IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 21, 2020, accepted April 9, 2020, date of publication April 14, 2020, date of current version April 30, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2987870*

# Fast Temporal Video Segmentation Based on Krawtchouk-Tchebichef Moments

**SADIQ H. ABDULHUSSAIN[1], (Member, IEEE), SYED ABDUL RAHMAN AL-HADDAD[2], (Senior Member, IEEE), M. IQBAL SARIPAN[2], (Member, IEEE), BASHEERA M. MAHMMOD[1], AND ASEEL HUSSIEN[3]**

[1]Department of Computer Engineering, University of Baghdad, Baghdad 10071, Iraq
[2]Department of Computer and Communication System Engineering, Universiti Putra Malaysia, Selangorm 43400, Malaysia
[3]Department of Computer Science, Liverpool John Moores University, Liverpool L3 3AF, U.K.

Corresponding author: Syed Abdul Rahman Al-Haddad (sar@upm.edu.my)

**ABSTRACT** With the increasing growth of multimedia data, the current real-world video sharing websites are being huge in repository size, more specifically video databases. This growth necessitates to look for superior techniques in processing video because video contains a lot of useful information. Temporal video segmentation (TVS) is considered essential stage in content-based video indexing and retrieval system. TVS aims to detect boundaries between successive video shots. TVS algorithm design is still challenging because most of the recent methods are unable to achieve fast and robust detection. In this regard, this paper proposes a TVS algorithm with high precision and recall values, and low computation cost for detecting different types of video transitions. The proposed algorithm is based on orthogonal moments which are considered as features to detect transitions. To increase the speed of the TVS algorithm as well as the accuracy, fast block processing and embedded orthogonal polynomial algorithms are utilized to extract features. This utilization will lead to extract multiple local features with low computational cost. Support vector machine (SVM) classifier is used to detect transitions. Specifically, the hard transitions are detected by the trained SVM model. The proposed algorithm has been evaluated on four datasets. In addition, the performance of the proposed algorithm is compared to several state-of-the-art TVS algorithms. Experimental results demonstrated that the proposed algorithm performance improvements in terms of recall, precision, and F1-score are within the ranges (1.31 - 2.58), (1.53 - 4.28), and (1.41 - 3.03), respectively. Moreover, the proposed method shows low computation cost which is 2% of real-time.

**INDEX TERMS** Temporal video segmentation, shot boundary detection, orthogonal polynomials, orthogonal moments.

## I. INTRODUCTION

The immense growth of computer performances and the low cost of storage devices during the past decades led to the dominance of multimedia data in the cyberspace, rise in the volume of transmitted data, and the size of repositories [1]. However, video, among multimedia data, is considered the most consumed concerning the storage space [2]. Nowadays, the size of video databases is dramatically increasing annually. For example YouTube, one of the popular VSW globally [3], approximately 300 video hours every minute were uploaded in 2019 and 5 billion videos watched every day [1].

Video indexing and retrieval are used to appropriately and swiftly save and arrange video data [4]. Thus, and due to the enormous of video databases, a necessity for automated and robust analysis of video content is demanded [4]–[7]. Video content analysis comprises video indexing and retrieval with respect to their spatiotemporal, visual and semantic contents [8]. Content based video indexing and retrieval (CBVIR) have various and wide applications. For example, browsing videos, management in video sharing websites, digital museums, news event analysis, and video surveillance [4].

A video shot is the basic building block of the video [1], [9]. Video shot is defined as a consecutive sequence of frames that have temporal and scene connection. Frames of video shot are picked out by a single camera operation [5], [10], [11]. These frames are combined to form a scene during video production process. In addition, scenes are aggregated to form the entire video. Boundaries between shots is known as a shot transition. There are two main types of shot transitions: hard

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

and soft transitions. Hard transition (HT) is the process of aggregating two shots directly. On the other hand, soft transition (ST) is the process of aggregating two shots by involving multiple frames [12]. Generally, frames involved in a transition are not preferred for video indexing or summarization processes because they have low information content [7].

TVS, named also shot boundary detection, aims to partition a video into shots by detecting transitions between them. Then, video shots are forwarded to the CBVIR [13], [14]. In other words, TVS is utilized as an initial and substantial stage in CBVIR; where, its performance affects the results of the next CBVIR stages [7], [15], [16].

Detection of transitions in TVS algorithms is performed by the statistical machine learning-based and/or rules-based techniques. The machine learning-based technique includes supervised and unsupervised learning [17]. Feature extraction process is a substantial step in TVS algorithms which aims to acquire significant depiction of the visual information [12]. Feature extraction can be categorized based on the algorithm processing domain into: compressed and uncompressed domains [18]. TVS algorithms are primarily centered on the uncompressed domain, for instance, pixel-based algorithms [19]. Then they are developed to encompasses other approaches such as: edge information [20], histogram of video frames [21], transform coefficients [22], and local keypoint [1]. Several researchers employed the coefficients of discrete transforms as a feature extraction tool, such as discrete Fourier transform, discrete Wavelet transform, and discrete Walsh-Hadamard transform. These methods exhibits a good performance in detecting video shot transitions [23]; however, their computational cost is considered high [17].

Generally, the ranking of a TVS algorithm depends on the algorithm ability to swiftly detect the shot transition (shot boundary). That is, the performance of TVS algorithms can be calculated by their accuracy in discovering correct transition and the time required to detect transitions [23], [24]. In addition, improvement in terms of the detection accuracy for HTs and STs is still demanded [17].

Motivated by these issues, this paper proposes a fast and accurate TVS algorithm based on discrete orthogonal moments. Where, fast block processing to extract is used to extract local discrete orthogonal moments. To increase the performance of the TVS algorithm, the embedded image kernel is used and combined with the fast block processing algorithm to extract multiple features.

This paper is organized as follows: Section II presents a survey on the related work. Section III describes the fundamentals of OPLs and their moments. Section IV provides the proposed TVS algorithm to detect HT. Section V displays the results that highlight the effectiveness of proposed TVS algorithm. Finally, Section VI concludes the paper.

## II. RELATED WORK

The performance of TVS algorithms show a trade-off between the computation cost and accuracy [24]. The existing TVS algorithms either show high recall at the expense of high false detection rate, i.e. low precision, or low false detection rate at the expenses of low recall [23]. The other significant factor that influences TVS performance is the algorithm computation cost which is always required to be lessen, i.e. the speed of TVS algorithm needs to be increased. Note that, within a shot, frames are very comparable in their content. Therefore, when a transition occurs, a variation in the values of te similarity will appear [24]. However, in a scene, the rate of change is very low; thus, miss detection is occurred [24]. In addition to that, there are special effects that occur in video scene such as; flashlights or light variations, object and camera motion, and camera operation. These effects impact on TVS algorithm performance. For better TVS algorithm performance, TVS algorithms during transition detection process should be able to detect transitions, minimize false alarm rate within a shot, and lessen miss detects. Accordingly, design of a TVS algorithm, which can combine the solutions to these problems, becomes a necessity.

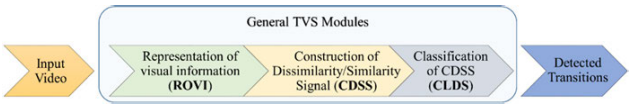


**FIGURE 1.** General TVS modules.

Generally, TVS module encompasses three sub-modules (see FIGURE 1) [24]: (1) feature extraction of visual information content, (2) establishing a similarity/dissimilarity values, and (3) classification of the similarity/dissimilarity values [25]. The previously mentioned modules may contain pre-processing and/or post-processing steps.

The features are extracted from video frame sequence to characterize the visual content of video frame. Visual content representation comprises several types based on pixel, histogram, edge, local key point, and transform.

TVS algorithms based on pixel information directly utilize pixel intensities of video frame sequence for video content characterization. Thus, they are considered fast and simple algorithms. Despite that, these algorithms are rated to be responsive to camera motion, object motion, global motion, and diverse types of camera operations [26]. Generally, high sensitivity of any algorithm lead to high false detection; accordingly low precision rate. In addition to their high sensitivity, pixel-based algorithm suffers from missed detection of transitions.

TVS algorithms based on histograms reflect the number of frame's intensities that are registered in a predefined range. Histograms are considered a substitution for algorithms based on pixel intensities because the former do not consider spatial information. Hence, histograms, partly, are considered constant to small local and global motions compared with pixel-based algorithms [4], [27], [28]. TVS algorithms based on histogram assumes that the histograms are comparable for two consecutive frames having stationary object and background.

Compared to algorithms based on pixel intensities, histogram-based algorithms are not responsive to object motion and camera motion [24]. However, large object motion and camera motion make a variation in the similarity/dissimilarity values. Thus, a false detection is declared [13], [29]. In addition, flash light occurrence, panning, tilting, and zooming leads to false positives [30]. Consequently, employing histogram to detect HTs without false positives and negatives is considering insufficient [4], [25].

Edge-based techniques (EBTs) considered a low-level feature of a frame and more invariant to illumination changes. EBTs are designed to detect HTs. In EBTs, a transition is detected when a large distance between edges are exhibited between the current and previous frames. The required processes for computing edge changes are: edge detection for both current and previous frames, edge change ratio, and motion compensation [31]. EBTs are less reliable in terms of computational cost and performance when compared to other algorithms [24]. EBTs are prone to high rates of false alarms due to different factors, such as camera operations [24].

Local key point (LKP) and their descriptors are employed by many computer vision applications. Surrounding region of LKP is scale-invariant and LKP descriptor can be computed from that area. Speeded up robust features (SURF) [32], scale-invariant feature transform (SIFT) [33], and Harris corner detector [34] are methods for LKP extraction. The idea behind using local descriptors is that the LKP matching of objects or background within intra-shot frames are high, while LKP shows high variation within inter-shot frames. To find dissimilarity (DS), LKPs were extracted and matched for two successive frames. The alteration in the number of matched LKPs, i.e. DS, were observed in order to detect transitions. One of the earliest LKP implementation is proposed in [35]. They utilized SIFT for TVS algorithm by assuming high NoMK within shot and close zero between shots.

