Journal of AI and Data Mining
Vol 7, No 4, 2019, 521-535

DOI: 10.22044/JADM.2019.7850.1928

Video Abstraction in H.264/AVC Compressed Domain

A. Yamghani[1] and F. Zargari[2*]

1. Department of Computer Engineering, Science and Research branch, Islamic Azad University, Tehran, Iran.
2. Department of information technology of Iran Telecom Research Center (ITRC), Tehran, Iran.

Received 24 December 2018; Revised 08 March 2019; Accepted 10 April 2019

**Corresponding author: zargari@itrc.ac.ir (F.Zargari).*

Abstract

Video abstraction allows searching, browsing, and evaluating videos only by accessing their useful contents. Most of the studies performed have used pixel domain, which requires the decoding process and needs more time and process than the compressed domain video abstraction. In this paper, we present a new video abstraction method in the H.264/AVC compressed domain, AVAIF. This method is based upon the normalized histogram of the extracted I-frame prediction modes in the H.264 standard. The frames' similarities are calculated by intersecting their I-frame prediction modes' histogram. Moreover, the fuzzy c-means clustering is employed to categorize similar frames and extract key frames. The results obtained show that the proposed method achieves, on average, 85% accuracy and 22% error rate in compressed domain video abstraction, which is higher than the other tested methods in the pixel domain. Moreover, on average, it generates video key frames that are closer to human summaries, and it shows robustness to coding parameters.

Keywords: *Video Abstraction, Clustering, Prediction Modes' Histogram, Compressed Video, Key Frame Extraction, Compressed Domain Feature Vector.*

1. Introduction

Development of new methods for an efficient storage and compression, and transferring videos, especially on the Internet, has caused massive volume of video libraries to emerge. The access to these video libraries needs to be in a way that the users can decide to watch a whole video easily and quickly. Moreover, by removing the redundant data, we can generate a simple form of video content to be saved, retrieved, and indexed. Removing the redundant data and producing a simplified version of a video is called video summarization. A video summary can be in two forms: dynamic and static. The dynamic video summary is created from small video shots and keeps their timing sequence. Various parts of a video can be separated conceptually by shot boundary detection. Then representatives for each one of the parts can be chosen. The static summarization is another form, which is defined based on frames, and is called abstraction, as well. Video abstraction is a kind of video summarization that extracts key frames containing the most distinguished content of a video [1, 2]. In video abstractions, extracting the frames' features and calculating the distance between these frames are performed to obtain the similar frames. Finally, the key frames are selected as representatives from each group of similar frames. The key frames are extracted by different methods such as key point-based selection [3], PCA [4], threshold-based [5], video segmentation [6], models based on content-based [7], and data clustering [8, 9]. Although selection of the cluster numbers is always an issue, the most common method of key frame extraction is clustering. The key frame determination in all the techniques mentioned above needs to extract features from the constituent frames of the original video.

The main goal of video abstraction is to find the similar components and select a representative for them such that the volume of the original video is decreased as well as preserving the primary video message. One of the advantages of video abstraction over dynamic summarization is consuming less memory to keep the results. Video abstraction can be performed in both the pixel (uncompressed) and compressed domains. In most uncompressed domain abstraction approaches, the whole frame is used for feature extraction. These features can be visual such as color or texture changes or audio and voice. Although some of these methods may achieve a reasonably acceptable result, they are time-consuming and require a large memory to access the compressed video in the pixel domain. Nowadays, most of the videos are stored in the compressed domain due to its better storage and transmission bandwidth management. Therefore, it is beneficial to use features directly available in the compressed bit-stream to avoid decoding and storage of a decompressed video.

H.264/AVC is one of the most common international standards that is used in video storage, transmission, and streaming. It has improved compression performance compared to the earlier standards using some features such as greater flexibility on compression options, transmission support, multiple reference frames, and flexible macroblock.

In this paper, we proposed a new method for the H.264/AVC compressed domain video abstraction based on various prediction modes in I-frames called AVAIF (AVC Video Abstraction by I-frame Features). AVAIF groups similar I-frames in clusters using the compressed domain features, and then it specifies the key frames. Moreover, this method is designed to determine the key frames automatically, which is remarkable compared to the other methods.

The main contributions of this paper are as follow:

- Extracting features by partial decompression of video
- Using fuzzy clustering to determine key frames
- Automatically estimating the cluster number
- Removing redundant key frames using the compressed domain features

The rest of the paper is organized as what follows. In Section 2, we discuss the works related to video abstraction and their issues. In Section 3, first, a brief explanation about the H.264/AVC coding standard is provided, and then the proposed method is given. Section 4 includes the experimental results, followed by concluding remarks in Section 5.

2. Related works

A video abstract is a simplified version of the original video, and consists of a set of frames extracted from it. There are different approaches for video abstraction but all summarize the video sequence as a set of key frames. In this section, some of the leading techniques and related issues of a key frame extraction are explained. For more information about the existing methods, [11, 12] can be referred to. The most common key frame extraction technique is to use the low-level features to compute the frame differences and remove the frames whose differences are less than a threshold. These low-level features can be color histogram, edge histogram, pixelate difference, etc. [13].

Yan and Hauptmann [14] first split the frame into 5×5 blocks to capture local color information. Then in each block, the color histogram and color moments are extracted for video retrieval. Sun *et al*. [15] have constructed a maximum occurrence frame for a shot. Then a weighted distance is calculated between each frame in the shot and the constructed frame. The key frames are extracted at the peaks of the distance curve. DeMenthon *et al*. [16] considered a video sequence as a trajectory curve in a high-dimensional feature space. They introduced a method to extract the key frames by finding discontinuities on the curve. Block intensity Comparison Code (BICC) is a new feature proposed in [17] for video classification to compute the average intensity in k × k blocks of each frame. After classifying the genres, an unsupervised algorithm is applied to detect shot transition. Irtaza *et al*. [18] applied wavelet packets tree, Gabor analysis, and curvelet transform as the feature vectors, and generated the corresponding feature vector by fusing them. Then for similarity detection, they used the Pearson correlation-based similarity calculation. Although this is an approach they used for CBIR, it can be used to detect the key frames as well. Zhang *et al*. [19] first extracted a feature curve by selecting a group of new distance characteristics. Then they obtained the key frame set based on the amplitude of the curve separation automatically. This approach is adaptive, and extracts the less number of key frames in the slow motion and extracts more in the intense movement. Ejaz *et al.* [20] presented a new method to extract the key frames based on the combined visual features. Correlation of RGB color channels, color histogram, and moments of inertia are pulled together to define an adaptive measure. Using this combined measure, the difference between the current frame and the last key frame is calculated. In this method, Euclidean distance in HSV color histogram is used to remove the redundant frames. The merits of the comparison-based algorithms include their simplicity, intuitiveness, and adaptation of the number of key frames to the length of the shot. The limitations of these algorithms include the followings. 1) The key frames represent the local properties of the shot rather than the global properties. b) The irregular distribution and uncontrolled number of key frames make these algorithms unsuitable for applications that require an even distribution or a fixed number of key frames. c) Redundancy can occur when there are contents appearing repeatedly in the same shot.

