IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 10, 2021, accepted July 21, 2021, date of publication July 26, 2021, date of current version August 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100139

Fast Shot Boundary Detection Based on Separable Moments and Support Vector Machine

ZINAH N. IDAN[1], SADIQ H. ABDULHUSSAIN[1], BASHEERA M. MAHMMOD[1], KHALED A. AL-UTAIBI[2], (Member, IEEE), SYED ABDUL RAHMAN AL-HADAD[3], (Senior Member, IEEE), AND SADIQ M. SAIT[1,4]

[1]Department of Computer Engineering, University of Baghdad, Al-Jadriya, Baghdad 10071, Iraq
[2]Department of Computer Engineering, University of Ha'il, Ha'il 81451, Saudi Arabia
[3]Department of Computer and Communication Systems Engineering, Universiti Putra Malaysia, Serdang 43400, Malaysia
[4]Center for Communications and IT Research, Department of Computer Engineering, Research Institute, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Khaled A. Al-Utaibi (alutaibi@uoh.edu.sa)

ABSTRACT The large number of visual applications in multimedia sharing websites and social networks contribute to the increasing amounts of multimedia data in cyberspace. Video data is a rich source of information and considered the most demanding in terms of storage space. With the huge development of digital video production, video management becomes a challenging task. Video content analysis (VCA) aims to provide big data solutions by automating the video management. To this end, shot boundary detection (SBD) is considered an essential step in VCA. It aims to partition the video sequence into shots by detecting shot transitions. High computational cost in transition detection is considered a bottleneck for real-time applications. Thus, in this paper, a balance between detection accuracy and speed for SBD is addressed by presenting a new method for fast video processing. The proposed SBD framework is based on the concept of candidate segment selection with frame active area and separable moments. First, for each frame, the active area is selected such that only the informative content is considered. This leads to a reduction in the computational cost and disturbance factors. Second, for each active area, the moments are computed using orthogonal polynomials. Then, an adaptive threshold and inequality criteria are used to eliminate most of the non-transition frames and preserve candidate segments. For further elimination, two rounds of bisection comparisons are applied. As a result, the computational cost is reduced in the subsequent stages. Finally, machine learning statistics based on the support vector machine is implemented to detect the cut transitions. The enhancement of the proposed fast video processing method over existing methods in terms of computational complexity and accuracy is verified. The average improvements in terms of frame percentage and transition accuracy percentage are 1.63% and 2.05%, respectively. Moreover, for the proposed SBD algorithm, a comparative study is performed with state-of-the-art algorithms. The comparison results confirm the superiority of the proposed algorithm in computation time with improvement of over 38%.

INDEX TERMS Multimedia databases, image processing, mathematics, orthogonal polynomial, orthogonal moments, shot boundary detection, multimedia, cut transitions.

I. INTRODUCTION

With video multimedia data growth, internet traffic is rising from a moderately consistent stream to a dynamic traffic pattern [1], [2]. Searching the entire video for specific content is time-consuming. Moreover, video production and creation remain challenging [3]. Therefore, in the last decade, efficient management systems for indexing, browsing, and retrieval of videos have been considered. Content-based video indexing and retrieval aim to automate video structure analysis. Consequently, shot boundary detection (SBD) is a pre-processing step in video analysis [4].

Video is a three dimensional (3D) signal that involves a combination of text, audio, and images with a time

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

dimension. A video sequence generally comprises frames, shots, scenes, and stories. Video shots, or simply shots, are the basic units in a video and are considered a useful tool in providing semantic search [3]. Each shot consists of a sequence of frames, and each video frame represents a single image [5]. In a video sequence, shot boundaries are divided into a cut transition (CTR) and a gradual transition (GTR). CTR is the sudden transition from one video shot to another, and GTR represents the gradual transition between two shots and involves multiple frames. GTR are categorized into fade, dissolve, and wipe transitions [6]. Transition detection in SBD algorithms can be categorized based on the classification technique used, which are: 1) machine learning technique, 2) rules-based technique, and 3) combination of machine learning and rule-based techniques [7]. The SBD algorithm based on machine learning can be divided into supervised and unsupervised.

Generally, the essential process in the SBD algorithms is the feature extraction process. In this process, the significant representation of the visual information is considered the main goal [8]. Feature extraction can be classified based on the domain where the frame is processed. The compressed and uncompressed domains are the existing processing domains [9]. Due to the valuable information in the uncompressed domain, the SBD algorithms are primarily centered on it [7]. Different algorithms are used in the uncompressed domain such as: pixel-based algorithms [10], transform-based algorithms [11], histogram-based algorithms [12], and edge-based algorithms [13].

Transform-based algorithm have been employed by many researchers where the transform coefficients are extracted and considered as features. Discrete Walsh-Hadamard transform, discrete Wavelet transform, and discrete Fourier transform are examples of transforms used in SBD algorithms which achieved acceptable performance in detecting transitions [14]. However, these algorithm are computational expensive [15] because of processing the frames in a video sequence.

Processing all video frames will result in high detection accuracy. However, considerable computation time is consumed in processing non-boundary frames. To reduce the computation cost by eliminating the non-boundary frames, an additional level in video processing operation is applied. This level is considered a pre-processing stage in reducing computation in subsequent stages.

Our contribution can be summarized as follows:

- Develop an efficient candidate segment selection algorithm to eliminate non-transition frames and preserves candidate segments. The candidate segment selection algorithm applies bisection technique. In addition, the criteria used for selecting the candidate segment is developed and shows improvement in the performance compared to existing algorithms, which reduces the time required to process the entire video for shot boundary detection.
- Present a new active area selection technique. The presented technique utilizes 2/3 and 7/8 of the frame's height and width, respectively. The active area ratio is selected after performing several experiments. These experiments reveals that the selected active area shows better performance in terms of processing time for the shot boundary detection.
- In the design of the proposed SBD algorithm, the squared Tchebichef-Krawtchouk polynomials (STKP) is employed for the first time in SBD algorithm. The employed polynomials show better performance in terms of localization property and energy compaction when compared to existing algorithms. Also, the employed polynomials have been proved in other computer vision algorithms.
- Finally, the design of the proposed SBD is designed based on the previously mentioned tools. In addition, the proposed algorithm shows a noticeable improvement in terms of computation speed while preserving the accuracy.

This paper is organized as follows. Section II describes related work. Section 3 presents the mathematical fundamentals of STKP and moment computations. Section 4 introduces the proposed fast video processing method. Section 5 provides a performance evaluation of the proposed method. In Section 6 we present some conclusions.

II. RELATED WORK

Reducing computation cost without sacrificing accuracy is a challenging task in big multimedia data analytics [16]. In the past two decades, several algorithms have been proposed to develop detection accuracy with fast video processing. This development is considered valuable support for a high-level application of video structure analysis [17]. Most existing algorithms consist of three levels to detect transitions [6]. Feature extraction is the primary level to represent rich content (visual information) of the video frames [18]. Different approaches have been proposed for visual information representation (feature extraction). Pixel-based [19], histogram-based [20], and transform-based [6] are examples of different features used in these approaches. The transform domain (moment domain) offers a powerful ability to analyze a signal's components and suppress noise [21]. The next step in SBD is constructing similarity and dissimilarity signals by extracting temporal characteristics. These characteristics will specify the variations between consecutive frames or between frames with inter-frame distances [6]. At the shot transition, the similarity signal (SS) has low values, whereas the dissimilarity signal (DS) has high values. The opposite is applied within the same shot [6]. The last level in identification is to classify the SS/DS values into cut/gradual transitions and non-transitions [6].

All video frames must be processed using SBD algorithms. For further computational cost reduction, Li *et al.*in [22] proposed a pre-processing method, in which only the candidate segments are selected before the SBD main levels. Every 21 frames are formed as a segment and every 10 segments are grouped. The adaptive threshold with bisection-based comparisons was used to eliminate the non-boundary transitions. However, this method showed sensitivity to object and camera motion when the pixel-based approach is used. Moreover, it only approximated the location of the transitions [23].

In [23], an improved adaptive threshold was used. A histogram-based approach and singular value decomposition were applied to determine the actual transitions. In addition, the filtering process is implemented to eliminate false alarms. However, because of the histogram method, the detection was not accurate due to the possibility of mis-detection. In [24] the same adaptive threshold as in [23] was implemented to improve the speed of the SBD algorithm. However, its accuracy needs to be improved [15].

Tippaya *et al.* in [15], [25] selected the candidate segments by extracting the temporal characteristics using speeded-up robust features (SURF) descriptors and histograms. Despite the high accuracy from the multi-modal features, the execution speed was very low.

Dhiman *et al.* in [26] proposed a method similar to the one in [23]. In addition, only the blue plane is used for feature extraction to reduce the computations. However, here, an SBD algorithm needs to be devised to achieve detection accuracy with low computation cost. In addition, there are SBD algorithms employ deep neural networks. Xu *et al.* [24] presented a SBD algorithm to detect transitions based on features extracted from convolutional neural network (CNN); however, the detection accuracy still low. Gygli [27] and Hassanien *et al.* [28] proposed an SBD algorithm using CNN with fully connected architecture. Hassanien *et al.* predicted a likelihood of transition within a sequence of 16 frames using the convolutional 3D network [29]. Compared to Hassanien *et al.*, Gygli's algorithm has a smaller network and without post-processing. On the other hand, in Hassanien *et al.*, a support vector machine (SVM) classifier is utilized (the predictions are not used directly).The performance of Hassanien *et al.*algorithm outperformed Gygli's algorithm and the reported F1-score was 92.5%.. In addition, the algorithms based on deep neural networks are significantly are running on expensive GPU and require a very large training dataset.