Discrete transform allows to view signals in different domains and gives the ability to analyze the components of various signals [36]. Discrete transforms such as DFT and DCT are characterized by their EC capability and localization property. Transform-based techniques (TBTs), transform a frame from the spatial domain into the transform domain [24]. For example, [37] used FFT, while [22], [38] employed Wlash-Hadamard Transform. Although this approach has good detection accuracy, they are considered consuming in terms of computational cost.

## III. PRELIMINARIES

In this section the utilized orthogonal polynomials (OPLs) and their mathematical model are introduced. In addition, the mathematical expression for the computation of orthogonal moments is illustrated.

### A. ORTHOGONAL POLYNOMIALS

Orthogonal polynomial is a square matrix with two axes, namely polynomial order ($n$) and signal index ($x$). The values of the matrix coefficient is called orthogonal polynomial coefficients (OPCFs). Generally, These coefficients is defined by hypergeometric series and gamma function. Tchebichef polynomials (TPLs) and Krawtchouk polynomials (KPLs) are widely used for their ability in signal compression and image representation [39], [40]. Due to their powerful capability, different hybrid are formed from them such as Krawtchouk-Tchebichef polynomial (KTP) [36] and sqaured Krawtchouk-Tchebichef polynomial (SKTP) [41]. In this paper, SKTP is employed as an OPL because of its performance in terms of energy compaction and localization property over other existing OPLs [41]. The SKTP ($\mathscr{U}$) is expressed as follows:

$$\mathscr{U}_n(x;p) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} \mathscr{K}_j(i;p)\,\mathscr{T}_j(x)\,\mathscr{K}_l(n;p)\,\mathscr{T}_l(i) \qquad n, x = 0, 1, \cdots, N-1 \quad (1)$$

where $\mathscr{K}$ and $\mathscr{T}$ are the KPLs and TPLs functions, respectively, and $p$ is the controlling parameter. The normalized KPL functions are defined as follows [42]:

$$\mathscr{K}_n(n;p) = \sqrt{\frac{\omega_{\mathscr{K}}(x)}{\rho_{\mathscr{K}}(n)}}\,{}_2F_1\left(-n, -x; -N+1; \frac{1}{p}\right) \quad (2)$$

where $\rho_{\mathscr{K}}$, $\omega_{\mathscr{K}}$ are the norm and eight functions of the KPL, respectively. The normalized TPL functions are given by [43]:

$$\mathscr{T}_n(x) = \sqrt{\frac{\omega_{\mathscr{T}}(x)}{\rho_{\mathscr{T}}(n)}}\,(1-N)_n\,{}_3F_2\,(-n, -x, 1+n; 1-N; 1) \quad (3)$$

where $\rho_{\mathscr{K}}$ $\omega_{\mathscr{K}}$ are the norm and eight functions of the TPL, respectively.

To compute the KPL and TPL, the three-terms recurrence relations are utilized. The computing KPL and TPL which are employ Equations (2) and (3) are computationally cost and unstable because of the hypergeometric series (${}_2F_1$ and ${}_3F_2$) and gamma functions [41], [44].

### B. ORTHOGONAL MOMENTS

Orthogonal moments (OMs) are defined as set of scalar quantities which are efficient and superior data descriptor [45]. They are utilized for signal information representation without redundancy. In addition, OMs are employed to reveal small variations in the signal intensity [46]. Generally, the most energy of the signal (information) are contained in the low-order moments contain, whereas the signal details are carried out by the higher-order moments [47]. For a 2D signal, the SKTP moments ($\mathscr{M}_{nm}$) are computed as follows:

$$\mathscr{M}_{nm} = \sum_{x=0}^{N_1-1}\sum_{y=0}^{N_2-1} \mathscr{U}_n(x;p,N_1)\mathscr{U}_m(y;p,N_2)f(x,y)$$
$$n = \frac{N_1}{2} - 1, \frac{N_1}{2}, \ldots, \frac{N_1 - O_n}{2}, \frac{N_1 + O_n}{2} - 1$$
$$m = \frac{N_2}{2} - 1, \frac{N_2}{2}, \ldots, \frac{N_2 - O_m}{2}, \frac{N_2 + O_m}{2} - 1 \quad (4)$$

where $O_n$ and $O_m$ are the maximum order of moments which are used to represent the signal, and $f(x, y)$ represents the 2D signal (image). In practice, matrix multiplication can be used to compute moments, as follows:

$$\mathbf{M} = \mathbf{U}_1 \, \mathbf{F} \, \mathbf{U}_2^T \tag{5}$$

where $\mathbf{F}$ represents the matrix form of the image $f(x, u)$, $\mathbf{M}$ demonstrates the matrix form of the moments $\mathscr{M}_{nm}$, and $\mathbf{U}_1$ and $\mathbf{U}_2$ represent the matrix form of OPLs ($\mathscr{U}_n$ and $\mathscr{U}_m$), respectively. It is noteworthy that the basis functions of OPs can be utilized as an approximate solution for differential equations [48].

## IV. THE PROPOSED TVS METHOD

In this section, the TVS algorithm is presented. The presented TVS algorithm involves three stages. These stages are: 1) feature extraction, 2) dissimilarity signal representation, and 3) transition detection. The significant stage in any TVS algorithm is the feature extraction (first stage). The feature extraction is based on fast block processing OPLs and embedding image kernels. In most TVS algorithms, the features are extracted from the entire image (video frame). However, in the presented algorithm, frame active area is suggested to lessen the effect of persistent and variable visual materials. This is done by considering the frame region that hold most of the information and eliminate regions that affect the accuracy of transition detection.

### A. FRAME ACTIVE AREA

In different types of video, there are two types of embedded visual materials: persistent and variable visual materials. Persistent materials such as fixed station logo, persistent subtitle, and persistent regions which is fixed intensity (commonly black) area usually appears at the upper and lower portion of the frame as shown in FIGURE2.

These visual materials are obviously similar within inter-shot and intra-shot frames and thus additional similarity between features is collected in the CDSS process. On the other hand, variable visual materials such as: animated logo, animated subtitle, and transcript are dissimilar within inter-shot and intra-shot frames as shown in FIGURE3.

Hence, the DS may increase or decrease (conversely, similarity signal (SS) may decrease or increase). Frame active area is suggested to alleviate the effect of the aforementioned visual materials by considering a frame region that hold most of the visual information and eliminate regions that affect the DS/SS.

For an image (video frame) $I$ of size $N_1 \times N_2$, where its pixels are defined in the region $[0, N_1 - 1]$ and $[0, N_2 - 1]$ in the $y$- and $x$- directions, respectively; the frame active area is defined as follows:

$$FAA_y \in \left[\frac{N_1}{8}, N_1 - \frac{N_1}{8} - 1\right] \tag{6}$$

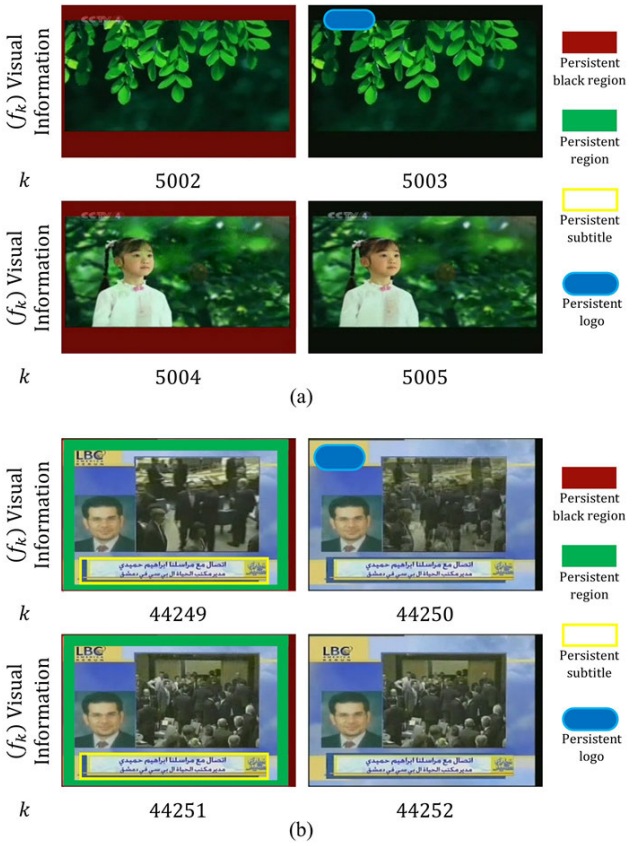


**FIGURE 2.** Persistent visual material extracted from (a) Video ID 01 from Dataset TRECVID 2005, and (b) Video ID 01 from Dataset TRECVID 2006.

$$FAA_x \in \left[\frac{N_2}{16}, N_2 - \frac{N_2}{16} - 1\right] \tag{7}$$

where $FAA_y$, and $FAA_x$ are the active region in y and x-directions, respectively. Mathematically, the active image dimension is $N_{1A} \times N_{2A} = \frac{7}{8}N_1 \times \frac{3}{4}N_2$. FIGURE4 demonstrates the considered frame active area from which the features are extracted and utilized in the proposed TVS algorithm. Besides, features extracted from frame active area are more reliable to that in the inactive or uninformative region. These inactive regions are usually at the frames border, thus eliminating these regions will positively determine the active region of the frame without affecting the significant visual information.