These methods work based on the sufficient content change between the consecutive frames or the current frame and the last key frame. Therefore, the resulting key frames do not adequately represent the precedent video portion [11]. The other most common video abstraction method is clustering to organize the key frames. Similar frames are grouped using video frames' features. The key frames are determined in each cluster by a selection measure.

Song *et al*. [8] have proposed a method based on fast clustering of the regions of interest. The key frames in each shot are extracted using the average histogram algorithm. Then based on the key points, several regions of interest are expanded. Finally, the fast clustering method is performed on the key frames by utilizing their ROIs. Zhang *et al*. [21] presented a motion-based clustering algorithm that first segmented the video to shots. Some key frames are selected for each video shot, and a clustering algorithm is used based on the motion compensation error to represent the video key frames. Mundur *et al*. [22] have proposed a clustering algorithm based on Delaunay Triangulation (DT), which presents frames by the HSV domain features. The clusters are selected by edge separation in a Delaunay algorithm. Cheng [23] improved the previous approach by clustering all frames in a shot. Then large enough clusters are selected, and the closest frame to each cluster center is considered as a key frame. Furini *et al*. [24] proposed a technique for video summarization based on the Furthest-point-First (FPF) algorithm. The number of clusters was calculated based on the dissimilarity between the consecutive frames. To do so, the generalized Jaccard distance was used. Unexpected movement or a scene change may cause maximum dissimilarity. Avial *et al*. [10] have proposed a method called VSUMM, in which the HSV color features are extracted after pre-sampling. After removing the irrelevant frames, the remaining ones are clustered by the k-means algorithm. Then the key frames are selected from each cluster, and the similar ones are removed. Asadi and Moghadam [25] have introduced a new technique based on fuzzy clustering. It first detects the shots and then applies the fuzzy c-means clustering. It extracts the key frames whose membership degrees are the highest in each cluster. Khara *et al*. [26] have proposed a method in which a combined feature from color, texture, and shape is used. It performs clustering using a density-based clustering algorithm.

One of the main problems with the clustering methods is determining the number of clusters before applying the algorithm. Moreover, they may not preserve the visual temporal order of the video frames. There are other methods that use compressed domain features in video analysis to avoid much time and memory consumption.

De Bruyne *et al*. [6] have presented a method to detect the shot boundary in compressed domain using the H.264/AVC standard. In this approach, first, a shot boundary detection algorithm is employed to segment the H.264/AVC bit streams based on temporal dependencies and spatial dissimilarities. It was developed to extract hierarchical coding patterns. Herranz and Martinez [9] have presented an algorithm based on clustering and ranking. It extracts DC color descriptor for I-frame in each GOP. Then the input sequence is segmented into video shots based on the criterion proposed in [6]. The threshold is adaptive instead of the fixed one in [6]. The video shots are hierarchically clustered and ranked to generate scalable summaries. Xiang-wei *et al*. [27] have proposed an approach in the compressed domain, where the DCT and DC coefficients are extracted from video frames. These factors are combined with the Rough set to generate an information system. They applied the Rough set to reduce the redundancy. Then this information system creates the final video abstraction. Almeida *et al*. [28] have introduced an approach that operates directly in the compressed domain. It first calculates the feature vectors from HSV color histogram on DC image for each I-frame. Then it compares the consecutive frames and selects a representative frame per each group. The redundant or meaningless frames are removed with color histogram and gradient orientation.

Recently, deep learning methods have become popular for video summarization. The intuition of using recurrent models is to effectively capture long-range dependencies among video frames.

Wu *et al*. [29] combined eye movement with video content by imitating the visual processing in human cortex. They proposed a model based on foveated two-stream deep ConvNets. In the spatial stream, the foveated images are constructed based on subjects' fixation points to convey the visual appearance of the video. In the temporal stream, the multi-frame motion vectors are built up to extract movement information of the input video. Rochan *et al*. [30] proposed a model with three properties:

1) convolutional across the temporal domain.
2) Process all frames simultaneously.
3) Semantic segmentation.

In this method, the encoder is used to process the frames to extract both high-level semantic features and long-term structural relationship information among frames, while the decoder is used to produce a sequence of 0/1 labels. Otani *et al*. [31] used deep features that could encode various levels of content semantics including objects, actions, and scenes. To do so, they designed a deep neural network that mapped videos as well as descriptions to a common semantic space, and jointly trained it with associated pairs of videos and descriptions. Then the deep features were extracted from each segment of the original video and apply a clustering-based summarization technique to them.

These methods are based on the semantic and require powerful processors and a lot of training data to work reasonably on keyframe extraction.

In this paper, we proposed a new video abstraction method in the H.264/AVC compressed domain. Apart from semantic and more focused on the frame structure, this method is based on the prediction mode histograms of I-frames in H.264/AVC encoded video. Moreover, it applies a new technique to determine the cluster number using the compressed domain features. Besides, temporal order of key frames is considered in the final result. The experimental results indicate that the proposed method achieves an impressive quality in the extracted key frames.