In the algorithms mentioned above, the entire video frame is processed for each candidate segment. In [7], Abdulhussain *et al.* proposed an SBD algorithm was based only on the selection of frame active area. By eliminating the persistent and variable materials, only the valuable material with rich content is preserved by considering 7/8 of the frame height and 3/4 of frame width. Furthermore, the candidate segment selection is applied based on a threshold-based approach and the inequality criteria. The SBD features are extracted based on the moment computation. The local moment is computed by applying the moment block processing proposed in [8]. Then, a group of three local moments is computed by applying embedding operators. However, further elimination for the non-boundary frames can be applied by applying additional comparison criteria. Therefore, additional effort must be made to provide a compromise between the detection accuracy and the computation cost.

III. PRELIMINARIES

Orthogonal polynomials (OPs) are considered efficient tools in several applications such as information hiding [30]–[32], face recognition [21], SBD [7], speech enhancement [33], [34], and handwritten numerical recognition [35]. The moments are the projection of signals on OPs [21], [36], [37]. In this section, the mathematical definitions of STKP and associated separable moments are presented. These moments are used to transform the video frames into the moment domain and form the features. In this paper, STKP is utilized as an OP since it has been proven that its performance outperforms other OPs in terms of energy compaction and localization property over other existing OPs [38]. In addition, the extracted features affect the selection of the candidate transition, as shown in Section IV-B.

A. STKP DEFINITION

A new orthogonal polynomial is generated by multiplying two orthogonal polynomials [39]. From this perspective, STKP is generated by multiplying two hybrid forms of polynomials, namely, Krawtchouk–Tchebichef orthogonal polynomial (KTOP) [40] and Tchebichef–Krawtchouk orthogonal polynomial (TKOP) [39]. These hybrid forms are generated from Krawtchouk orthogonal polynomial (KOP) [41] and Tchebichef orthogonal polynomial (TOP) [42]. KTOP is defined as follows [40]:

$$X_n(x; p, N) = \sum_{h=0}^{N-1} T_h(n)\, K(x; p) \qquad n, x = 0, 1, \cdots, N-1 \qquad (1)$$

where T and K are the TOP and KOP, respectively. p is the control parameter of KOP. TKOP can be expressed as follows [39]:

$$Y_n(x; p, N) = \sum_{h=0}^{N-1} T_h(x)\, K(n; p) \qquad n, x = 0, 1, \cdots, N-1 \qquad (2)$$

The STKP form can be defined by combining (1) and (2) as follows [38]:

$$Z_n(x; p, N) = \sum_{a=0}^{N-1}\sum_{b=0}^{N-1}\sum_{c=0}^{N-1} T_b(a)K_b(x; p)T_c(n)K_c(a; p) \qquad n, x = 0, 1, \cdots, N-1 \qquad (3)$$

B. MOMENT COMPUTATION

The moments are considered a beneficial tool to characterize data by preventing data redundancy. Subsequently, 1D and 2D signals are described by the moments. For a two dimensional (2D) signal $I(x, y)$ with a size of $N \times N$, the moment can be defined as [38]:

$$\eta_{nm} = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} Z_n(x; p, N_1) Z_m(y; p, N_2) I(x, y)$$
$$n = 0, 1, \cdots, \quad N_1 - 1$$
$$m = 0, 1, \cdots, \quad N_2 - 1 \qquad (4)$$

where $Z_n(x)$, and $Z_m(y)$ are the orthogonal polynomials.

The reconstruction of the 2D signal can be computed as follows:

$$\hat{I}(x, y) = \sum_{n=0}^{N_1-1} \sum_{m=0}^{N_2-1} \eta_{nm} Z_n(x; p, N_1) Z_m(y; p, N_2)$$
$$x = 0, 1, \cdots, \quad N_1 - 1$$
$$y = 0, 1, \cdots, \quad N_2 - 1 \qquad (5)$$

The matrix representation is used to provide a second faster and simplified form of the moment in Equation (4) as follows [38]:

$$\eta = Z_1 \, I \, Z_2^T \qquad (6)$$

where Z_1 and Z_2 represent the matrix form of $Z_n(x; p, N_1)$ and $Z_m(y; p, N_2)$, respectively. The transform coefficients (moments) are considered the features that describe different types of signals [43]. To select the low-order moment with high energy coefficients, the specific order (ord) is specified. The coefficients row-wise for each generated polynomial (Z_1 and Z_2) are selected such that the selected coefficients are equal to the required order [38].

IV. PROPOSED FAST VIDEO PROCESSING METHOD

The framework of proposed algorithm includes three steps which are: the feature extraction, constructing the DS, and the identification of CTR. A preliminary processing is performed to decode the video frames. Then, only the active area of the frames is considered to preserve the valuable visual information and provide a reduction in computation time.

In the proposed algorithm, STKP is used to extract features. Two types of features are employed in the proposed algorithm which are: global and local features. The global features are used to specify the candidate transition locations in the process of candidate segments selection. The local features (moments) are considered for visual representation because they are robust against disturbance factors. Therefore, an OP block processing [8] module is employed to reduce the effect of video sequence disturbance as well as the computational cost.

Thereafter, DS is constructed to form a feature vector which dissimilarity of the contextual (temporal) information to improve the detection accuracy. The resulted feature vector is used in the next stage to identify the CTR.

Finally, the CTRs are detected in the identification stage, which uses a binary SVM model to classify the video sequence into transition (CTR) and nontransition frames. The general framework of the proposed algorithm is shown in Figure 1.

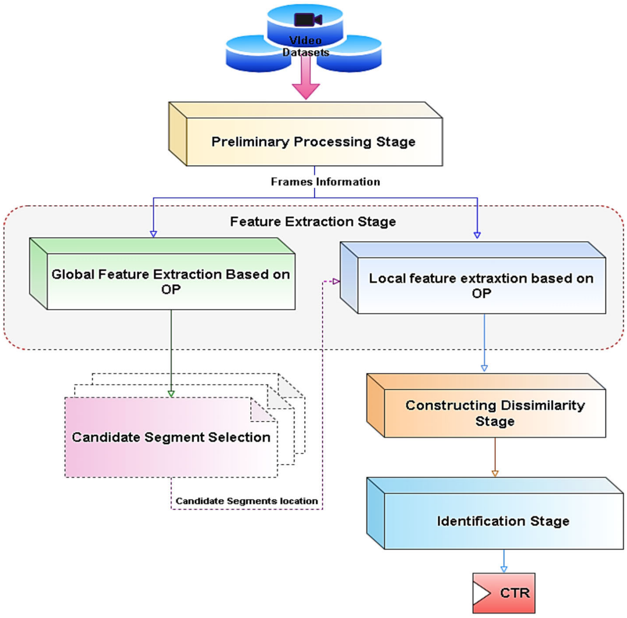

Video Datasets
Preliminary Processing Stage
Frames Information
Feature Extraction Stage
Global Feature Extraction Based on OP
Local feature extraxtion based on OP
Candidate Segment Selection
Candidate Segments location
Constructing Dissimilarity Stage
Identification Stage
CTR

FIGURE 1. The block diagram of the proposed algorithm.

In this section, the details of the proposed algorithm are presented. First, the active area of the frames is used to reduce the processed video frame area. Second, the proposed pre-processing module is applied to select the candidate segments.

A. MODIFIED FRAME ACTIVE AREA SELECTION

The active area selection is used to select the frame's region that holds most of the visual information. The visual material in the frames can be divided into three types: persistent, variable, and valuable material. The persistent material usually appears at the upper and lower ends of the frames. Fixed-logo, fixed-subtitle, and fixed-intensity regions are examples of persistent material. These persistent visual materials affect the similarity measurements because they are similar with different shots, as shown in Figure 2(a). The variable material has a significant effect on the dissimilarity measurements. Animated logos, animated subtitles, and transcripts are examples of variable material, as shown in Figure 2(b). The last type, valuable material, contains the rich features and considers the active part of the frame, as shown in Figure 2(c). Therefore, the method presented in [7] is modified and used to select the frame active area by removing a portion of the persistent and variable visual material.

For each video frame of size $N_1 \times N_2$, the frame active area selection is defined as follows and the resulting frame is of size $N_{A1} \times N_{A2}$:

$$N_{A1} = prc_1 \, N_1 \qquad (7)$$
$$N_{A2} = prc_2 \, N_2 \qquad (8)$$

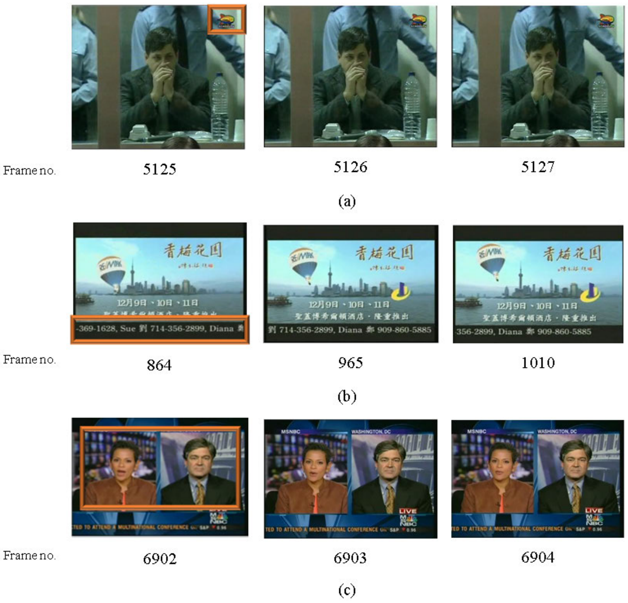


FIGURE 2. **Frame visual material: (a) persistent logo from video ID 05 from Dataset TRECVID 2007, (b) variable subtitle from video ID 10 from Dataset TRECVID 2006, and (c) valuable material from video ID 04 from Dataset TRECVID 2005. The material examples are surrounded by orange rectangle.**

where prc_1 and prc_2 are the selected percentages for the frame active area regions.