Briefly, frame active area is suggested to lessen the effect of persistent and variable visual materials by considering the frame region that hold most of the information and eliminate regions that affect the DS/SS. In addition, features extracted from frame active area are more informative compared to that exist in the inactive or uninformative region. These inactive regions are usually at the frames border, thus eliminating these regions will definitely determine the active region of the frame without affecting the significant visual information. This will not only consider the important frame information but also increase the speed of the TVS algorithm.

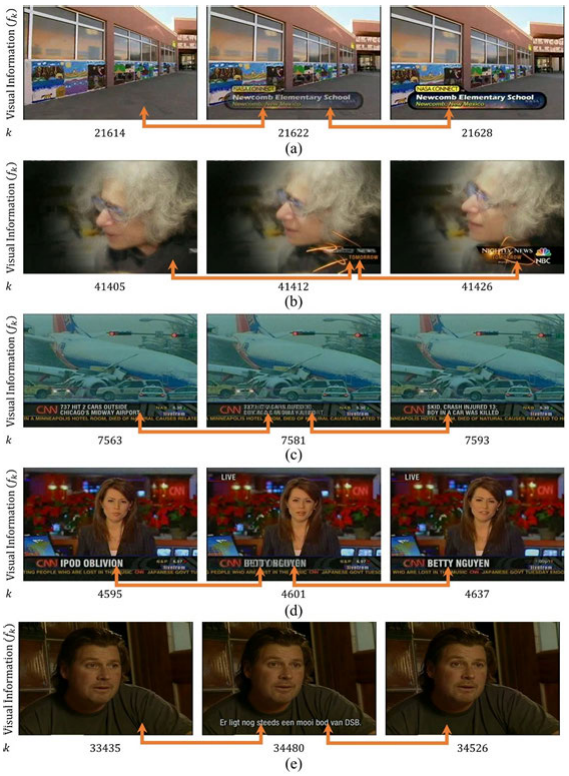


**FIGURE 3.** Variable visual material within intra-shot frames extracted from (a) Video ID 09 from Dataset TRECVID 2005, (b), (c), and (d) Video ID 08, 09, and 12 from Dataset TRECVID 2006, and (e) Video ID 17 from Dataset TRECVID 2007. Note: The orange arrow shows the change in visual information.

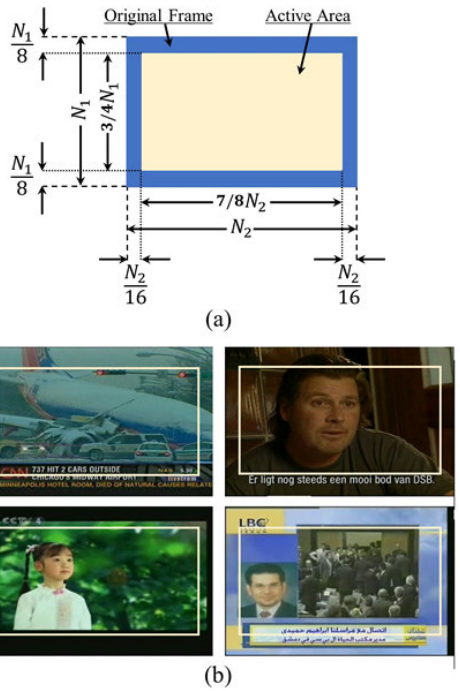


**FIGURE 4.** Frame active area (a) Schematic diagram, and (b) active area examples.

### B. CANDIDATE SEGMENTS SELECTION BASED ON SKTP

Generally, a video frames sequence, in this paper termed as video-frame-level-0 (VFL0), has multiple shots separated by transitions. Each shot has multiple non-transition frames and there are no or some transition frames between consecutive shots. In this work, the first step toward TVS is the selection of candidate transition segments. The main goal of this stage is to exclude the non-transition frames which in turn minimizes the computation cost. Commonly, in a small shot portion, the frames within that period is generally have a high similarity [49]–[51]. Therefore, a segment with a high correlation between first and last segment frames is considered a non-transition segment. While, a low correlation is considered as inter-shot frames.

The candidate transition selection technique presented in [11], [49] is modified and utilized as a first stage in the proposed TVS algorithm. The modified technique is based on a new adaptive threshold, inequalities criteria, and features extracted using SKTP. The modified adaptive threshold and criteria are proposed to select all the transition segments. The candidate transition segment selection can be performed as follows:

I. Video frame sequence is segmented into $N_{skip}$-frames by considering one overlap frame between consecutive segments for DS computation. i.e., last frame of the current segment is the first frame in the next segment. The total number of segments, $N_{segment}$, for video under processing are $\lceil \frac{N_f}{N_{skip}} + 1 \rceil$. Where $N_f$ is the total number of video frames.

II. Compute the distance between the first and last frame for each segment. To compute the distance, these frames are first transformed into moment domain using SKTP. The entire frame is used to compute the moments (global moments) to obtain a high DS between all inter-shot frames. Only the high energy moments are extracted and considered in the distance measure. The distance of the $k$th segment is computed as follows:

$$D(kN_{skip}, (k+1)N_{skip}) = \sum_{i=1}^{O_n} \sum_{j=1}^{O_m} \left| \mathbf{M}(f_k N_{skip}, i, j) - \mathbf{M}(f(k+1)N_{skip}, i, j) \right| \quad k = 0, 1, \cdots, N_{segement} - 1 \tag{8}$$

where $\mathbf{M}(f_i)$ is the moments of the $i$th frame that is computed using (5) by preset the highest order of the SKTP. $O_n$ and $O_m$ are the maximum moment orders in $n$ and $m$ direction. For convenience, the distance measure for each segment will be denoted as $D^{N_{skip}}(k)$ instead of $D(kN_{skip}, (k+1)N_{skip})$.

III. For each 10 group of segments, an adaptive threshold ($Th_{CS}$) is computed using the following suggested

formula:

$$Th_{CS} = m_L + \sqrt{\frac{m_G}{s_L}}\, m_L \tag{9}$$

where $m_G$ is the global mean of all computed distances in the video. $m_L$ and $s_L$ are the local mean and standard deviation for each grouped segment, respectively. The above adaptive threshold formula is proposed to select all segments that have transitions frames.

IV. Minor of transition segments distances have evident increase or decrease, they are missed falsely because their distances are less than the adaptive threshold. Therefore, the distance values between neighboring segments must be taken into account. This is performed using the following nested criteria:

$$\begin{aligned} D^{N_{skip}}(k) &> 5D^{N_{skip}}(k-1) \bigcup D^{N_{skip}}(k) \\ &> 5D^{N_{skip}}(k+1) \bigcup D^{N_{skip}}(k) > 0.7m_G \end{aligned} \tag{10}$$

The distance values that satisfy the above criteria are labeled as candidate transition segments.

In this stage, false alarms are considered better than miss detected transitions [11], [49]. This is because all candidate segments are processed in the following stages in which false alarms can be discarded, therefore, post-processing stages are employed. While segments that do comprise transitions if discarded directly as non-transition segments, they cannot simply be retrieved.

The candidate segment selection algorithm is illustrated in FIGURE5. Note that the frames in the candidate segments are termed as video-frame-level-1 (VFL1).

### C. FEATURE EXTRACTION

Local features are utilized to lessen the effect of object and camera motion within shots [2]. In the traditional methods, the image is partitioned into blocks, then each block is individually processed to extract features. Note that, the extracted features is located in a matrix, such that the features and image block are located correspondingly. Thereafter, the extracted features are processed. Similarly, the local moments are extracted from image blocks. This process is performed sequentially [12]. In addition, partitioning the image into blocks during the moments (features) computation is translated into irregular access patterns which in turn lead to high cache misses and replacements. Therefore, the computational cost of features extraction process is increased which lead to an increases in the processing time for TVS algorithm. Thus, a fast block processing (FBP) algorithm for moment computation [12] is utilized in this paper to reduce the computation time for the computation of moments. The mathematical form of the FBP is given by:

$$\mathbf{M} = \mathbf{U}_{B_1} \times \mathbf{F} \times \mathbf{U}_{B_2}^T \tag{11}$$

where $\mathbf{U}_{B_1} \in \mathbb{R}^{Ord\cdot v_1 \times N_1}$ and $\mathbf{U}_{B_2} \in \mathbb{R}^{Ord\cdot v_2 \times N_2}$, and $\mathbf{F} \in \mathbb{R}^{N_1 \times N_2}$. The matrices $\mathbf{U}_{B_1}$ and $\mathbf{U}_{B_2}$ are established from $\mathbf{R}_{B_1} \in \mathbb{R}^{Ord \times B_1}$ and $\mathbf{R}_{B_2} \in \mathbb{R}^{Ord \times B_2}$. $B_1 \times B_2$ is the image block size and $v_1 \times v_2$ represents the number of blocks, where $v_1 = N_1/B1$ and $v_2 = N_2/B2$.