3. Proposed method

In this section, we first provide a brief explanation about the H.264/AVC standard to the extent that is required for understanding the rest of the paper. For further explanation about this standard, we refer the interested readers to [32]. H.264/AVC employs intra- and inter-prediction to provide estimation for a block from the current frame or previously coded frames, respectively. The frames in H.264/AVC are divided into the I, P, and B frames. I-frames only employ intra-prediction, whereas in the P and B frames, inter-prediction can be used as well (figure 1). In this work, we employed the H264/AVC intra-frame coding for video abstraction. Hence, we describe I-frame coding of H.264/AVC in more detail.

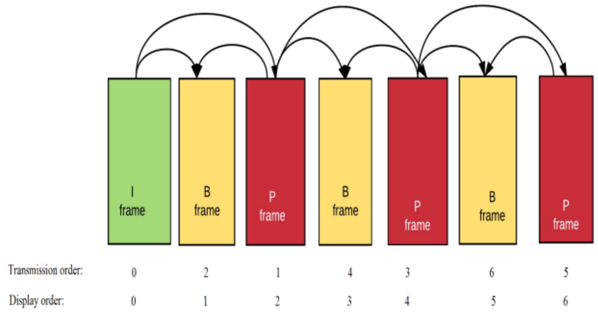

I frame
B frame
P frame
B frame
P frame
B frame
P frame
Transmission order: 0 2 1 4 3 6 5
Display order: 0 1 2 3 4 5 6

Figure 1. Frame sequence.

In I-frame coding, the pixels in the left and upper boundaries of a block are used to provide intra-prediction for the pixels inside the block. There are nine prediction modes for 4×4 blocks (Figure 2) and four prediction modes for 16×16 blocks (Figure 3) in the H.264/AVC standard. To provide a clear view of the intra-prediction and the mode selection in the H.264/AVC standard, suppose that the block, which is coded, includes horizontal lines. In this case, when we use the left boundary pixels of the block as a prediction for the pixels and subtract each pixel in the left boundary from the pixels in the same row of the block, the resulting residues are lower than the residues produced by using other boundary pixels and prediction directions. Hence, the horizontal mode is used for coding horizontal textures. As a result, the selected mode for coding each block in the image indicates its texture. For example, coding a block by vertical, horizontal, 45° or 135° show textures in the direction of vertical, horizontal, 45° or 135°, respectively. As colored samples are smooth in large areas, the color components are predicted the same as the 16x16 blocks.

The histogram of prediction modes is an appropriate descriptor for structure of I-frames in H.264/AVC coded videos and images [30]. It was employed for classification and retrieval of radiology images [34], and in this work, it was used as an effective feature vector for compressed domain video abstraction.

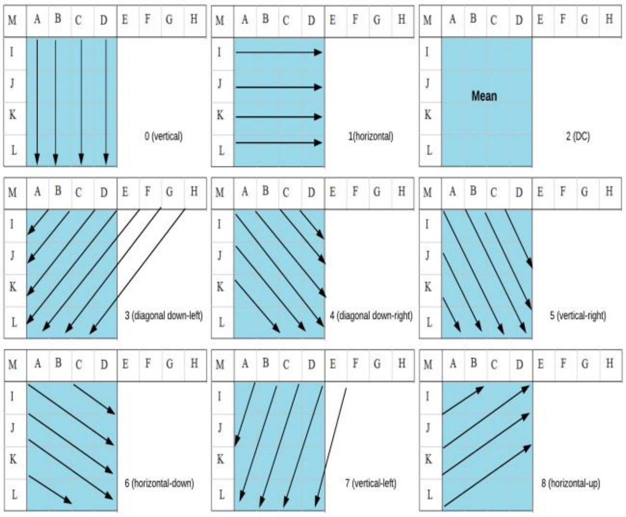


Figure 2. 4x4 intra-prediction modes in H.264.

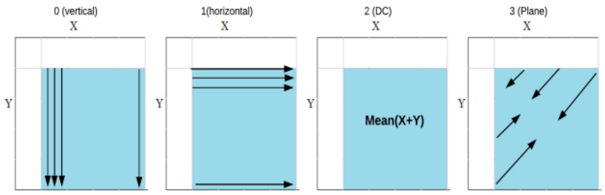


Figure 3. 16×16 intra-prediction modes in H.264.

Figure 4 illustrates the steps to produce video abstraction in the compressed domain using the prediction modes of I-frames. First, a feature vector is generated based on the prediction modes for each frame using I-frames from videos in the compressed domain. Next, a simple algorithm is used to detect the number of clusters, and then it groups video frames with a similar content. The fuzzy clustering algorithm is used to group the similar frames based on the prediction modes' histogram intersection. Then the frame with the highest membership degree is selected as a key frame per cluster. Moreover, the selected key frames are filtered to remove the possible redundant frames in the result. Finally, the key frames are organized based on their original time sequence. In the following, each sub-section explains the proposed method steps.

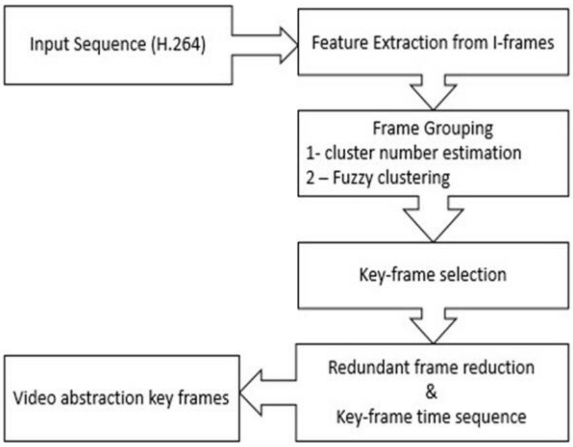

Input Sequence (H.264)
Feature Extraction from I-frames
Frame Grouping
1- cluster number estimation
2 – Fuzzy clustering
Key-frame selection
Redundant frame reduction
&
Key-frame time sequence
Video abstraction key frames

Figure 4. Our proposed approach.