This selection will guarantee that the extracted features are accomplished from more reliable regions and that the computation is reduced by reducing the frame size, increasing the video processing speed.

B. DEVELOPED CANDIDATE SEGMENT SELECTION TECHNIQUE

The primary step used to provide a low computation cost is the selection of the boundary transition and the elimination of the non-boundary transitions [22]. This selection will prevent a frame-by-frame linear scan.

Given that the frames within short temporal segments have a high similarity, the first and last frames of each segment are checked by measuring their similarity. If they are similar, the segment is marked as a non-boundary segment. If the measurement shows they are dissimilar, the segment candidate is marked as a transition segment. The non-boundary transitions require no further processing, thereby considerably reducing processing time.

The method proposed in [7] is developed to exclude additional non-transition frames. In the proposed method, the video frames with an active area are transformed into the moment domain using squared Tchebichef-Krawtchouk transform (STKT). Then, the global moment is computed using Equation (6) to form the features. The high-energy moment coefficients (features) are selected to describe each frame, as shown in Section V-A. This selection will also reduce computation complexity.

A pre-processing module based on an adaptive threshold, inequality criteria, and bisection-based comparison is designed to filter out the transitions. Then, the candidate segment location is obtained and used in the next sub-stages. The pre-processing module is described as follows.

1) ADAPTIVE THRESHOLD

The adaptive threshold is used to filter out the non-boundary video segments. The first step in the adaptive threshold is to partition the video into certain skipped frames (S_f) for each segment. Between every two consecutive segments, one overlap frame provides the temporal continuity [44]. Therefore, the segment length is equal to $S_f + 1$. Then, the distance of the first and last frames in the k^{th} segment is measured as follows:

$$D(S_{f(k)}, S_{f(k+1)}) = \sum_{i=1}^{ord_1}\sum_{j=1}^{ord_2} | \eta(f_{kS_f}, i, j) - \eta(f_{(k+1)S_f}, i, j) | \quad k = 0, 1, \cdots, (N_{segment} - 1) \tag{9}$$

where $\eta(f_i)$ is the moment of the ith frame. Every ten segments are grouped, and the local threshold of these segments is computed. The local measurement is computed for each group, and the global measurement is computed for all the segments. The local and global statistics (mean and standard deviation) are used to adaptively calculate the threshold as follows:

$$TH = 2ml + 0.5\left(\frac{mg}{sl}\right)ml \tag{10}$$

where mg is the global mean computed from all segment distances in the video, and, ml and sl are the local mean and local standard deviation, respectively. When the distance value of the segment is greater than the threshold, the segment is classified as a boundary segment. For these boundary segments, no additional comparisons are needed.

2) INEQUALITY NESTED CRITERIA

The false-positive detected boundary segments are better than false negatives (miss-detected segments) [22]. Therefore, further inequality criteria are used for the non-boundary segments because the discarded segments cannot be retrieved. These criteria will provide the relationship between the segments with their neighbors, defined as

$$(D(k) > 5D(k-1)) \ or \ (D(k) > 5D(k+1)) \ or \ (D(k) > (1.5mg)) \tag{11}$$

When the distance of the previously classified non-boundary segment satisfies the criteria, the segment is re-classified as a boundary segment.

3) BISECTION-BASED COMPARISONS

For further elimination of the non-boundary segments in each candidate segment, two rounds of bisection comparisons are accomplished. Given that the cut transition (CTR) may occur in 1 or 2 frames, the scope of CTR search must be precisely identified. Therefore, the first round in the bisection comparisons is to divide the segment frames into two sub-segments. The forward distance (D_f) is measured between the middle frame and the first frame. However, the backward distance (D_b) is computed between the middle frame and the last frame as follows:

$$\begin{aligned} D_f &= \sum_{i=1}^{ord_1}\sum_{j=1}^{ord_2} \mid \eta(f_{kS_f+\frac{S_f}{2}}, i, j) - \eta(f_{kS_f}, i, j) \mid \\ D_b &= \sum_{i=1}^{ord_1}\sum_{j=1}^{ord_2} \mid \eta(f_{kS_f+\frac{S_f}{2}}, i, j) - \eta(f_{(k+1)S_f}, i, j) \mid \\ & k = 0, 1, \cdots, (N_{segment} - 1) \end{aligned} \tag{12}$$

Then, the relationship between the obtained three distances in (9), (11), and (12) are used to identify the candidate segment type as follows:

$$\begin{array}{cl} D_f/D_b > 1.6 \; and \; D_f/D(k+1) > 0.7 & \text{Type 1} \\ D_b/D_f > 1.6 \; and \; D_b/D(k+1) > 0.7 & \text{Type 2} \\ Df/(D(k+1)) < 0.55 \; and \; D_b/D(k+1) < 0.55 & \text{Type 3} \\ Elsewhere & \text{Type 4} \end{array} \tag{13}$$

where:

Type 1: means that the forward distance is larger compared to segment distance. In addition, the variance between the forward and backward is distinct. Therefore, the shot transition is in the first sub-segment frames only.

Type 2: means that the backward distance is larger compared to the segment distance. The variance between the forward and backward is also distinct. Therefore, the shot transition is in the second sub-segment frames only.

Type 3: means that no transition in the candidate segment occurs, and the transition is incorrectly identified. Thus, the segment is classified as non-boundary.

Type 4: means that the entire segment (the segment length is equal to $S_f + 1$) is preserved because the segment may contain a shot transition.

The second bisection round is applied every ($S_f/2 + 1$) segment-length to obtain two segments with a length of ($S_f/4 + 1$) and repeat (13). For Type 4 in the second round, the segment is also perceived with ($S_f + 1$) segment length. The bisection with two rounds is shown in Figure 3.

By the elimination of a large number of non-boundary frames, the computational cost is reduced. In addition, the obtained candidate segments are suspected for CTR because its length is only a small number of frames, and an accurate CTR position can be obtained. Figure 4 demonstrates the procedure for the candidate segment selection.

C. SBD WITH THE FAST-PROPOSED METHOD

The SBD framework includes three steps: feature extraction, DS construction, and CTR identification. Furthermore, a preliminary processing step is required to decode the video frames and perform color space conversion. Then, only the active area of the frames is considered in the subsequent stages to preserve the valuable visual information and provide the first step for computation time reduction. Therefore, the proposed frame active area is considered a sub-stage in the preliminary processing step.

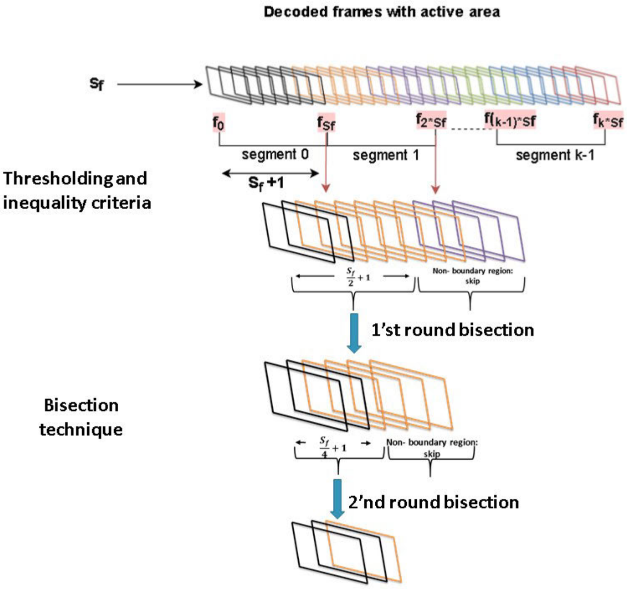


FIGURE 3. The bisection with two rounds.

1) FEATURE EXTRACTION OF SBD

Feature extraction plays a significant role in SBD because the content of the video frames is described by their features [18]. Therefore, in SBD, global and local features are extracted. The global features are extracted and used to find the location of the candidate segments. Thereafter, the local features of the candidate segments are considered to form the feature vector. The local features provide a robust visual representation compared with global features. Moreover, the local features resist object and camera motion and flash light effects, thereby improving detection accuracy [8]. Consequently, the mathematical model of moment block processing (MBP) proposed in [8] is adopted for direct local feature extraction with the reduction of the computation time. In this mathematical model, one matrix multiplication set is required to obtain the block processing result without requiring the video frame to be partitioned. The mathematical expression of MBP is defined as follows [8]:

$$\eta = P_{B1}\, I\, P_{B2}^{T} \tag{14}$$

where I is an image of size $N_{A1} \times N_{A2}$ with a block size of $B_1 \times B_2$ and the number of blocks is equal to $v_1 \times v_2$, where $v_1 = N_{A1}/B_1$ and $v_2 = N_{A2}/B_2$, and P_{B1} and P_{B2} is a single set of matrices for all the image two matrices are constructed from Z_{B1} of size $ord_1 \times B_1$ and Z_{B2} of size $ord_2 \times B_2$, respectively, by first performing a horizontal concatenation with a zero matrix and circular shift by $(v - 1)$ times. Then, we perform a vertical concatenation. In this study, the number of blocks is selected empirically as follows: $v_1 = v_2 = 8$ to compute the local features.