The proposed algorithm for HT detection is based on moments computed (features) using SKTP. However, smoothed image are considered better to extract features [26] because they show less sensitivity to OCM [52]. In addition, gradient images are used to extract features because they are invariant to illuminance change [53]. Thus, moments are extracted from the aforementioned versions of images, namely moments of smoothed images (MoSI) and moments of gradient images (MoGI). Instead of smoothing video frames (images) and computing the gradient of images prior to extract features (MoSI and MoGI), the orthogonal embedding image kernel (OEIK) algorithm [54] is utilized. The mathematical model used to compute MoSI using OEIK is given by [54]:

$$\mathbf{M}_S = \mathbf{U}_{YS} \times \mathbf{F} \times \mathbf{U}_{XS}^T \tag{12}$$

where $\mathbf{U}_{YS}$ and $\mathbf{U}_{XS}$ represent the OP embedded with smoothed image kernels in $y-$ and $x-$directions, respectively. The embedded OPs are computed as follows [54]:

$$\mathbf{U}_{YS} = R_1 \times H_{YS} \tag{13}$$

$$\mathbf{U}_{XS} = R_2 \times H_{XS} \tag{14}$$

where $H_{YS}$ and $H_{XS}$ are the Toeplitz matrices of the smoothed image kernels $h_{ys}$ and $h_{xs}$, respectively. To compute the MoGI in $x-$ and $y-$directions, the aforementioned model can be employed but the image kernels are replaced by the gradient image kernels. However, to reduce the computation cost for computing MoGIs, this can be performed by computing MoGIs from the MoSI [54]. The MoGIs can be computed from MoSI as follows [54]:

$$\mathbf{M}_{GX} = \mathbf{M}_S \times \mathbf{U}_{XSG} \tag{15}$$

$$\mathbf{M}_{GY} = \mathbf{U}_{YSG} \times \mathbf{M}_S \tag{16}$$

where $\mathbf{M}_{GX}$ and $\mathbf{M}_{GY}$ are the MoGIs in $x-$ and $y-$directions, respectively, and $\mathbf{U}_{XSG}$ and $\mathbf{U}_{YSG}$ are computed as follows:

$$\mathbf{U}_{XSG} = \mathbf{U}_{XS} \times \mathbf{U}_{XG}^T \tag{17}$$

$$\mathbf{U}_{YSG} = \mathbf{U}_{YG} \times \mathbf{U}_{YS}^T \tag{18}$$

In this paper, to reduce the computation of computing local features of MoSI and MOGIs, the OEIK algorithm and combined with FBP algorithm are used. The flow process of the presented algorithm is as follows:

I. For an input video ($\mathscr{V}$), the video information are acquired. These information include: the video frame size ($N_1$ and $N_2$) and number of frames ($N_f$).

II. The local moments are extracted for the frame active area of size $N_{A1} \times N_{A2}$ with a number of blocks equal to $v_1 \times v_2$, i.e. video frame block of size equal to $B_1 \times B_2 = N_{A1}/v_1 \times N_{A2}/v_2$. Gaussian smoothing operators $h_{xs}$ and $h_{ys}$ are defined as follows:

$$h_{xs} = \frac{1}{2\pi\sigma_x^2} e^{-\frac{(x-\mu_X)^2}{2\sigma_x^2}} \tag{19}$$

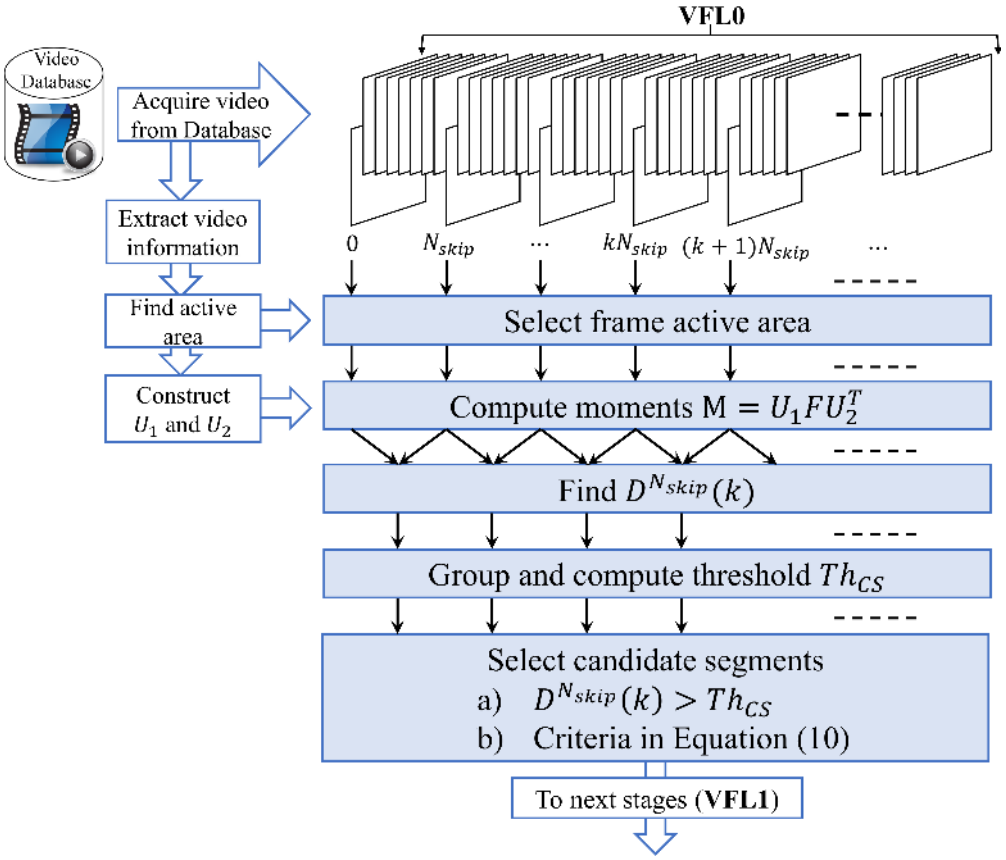


**FIGURE 5.** The flow diagram of candidate segment selection.

$$h_{ys} = \frac{1}{2\pi\sigma_y^2} e^{-\frac{(y-\mu_y)^2}{2\sigma_y^2}} \tag{20}$$

where $\sigma$ is the standard deviation and $\mu$ is the mean of the distribution. The used gradient operators are $h_{GX} = [-1, 1]$ and $h_{GY} = [-1, 1]^T$ and the moment orders are $O_n$ and $O_m$. Then, the SKTP polynomials for block processing $\mathbf{U}_{XS}$, $\mathbf{U}_{YS}$, $\mathbf{U}_{XSG}$, and $\mathbf{U}_{YSG}$ are implemented. The flow process is shown in FIGURE 6.

III. For each frame ($f$) in the candidate segments, i.e. VFL1, the local MoSI and MoGIs are computed using (12), (15), and (16).

### D. COMPUTATION OF DISSIMILARITY SIGNAL

For each moment set in the group of moments ($\mathbf{M}_S$, $\mathbf{M}_{GX}$, and $\mathbf{M}_{GY}$) in the VFL1, the dissimilarity signal between moments of consecutive frames ($f_k$ and $f_{k+1}$) is computed using city-block distance as follows:

$$\begin{aligned}\mathscr{FV} &= DS\big(\mathbf{M}(f_k), \mathbf{M}(f_{k+1})\big) \\ &= \sum_{i=1}^{O_n}\sum_{j=1}^{O_m} \|\mathbf{M}(f_k, i, j) - \mathbf{M}(f_{k+1}, i, j)\| \end{aligned} \tag{21}$$

where $\mathscr{FV}$ represents the DS. $\mathscr{FV}$ is considered a feature vector for the next step. For each moment group, a corresponding feature vector is computed using Equation (21), such that $\mathscr{FV}_S$, $\mathscr{FV}_{GX}$, and $\mathscr{FV}_{GY}$ are computed from $\mathbf{M}_S$, $\mathbf{M}_{GX}$, and $\mathbf{M}_{GY}$, respectively. The size of each feature vector is $1 \times N_{FCSi}$, where $N_{FCS_i}$ is the number of frames in the $i$th candidate segment. Note that, the total number of frames ($N_{TFCS}$) are the sum of all frames in each candidate segment ($N_{TFCS} = \sum_{i=1}^{N_{CS}} N_{FCS_i}$). where $N_{CS}$ is the total number of segments obtained from candidate segment selection stage.

The feature vectors $\mathscr{FV}_S$, $\mathscr{FV}_{GX}$, and $\mathscr{FV}_{GY}$ are concatenated to form a single feature vector $\mathscr{FV}_{SG}$ as follows:

$$\begin{aligned}\mathscr{FV}_{SG} &= \begin{bmatrix} \mathscr{FV}_S \\ \mathscr{FV}_{GX} \\ \mathscr{FV}_{GY} \end{bmatrix} \\ &= \begin{bmatrix} \mathscr{FV}_S(1) & \mathscr{FV}_S(1) & \cdots & \mathscr{FV}_S(N_{FCS_i}) \\ \mathscr{FV}_{GX}(1) & \mathscr{FV}_{GX}(2) & \cdots & \mathscr{FV}_{GX}(N_{FCS_i}) \\ \mathscr{FV}_{GY}(1) & \mathscr{FV}_{GY}(2) & \cdots & \mathscr{FV}_{GY}(N_{FCS_i}) \end{bmatrix} \end{aligned} \tag{22}$$

Obviously, the size of the feature vector $\mathscr{FV}_{SG}$ is $3 \times N_{FCS_i}$. Contextual information is considered a significant factor to detect transitions [26]. In contextual information, the features of the previous and next frames are considered to improve the accuracy of a TVS algorithm [2]. Therefore,