3.1. Feature extraction

Extracting features to describe video frames is the first step in video abstraction. As block prediction modes in H.264/AVC can be a proper descriptor for I-frames' texture, their histogram for I-frame is considered as the main feature of each video frame [33]. We used the proposed method in [33] to evaluate the similarity between I-frames, which is explained in the following. The prediction modes zero to eight are prediction modes of 4×4 blocks, whereas the prediction modes nine to twelve are dedicated to 16×16 blocks. The prediction mode histogram is generated as:

$$H_i'(L)=\begin{cases} h_i & 0\le i\le 8 \\ h_i\times 16 & 9\le i\le 12 \end{cases} \qquad (1)$$

where *H'i (L)* is the bin of the histogram for the i-th mode and h_i indicates the number of 4×4 blocks in the coded picture, which are predicted by the i-th mode. Multiplying h_i by 16 for the 16×16 blocks is used to indicate the number of 4×4 blocks that are encoded either by 16×16 prediction modes or 4×4 prediction modes. The normalized histogram is generated as:

$$H_i(L)=\frac{H'_i(L)}{W_L} \qquad (2)$$

where H_i is the i-th normalized bin of the histogram, which is generated by dividing the i-th bin of histogram ($H'_i(L)$) by W_L-the total number of 4×4 blocks in the image. W_L is calculated as:

$$W_L = \text{(frame height} \times \text{frame width) / 16} \qquad (3)$$

Also there are four prediction modes for 8x8 Chroma components. One histogram is used for both color components because the prediction modes for them are the same. The normalized histogram for Chroma components is generated as:

$$H_i(C) = (\sum_{j=1}^{Wc} Mode.c_i(j))/W_c \qquad (4)$$

where $H_i(C)$ is the normalized bin i of the histogram and $i \in [0,...,3]$.

$$Mode.c_i = \begin{cases} 1 & \text{If jth of Intra\_8x8 equal i} \\ 0 & \text{Otherwise} \end{cases} \qquad (5)$$

W_c is the total number of Chroma components in the frame, and is generated as:

$$W_c = (\text{frame height} \times \text{frame width})/256 \qquad (6)$$

Therefore, each frame has a prediction histogram containing a Luma component with 13 bins and a Chroma component with four bins as a feature vector. Considering the higher sensitivity of human vision to the Luma rather than the Chroma components, two parameters-α and β-are used for the Luma and Chroma impacts on the final normalized histogram, respectively. Figure 5 indicates a normalized histogram for a video frame.

$$H_{img} = \begin{cases} \alpha * H_i(L) & 0 \le i \le 12 \\ \beta * H_{i-13}(C) & 13 \le i \le 16 \end{cases}, \alpha + \beta = 1 \qquad (7)$$

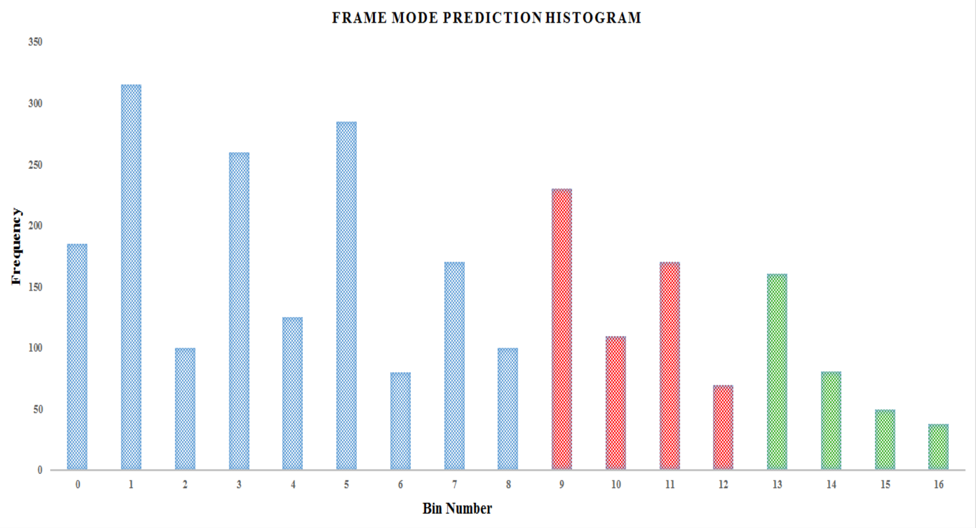


Figure 5. Frame feature vector - prediction modes histogram.

3.2. Frame grouping and key frame selection

The proposed method categorizes similar frames in one cluster and then selects the key frames from them. One of the challenges in unsupervised learning is that a number of clustered data are grouped. Although some techniques estimate the cluster numbers ([21], [22]) or analyze the cluster validity [35], they are in the pixel domain, and this makes the process computationally expensive. For example, in [22], the computation of the summaries takes around ten times the video length [10].

In this paper, a method is proposed for automatic video abstraction by estimating the number of clusters in the compressed domain. We applied the feature vector extracted with an adaptive threshold to compare video frames. Similarity measure is defined as the histogram intersection of two frames:

$$S_{pq} = \sum_{i=0}^{k} \{H_p \cap H_q\} = \sum_{i=0}^{k} \{\min(H_p(i), H_q(i))\} \qquad (8)$$

where $\mathbf{S_{pq}}$ is the histogram intersection that selects the minimum i-th bin from the two compared frames' features vector. In this way, an m x m matrix -m is the number of video frames-is generated whose items are the sum of minimum corresponding feature vector values. This matrix is diagonal symmetric, and each item is between 0 and 1 based on the normalized features vector.

$$\mathbf{S} = \begin{bmatrix} S_{11} \Lambda\ S_{1m} \\ M \\ S_{m1} \Lambda\ S_{mm} \end{bmatrix} \qquad (9)$$

$$0 < S_{pq} \le 1$$

If the similarity between two frames- $S_{pq}(p \ne q)$ –is less than a predefined threshold, a new cluster is added. The following algorithm is used to compute the number of clusters:

```
  Initialization
        • Number of clusters C = 1
  Loop to find number of clusters:
    • For p = 2 to MAX(I-frame)
        • For q = 1 to (p -1)
If (sum of intersections of frames p & q: $S_{pq} > \theta$ )
            • I-frame(p) ϵ cluster(C)
            • break
          End for
        If (I-frame(p) == null)
            • C = C + 1
            • I-frame(p) ϵ cluster(C)
    End for
```

We compared each frame with all the preceding frames. If any similarity occurs, the current frame belongs to the similar frame's cluster; otherwise, a new cluster is generated. Comparing a frame with all the precedent ones makes our proposed points near the clusters' center have higher membership grades. Clusters are identified with the similarity measure defined in (8). The frame with the highest membership degree in each cluster is selected as a key frame.