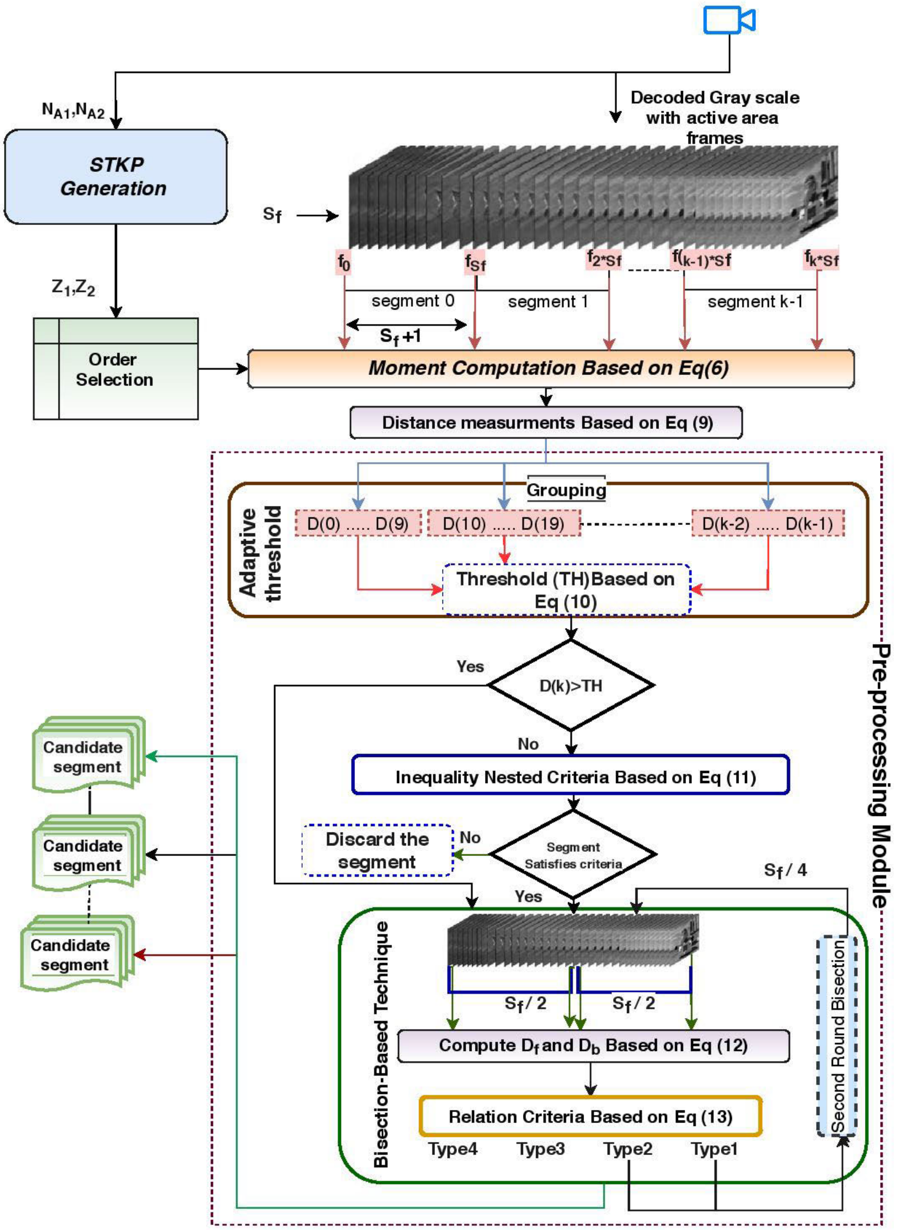


FIGURE 4. Candidate segment selection.

2) DISSIMILARITY MEASUREMENTS OF SBD

After obtaining the local features, the next subsequent stage in SBD is to find the DS between two computed moments of consecutive frames (f_k and $f(k+1)$). The DS is computed using the city-block distance metric, and the dissimilarity feature vectors (DFV) are formed for each of the candidate frames and defined as follows:

$$D(S_{f(k)}, S_{f(k+1)}) = \sum_{i=1}^{ord_{B1}} \sum_{j=1}^{ord_{B2}} \mid \eta(fs_{S_{f(k)}}, i, j) - \eta(fs_{S_{f(k+1)}}, i, j) \mid k \in loc_{candidate} \qquad (15)$$

where $loc_{candidate}$ represents the location of the candidate frames in the candidate segments, and ord_B is equal to $ord \times v$.

To improve detection accuracy, contextual information is considered [6]. This temporal information represents the features of previous (Pre) and posterior (Pos) candidate frames. The resultant feature vector is used in the next stage to identify the CTR transition. However, the dynamic range of the resulting features is considered a problem in detection. Therefore, feature vector normalization is important to retain features within a similar range [41], [45]. The mapping processes matrices by transforming the mean (x_{mean}) and standard deviation (x_{std}) of each row to desired mean (y_{mean}) and desired std (y_{std}) of the k^{th} feature vector (FV_c) as follows:

$$FV_n = (FV_c(k) - x_{mean}(k)\,) \times (y_{std}/(x_{std}(k))) + y_{mean} \qquad (16)$$

In this study, the feature vector is normalized with the desired mean and desired std equal to 0 and 1.2, respectively. This feature vector is used in the identification stage.

3) CTR DETECTION OF SBD

For the classification task, support vector machine (SVM), which is a popular supervised machine learning technique, is implemented [45], [46]. SVM has a significant advantage and is easier to implement than certain classifiers, as follows. First, compared with K-nearest neighbor (KNN), SVM provides better classification results for noise-free and noisy environments [38]. Second, compared with a neural network, SVM solves the local convergence problem [47]. Finally, compared with classical deep learning, SVM requires fewer parameters to be initialized with a smaller number of training samples; therefore, less training time is required [48], [49]. Moreover, deep learning requires many more logistical architecture requirements; hence, it needs much more computational power and resources than SVM does [49], [50].

In SVM, data are usually divided into training and testing sets in the classification task. The SVM generates a model from the training sets because they contain numerous attributes (features) and a label (target value) for each case. This model will be used to predict the test labels depending on the testing features only [45]. To improve performance, the radial basis function (RBF) is used with SVM. RBF represents a suitable choice because it provides non-linear mapping [45]. Moreover, it reduces the complexity by reducing the number of generated hyperplanes [45]. X-fold cross-validation is applied to tune the SVM cost parameter and RBF gamma parameter and overcome the over-fitting problem. SVM is implemented by using the LIB-SVM package [45], [51]. Figure 5 shows the block diagram of the SBD with fast video processing method.

V. EXPERIMENTAL RESULTS

This section discusses the performance of the proposed method in terms of speed and transition accuracy. Subsequently, a comparative analysis is performed to demonstrate the efficiency of the proposed method. The performance of the proposed fast and accurate method is evaluated by using a well-known dataset. TREC Video Retrieval Evaluation (TRECVID) 2001, 2005, 2006, and 2007 test data, which is co-sponsored by the National Institute of Standards and Technology, are used [52]. The test sets comprise many CTRs and GTRs, and the video sequences are reformed into the uncompressed AVI format. In this study, CTR is used to test the ability of the proposed method because it is more predominant than GTR [6], [53]. Table 1 shows the details of the TRECVID datasets. The table shows that the number of non-boundary frames is very large compared to the transition frames (CTR). Therefore, the elimination of these frames will highly reduce computation. The experimental results are performed using MATLAB on a Windows 10 PC with an Intel Core i7 2.4 GHz CPU and 16 GB of RAM.

Algorithm 1 Proposed Algorithm

```
Input: input video sequence (V)
Output: cut transition (CTR)
    Notation: $f$:video frame, $N_f$: total number of frames, $f_A$:frame with active area, $prc$: selected percentage, $S_f$:skipped frames, $Loc_{candidate}$: candidate frame location, $N_{A1}$:frame height with active area, $N_{A2}$:frame width with active area, $v$: no. of blocks, $Z$: the OP, $P_B$: the OP with block processing, $TH$: threshold, $D$: distance, $mg$: global mean, $ml$:local mean, $sl$: local standard deviation $k$:frame no., $DFV$: dissimilarity feature vector, $G_{truth}$: cut transitions positions in ground truth
procedure SHOT_BOUNDARY_DETECTION(V)
$\{f_1 \dots f_{N_f}\} \leftarrow$ Decode(V)
for $i = 1$ to $N_f$ do
    $f_A(i) \leftarrow$ Select frame active area ($f(i), prc1, prc2$)
end for
$\{Z_1, Z_2\} \leftarrow$ STKP generation($N_{A1}, N_{A2}$)    ▷ to extract global feature
Candidate_Segments_Selection($f_A$, $S_f$)    ▷ process all frames
$\{P_{B1}, P_{B2}\} \leftarrow$ generation($N_{A1}/v_1, N_{A2}/v_2$)    ▷ to extract local feature
Apply OP block processing
$\{\eta\} \leftarrow$ moment extraction    ▷ smooth & gradient moments

$\{D\} \leftarrow$ Dissimilarity($f_k, f_{k+1}$)
$FV_n \leftarrow$ (contextual information and feature normalization)
Cut_Transition_Detection( $FV_n$, $N_{gt}$)
end procedure
function Candidate_Segments_Selection($f_A$, $S_f$)
$segment\_lenght = S_f + 1$    ▷ segment video frames
for i = 1 to 10 do
    Group($segment(i)$)
end for
$TH = 2ml(k) + 1.5(mg/sl(k)(ml(k)))$    ▷ adaptive threshold

if $D(segment) > TH$ then
    candidate segment
else
    if $(D(k) > 5D(k-1)) or (D(k) > 5D(k+1)) or (D(k) > (1.5mg))$ then
        candidate segment
    end if
end if
if (candidate segment) then
    Apply two-round of bisection
    return $Loc_{candidate}$
else
    Discard the segment
end if
end function
function Cut_Transition_Detection($FV_n$, $N_{gt}$)
$predict_{label} \leftarrow$ svmpredict($FV_n$,model)
if ($predict_{label} = N_{gt}$) then
    Declare CTR
    return CTR
end if
end function
```