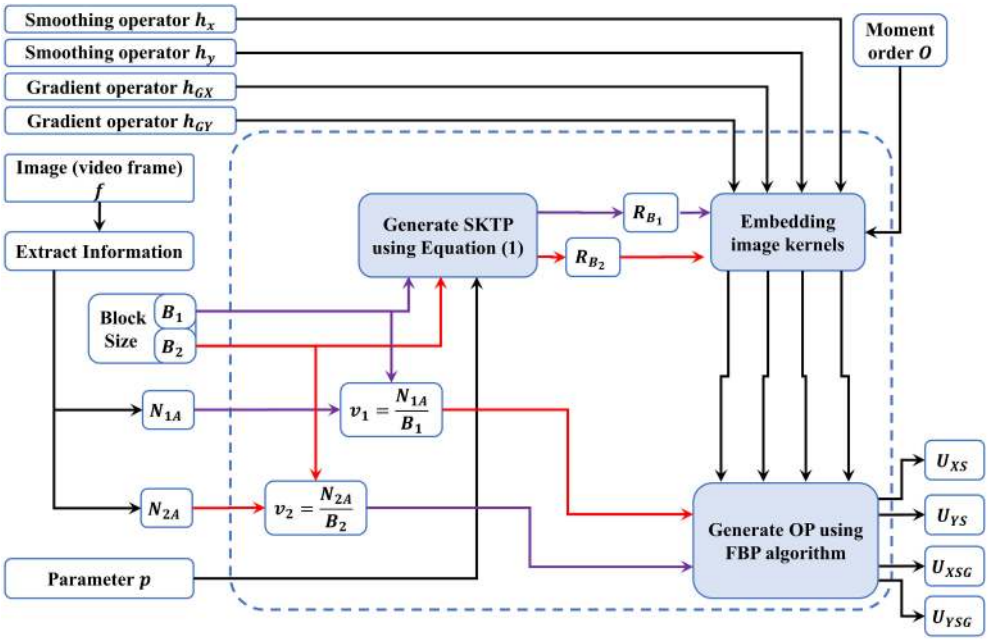


**FIGURE 6.** The flow process of the OPs generation which are employed for feature extraction.

the contextual information of previous ($N_{PRE}$) and next ($N_{POS}$) frames features are considered. The feature vector ($\mathscr{FV}_Z$) used for training and detection phases is given by the following mathematical expression:

$$\mathscr{FV}_Z = \begin{bmatrix} \mathscr{FV}_{SG}(k - N_{PRE}) \\ \mathscr{FV}_{SG}(k - N_{PRE} + 1) \\ \vdots \\ \mathscr{FV}_{SG}(k - 1) \\ \mathscr{FV}_{SG}(k-) \\ \mathscr{FV}_{SG}(k + 1) \\ \vdots \\ \mathscr{FV}_{SG}(k + N_{POS} - 1) \\ \mathscr{FV}_{SG}(k + N_{POS}) \end{bmatrix} \tag{23}$$

The size of the feature vector ($\mathscr{FV}_Z$) is $(N_{PRE} + N_{POS} + 1) \times N_{FCS}$.

### E. HTS DETECTION PROCESS

Support vector machine (SVM) is utilized to detect HTs, because of its ability in classification [55], [56]. The grid-search methods are utilized used to achieve the optimal SVM kernel parameters.

The normalization of the feature vector is a substantial process; thus, it should be utilized for both training and testing feature vectors [26], [57]. The utilization of feature vector normalization because features values lie within different ranges, hence the impact of large features values dominate small features values [26], [58]. To tackle this problem, the normalization of features is utilized so that features values lies within a similar ranges. This can be performed by transforming the $k$th feature $\mathscr{FV}_Z$ of mean $\mu_{\mathscr{FV}_Z}$ and standard deviation $\sigma_{\mathscr{FV}_Z}$ into the desired mean $\mu_{des}$ and standard deviation $\sigma_{des}$ as follows:

$$\mathscr{FV}_{ZN}(k) = \left(\mathscr{FV}_Z(k) - \mu_{\mathscr{FV}_Z}(k)\right)\left(\frac{\sigma_{des}}{\sigma_{\mathscr{FV}_Z}}\right) + \mu_{des} \tag{24}$$

To sum up, FIGURE 7 shows the process of the suggested HTs detection method based on SKTP.

## V. RESULTS AND DISCUSSION

This section provides systematic assessments to show the capability of the proposed TVS stages.

The dataset used in the evaluation of the proposed TVS algorithms is important and their examination and selection is required [26]. Standard datasets is suitable in the evaluation process; however, the datasets provided by researchers are limited by factors [26]. TRECVID evaluation [59] is co-sponsored by the National Institute of Standards and Technology (NIST). Thus, the most famous datasets of TRECVID (TRECVID 2001, TRECVID 2005, TRECVID 2006, and TRECVID 2007) are utilized to evaluate the performance of the proposed TVS algorithm.

TRECVID 2001 video dataset contains 6 videos. In these videos, there are 97,808 frames with a frame size of $240 \times 352$ and has 301 HTs. TRECVID 2005 includes 12 videos which comprises 744,593 frames with a size of $240 \times 352$. There are 2813 of HTs. TRECVID 2006 dataset contains 13 videos which has 597,042 frames with a size of $240 \times 352$. There are 1907 HTs in TRECVID 2006 dataset. Finally, TRECVID 2007 dataset includes 17 videos which have 637,738 frames with a size of $288 \times 352$. This dataset comprises 2253 of HTs. TABLE 1 summarize the TRECVID datasets.

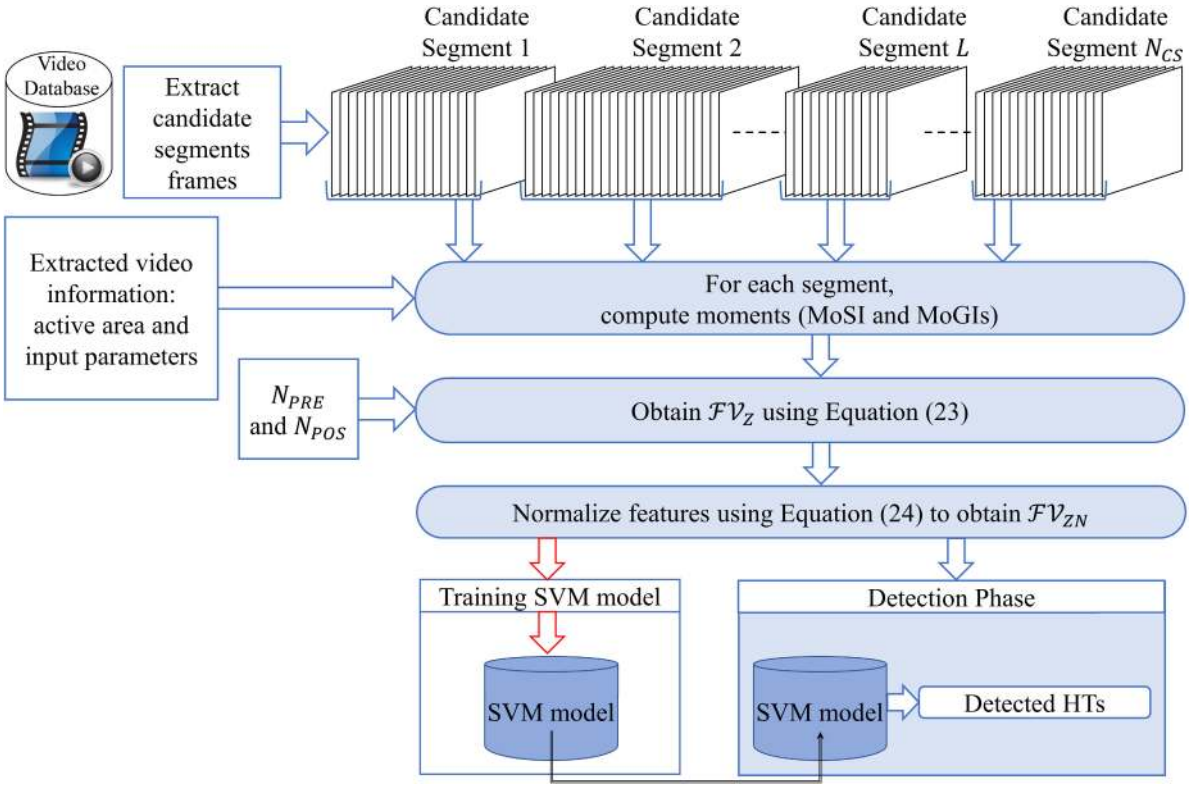


**FIGURE 7.** The proposed HTs detection method.

**TABLE 1.** TRECVID datasets details.

| TRECVID Dataset | Number of videos | Number of frames | Frames/sec | Number of HTs | Frame size |
|---|---|---|---|---|---|
| 2001 | 6 | 97,808 | 29.97 | 301 | $240 \times 352$ |
| 2005 | 12 | 744,604 | 29.97 | 2759 | $240 \times 352$ |
| 2006 | 13 | 597,043 | 29.97 | 1844 | $240 \times 352$ |
| 2007 | 17 | 637,805 | 25 | 2236 | $288 \times 352$ |
| Total | 48 | 2,077,260 | - | 7,140 | - |

Candidate segment selection is considered a preprocessing stage for transition detection in which the non-transition frames are eliminated to lessen the number of frames to be processed. To evaluate the performance of the proposed candidate segment section (CSS) technique based on SKTP, three preprocessing techniques used for comparison purpose which are improved pixel wise based candidate segment selection technique (IPW) [11], and modified pixel wise based candidate segment selection technique (MPW) [60]. TABLE 2 reports the results of the percentage of HTs (HT%) in the candidate segment to the total transitions for all videos in each dataset. In addition to that, the percentage value of the total frames in the candidate segments to the total frames for videos in the dataset (Frame%) is also reported. The block size $B_1 \times B_2$ is selected to be equal to the image size $N_{A1} \times N_{A2}$, i.e. the number of blocks $v_1 \times v_2$ is equal to $1 \times 1$. In addition, 5% of the moments coefficient ($O_n$, and $O_m$) are used. The algorithm is implemented on HP dv6 with core i7-2670QM CPU and 8GB of RAM.

The reported results of the two techniques show comparable computation time compared to the modified technique. For instance, the total time needed to process all the datasets is 118.92 sec for the modified technique, while for other techniques is ≈116 sec. However, the modified technique shows higher percentage of HTs and STs detection in candidate segments than the two techniques. For example, the total results of the modified technique show that the detection percentage are 99.92% for HTs in the candidate segment, while the best results of the other techniques is 96.16% for HTs in the candidate segments. In addition, the frames to be processes in the subsequent stages are less in the modified technique, which are 39.72% compared to ≈42.8% for the other techniques. To sum up, the results in TABLE 2 show that the modified technique based on adaptive threshold, nested criteria, and features based on SKTP affects positively on the candidate segment selection stage.

