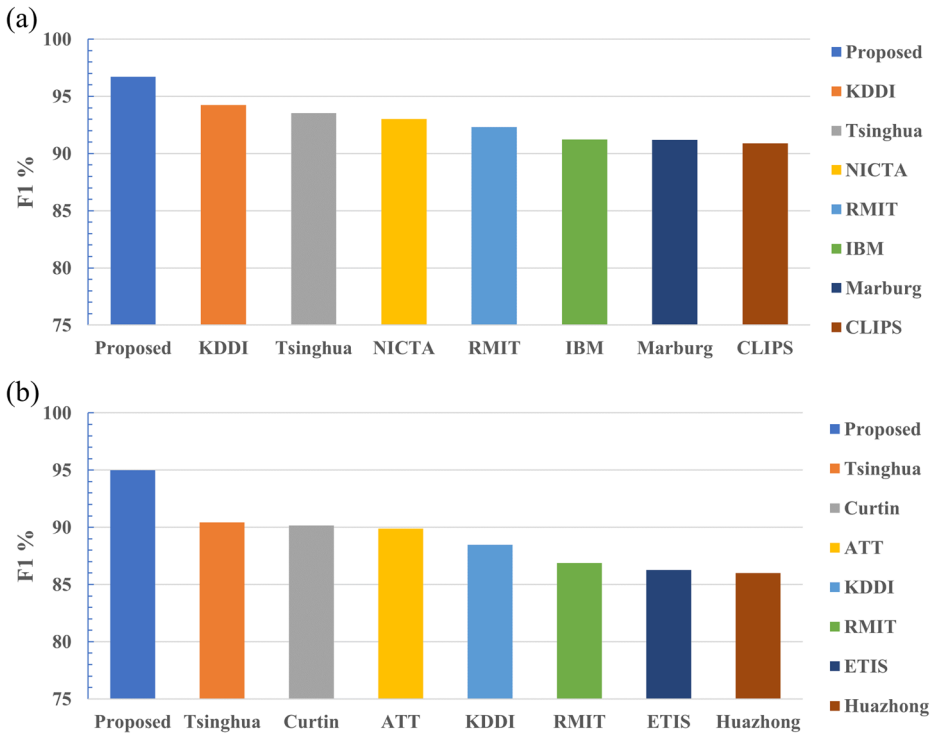


Fig. 6 Comparison between proposed algorithm and top competitors in TRECVID **a** 2005, and **b** 2006

in Fig. 4 as well as the occurrence of HT between them. As shown in the figure, the proposed algorithm is able to distinguish between HTs and disturbance factors such as local object motion, global object motion, and object occlusion. Furthermore, the proposed algorithm is able to distinguish HTs between two highly similar shots as depicted in the last row of Fig. 4.

For more illustration, Fig. 5 shows different sampled frame sequences with disturbance factors in the datasets which are correctly classified as non-transition sequence. These disturbance factors include camera flash light, explosion, fast object motion, and different camera operations.

For comparison purpose, the proposed SBD is first compared to the top 7 competitors in TRECVID 2005 [30] and TRECVID 2006 [31] as shown in Fig. 6. It can be observed that the proposed method outperforms the top competitors in TRECVID 2005 and 2006

Table 3 Accuracy comparison using TRECVID2007 dataset

Algorithm	Processing time per Frame (msec)	$P\%$	$R\%$	$F1\%$
Proposed SBD	5.7	97.13	97.56	97.34
SVMNSCT-SBD [27]	142.48	96.36	97.66	97.01
WHT-SBD [36]	96.63	97.42	97.79	97.61
THNSCT-SBD [37]	33,948.73	84.71	71.09	77.30

Table 4 Accuracy comparison using TRECVID2005 dataset

Algorithm	Processing time per frame (msec)	$\mathcal{P}\%$	$\mathcal{R}\%$	$F1\%$
Proposed SBD	5.6	95.35	98.14	96.72
CBBH-SBD [50]	40.77	96.00	95.00	95.50

competitions in terms of harmonic mean ($F1\%$). For TRECVID 2005, as shown in Fig. 6a, the proposed algorithm outperforms the top 7 competitors in TRECVID 2005 SBD task. The presented algorithm shows an improvement of $\sim 2.5\%$ when compared with KDDI algorithm (highest accuracy in top TRECVID 2005), and an improvement of $\sim 5.8\%$ when compared with CLIPS algorithm (lowest accuracy in top TRECVID 2005). For TRECVID 2006, as shown in Fig. 6b, the improvements of the proposed algorithm are $\sim 4.5\%$ and $\sim 9.0\%$ when compared with Tsinghua (highest accuracy in TRECVID 2006) and Huazhong (lowest accuracy in TRECVID 2006), respectively.