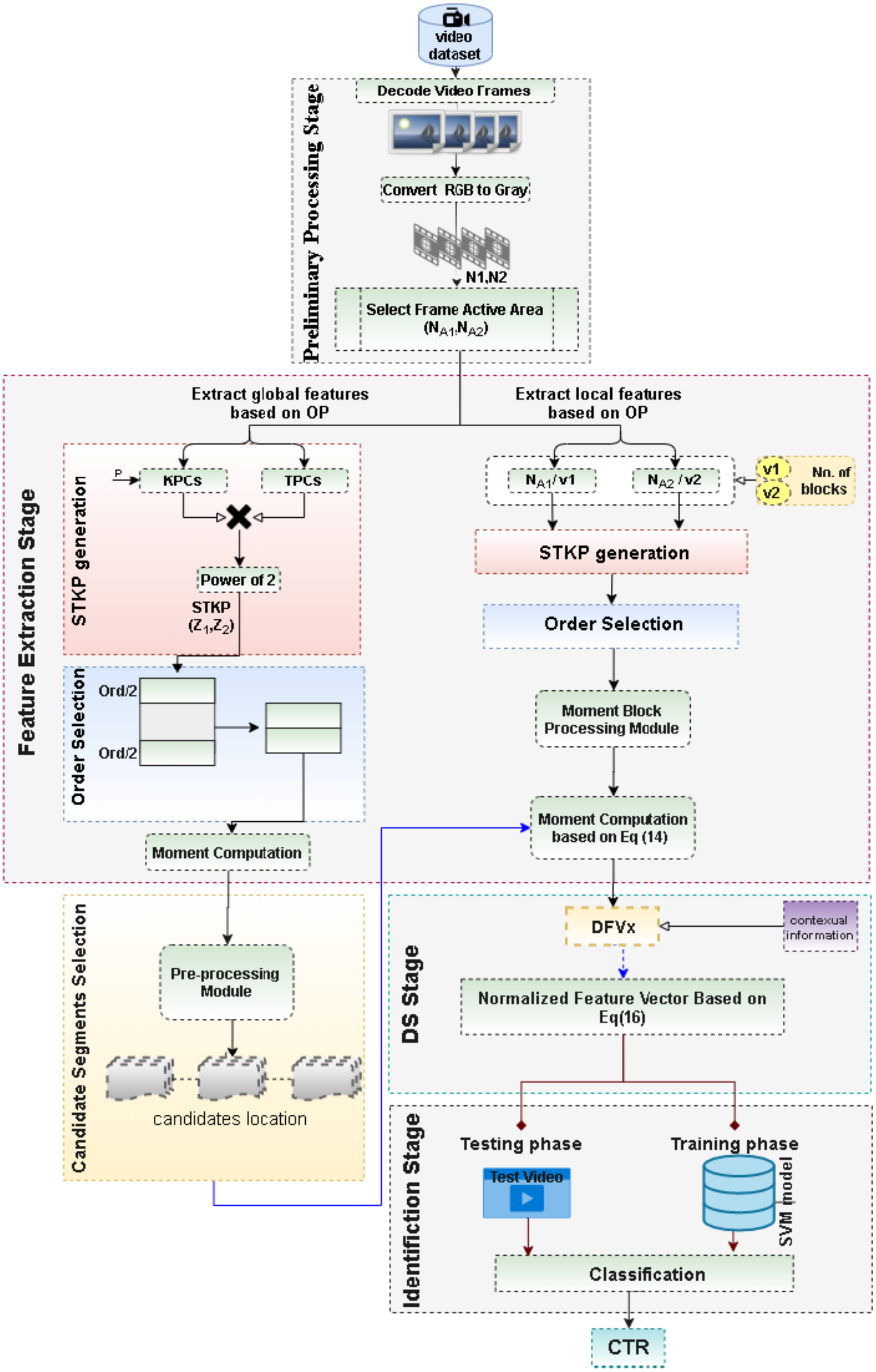


FIGURE 5. Complete flow diagram of CTR detection.

A. EVALUATION OF FRAME ACTIVE AREA SELECTION

For each video frame of size $N_1 \times N_2$, three cases of the frame active area selection are given in Table 2, and the resultant frame has the size $N_{A1} \times N_{A2}$.

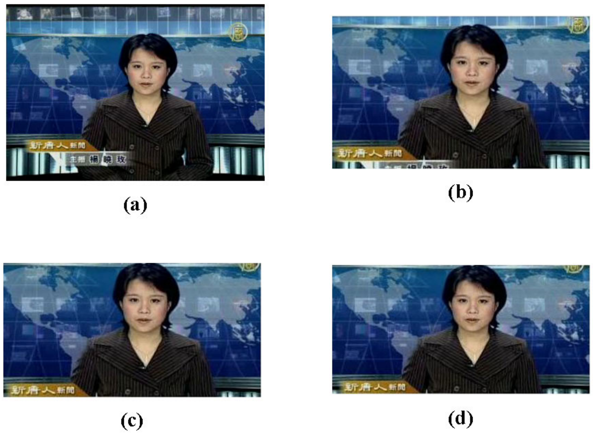
For the frame active area experiment, we chose the TRCE-VID 2005 dataset that consists of twelve videos. Three cases for frame active areas shown in Figure 6 are tested with the highest energy coefficients.

TABLE 1. Details of the TRECVID datasets.

Dataset	Total # of videos	Total # of frames	Total # of CTR	Total # of non-transition frames	Frame Height $(N1)$	Frame Width $(N2)$
2001	6	97,808	300	97,508	240	320
2005	12	744,604	2,815	741,789	240	352
2006	13	597,043	1,909	595,134	240	352
2007	17	637,805	2,236	635,569	288	352
Total	48	2,077,260	7,260	2,070,000	-	-

TABLE 2. Frame active area selection.

Case number	N_{A1}	N_{A2}	$N_{A1} \times N_{A2}$
case1	$75\% N_1$	$87\% N_2$	3/4 $N_1 \times$ 7/8 N_2
case2	$66\% N_1$	$87\% N_2$	2/3 $N_1 \times$ 7/8 N_2
case3	$66\% N_1$	$91\% N_2$	2/3 $N_1 \times$ $10/11 N_2$


(a)

(b)

(c)

(d)

FIGURE 6. Frame active area cases for examples extracted from TRECVID2005 from video ID 05. (a) Original frame, (b) Case1, (c) Case2, and (d) Case3.

For STKP, to select the high-order coefficients, we implement the following steps.

1) We compute STKP using Equation (3) for each frame's dimension (N_1 and N_2). The results are two polynomials of size equal to $N_1 \times N_1$ and $N_2 \times N_2$, respectively.
2) For STKP, the priority of the selection order in (n-direction) is $n = 0, N-1, 1, N-2, \ldots, N/2 + ord/2 - 1, N/2 - ord/2$. Therefore, we select the coefficients row-wise for each generated polynomial such that the selected coefficients equal the required order. Figure 7 shows the highest energy coefficient generation. The selection of these coefficients will reduce the computation cost because using the entire moment coefficients in the next computations is not needed.

The three cases for frame active area are tested with 6%, 12%, and 25% of the high energy moment's coefficients. The execution time is computed for each case and repeated three times for accuracy appraisal. Table 3 shows the average execution time for the frame active area cases. From Table 3, it can be inferred that as the size of the frames increases, the time required for the processing is increased with comparable visual material, as shown in Figure 6. The results show the superiority of Case2 in terms of average time for different values of moments coefficients percentages (6%, 12%, and 25%). In addition, it is clear that when the selected moment order is increased, the computational cost will also be increased because the size of the OP matrix is increased in the n-direction. Therefore, from the results, the lowest total execution time for Case2 is 1125.11 Seconds when the selected moment coefficients is 6%. On the basis of the obtained results, Case2 with a moment order of only 6% will be considered in the next experiments.

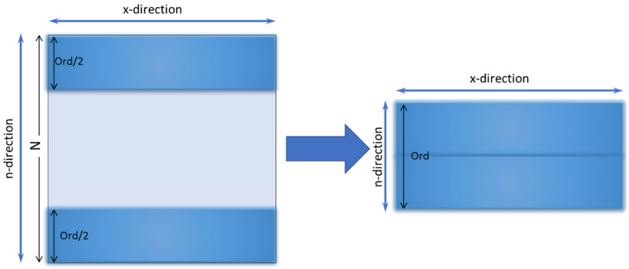


FIGURE 7. High-order coefficient selection for STKP.