3.3. Redundant Key frames reduction and time sequence

The resulting Key frames are compared with each other to reduce the redundancy. The similarity measure is defined for the Key frames based on (8) as:

$$S_{kf1,kf2} = \sum_{i=0}^{k} \{H_{kf1} \cap H_{kf2}\} = \sum_{i=0}^{k} \{\min(H_{kf1}(i), H_{kf2}(i))\} \quad (10)$$

where $S_{kf1.kf2}$ is the histogram intersection that selects the minimum i-th bin from the two compared key frame feature vector.

If the similarity is more than a threshold, η, one of the compared key frames is removed from the final result. As the frame numbers are kept during the feature extraction phase, the final key frames are arranged based on their time sequence.

4. Results and discussion

In this work, the proposed video abstraction algorithm was implemented in the H.264 standard reference software. To evaluate it, two sets of experiments were performed:

method able to group similar frames in the whole set of I-frames rather than the consecutive frames. It is worth noting that comparing the consecutive frames in similar video frames' sequence like news videos generates the redundant clusters. In other words, the advantage of this method over other algorithms that are based on shots and consecutive frames' comparison is to create non-redundant clusters. Also on the fade, the threshold determination is too hard for shot detection.

In the clustering phase, we applied the estimated number of clusters and extracted a feature vector for each frame to group similar video frames. Since each frame can belong to a ratio to each cluster, we used fuzzy clustering. Fuzzy C-means is a data clustering algorithm that groups a set of data items into n clusters such that every data point in the dataset belongs to every cluster with a certain degree. Data

1) Experiments were carried out to show the proposed method's results and compared with other methods.
2) Experiments were carried out to demonstrate the parameters and sensitivity analysis.

The simulation tests were performed on the video dataset that has been made available by Avila *et al.* [10]. The user summaries for each video were also applied to evaluate the proposed method and to compare with other techniques. This video dataset contains 50 videos in different genres–documentary, educational, ephemeral, historical, and lecture-that were selected from the open video project (www.open-video.org). All videos are in the MPEG-1 format (30 fps, 352×240 pixels) colored with voice and duration between 1 to 4 min. First, all videos were transformed into the H264/AVC format with QP = 32 and low complexity features to apply the compressed domain features. Then we extracted the prediction modes' histogram. Table 1 indicates the timing specifications of experiments performed on an AMD X6 3.2 GHz, RAM 8 GB, with 64-bit OS. The timing results in table 1 indicate that the mode extraction time ratio (METR) is 8%, which means that the mode extraction time is insignificant compared to the total decoding time. Objective evaluation based on the user judgment for video abstract quality is used as an evaluation method. Users have selected the key frames, and these video abstracts are compared with the ones generated automatically by different techniques. Each key frame in the generated abstract is compared with the selected user key frames [10]. In our method, if the frames' histogram intersection is higher than a threshold, they are considered as similar. The similarity threshold was counted to be 0.9. The quality of the automatically generated abstract was measured by two metrics named Accuracy Rate (CUS_A) and Error Rate (CUS_E), proposed by Avila *et al.* [10] and defined as follow:

$$CUS_A = \frac{n_{mAGA}}{n_{US}} \quad , \quad CUS_E = \frac{n_{m'AGA}}{n_{US}} \qquad (11)$$

where:

n_{mAGA} is the number of matching key frames from the automatically generated abstract (AGA),

$n_{m'AGA}$ is the number of non-matching key frames from the automatically generated abstract,

and n_{US} is the number of key frames from user summary.

The highest quality of video abstract occurs when all key frames in the user summary and the automatically generated one are the same, and there is no unmatched key frame. In this case: $CUS_E = 0$, $CUS_A = 1$

4.1 Comparison with other techniques

We compared our method (AVAIF) with DT [21], OV [16], STIMO [24], VSUMM [10], and VSUKFE [20] on video database [10]. The parameters θ and η were considered as 0.9 and 0.93, respectively. To generate the normal histogram for each frame, α and β were selected as 0.8 and 0.2, respectively. Table 2 indicates the mean values for CUS_A and CUS_E for all methods. The results in table 2 show that the proposed method reached the highest Accuracy Rate as well as the lowest Error Rate. Although VSUMM has an equal Accuracy Rate, it still has a high Error Rate. DT generates much shorter summaries than the summaries produced by human users, and thus the DT summaries have a low Error Rate between all techniques except for our proposed method. Hence, AVAIF provides better results compared to other methods based on the user summaries.

Table 1. Timing and memory specifications of the processes performed in the proposed method.

Average Compression Time (per frame) sec	Average Decompression Time (per frame) sec	Average Mode Extraction Time (per frame) sec	Average Video Abstraction Time(per frame) sec	Average Compression Rate(per video)
1.3	0.05	0.004	0.025	% 69.8

Table 2. Comparing different methods' results.

	DT [21]	OV [16]	STIMO [24]	VSUMM [10]	VSUKFE [20]	AVAIF
CUS_A	0.53	0.7	0.72	0.85	0.8	0.85
CUS_E	0.29	0.57	0.58	0.38	0.32	0.22

Figure 8 shows the video ''The Voyage of the Lee, segment 05'' abstracts created by five users. Figure 9 shows the video key frames generated by different methods used in comparison. By considering figure 9, it can be found out that the highest accuracy rate ($CUS_A = 0.9$) is achieved by VSUKFE, VSUMM, and AVAIF but the proposed method has a lower Error Rate ($CUS_E = 0.33$) compared to VSUMM. Although in this video, VSUKFE has the best Error Rate compared to all other methods, the mean Error Rates in table 2 indicate that the proposed method has overall the best performance to minimize the error during the process.

Also it could be observed that STIMO achieved the Error Rate very close to our proposed method but its Accuracy Rate was significantly lower than that for our proposed method. Therefore, AVAIF generates the best video abstract using the compressed domain features compared to other methods based on the user summaries.

4.2 Parameter analysis

In the compressed domain video abstraction techniques, the robustness to video compression parameters is important as the final result may change by their variation. These parameters are based upon the proposed method and compressed domain standard.