In addition, the proposed algorithm is compared to the state-of-the-art algorithms which are: SVM and Non-Subsampled Contourlet Transform-based SBD (SVMNSCT-SBD) [27], threshold and Non-Subsampled Contourlet Transform-based SBD (THNSCT-SBD) [37], Walsh-Hadamard transform-based SBD (WHT-SBD) [36], and concatenated block based histograms-based SBD (CBBH-SBD) [50]. The comparison is presented in Tables 3 and 4 in terms of accuracy and computation cost. It can be noted that the proposed algorithm shows an improvement compared to THNSCT-SBD and CBBH-SBD algorithms, and demonstrates a comparable result compared to SVMNSCT-SBD and WHT-SBD algorithms in terms of harmonic mean ($F1 - score$). On the other hand, the proposed algorithm shows a remarkable progress in terms of the time required to process video frames (please see Tables 3 and 4).

Although, the proposed algorithm shows promising results in terms of accuracy and computational cost, the performance of the algorithm can be further improved in terms of computational cost by using multi-core CPU/GPU [54, 55] and disturbed optimization [47, 48].

6 Conclusion

In this paper, a new SBD algorithm based on new features, that are extracted using OP, is presented. Smoothing frames is used to eliminate noise; while computing frame gradient in x -direction and y -direction is used for frame feature extraction. This work is based on moments extracted from smoothed and gradients video frames to detect HT effectively. The temporal information is taken into account when feature vector is constructed. The results shows that the presented algorithm outperforms the TRECVID competitors and different state-of-the-art algorithms. However, it is noticed that the computation cost is increased when convolving multiples image kernels with each video frame. Therefore, our future work is going towards further reducing the computational cost of the SBD algorithm as well as the detection of soft transitions so that the algorithm can be generalized to detect different transitions.