B. EVALUATION OF CANDIDATE SEGMENT SELECTION

The parameters have a considerable effect on candidate segment selection and detection performance. Therefore, the selection of the threshold criteria and the bisection factors are to be suitable for complex video content. Decreasing the threshold parameters will lead to considering segments with lower distance as candidate segments. The number of S_f in each segment is selected to adjust between detection accuracy and speed. The small numbers of frames will increase the distance calculations, thereby increasing the computation complexity. By contrast, large values will reduce the accuracy by incorrectly considering a non-boundary segment as a boundary segment. To specify the best S_f, a matrix is formed with ones in the position of the frames in the candidate segments and zero elsewhere. Then, the matrix is compared with the CTRs in the ground truth. The comparison is demonstrated on two concepts: the frame percentage (FRP) and the transition accuracy percentage (TAP). Lower FRP values with higher TAP values are needed. FRP is defined as the number of frames that will be processed and defined as follows:

$$FRP = \frac{F_{seg}}{N_f} \times 100\% \tag{17}$$

TABLE 3. Average execution time for three cases of the frame active area.

Dataset 2005	Average (3 times) Processing time in seconds for 6% of moment coefficients			Average (3 times) Processing time in seconds for 12% of moment coefficients			Average (3 times) Processing time in seconds for 25% of moment coefficients		
Video ID	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
01	162.87	155.37	157.93	213.76	188.30	188.99	280.33	242.90	314.54
02	79.69	76.00	79.07	99.90	92.90	90.21	119.67	100.60	123.22
03	81.02	78.51	78.78	98.30	92.69	89.70	124.56	99.55	110.41
04	55.47	50.06	52.08	66.02	60.75	61.70	79.82	65.68	69.02
05	60.69	58.34	59.18	73.13	70.21	68.50	88.52	73.10	77.40
06	79.82	75.31	79.41	100.81	102.49	92.00	120.93	112.61	104.49
07	79.52	76.35	78.91	99.59	104.27	93.54	117.43	120.04	104.95
08	161.19	160.56	164.23	203.11	207.53	221.93	296.86	254.28	277.62
09	81.84	78.13	82.44	96.39	99.30	100.08	119.77	118.81	117.77
10	80.71	79.20	81.83	95.49	100.76	105.50	111.79	110.54	108.68
11	80.29	78.60	81.25	94.18	92.08	108.76	118.18	110.04	106.56
12	161.63	158.68	160.93	196.98	192.98	214.09	307.52	262.80	306.07
Total Time	1164.74	1125.11	1156.05	1437.66	1404.26	1434.99	1885.38	1670.95	1820.73

TABLE 4. Frame-skipping selection.

Dataset 2005	$S_f = 10$		$S_f = 14$		$S_f = 20$	
Video ID	TAP%	FRP%	TAP%	FRP%	TAP%	FRP%
01	99.35	20.67	98.70	21.34	95.78	22.55
02	99.21	22.69	98.82	23.66	96.46	25.63
03	100.00	21.60	100.00	21.99	99.03	23.55
04	100.00	22.04	100.00	23.02	100.00	24.34
05	99.41	24.30	98.22	25.23	97.04	26.65
06	98.35	24.91	95.71	24.75	89.77	25.55
07	98.94	23.00	97.54	24.18	93.66	24.78
08	100.00	18.76	100.00	19.50	99.53	20.57
09	100.00	22.31	100.00	22.92	100.00	23.30
10	100.00	24.11	100.00	24.93	98.61	26.86
11	97.79	20.18	98.53	20.02	97.06	20.73
12	100.00	19.76	99.67	20.71	99.34	22.52
Average	99.42	22.03	98.93	22.69	97.19	23.92
Processing Time (in seconds)	104.64		101.50		102.12	

where F_{seg} is the total number of frames in the candidate segments and N_f is the total number of frames in the video sequence.

TAP is the percent of correctly predicted transitions and it is computed as follows:

$$TAP = \frac{T_{correct}}{T_{groundtruth}} \times 100\% \qquad (18)$$

where $T_{correct}$ is the correctly predicted transition and $T_{groundtruth}$ is the total number of transitions in the ground truth. Different values of S_f are applied to the TRECVID 2005 dataset and the values of TAP and FRP are computed and recorded for each video, as shown in Table 4. Note that, only the adaptive threshold and the inequality criteria are used to select the best number of skipped frames. The results show that for small value of segment length, $S_f = 10$, the processing time increases because the number of segments are increased; and thus, the distance calculation are increased. On the other hand, for large values of skipped frames, $S_f = 20$, although the computation is reduced, some CTRs will be not recognized and judged as non-boundary. Hence, the TAP is reduced, and among the cases, the FRP shows the highest value of 23.92. The interesting finding is when $S_f = 14$, a tradeoff between FRP and TAP occurs. Therefore, this value of S_f (14) i considered the suitable selection because it affects the total number of frames and the accuracy of transitions detection.

Moreover, for more clarification, the performance of candidate segment selection technique is evaluated using 48 videos from the TRECVID dataset. Table 5 shows the results of FRP and TAP of the proposed method. The results are computed first for the first two steps (the adaptive threshold and inequality criteria) of the candidate segment. Then, the bisection-based method is applied. Without the bisection comparison, the eliminated frames are $\simeq$ 77% of the total frames. In other words, only $\simeq$ 23% of the frames $(475, 485$ frames out of $2, 077, 260)$ are processed in the next stages. These processed frames are further reduced when applying the bisection, and the frame percentage to be processed is approximately 17% (Only $353, 134$ frames needed to be processed from the number of frames $2, 077, 260)$ as reported in Table 1.

In conclusion, the results show a high TAP when applying the first two steps. Moreover, an upgrade in FRP when applying the bisection comparison occurs, eliminating further non-boundary transitions.

TABLE 5. Percentages of TAP and FRP in each step of the proposed method.

	Thresholding + inequality criteria		Bisection-based comparison	
Dataset	TAP%	FRP%	TAP%	FRP%
2001	97.77	23.30	94.82	17.90
2005	98.93	22.69	98.52	16.93
2006	96.77	22.87	93.58	17.17
2007	96.44	22.70	95.63	18.37
Average	97.48	22.89	96.39	17.59

To validate the superiority of the proposed scheme in terms of speed (FRP) and accuracy (TAP), the results of the proposed candidate segment selection technique is compared with two existing techniques, which are: the pre-processing method [22] and single-plane method [26]. The comparison is performed on the 2001, 2005, 2006, and 2007 datasets in terms of FRP and TAP for all methods, as shown in Table 6. For method in [22], it shows lower FRP rate than the proposed method; however, the TAP% is not satisfactory as many missed transitions occur in the candidate segments. By contrast, the method in [26] shows higher FRP rate and lower TAP% than the proposed method. For example, when considering the TREVID 2001 dataset, the proposed technique shows $\simeq 5\%$ fewer frames than the method in [26]. In other words, the frames are fewer by $5\% \times 97,808$ (from Table 1) $= 4,890$ frames, and the TAP is higher. Therefore, the results show the superiority of the proposed method to the existing methods.

TABLE 6. Comparison of candidate segment selection techniques.

	Proposed Method		Pre-Processing Method [22]		Single-Plane Method [26]	
Dataset	TAP%	FRP%	TAP%	FRP%	TAP%	FRP%
2001	94.82	17.90	97.81	16.63	97.39	22.19
2005	98.52	16.93	93.05	13.76	94.77	18.29
2006	93.58	17.17	84.72	14.67	90.08	19.56
2007	95.63	18.37	95.39	16.23	96.53	21.74
Average	95.63	18.37	95.39	16.23	96.53	21.74

For more elucidation, the average results of all datasets, which contains different video genres, are reported in Table 7. In addition, the improvement of the proposed method over the existing methods i also reported. From Table 7, the achieved improvement of FRP and TAP are 1.63% and 2.05%, respectively, which depicts that the proposed candidate segment selection technique is able to reduce the number of processes frames by 1.63% and increase the number of CTR transitions in the selected candidates by 2.05%, i.e. the trade-off is demonstrated by increasing the TAP with the reduction of FRP when the proposed technique is used.

Moreover, the proposed SBD method has several advantages over existing methods. First, the proposed SBD algorithm uses the active area for the frames in the candidate segment which in turn reduces computation cost. Second, the skipped frames are fewer compared to existing methods, in this case the number of processed frames is decreased and the computation cost will be reduced.

TABLE 7. Improvement ratio between the proposed candidate segment selection technique over existing methods.

Average results for the proposed method		Average results for the methods in [22] and [26]	
TAP%	FRP%	TAP%	FRP%
95.64	17.59	93.72	17.88

C. PERFORMANCE EVALUATION OF THE PROPOSED SBD ALGORITHM

For the SBD, the local features are extracted based on STKP with 6% of moment order. Then, the dissimilarity feature vector is formed with contextual information. We experimentally found that the values of $pre = pos = 2$ lead to a reasonable trade-off in results. The normalized feature vector is used as input in the SVM classifier that is trained with 50% training videos. The remaining videos of the datasets, 50% testing videos, are used for testing. Notably, the proposed SBD algorithm based on the proposed method overcomes the imbalance problem. This problem occurs due to the instability between the two classes [51]. The non-transition frames are more than the transition frames. By using the candidate segments with the bisection, many non-transition frames are excluded. Therefore, a balance occurs between the transition and the non-transition classes. The performance of the SBD is assessed in terms of computation cost and CTR detection. The evaluation metrics are precision (P), recall (R), F_{score}, and computation time. These metrics can be defined as follows [7]:

$$P = \frac{N_{cd}}{N_{td}} \times 100\% \tag{19}$$

$$R = \frac{N_{cd}}{N_{gt}} \times 100\% \tag{20}$$

$$F_{score} = \frac{2N_{cd}}{N_{td} + N_{gt}} \times 100\% \tag{21}$$

where N_{cd}, N_{td}, and N_{gt} are the correctly detected, totally detected, and ground truth transitions, respectively. Note that the computation time is reported for entire stages of the proposed SBD algorithm. Table 8 summarizes the performance accuracies and the computation time of the SBD algorithm. The results demonstrate that the CTR scheme provides good detection performance for P and R. The improvement in P rate means the reduction in false detected transitions. The improvement in R rate is from the correctly detected transitions. Good F_{score} results, which is the harmonic average of R and P, are obtained. Therefore, promising detection performance can be obtained regardless of the type of video dataset. For more clarification, the confusion matrices for the datasets used in the experiment are depicted in Figure 8.