4.2.1 Proposed method parameters

As mentioned in Section 3, there are used two different parameters in the proposed method: similarity and feature vector. The technique was applied to 20 different videos including 62,080 frames to select the optimal value for these parameters. Each video was tested for selecting the most appropriate similarity parameters (SPs)- θ and η– and feature vector parameters (FVPs)-α and β.

Figures 6 and 7 show the changes in SPs and FVPs averaged on 20 videos against F-Score, respectively. It was observed that the Precision and Recall were gradually increased by increasing the values for θ and α. It could also be observed that in θ = 0.9 and α = 0.8, the F-Score started decreasing. Therefore, it can be deduced that the optimal values for these parameters are the values where F-Score is maximum. Another test also showed that η had the same behavior as θ and since it required more precision for the similarity between key-frames, the value was a little more than θ.

4.2.2 Compressed domain parameters

The quantization parameter and the rate-distortion optimization method for intra-prediction are the robustness parameters for video compression. The quantization parameter is used for rate control of the compressed video. The higher this parameter, the lower is the output rate and the quality of the decoded video frames.

Also there are two methods to optimize the intra-frame prediction mode in the H.264 standard. The first method is to select the intra-prediction mode in order to minimize the residual values after compensation. The second one selects the intra-prediction mode that optimizes the rate-distortion (RD). We studied the robustness of the proposed method to these parameters as their variation may affect the intra-prediction modes, and consequently, the frames’ feature vector.

To analyze the effects of the parameters, the proposed method was applied to nine videos in different genres–3 videos from each documentary, educational, and lecture genres- containing 30912 frames entirely to test the performance.

The quantization parameter variation and encoding algorithm complexity variation were tested for each video. We analyzed the generated abstract from the original video by changing these parameters. Figure 10 illustrates the values for CUSA and CUSE based on the QP variation from 26 to 38 for each genre. The accuracy rate and error rate for three genres are indicated at the top and bottom parts of the graph, respectively. Figure 10 shows that by changing QP from 26 to 38, the accuracy rate varies at most 0.06, 0.05, and 0.08 for education, lecture, and documentary genres; and the maximum change of the error rates are 0.07, 0.05, and 0.08, correspondingly. Due to these values, it can be inferred that the proposed method has acceptable robustness to QP variation and also has satisfactory results even for higher quantization parameters.

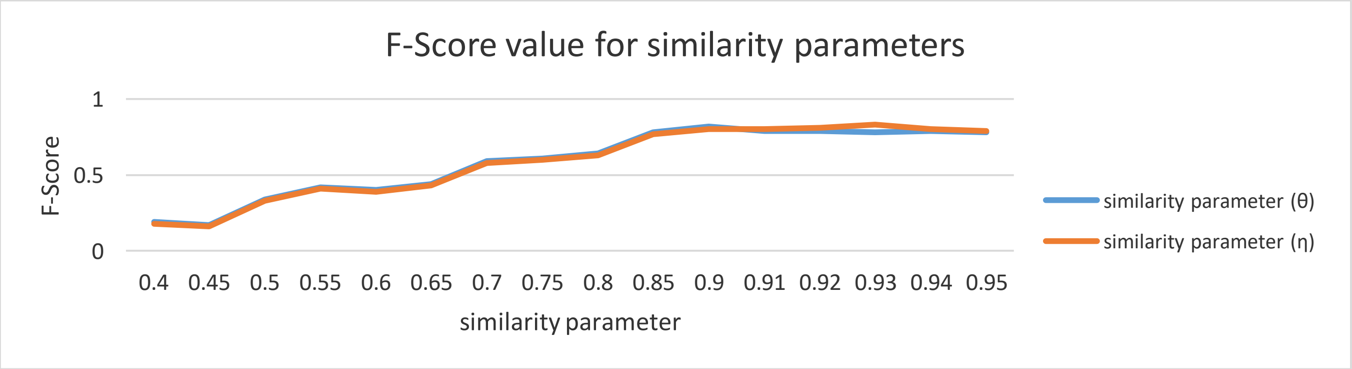


Figure 6. Average similarity parameter against F-Score.