Acknowledgements We would like express our heartfelt thanks to all of the anonymous reviewers for their efforts, valuable comments and suggestions.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Abdulhussain SH (2017) On computational aspects of Tchebichef polynomials for higher polynomial order. *IEEE Access* 5(1):2470–2478
2. Abdulhussain SH, Ramli AR, Mahmmod BM, Al-Haddad SAR, Jassim WA (2017) Image edge detection operators based on orthogonal polynomials. *Int J Image Data Fus* 8(3):293–308
3. Abdulhussain SH, Ramli AR, Al-Haddad SAR, Mahmmod BM, Jassim WA (2018) Fast recursive computation of Krawtchouk polynomials. *J Math Imaging Vis* 60(3):285–303
4. Abdulhussain SH, Ramli AR, Saripan MI, Mahmmod BM, Al-Haddad S, Jassim WA (2018) Methods and challenges in shot boundary detection: a review. *Entropy* 20(4):214
5. Birinci M, Kiranyaz S (2014) A perceptual scheme for fully automatic video shot boundary detection. *Signal Process Image Commun* 29(3):410–423
6. Camara-Chavez G, Precioso F, Cord M, Phillip-Foliguet S, de A Araujo A (2007) Shot boundary detection by a hierarchical supervised approach. In: 2007 14th international workshop on systems, signals and image processing and 6th EURASIP conference focused on speech and image processing, multimedia communications and services, pp 197–200
7. Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol (TIST)* 2(3):27
8. Chaves GC (2007) Video content analysis by active learning. PhD Thesis, Federal University of Minas Gerais
9. Chen X, He F, Yu H (2018) A matting method based on full feature coverage. *Multimed Tools Appl*, 1–29
10. Cooper M, Foote J, Adcock J, Casi S (2003) Shot boundary detection via similarity analysis. In: Proceedings of the TRECVID workshop
11. Gargi U, Kasturi R, Strayer SH (2000) Performance characterization of video-shot-change detection methods. *IEEE Trans Circ Syst* 8(10):4761–4766
12. Hsu C-W, Chang C-C, Lin C-J et al (2003) A practical guide to support vector classification. Department of Computer Science and Information Engineering
13. Hu W, Xie N, Li L, Zeng X, Maybank S (2011) A survey on visual content-based video indexing and retrieval. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 41(6):797–819
14. Jaffrè G, Joly P, Haidar S (2004) The Samova shot boundary detection for TRECVID evaluation 2004
15. Janwe NJ, Bhoyar KK (2013) Video shot boundary detection based on JND color histogram. In: 2013 IEEE Second international conference on image information processing (ICIIP), pp 476–480
16. Ji QG, Feng JW, Zhao J, Lu ZM (2010) Effective dissolve detection based on accumulating histogram difference and the support point. In: 2010 First international conference on pervasive computing signal processing and applications (PCSPA), pp 273–276
17. Kar T, Kanungo P (2017) A motion and illumination resilient framework for automatic shot boundary detection. *Signal Image Vid Process* 11(7):1237–1244
18. Koprinska I, Carrato S (2001) Temporal video segmentation: a survey. *Signal Process Image Commun* 16(5):477–500
19. Krulikovska L, Pavlovic J, Polec J, Cernekova Z (2010) Abrupt cut detection based on mutual information and motion prediction. In: ELMAR 2010 proceedings, pp 89–92
20. Küçükünç O, Gündükbay U, Ulusoy Ö (2010) Fuzzy color histogram-based video segmentation. *Comput Vis Image Underst* 114(1):125–134
21. Li K, He F, Yu H, Chen X (2017) A parallel and robust object tracking approach synthesizing adaptive Bayesian learning and improved incremental subspace learning. *Front Comput Sci*, 1–20
22. Li K, He F-Z, Yu H-P (2018) Robust visual tracking based on convolutional features with illumination and occlusion handling. *J Comput Sci Technol* 33(1):223–236
23. Liu C, Wang D, Zhu J, Zhang B (2013) Learning a contextual multi-thread model for Movie/TV scene segmentation. *IEEE Trans Multimed* 15(4):884–897