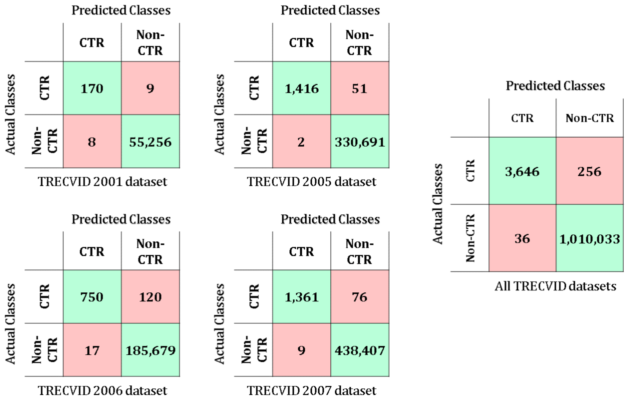
The robustness of the obtained results is verified by a comparison with the state-of-the-art SBD algorithm proposed

TABLE 8. SBD accuracy measurement and computation time.

Dataset	Ground Truth	Total Detected	Correctly Detected	P	R	F_{score}	Computation Time in (Sec)
2001	179	178	170	95.51	94.97	95.24	6.64
2005	1416	1367	1365	99.85	96.40	98.10	35.72
2006	870	767	750	97.78	86.21	91.63	23.62
2007	1437	1370	1361	99.34	94.71	96.97	66.01
Total	3902	3682	3646	99.02	93.44	96.15	131.99

TABLE 9. Comparison measurements of SBD.

Dataset	Proposed algorithm				SBD algorithm [7]			
	P	R	F_{score}	Computation Time in (Sec)	P	R	F_{score}	Computation Time in (Sec)
2001	95.51	94.97	95.24	6.64	94.65	97.79	96.20	12.53
2005	99.85	96.40	98.10	35.72	96.54	99.12	97.81	62.53
2006	97.78	86.21	91.63	23.62	95.33	97.87	96.58	38.99
2007	99.34	94.71	96.97	66.01	98.91	99.07	98.99	99.41



TRECVID 2001 dataset

Actual Classes \ Predicted Classes	CTR	Non-CTR
CTR	170	9
Non-CTR	8	55,256

TRECVID 2005 dataset

Actual Classes \ Predicted Classes	CTR	Non-CTR
CTR	1,416	51
Non-CTR	2	330,691

TRECVID 2006 dataset

Actual Classes \ Predicted Classes	CTR	Non-CTR
CTR	750	120
Non-CTR	17	185,679

TRECVID 2007 dataset

Actual Classes \ Predicted Classes	CTR	Non-CTR
CTR	1,361	76
Non-CTR	9	438,407

All TRECVID datasets

Actual Classes \ Predicted Classes	CTR	Non-CTR
CTR	3,646	256
Non-CTR	36	1,010,033

FIGURE 8. Confusion matrices for the obtained results in Table 8.

in [7]. Table 9 shows a comparison of the accuracy measurements and computation time. The results indicate an excellent runtime of the proposed algorithm by reporting a 38.17% improvement over the SBD algorithm. This improvement is related to the high reduction in the number of processed frames. However, the proposed algorithm shows slightly less accuracy than SBD does because of the number of features that are used. The proposed algorithm used only one feature based on the moment of STKP, but the SBD algorithm used three features based on embedding operators. Overall, by using the proposed algorithm, the computational cost is highly reduced with acceptable detection accuracies. Therefore, the proposed algorithm is superior to the state-of-the-art SBD algorithm.

To prove the efficiency of the proposed algorithm, additional comparisons are performed between the proposed algorithm and different existing algorithms. These algorithms are: SBD algorithm concatenated block-based SBD algorithm (CBB-SBD) [54], Walsh-Hadamard transform-based SBD algorithm (WHT-SBD) [11], SBD algorithm based on convolutional neural networks (CNN-SBD) [28], and SBD using Non-Subsampled Contourlet Transform (NSCT-SBD) [14].

TABLE 10. Comparison between the proposed algorithm and existing algorithm using TRECVID 2007 dataset.

Algorithm	Evaluation Metrics			
	$P\uparrow$	$R\uparrow$	$F_{score}\uparrow$	Time ↓ in Secs
WHT-SBD [11]	97.42	97.79	97.42	8003.12
NSCT-SBD [14]	96.36	97.66	97.01	11679.19
CNN-SBD [28]	-	-	97.36	729.82
Proposed	99.34	94.71	96.97	66.01

Table 10 reports a comparison between the proposed algorithm and existing algorithms using the TRECVID 2007 dataset. From Table 10 it can be seen that the highest F_{score} is reported for the WHT-SBD algorithm which is 97.42%. While the reported F_{score} for the proposed algorithm is 96.97%, which is only 0.45% less than the reported F_{score} of the WHT-SBD algorithm. However, regarding the computation time, the proposed algorithm offers a noticeable reduction when compared with the existing methods. The computation time reported for the proposed algorithm is 66.01 Secs which represents an improvement of 11 times than the CNN-SBD algorithm.

Another comparison is reported in Table 11 between the proposed algorithm, CBB-SBD algorithm, DeepSBD, and TSSBD using the TRECVID 2005 dataset. From Table 11, it is clear that the CBB-SBD algorithm outperforms the proposed algorithm by 0.26% only. However, for the computation time required to detect transitions, the proposed

TABLE 11. Comparison between the proposed algorithm and existing algorithm using TRECVID 2005 dataset.

Algorithm	Evaluation Metrics			
	$P\uparrow$	$R\uparrow$	$F_{score}\uparrow$	Time ↓ in sec
WHT-SBD [11]	96.99	95.00	95.50	6816.63
DeepSBD [28]	-	-	93.4	-
TSSBD [55]	93.2	93.8	93.5	-
Proposed	95.51	94.97	95.24	35.72

algorithm outperforms the CBB-SBD algorithm with an improvement of 190. In addition, the proposed algorithm outperforms the DeepSBD and TSSBD algorithms in terms of transition detection accuracy.

From the reported results it can be observed that the proposed algorithm shows a remarkable improvement in time to detect transitions, with a slight decrease in the accuracy. The proposed algorithm reduces the computation time remarkably (see Tables 9, 10 and 11) because of: 1) the number of frames processed is reduced using the developed segment selection technique, 2) the size of the frame which is reduced using the frame active area selection technique, and 3) the less number of moments (features) used to represent the frame.

VI. CONCLUSION

In this study, a fast video processing method for SBD based on frame active area and candidate segment selection technique is proposed. The frame active area is considered to preserve valuable visual information and provide the first step in computation time reduction. The candidate segment selection is implemented to reduce the computation for the successive stages by eliminating the non-boundary frames. To perform the elimination, adaptive threshold, inequality criteria, and two rounds of bisection comparisons are implemented as pre-processing module. Therefore, the proposed method demonstrates superior performance in terms of video processing speed and accuracy. The SBD application is implemented with the proposed method to detect CTR. STKP is used to extract SBD features and the transitions are detected using supervised machine learning. Compared with existing methods, the proposed method achieves remarkable results in terms of TAP and FRP by trade-off. Furthermore, an excellent reduction in computational cost is reached. For future effort, the proposed method is examined with several features to improve detection accuracy. In addition, in our future work, our goal is to generalize the proposed method to detect different types of shot transitions. Furthermore, different applications, especially for video processing and multimedia analytics, will be implemented based on proposed method.

ACKNOWLEDGMENT

The authors acknowledge the University of Baghdad, the University of Ha'il, and the King Fahd University of Petroleum & Minerals, Dhahran, for all support.

REFERENCES

[1] G. M. D. T. Forecast, "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," *Update*, vol. 2017, p. 2022, Feb. 2019.

[2] S. Aljawarneh and M. B. Yassein, "A multithreaded programming approach for multimedia big data: Encryption system," *Multimedia Tools Appl.*, vol. 77, no. 9, pp. 10997–11016, 2018.

[3] M. Wang, G.-W. Yang, S.-M. Hu, S.-T. Yau, and A. Shamir, "Write-a-video: Computational video montage from themed text," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–13, Nov. 2019.

[4] 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.

[5] G. G. L. Priya and S. Domnic, "Video shot cut detection using least square approximation method," in *Proc. Int. Conf. Inf. Process.*, Bangalore, India. Berlin, Germany: Springer-Verlag, 2011, pp. 161–170.

[6] 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, 2019.

[7] S. H. Abdulhussain, S. A. R. Al-Haddad, M. I. Saripan, B. M. Mahmmod, and A. Hussien, "Fast temporal video segmentation based on krawtchouk-tchebichef moments," *IEEE Access*, vol. 8, pp. 72347–72359, 2020.

[8] S. H. Abdulhussain, 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.*, Jul. 2019, pp. 1–8.

[9] 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.

[10] 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.

[11] G. G. L. 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. Iyengar, "Multimedia big data analytics: A survey," *ACM Comput. Surveys*, vol. 51, no. 1, pp. 1–34, 2018.