The candidate segments resulted from the previous stage (VFL1) contains HTs. The aim of this stage is to detect HTs

**TABLE 2. Comparison of candidate segment selection techniques.**

| Dataset | Results | Proposed CSS Technique | IPW [11] | MPW [60] |
|---|---|---|---|---|
| 2001 | HT% | 100 | 98.66 | 99.33 |
| | Frames% | 44.21 | 44.32 | 44.57 |
| | Time (sec) | 5.94 | 5.8 | 5.82 |
| 2005 | HT% | 99.96 | 97.76 | 98.97 |
| | Frames% | 36.72 | 40.56 | 39.49 |
| | Time (sec) | 41.11 | 40.1 | 39.77 |
| 2006 | HT% | 99.89 | 95.06 | 94.7 |
| | Frames% | 38.19 | 41.09 | 42.01 |
| | Time | (sec) | 32.3 | 32.7 |
| 2007 | HT% | 99.87 | 93.43 | 93.47 |
| | Frames% | 42.5 | 47.12 | 47.23 |
| | Time (sec) | 38.91 | 38.11 | 37.76 |
| Total | HT% | 99.92 | 95.75 | 96.16 |
| | Frames% | 39.27 | 42.9 | 42.83 |
| | Time (sec) | 118.92 | 116.31 | 116.05 |

**TABLE 3. Feature vectors parameters setting.**

| Parameters | Symbol | Value |
|---|---|---|
| Block size | $v_1 \times v_1$ | $8 \times 8$ |
| Image smoothing operator in x-direction | $S(x)$ | $\sigma_x = 1$ $length = 3$ |
| Image smoothing operator in y-direction | $S(y)$ | $\sigma_y = 1$ $length = 3$ |
| Image gradient operator in x-direction | $h_{GX}$ | $[-1, 1]$ |
| Image gradient operator in y-direction | $h_{GY}$ | $[-1, 1]^T$ |
| Moments order | $O_n$ | $\frac{5}{100} N_{A1}$ |
| | $O_m$ | $\frac{5}{100} N_{A2}$ |
| Contextual information | $N_{PRE}$ | $(1, 2, 3, 4)$ |
| | $N_{POS}$ | $(1, 2, 3, 4)$ |
| Feature vector normalization parameter | $\mu_{des}$ | 0 |
| | $\sigma_{des}$ | 1.6 |

in the candidate segments. The moments order is selected such that only 5% of the moments coefficient are extracted. The selection of 5% of moment coefficient is based on the fact that high energy moments using SKTP are adequate for representing video frames and at the same time providing less computation time [41]. The parameters settings for computing the feature vectors for this stage are listed in TABLE 3. However, the contextual information parameters $N_{PRE}$ and $N_{POS}$ have four different values to show their effect on recall ($\mathscr{R}$), precision ($\mathscr{P}$), F1-score ($F1$), and computation time. The recall, precision, and F1-score are computed as follows:

$$\mathscr{R} = \frac{N_{Correct}}{N_{Correct} + N_{MSD}} = \frac{N_{Correct}}{N_{Gorund}} \tag{25}$$

$$\mathscr{P} = \frac{N_{Correct}}{N_{Correct} + N_{FAM}} = \frac{N_{Correct}}{N_{TD}} \tag{26}$$

$$F1 = \frac{2\mathscr{R}\mathscr{P}}{\mathscr{R} + \mathscr{P}} = \frac{2N_{Correct}}{N_{TD} + N_{Gorund}} \tag{27}$$

The design of the HT detection stage comprises SVM model. To obtain SVM model, training dataset is required in the SVM training phase. The training dataset is selected from the video datasets in which 30% of the videos are used for training phase and 70% for evaluation (testing) phase. The selected videos for training phase are listed in TABLE 4. The remaining videos (test videos) are used for evaluation

**TABLE 4. The training video set.**

| | Dataset | | | |
|---|---|---|---|---|
| | 2001 | 2005 | 2006 | 2007 |
| Video ID | V01/2001 | V02/2005 | V03/2006 | V03/2007 |
| | V02/2001 | V03/2005 | V05/2006 | V08/2007 |
| | V03/2001 | V11/2005 | V06/2006 | V09/2007 |
| | — | — | V10/2006 | V15/2007 |
| Total Frames | 40,257 | 153,199 | 205,595 | 112,977 |
| Total HTs | 120 | 592 | 531 | 405 |

**TABLE 5. The details of the test videos for each dataset.**

| Dataset | Frames | Time (Sec) | HT |
|---|---|---|---|
| 2001 | 57,551 | 1920.29 | 181 |
| 2005 | 591,405 | 19733.23 | 2167 |
| 2006 | 391,448 | 13061.33 | 1313 |
| 2007 | 524,828 | 20993.12 | 1831 |
| Total | 1,565,232 | 55707.97 | 5492 |

**TABLE 6. Experimental results of HT detection for different $N_{PRE}$ and $N_{POS}$ values.**

| $N_{PRE}, N_{POS}$ | Dataset | Total Detected | Correctly Detected | $\mathscr{R}$ | $\mathscr{P}$ | $F1$ | Computation time in (Sec) |
|---|---|---|---|---|---|---|---|
| 1, 1 | 2001 | 204 | 177 | 86.76 | 97.79 | 91.95 | 42.4 |
| | 2005 | 2273 | 2135 | 93.93 | 98.52 | 96.17 | 391.1 |
| | 2006 | 1420 | 1272 | 89.58 | 96.88 | 93.08 | 267.5 |
| | 2007 | 1862 | 1799 | 96.62 | 98.25 | 97.43 | 433.4 |
| 2, 2 | 2001 | 187 | 177 | 94.65 | 97.79 | 96.20 | 44.5 |
| | 2005 | 2225 | 2148 | 96.54 | 99.12 | 97.81 | 394.2 |
| | 2006 | 1348 | 1285 | 95.33 | 97.87 | 96.58 | 272.1 |
| | 2007 | 1834 | 1814 | 98.91 | 99.07 | 98.99 | 440.5 |
| 3, 3 | 2001 | 189 | 177 | 93.65 | 97.79 | 95.68 | 45.6 |
| | 2005 | 2225 | 2146 | 96.45 | 99.03 | 97.72 | 406.5 |
| | 2006 | 1336 | 1284 | 96.11 | 97.79 | 96.94 | 281.4 |
| | 2007 | 1834 | 1812 | 98.80 | 98.96 | 98.88 | 451.8 |
| 4, 4 | 2001 | 187 | 177 | 94.65 | 97.79 | 96.20 | 46.4 |
| | 2005 | 2228 | 2146 | 96.32 | 99.03 | 97.66 | 413.5 |
| | 2006 | 1339 | 1284 | 95.89 | 97.79 | 96.83 | 285.8 |
| | 2007 | 1839 | 1815 | 98.69 | 99.13 | 98.91 | 458.2 |

phase and their description is shown in TABLE 5 for the four datasets.

The test videos are used as inputs to the obtained SVM models. The experimental results of the HTs detection for different values of $N_{PRE}$ and $N_{POS}$ are listed in TABLE 6 with the total processing time including moment extraction.

It can be concluded from TABLE 6 that when the contextual information is increased, the precision, recall, and F1-score measures are also increased. In turn the number of false positive decrease and the number of MSD decrease. For example, in TABLE 6 ($N_{PRE} = 1$ and $N_{POS} = 1$) for 2005 dataset results, the number of false positives are 138 and miss detected transition are 32. While for $N_{PRE}$, $N_{POS} = 2$ the number of false positives are 77 and miss detected transition are 19. For $N_{PRE}$, $N_{POS} = 3$ and $N_{PRE}$, $N_{POS} = 4$, comparable results are obtained when $N_{PRE}$, $N_{POS} = 2$.

To realize the optimum value of $N_{PRE}$ and $N_{POS}$ that provide maximum accuracy and less computation cost, FIGURE 8 is plotted. In this Figure, the total F1-score and computation time, excluding the moment extraction process time, are plotted against the parameters of contextual information ($N_{PRE}$

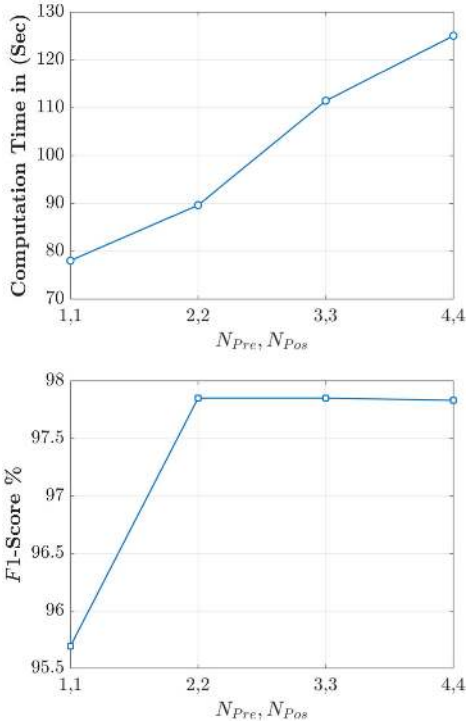


**FIGURE 8.** The effect of $N_{PRE}$ and $N_{POS}$ on the accuracy and computation time.

**TABLE 7.** Experimental results for different number of blocks.

| Dataset | $v_1 \times v_2 = 8 \times 8$ | | | $v_1 \times v_2 = 4 \times 4$ | | |
|---|---|---|---|---|---|---|
| | $\mathcal{R}$ | $\mathcal{P}$ | $F1$ | $\mathcal{R}$ | $\mathcal{P}$ | $F1$ |
| 2001 | 94.65 | 97.79 | 96.20 | 94.12 | 97.24 | 95.65 |
| 2005 | 96.54 | 99.12 | 97.81 | 95.84 | 98.85 | 97.32 |
| 2006 | 95.33 | 97.87 | 96.58 | 94.12 | 97.64 | 95.85 |
| 2007 | 98.91 | 99.07 | 98.99 | 98.04 | 98.36 | 98.20 |

and $N_{POS}$). From FIGURE 8, it is obvious that $N_{PRE} = 2$ and $N_{POS} = 2$ are a reasonable choice to give an acceptable tradeoff between accuracy and computation time, and are considered in the following sections.