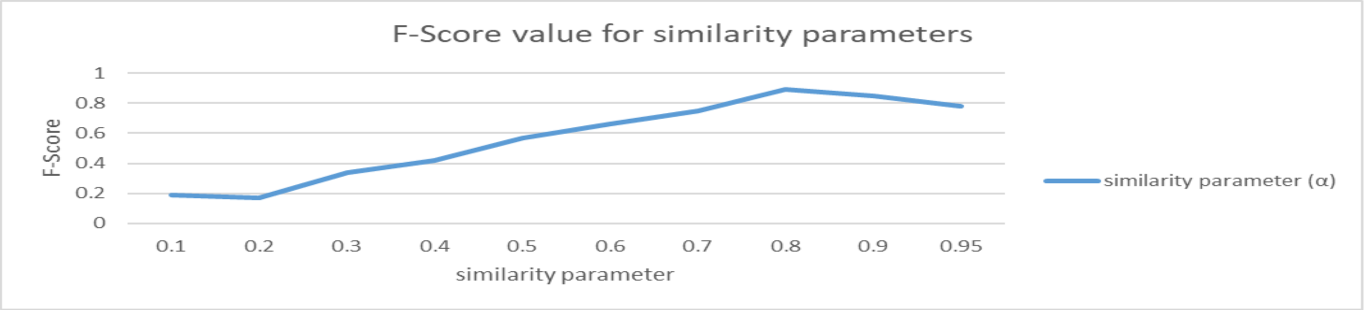
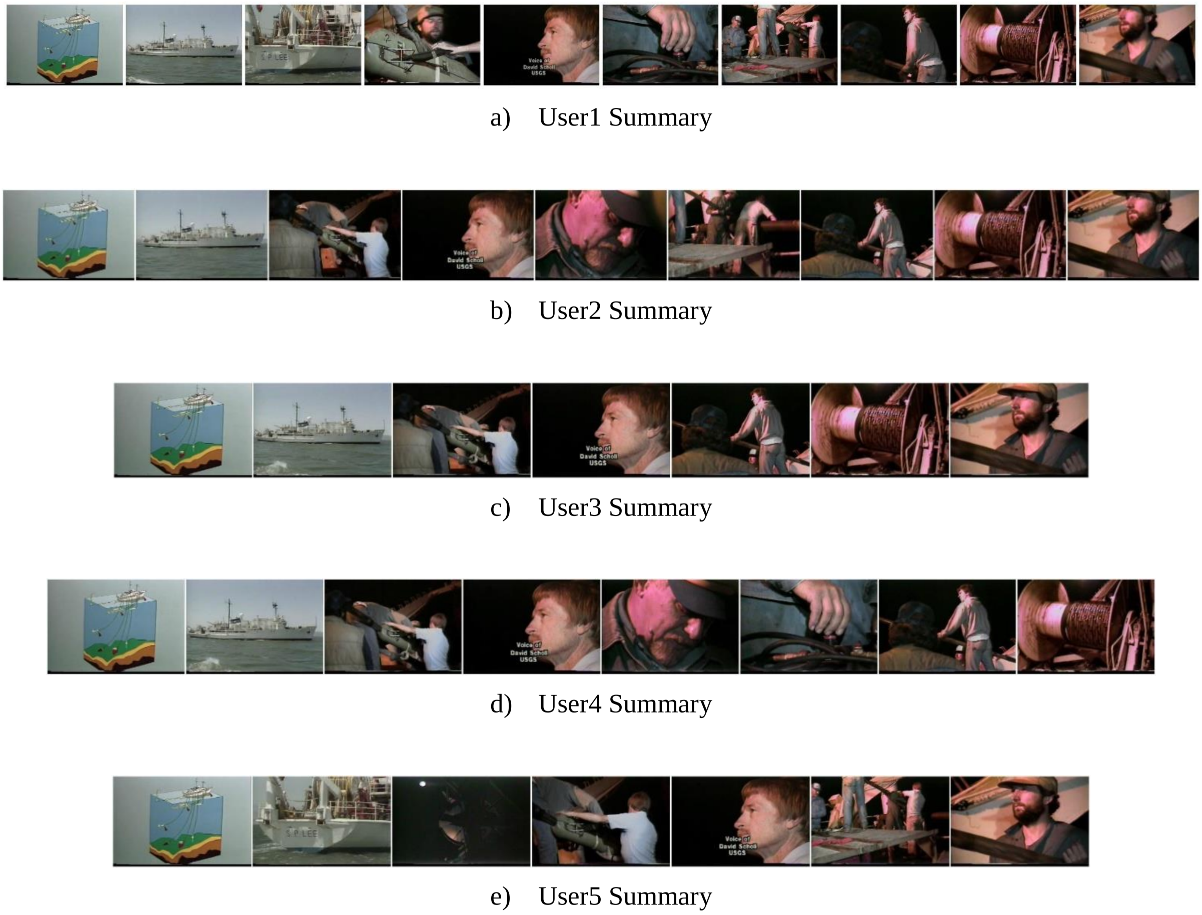


Figure 7. Average feature vector parameter against F-Score.

\



a) User1 Summary

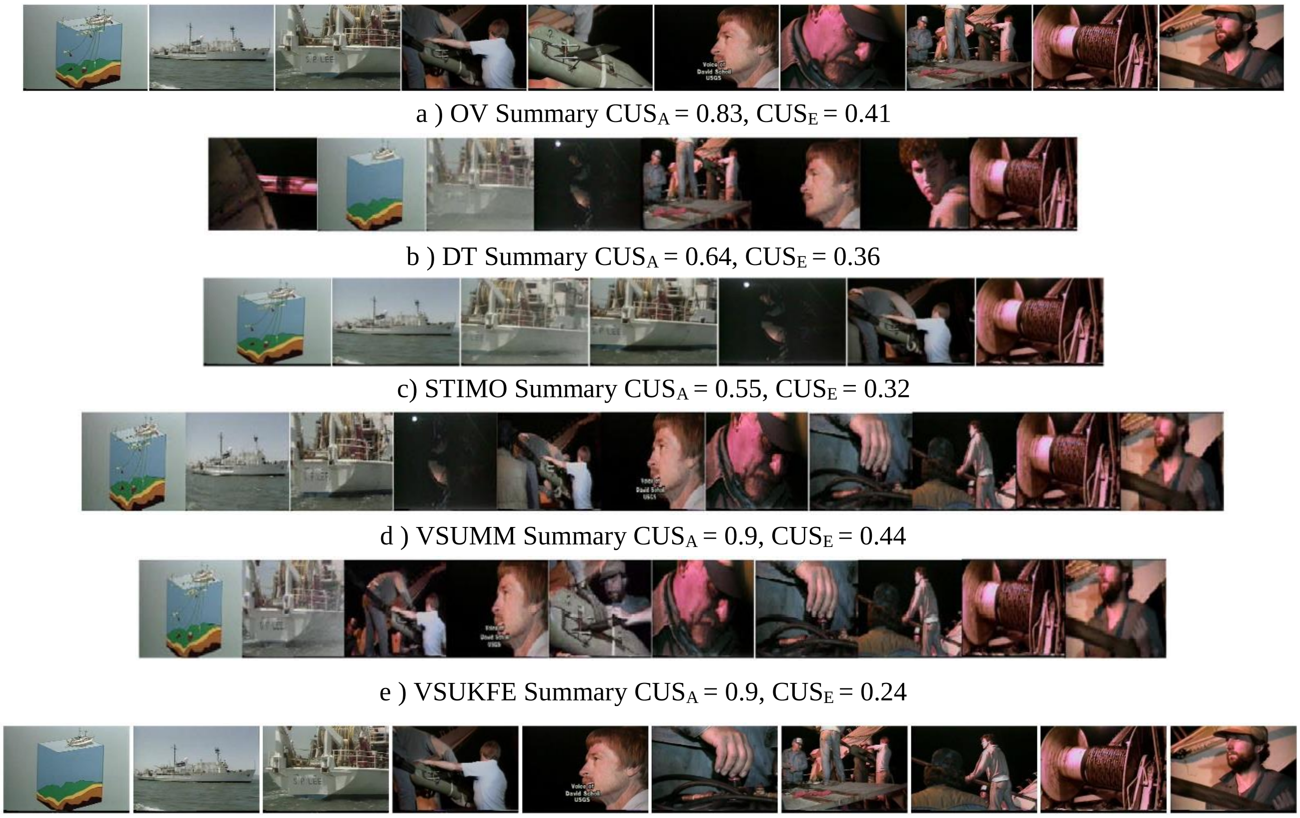
b) User2 Summary

c) User3 Summary

d) User4 Summary

e) User5 Summary

Figure 8. User Summaries of "The voyage of the lee, segment 05".