[17] Y. Pan, Z. Niu, J. Wu, and J. Zhang, "InSocialNet: Interactive visual analytics for role—Event videos," *Comput. Vis. Media*, vol. 5, no. 4, pp. 375–390, Dec. 2019.

[18] R. S. Choras, "Image feature extraction techniques and their applications for CBIR and biometrics systems," *Int. J. Biol. Biomed. Eng.*, vol. 1, no. 1, pp. 6–16, 2007.

[19] C.-W. Su, H.-Y. M. Liao, H.-R. Tyan, K.-C. Fan, and L.-H. Chen, "A motion-tolerant dissolve detection algorithm," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1106–1113, Dec. 2005.

[20] C.-C. Lo and S.-J. Wang, "Video segmentation using a histogram-based fuzzy c-means clustering algorithm," *Comput. Standards Interfaces*, vol. 23, no. 5, pp. 429–438, Nov. 2001.

[21] 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, 2019.

[22] 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.

[23] 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.

[24] 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.

[25] S. Tippaya, S. Sitjongsataporn, T. Tan, K. Chamnongthai, and M. Khan, "Video shot boundary detection based on candidate segment selection and transition pattern analysis," in *Proc. IEEE Int. Conf. Digit. Signal Process. (DSP)*, Jul. 2015, pp. 1025–1029.

[26] S. Dhiman, R. Chawla, and S. Gupta, "A novel video shot boundary detection framework employing DCT and pattern matching," *Multimedia Tools Appl.*, vol. 78, no. 24, pp. 1–17, 2019.

[27] M. Gygli, "Ridiculously fast shot boundary detection with fully convolutional neural networks," in *Proc. Int. Conf. Content-Based Multimedia Indexing (CBMI)*, Sep. 2018, pp. 1–4.

[28] A. Hassanien, M. Elgharib, A. Selim, S.-H. Bae, M. Hefeeda, and W. Matusik, "Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks," 2017, *arXiv:1705.03281*. [Online]. Available: http://arxiv.org/abs/1705.03281

[29] T. Souček and J. Lokoč, "TransNet v2: An effective deep network architecture for fast shot transition detection," 2020, *arXiv:2008.04838*. [Online]. Available: http://arxiv.org/abs/2008.04838

[30] 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)*, 2019, pp. 147–153.

[31] A. M. Abdul-Hadi, S. H. Abdulhussain, and B. M. Mahmmod, "On the computational aspects of charlier polynomials," *Cogent Eng.*, vol. 7, no. 1, Jan. 2020, Art. no. 1763553.

[32] B. M. Mahmmod, A. M. Abdul-Hadi, S. H. Abdulhussain, and A. Hussien, "On computational aspects of Krawtchouk polynomials for high orders," *J. Imag.*, vol. 6, no. 8, p. 81, Aug. 2020.

[33] 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.

[34] B. M. Mahmmod, S. H. Abdulhussain, M. A. Naser, M. Alsabah, and J. Mustafina, "Speech enhancement algorithm based on a hybrid estimator," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1090, no. 1, Mar. 2021, Art. no. 012102.

[35] S. H. Abdulhussain, B. M. Mahmmod, M. A. Naser, M. Q. Alsabah, R. Ali, and S. A. R. Al-Haddad, "A robust handwritten numeral recognition using hybrid orthogonal polynomials and moments," *Sensors*, vol. 21, no. 6, p. 1999, Mar. 2021.

[36] I. M. Hameed, S. H. Abdulhussain, and B. M. Mahmmod, "Content-based image retrieval: A review of recent trends," *Cogent Eng.*, vol. 8, no. 1, 2021, Art. no. 1927469.

[37] S. H. Abdulhussain and B. M. Mahmmod, "Fast and efficient recursive algorithm of meixner polynomials," *J. Real-Time Image Process.*, pp. 1–13, Apr. 2021, doi: 10.1007/s11554-021-01093-z.

[38] 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.

[39] W. A. Jassim, P. Raveendran, and R. Mukundan, "New orthogonal polynomials for speech signal and image processing," *IET Signal Process.*, vol. 6, no. 8, pp. 713–723, Oct. 2012.

[40] 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, Aug. 2018.

[41] 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, 2018.

[42] 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.

[43] S. H. Abdulhussain, A. R. Ramli, A. Hussain, B. Mahmmod, and W. Jassim, "Orthogonal polynomial embedded image kernel," in *Proc. Int. Conf. Inf. Commun. Technol.*, 2019, pp. 215–221.

[44] C.-M. Chew and M. S. Kankanhalli, "Compressed domain summarization of digital video," in *Proc. Pacific-Rim Conf. Multimedia*, Beijing, China. Berlin, Germany: Springer, 2001, pp. 490–497.

[45] C.-W. Hsu, C.-C. Chang, C.-J. Lin, and Others, "A practical guide to support vector classification," Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2003.

[46] R. Muralidharan and C. Chandrasekar, "Object recognition using SVM-KNN based on geometric moment invariant," *Int. J. Comput. Trends Technol.*, vol. 1, no. 1, pp. 215–220, 2011.

[47] X. Ling, O. Yuanxin, L. Huan, and X. Zhang, "A method for fast shot boundary detection based on SVM," in *Proc. Congr. Image Signal Process.*, vol. 2, 2008, pp. 445–449.

[48] P. Liu, K.-K. R. Choo, L. Wang, and F. Huang, "SVM or deep learning? A comparative study on remote sensing image classification," *Soft Comput.*, vol. 21, no. 23, pp. 7053–7065, 2017.

[49] A. Varghese, G. Agyeman-Badu, and M. Cawley, "Deep learning in automated text classification: A case study using toxicological abstracts," *Environ. Syst. Decisions*, vol. 40, pp. 1–15, Feb. 2020.

[50] T. L. Duc, R. G. Leiva, P. Casari, and P.-O. Östberg, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey," *ACM Comput. Surveys*, vol. 52, no. 5, pp. 1–39, Oct. 2019.

[51] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

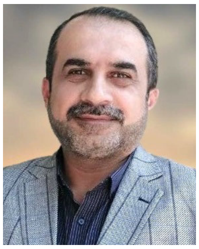
[52] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of TRECVid activity," *Comput. Vis. Image Understand.*, vol. 114, no. 4, pp. 411–418, 2010.

[53] L. Krulikovská, J. Pavlovič, J. Polec, and Z. Černeková, "Abrupt cut detection based on mutual information and motion prediction," in *Proc. ELMAR*, 2010, pp. 89–92.

[54] Y. Bendraou, F. Essannouni, D. Aboutajdine, and A. Salam, "Shot boundary detection via adaptive low rank and SVD-updating," *Comput. Vis. Image Understand.*, vol. 161, pp. 20–28, Aug. 2017.

[55] L. Wu, S. Zhang, M. Jian, Z. Lu, and D. Wang, "Two stage shot boundary detection via feature fusion and spatial-temporal convolutional neural networks," *IEEE Access*, vol. 7, pp. 77268–77276, 2019.

ZINAH N. IDAN was born in Baghdad, Iraq, in 1989. She received the B.Sc. and M.Sc. degrees in computer engineering from the University of Baghdad, in 2011 and 2020, respectively. Her research interests include computer vision and signal processing.

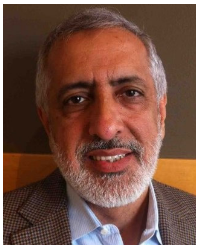


SADIQ H. ABDULHUSSAIN received the B.Sc. and M.Sc. degrees in electrical engineering from Baghdad University, in 1998 and 2001, respectively, and the Ph.D. degree from Universiti Putra Malaysia, in 2018. Since 2005, he has been a Staff Member with the Computer Engineering Department, Faculty of Engineering, University of Baghdad. His research interests include computer vision, signal processing, as well as speech and image processing.

BASHEERA M. MAHMMOD 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.

KHALED A. AL-UTAIBI (Member, IEEE) was born in Riyadh, Saudi Arabia, in 1973. He received the B.S., M.S., and Ph.D. degrees in computer engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 1997, 2002, and 2019, respectively. From 2002 to 2004, he worked as a Lecturer with the Department of Computer Engineering, King Fahd University of Petroleum and Minerals. He joined the Department of Computer Engineering, University of Hail, Hail, Saudi Arabia, as a Lecturer, in 2004, where he became an Assistant Professor, in 2020. His current research interests include image cryptography, optimization algorithms, machine learning, and artificial intelligence.

SYED ABDUL RAHMAN AL-HADAD (Senior Member, IEEE) received the Ph.D. degree in electrical, electronic and systems engineering from the National University Malaysia. He is specialized in human speech processing, animal sound processing, al-Quran sound processing, sound media security, and biometric. Since 1997, he has been working with the Department of Computer and Communications Systems Engineering, Universiti Putra Malaysia, and promoted to an Associate Professor, in 2012. He is the Head of the Laboratory Information Engineering and Robotics. Furthermore, he had taught 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. He also actively joins professional society, such as the Deputy Chair of the IEEE Systems, Man and Cybernetics, MITS, MSET, and others.



SADIQ M. SAIT was born in Bengaluru. He received the bachelor's degree in electronics engineering from Bangalore University, in 1981, and the master's and Ph.D. degrees in electrical engineering from the King Fahd University of Petroleum and Minerals (KFUPM), in 1983 and 1987, respectively. He is currently a Professor of computer engineering and the Director of the Center for Communications and IT Research, Research Institute, KFUPM. He has authored over 300 research articles, contributed chapters to technical books, and lectured in over 25 countries. He is also the principle author of two books. He received the Best Electronic Engineer Award from the Indian Institute of Electrical Engineers, Bengaluru, in 1981.