24. Liu L, Hua Y, Zhao Q, Huang H, Bovik AC (2016) Blind image quality assessment by relative gradient statistics and adaboosting neural network. *Signal Process Image Commun* 40(Supplement C):1–15
25. Mahmmod BM, Ramli AR, Abdulhussain SH, Al-Haddad SAR, Jassim WA (2017) Low-distortion MMSE speech enhancement estimator based on Laplacian prior. *IEEE Access* 5(1):9866–9881
26. Mahmmod BM, bin Ramli AR, Abdulhussain SH, Al-Haddad SAR, Jassim WA (2018) Signal compression and enhancement using a new orthogonal-polynomial-based discrete transform. *IET Signal Process* 12(1):129–142
27. Mondal J, Kundu MK, Das S, Chowdhury M (2017) Video shot boundary detection using multiscale geometric analysis of nscst and least squares support vector machine. *Multimed Tools Appl*, 1–23
28. Mukundan R (2004) Some computational aspects of discrete orthonormal moments. *IEEE Trans Image Process* 13(8):1055–1059
29. Noble WS (2006) What is a support vector machine? *Nat Biotechnol* 24(12):1565–1567
30. Over P, Ianeva T, Kraaij W, Smeaton AF, Val UD (2005) TRECVID 2005 - an overview. NIST, 1–27
31. Over P, Ianeva T, Kraaij W, Smeaton AF (2006) TRECVID 2006 - an overview. NIST, 1–29
32. Pacheco F, Cerrada M, Sánchez R-V, Cabrera D, Li C, de Oliveira JV (2017) Attribute clustering using rough set theory for feature selection in fault severity classification of rotating machinery. *Expert Syst Appl* 71:69–86
33. Parmar M, Angelides MC (2015) MAC-REALM: a video content feature extraction and modelling framework. *Comput J* 58(9):2135–2170
34. Porter SV, Mirmehdi M, Thomas BT (2000) Video cut detection using frequency domain correlation. In: 2000 Proceedings 15th international conference on pattern recognition, vol 3. IEEE, pp 409–412
35. Priya GL, Domnic S (2012) Edge strength extraction using orthogonal vectors for shot boundary detection. *Procedia Technology* 6:247–254
36. Priya LGG, Domnic S (2014) Walsh – Hadamard transform kernel-based feature vector for shot boundary detection. *IEEE Trans Image Process* 23(12):5187–5197
37. Sasithradevi A, Roomi SMM, Raja R (2016) Non subsampled contourlet transform based shot boundary detection in videos. *Int J Control Theory Appl* 9(7):3231–3238
38. Sergios Theodoridis KK (2008) Pattern recognition, 4th edn. Academic Press
39. Smeaton AF, Over P, Doherty AR (2010) Video shot boundary detection: seven years of TRECVID activity. *Comput Vis Image Underst* 114(4):411–418
40. Solomon C, Breckon T (2011) Fundamentals of digital image processing: a practical approach with examples in matlab. Wiley-Blackwell
41. Swanberg D, Shu C-F, Jain RC (1993) Knowledge-guided parsing in video databases. In: IS&T/SPIE’s symposium on electronic imaging: science and technology. International Society for Optics and Photonics, pp 13–24
42. Thounaojam DM, Khelchandra T, Singh KM, Roy S (2016) A genetic algorithm and fuzzy logic approach for video shot boundary detection. *Comput Intell Neurosci* 2016(Article ID 8469428):14
43. Tong W, Song L, Yang X, Qu H, Xie R (2015) Ieee CNN-based shot boundary detection and video annotation. In: 2015 IEEE international symposium on broadband multimedia systems and broadcasting
44. Urhan O, Gullu MK, Erturk S (2006) Modified phase-correlation based robust hard-cut detection with application to archive film. *IEEE Trans Circ Syst Vid Technol* 16(6):753–770
45. Vlachos T (2000) Cut detection in video sequences using phase correlation. *IEEE Signal Process Lett* 7(7):173–175
46. Xu J, Tang YY, Zou B, Xu Z, Li L, Lu Y (2015) The generalization ability of online SVM classification based on Markov sampling. *IEEE Trans Neural Netw Learn Syst* 26(3):628–639
47. Yan X-H, He F-Z, Chen Y-L (2017) A novel hardware/software partitioning method based on position disturbed particle swarm optimization with invasive weed optimization. *J Comput Sci Technol* 32(2):340–355
48. Yan X, He F, Hou N, Ai H (2018) An efficient particle swarm optimization for large-scale hardware/software co-design system. *Int J Coop Inf Syst* 27(01):1741001
49. Yoo H-W, Ryoo H-J, Jang D-S (2006) Gradual shot boundary detection using localized edge blocks. *Multimed Tools Appl* 28(3):283–300
50. Youssef B, Fedwa E, Driss A, Ahmed S (2017) Shot boundary detection via adaptive low rank and svd-updating. *Comput Vis Image Understand* 161(Supplement C):20–28
51. Zabih R (1999) A feature-based algorithm for detecting and classifying production effects. *Multimed Syst* 7(2):119–128
52. Zabih R, Miller J, Mai K (1995) A feature-based algorithm for detecting and classifying scene breaks. In: Proceedings of the third ACM international conference on multimedia MULTIMEDIA 95, vol 95, pp 189–200

53. Zhang H, Kankanhalli A, Smoliar SW (1993) Automatic partitioning of full-motion video. *Multimed Syst* 1(1):10–28
54. Zhou Y, He F, Qiu Y (2017) Dynamic strategy based parallel ant colony optimization on gpus for tsps. *Sci Chin Inf Sci* 60(6):068102
55. Zhou Y, He F, Hou N, Qiu Y (2018) Parallel ant colony optimization on multi-core simd cpus. *Futur Gener Comput Syst* 79:473–487



Sadiq H. Abdulhussain was born in Baghdad, Iraq, in 1976. He received the B.Sc. and M.Sc. degrees in Electrical Engineering from Baghdad University, in 1998 and 2001, respectively. In 2018, he received the PhD degree in Computer and Embedded System Engineering from Universiti Putra Malaysia. Since 2005 until now, he is staff member at department of Computer Engineering, Faculty of Engineering, University of Baghdad. His research interests include computer vision, signal processing, speech and image processing.