On the other hand, the effect of number of blocks $v_1 \times v_2$ is also discussed. The parameter setting considered in this experiment is similar to that of TABLE 3 with two different values for number of blocks $v_1 \times v_2 = 8 \times 8$ and $4 \times 4$. That is, the block size for former is less than that of the later. The achieved results are reported in TABLE 7. It is clear that when the number of blocks is increased, the detection accuracy is increased and vice versa as shown in TABLE 7.

Note that, the computation time for both cases are equivalent because of implementing the FBP algorithm for computing moments. While in previous works such as [22], the computational cost has reduced by reducing the number of blocks; however, the accuracy is also decreased. Hence, in the proposed TVS algorithm, $N_{PRE} = 2$ and $N_{POS} = 2$, $v_1 \times v_2 = 8 \times 8$ will be considered.

To evaluate the performance of the proposed TVS algorithm, the proposed algorithm is compared to the state-of-the-art algorithms. The state-of-the-art algorithm are: TVS algorithm based on Non-Subsampled Contourlet Transform and SVM (NSCT) [23], TVS algorithm based on Walsh-Hadamard transform (WHT) [22], TVS algorithm based on concatenated block based histograms (CBBH) [61], and TVS algorithm based on orthogonal polynomial and FBP (OPFBP) [12]. The comparison is presented in TABLE 8 and TABLE 9 in terms of computation cost and accuracy. Tables 8 and 9 reveal that the proposed TVS algorithm in terms of F1-score has a comparable results to to NSCT and WHT algorithms, and demonstrates an improvement to OPFBP and CBBH algorithms. On the other hand, the proposed TVS algorithm exhibits an advancement in terms of the time required to process video frames (please see TABLE 8 and TABLE 9).

**TABLE 8.** Accuracy comparison and computation time using TRECVID2007 dataset.

| Algorithm | $\mathcal{R}\%$ | $\mathcal{P}\%$ | $F1\%$ | Processing time/frame (msec) |
|---|---|---|---|---|
| Proposed algorithm | 99.07 | 98.91 | 98.99 | 0.67 |
| OPFBP [12] | 96.58 | 96.91 | 96.74 | 1.5 |
| NSCT [23] | 97.66 | 96.36 | 97.01 | 142.48 |
| WHT [22] | 97.79 | 97.42 | 97.61 | 96.63 |

**TABLE 9.** Accuracy comparison and computation time using TRECVID2005 dataset.

| Algorithm | $\mathcal{R}\%$ | $\mathcal{P}\%$ | $F1\%$ | Processing time/frame (msec) |
|---|---|---|---|---|
| Proposed algorithm | 99.07 | 98.91 | 98.99 | 0.67 |
| OPFBP [12] | 97.54 | 95.76 | 96.64 | 1.35 |
| CBBH [61] | 95.00 | 96.99 | 95.50 | 40.77 |

## VI. CONCLUSION

In this paper, a new TVS algorithm is proposed. This work has different stages, where each stage has a significant impact on the performance of the proposed TVS algorithm. MoSI and MoGIs are used to reduce the effect of disturbance factors such as noise, object motion, camera motion, and flash lights. The proposed frame active area effecting in improving the performance of accuracy as well as the computational cost of the TVS algorithm. In addition, the modified CSS algorithm remarkably reducing the computational time through selecting video segments which contain transitions. The combination of FBP and OEIK algorithm highly reduced the computational cost of the TVS algorithm. The results shows that the proposed algorithm outperforms the state-of-the-art algorithms. Our future work is going towards the detection of soft transitions which in turn guarantee that the proposed algorithm can be utilized for detecting different types of transitions.

## REFERENCES

[1] M. Birinci and S. Kiranyaz, "A perceptual scheme for fully automatic video shot boundary detection," *Signal Process., Image Commun.*, vol. 29, no. 3, pp. 410–423, Mar. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596513001884

[2] C. Liu, D. Wang, J. Zhu, and B. Zhang, "Learning a contextual multithread model for movie/TV scene segmentation," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 884–897, Jun. 2013.

[3] (2019). *Alexa*. [Online]. Available: http://www.alexa.com/topsites

[4] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 6, pp. 797–819, Nov. 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5729374

[5] R. Priya and T. N. Shanmugam, "A comprehensive review of significant researches on content based indexing and retrieval of visual information," *Frontiers Comput. Sci.*, vol. 7, no. 5, pp. 782–799, Oct. 2013, doi: 10.1007/s11704-013-1276-6.

[6] K. Choroś, "Improved video scene detection using player detection methods in temporally aggregated tv sports news," in *Proc. Int. Conf. Comput. Collective Intell.*, D. Hwang, J. J. Jung, and N.-T. Nguyen, Eds. Cham, Switzerland: Springer, 2014, pp. 633–643, doi: 10.1007/978-3-319-11289-3_64.

[7] H. Bhaumik, S. Bhattacharyya, M. D. Nath, and S. Chakraborty, "Hybrid soft computing approaches to content based video retrieval: A brief review," *Appl. Soft Comput.*, vol. 46, pp. 1008–1029, Sep. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494616301314

[8] G. C. Chaves, "Video content analysis by active learning," Ph.D. dissertation, Dept. Comput. Sci., Federal Univ. Minas Gerais, Belo Horizonte, Brazil, 2007.

[9] N. J. Janwe and K. K. Bhoyar, "Video shot boundary detection based on JND color histogram," in *Proc. IEEE 2nd Int. Conf. Image Inf. Process. (ICIIP)*, Dec. 2013, pp. 476–480.

[10] U. Gargi, R. Kasturi, and S. H. Strayer, "Performance characterization of video-shot-change detection methods," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 10, pp. 1–13, Feb. 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=825852

[11] Z.-M. Lu and Y. Shi, "Fast video shot boundary detection based on SVD and pattern matching," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5136–5145, Dec. 2013.

[12] S. H. Abdulhussain, A. R. Ramli, B. M. Mahmmod, M. I. Saripan, S. A. R. Al-Haddad, T. Baker, W. N. Flayyih, and W. A. Jassim, "A fast feature extraction algorithm for image and video processing," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8851750/

[13] K. Choroś, "False and miss detections in temporal segmentation of TV sports news videos–causes and remedies," in *New Research in Multimedia and Internet Systems*, A. Zgrzywa, K. Choroś, and A. Siemiński, Eds. Cham, Switzerland: Springer, 2015, pp. 35–46, doi: 10.1007/978-3-319-10383-9_4.

[14] L. H. Iwan and J. A. Thom, "Temporal video segmentation: Detecting the end-of-act in circus performance videos," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 1379–1401, Jan. 2017, doi: 10.1007/s11042-015-3130-3.

[15] G. Gao and H. Ma, "To accelerate shot boundary detection by reducing detection region and scope," *Multimedia Tools Appl.*, vol. 71, no. 3, pp. 1749–1770, Aug. 2014.

[16] M. Tavassolipour, M. Karimian, and S. Kasaei, "Event detection and summarization in soccer videos using Bayesian network and copula," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 291–304, Feb. 2014.

[17] S. Tippaya, S. Sitjongsataporn, T. Tan, M. M. Khan, and K. Chamnongthai, "Multi-modal visual features-based video shot boundary detection," *IEEE Access*, vol. 5, pp. 12563–12575, 2017.

[18] A. Amiri and M. Fathy, "Video shot boundary detection using QR-decomposition and Gaussian transition detection," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, pp. 1–12, Dec. 2010.

[19] C.-W. Ngo, T.-C. Pong, and R. T. Chin, "Video partitioning by temporal slice coherency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 8, pp. 941–953, Aug. 2001.

[20] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying production effects," *Multimedia Syst.*, vol. 7, no. 2, pp. 119–128, Mar. 1999.

[21] M. Tkalcic and J. F. Tasic, "Colour spaces: Perceptual, historical and applicational background," in *Proc. IEEE Region 8 EUROCON-Comput. Tool*, vol. 1, Sep. 2003, pp. 304–308.

[22] L. G. G. Priya and S. Domnic, "Walsh–Hadamard transform kernel-based feature vector for shot boundary detection," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5187–5197, Dec. 2014.

[23] J. Mondal, M. K. Kundu, S. Das, and M. Chowdhury, "Video shot boundary detection using multiscale geometric analysis of NSCT and least squares support vector machine," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 8139–8161, Apr. 2018, doi: 10.1007/s11042-017-4707-9.

[24] S. Abdulhussain, A. Ramli, M. Saripan, B. Mahmmod, S. Al-Haddad, and W. Jassim, "Methods and challenges in shot boundary detection: A review," *Entropy*, vol. 20, no. 4, p. 214, 2018. [Online]. Available: http://www.mdpi.com/275270 http://www.mdpi.com/1099-4300/20/4/214

[25] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, "A formal study of shot boundary detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 2, pp. 168–186, Feb. 2007.

[26] S. H. Abdulhussain, A. R. Ramli, B. M. Mahmmod, M. I. Saripan, S. A. R. Al-Haddad, and W. A. Jassim, "Shot boundary detection based on orthogonal polynomial," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 20361–20382, Jul. 2019, doi: 10.1007/s11042-019-7364-3.

[27] R. Tapu and T. Zaharia, "Video segmentation and structuring for indexing applications," *Int. J. Multimedia Data Eng. Manage.*, vol. 2, no. 4, pp. 38–58, Oct. 2011, doi: 10.4018/jmdem.2011100103.