Abd Rahman Ramli received a Bachelor of Science in Electronics, Universiti Kebangsaan Malaysia in 1982. Then he proceeded with his Master Degree in Information Technology System at the University of Strathclyde, United Kingdom in 1985. In 1990, he pursued his doctoral studies in University of Bradford, United Kingdom. He was appointed as the Head of Computer and Communication System Engineering in August 1996 until July 1998. Abd Rahman had also served as Head of Intelligent Systems and Robotics Laboratory, Institute of Advanced Technology, Universiti Putra Malaysia where he leads a cutting edge research laboratory in Real-Time and Embedded Systems, Intelligent Systems and Perceptual Robotics. His research interests are in the area of image processing and electronic imaging, multimedia systems engineering, embedded system and intelligent systems.



Basheera M. Mahmmod was born in Baghdad, Iraq, in 1975. She received the B.Sc. in Electrical Engineering in 1998 from Baghdad University. Then, she proceeded with her Master Degree in Electronic and Communication Engineering / Computer from Baghdad University in 2012. In 2018, she received the PhD degree in Computer and Embedded System Engineering from Universiti Putra Malaysia. Since 2007 until now, she is staff member at department of Computer Engineering, Faculty of Engineering, University of Baghdad. Her research interests include speech enhancement, signal processing, computer vision, RFID, and cryptography.



M. Iqbal Saripan received the B.Eng. degree in electronic engineering from the Universiti Teknologi Malaysia in 2001 and the Ph.D. degree from the University of Surrey, Guildford, U.K., in 2006. Currently, he is a lecturer at the Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia. Also, he is Director Center for Quality Assurance (CQA), Universiti Putra Malaysia. His research interests are in the area of computer security, signal processing, image processing and embedded system. He is a member of the IEEE and IoP UK.



S. A. R. Al-Haddad is a PhD holder in Electrical, Electronic and Systems Engineering from National University Malaysia. Specialized in Human Speech Processing, Animal Sound Processing, AI-Quran Sound Processing, Sound Media Security and Biometric. Dr. Al-Haddad had served Department of Computer and Communications Systems Engineering, Universiti Putra Malaysia since 1997 and promoted to Associate Professor in year 2012. Furthermore, he had teach students for undergraduate and graduate in Malaysia and International over than 19 years. On research, he published more than hundreds journals and proceedings. In research, he has more than twenty International and national grants and had six patents and copyrights. On the other hand, Dr. Al-Haddad is Head of Laboratory Information Engineering and Robotics. He also actively join professional society such as IEEE as Senior Member, Deputy Chair IEEE Systems, Man and Cybernetics, MITS, MSET and others.



Wissam A. Jassim was born in Baghdad, Iraq, in 1976. He received the B.Sc. and M.Sc. degrees in Electrical Engineering from Baghdad University, in 1999 and 2002, respectively, and the Ph.D. degree in Electrical Engineering from the University of Malaya, Kuala Lumpur, Malaysia in 2012. From 2013 to 2015, he was a Visiting Research Fellow with the Department of Biomedical Engineering, University of Malaya, Kuala Lumpur, Malaysia. From 2015 to 2016, he was a Post-Doctoral Fellow with the Department of Electrical Engineering, University of Malaya, Kuala Lumpur, Malaysia. He is currently a Research Fellow with the ADAPT Center, School of Engineering, Trinity College Dublin, the University of Dublin, Dublin 2, Ireland. His current research interests include machine learning, speech, and image processing.

Affiliations

Sadiq H. Abdulhussain¹  · Abd Rahman Ramli² · Basheera M. Mahmmod¹ · M. Iqbal Saripan² · S. A. R. Al-Haddad² · Wissam A. Jassim³

Abd Rahman Ramli
arr@upm.edu.my

Basheera M. Mahmmod
basheera412@yahoo.com

M. Iqbal Saripan
iqbal@upm.edu.my

S. A. R. Al-Haddad
sar@upm.edu

Wissam A. Jassim
wissam.jassim@tcd.ie

- ¹ Department of Computer Engineering, University of Baghdad, Baghdad, Iraq
- ² Department of Computer and Communication System Engineering, Universiti Putra Malaysia, Selangor, Malaysia
- ³ ADAPT Center, School of Engineering, Trinity College Dublin, University of Dublin, Dublin 2, Ireland