[28] G. Ciocca and R. Schettini, "Dynamic storyboards for video content summarization," in *Proc. 8th ACM Int. Workshop Multimedia Inf. Retr. (MIR)*, 2006, p. 259. [Online]. Available: http://dl.acm.org/citation.cfm?id=1178677.1178713

[29] X. Jiang, T. Sun, J. Liu, J. Chao, and W. Zhang, "An adaptive video shot segmentation scheme based on dual-detection model," *Neurocomputing*, vol. 116, pp. 102–111, Sep. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231212007096

[30] A. Hanjalic, "Shot-boundary detection: Unraveled and resolved?" *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 2, pp. 90–105, Feb. 2002.

[31] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *Proc. 3rd ACM Int. Conf. Multimedia (MULTIMEDIA)*, vol. 95, 1995, pp. 189–200. [Online]. Available: http://portal.acm.org/citation.cfm?doid=217279.215266

[32] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2006.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[34] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–151.

[35] M.-H. Park, R.-H. Park, and S. W. Lee, "Shot boundary detection using scale invariant feature matching," *Proc. SPIE*, vol. 6077, Jan. 2006, Art. no. 0771N, doi: 10.1117/12.642244.

[36] B. M. Mahmmod, A. R. B. Ramli, S. H. Abdulhussain, S. A. R. Al-Haddad, and W. A. Jassim, "Signal compression and enhancement using a new orthogonal-polynomial-based discrete transform," *IET Signal Process.*, vol. 12, no. 1, pp. 129–142, Feb. 2018. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-spr.2016.0449

[37] S. V. Porter, M. Mirmehdi, and B. T. Thomas, "Video cut detection using frequency domain correlation," in *Proc. 15th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, Sep. 2000, pp. 409–412.

[38] G. G. L. Priya and S. Domnic, "Edge strength extraction using orthogonal vectors for shot boundary detection," *Procedia Technol.*, vol. 6, pp. 247–254, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212017312005750, doi: 10.1016/j.protcy.2012.10.030.

[39] H. S. Radeaf, B. M. Mahmmod, S. H. Abdulhussain, and D. Al-Jumaeily, "A steganography based on orthogonal moments," in *Proc. Int. Conf. Inf. Commun. Technol. (ICICT)*. New York, NY, USA: ACM, 2019, pp. 147–153, doi: 10.1145/3321289.3321324.

[40] B. M. Mahmmod, A. R. Ramli, T. Baker, F. Al-Obeidat, S. H. Abdulhussain, and W. A. Jassim, "Speech enhancement algorithm based on super-Gaussian modeling and orthogonal polynomials," *IEEE Access*, vol. 7, pp. 103485–103504, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8766088/

[41] S. H. Abdulhussain, A. R. Ramli, B. M. Mahmmod, M. I. Saripan, S. A. R. Al-Haddad, and W. A. Jassim, "A new hybrid form of Krawtchouk and Tchebichef polynomials: Design and application," *J. Math. Imag. Vis.*, vol. 61, no. 4, pp. 555–570, May 2019, doi:10.1007/s10851-018-0863-4.

[42] S. H. Abdulhussain, A. R. Ramli, S. A. R. Al-Haddad, B. M. Mahmmod, and W. A. Jassim, "Fast recursive computation of Krawtchouk polynomials," *J. Math. Imag. Vis.*, vol. 60, no. 3, pp. 285–303, Mar. 2018, doi: 10.1007/s10851-017-0758-9.

[43] S. H. Abdulhussain, A. R. Ramli, S. A. R. Al-Haddad, B. M. Mahmmod, and W. A. Jassim, "On computational aspects of Tchebichef polynomials for higher polynomial order," *IEEE Access*, vol. 5, pp. 2470–2478, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7856982/

[44] Z. N. Idan, S. H. Abdulhussain, and S. A. R. Al-Haddad, "A new separable moments based on Tchebichef-Krawtchouk polynomials," *IEEE Access*, vol. 8, pp. 41013–41025, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9018036/

[45] W. A. Jassim, R. Mukundan, and P. Raveendran, "New orthogonal polynomials for speech signal and image processing," *IET Signal Process.*, vol. 6, no. 8, pp. 713–723, Oct. 2012. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-spr.2011.0004

[46] K.-H. Thung, R. Paramesran, and C.-L. Lim, "Content-based image quality metric using similarity measure of moment vectors," *Pattern Recognit.*, vol. 45, no. 6, pp. 2193–2204, Jun. 2012. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0031320311004742

[47] W. A. Jassim, M. S. A. Zilany, and R. Paramesran, "Enhancing noisy speech signals using orthogonal moments," *IET Signal Process.*, vol. 8, no. 8, pp. 891–905, Oct. 2014.

[48] B. S. Journal, "Orthogonal functions solving linear functional differential equationsusing Chebyshev polynomial," *Baghdad Sci. J.*, vol. 5, no. 1, pp. 143–148, 2008.

[49] Y.-N. Li, Z.-M. Lu, and X.-M. Niu, "Fast video shot boundary detection framework employing pre-processing techniques," *IET Image Process.*, vol. 3, no. 3, pp. 121–134, Jun. 2009.

[50] C. Grana and R. Cucchiara, "Linear transition detection as a unified shot detection approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 483–489, Apr. 2007.

[51] H. Liu, H. Lu, and X. Xue, "A segmentation and graph-based video sequence matching method for video copy detection," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1706–1718, Aug. 2013.

[52] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, Jan. 1993.

[53] W. Zhou, L. Yu, W. Qiu, Y. Zhou, and M. Wu, "Local gradient patterns (LGP): An effective local-statistical-feature extraction scheme for no-reference image quality assessment," *Inf. Sci.*, vols. 397–398, pp. 1–14, Aug. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025517305479

[54] S. H. Abdulhussain, A. R. Ramli, A. J. Hussain, B. M. Mahmmod, and W. A. Jassim, "Orthogonal polynomial embedded image kernel," in *Proc. Int. Conf. Inf. Commun. Technol. (ICICT)*. New York, NY, USA: ACM, 2019, pp. 215–221, doi: 10.1145/3321289.3321310.

[55] J. Xu, Y. Yan Tang, B. Zou, Z. Xu, L. Li, and Y. Lu, "The generalization ability of online SVM classification based on Markov sampling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 628–639, Mar. 2015.

[56] W. S. Noble, "What is a support vector machine?" *Nature Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, Dec. 2006, doi: 10.1038/nbt1206-1565.

[57] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," *BJU Int.*, vol. 101, no. 1, pp. 400–1396, 2003. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[58] K. Koutroumbas and S. Theodoridis, *Pattern Recognition*, 4th ed. New York, NY, USA: Academic, 2008.

[59] (2018). *TRECVID*. [Online]. Available: http://trecvid.nist.gov/

[60] J. Xu, L. Song, and R. Xie, "Shot boundary detection using convolutional neural networks," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2016, pp. 1–4.

[61] B. Youssef, E. Fedwa, A. Driss, and S. Ahmed, "Shot boundary detection via adaptive low rank and SVD-updating," *Comput. Vis. Image Understand.*, vol. 161, pp. 20–28, Aug. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314217301157

**SADIQ H. ABDULHUSSAIN** (Member, IEEE) 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. He is currently pursuing the Ph.D. degree with the Department of Computer and Communication Systems Engineering, Universiti Putra Malaysia. Since 2005, he has been a staff member with the Computer Engineering Department, Faculty of Engineering, University of Baghdad. His research interests include computer vision, signal processing, as well as speech and image processing.

**SYED ABDUL RAHMAN AL-HADDAD** (Senior Member, IEEE) received the Ph.D. degree in electrical, electronic, and systems engineering from National University Malaysia. He has been serving the Department of Computer and Communications Systems Engineering, Universiti Putra Malaysia, since 1997, and was promoted to an Associate Professor, in 2012. Furthermore, he has been teaching students of undergraduate and graduate levels in Malaysia and International for over 19 years. On research, he has published more than hundreds of journals and proceedings. In research, he has more than 20 International and national grants, and holds six patents and copyrights. On the other hand, he is the Head of the Laboratory Information Engineering and Robotics. He is specialized in human speech processing, animal sound processing, Al-Quran sound processing, sound media security, and biometrics. He actively joined professional societies such as the Deputy Chair of the IEEE Systems, Man and Cybernetics, MITS, MSET, and others.

**M. IQBAL SARIPAN** (Member, IEEE) received the B.Eng. degree in electronic engineering from Universiti Teknologi Malaysia, in 2001, and the Ph.D. degree from the University of Surrey, Guildford, U.K., in 2006. He is currently a Lecturer and the Head of Embedded and Artificial Intelligent Systems Engineering Research Group, Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia. His research interests are in the area of computer security, signal processing, image processing, and embedded systems. He is a member of IoP UK.

**BASHEERA M. MAHMMOD** was born in Baghdad, Iraq, in 1975. She received the B.Sc. degree in electrical engineering and the master's degree in electronics and communication engineering from Baghdad University, in 1998 and 2012, respectively, and the Ph.D. degree in computer and embedded system engineering from Universiti Putra Malaysia, in 2018. Since 2007, she has been a Staff Member with the Department of Computer Engineering, Faculty of Engineering, University of Baghdad. Her research interests include speech enhancement, signal processing, computer vision, RFID, and cryptography.

**ASEEL HUSSIEN** received the B.Sc. degree in architecture engineering and the M.Sc. degree in computing information systems, and the Ph.D. degree from Liverpool John Moores University, titled ARGILE: A Conceptual Framework Combining Augmented Reality with Agile Philosophy for the UK Construction Industry. She is the Program Leader in building surveying and facilities management with Liverpool John Moores University. Her research interests include the virtual reality, smart city, augmented reality, agile project management, and building information model BIM.