a) OV Summary $CUS_A = 0.83$, $CUS_E = 0.41$

b) DT Summary $CUS_A = 0.64$, $CUS_E = 0.36$

c) STIMO Summary $CUS_A = 0.55$, $CUS_E = 0.32$

d) VSUMM Summary $CUS_A = 0.9$, $CUS_E = 0.44$

e) VSUKFE Summary $CUS_A = 0.9$, $CUS_E = 0.24$

f) AVAIF Summary $CUS_A = 0.9$, $CUS_E = 0.33$

Figure 9. Generated key frames by various methods for the video "The voyage of the lee, segment 05".

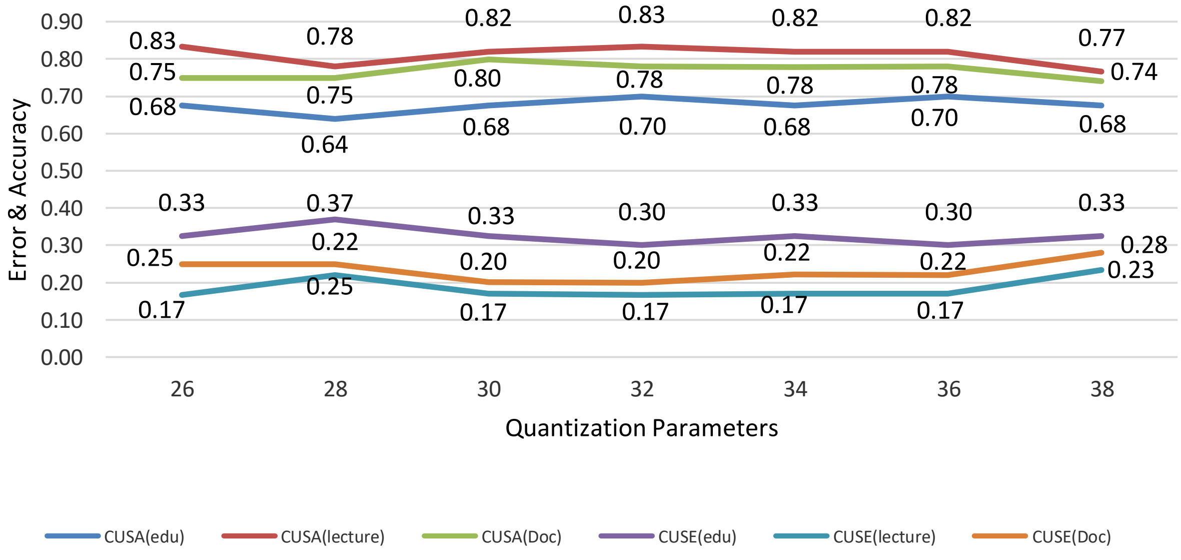


Figure 10. CUS_A and CUS_E variation based on QP changes.

Figures 11 to 13 illustrate the relation between the quality of the generated abstract and the complexity changes from QP = 26 to QP = 32. These graphs indicate that the maximum changes for CUS_A and CUS_E-between low and high complexity methods-are 0.05, 0.08 in education genre, 0.07, 0.07 in lecture genre, and 0.1, 0.07 in documentary genre. Although there are more differences in documentary genre, it is still acceptable. In addition, these graphs confirm that more accuracy rate and less error rate can be achieved in the low complexity encoding methods. Hence, the robustness of the proposed method to various coding parameters is acceptable.

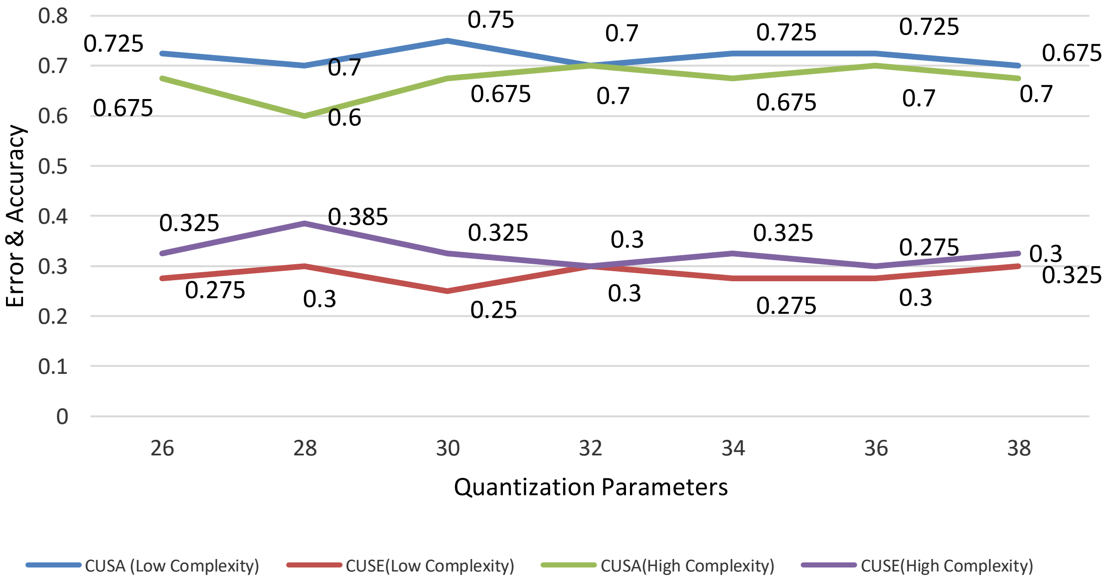


Figure 11. CUS$_A$ and CUS$_E$ variation based on encoding complexity changes in various QP (Educational).

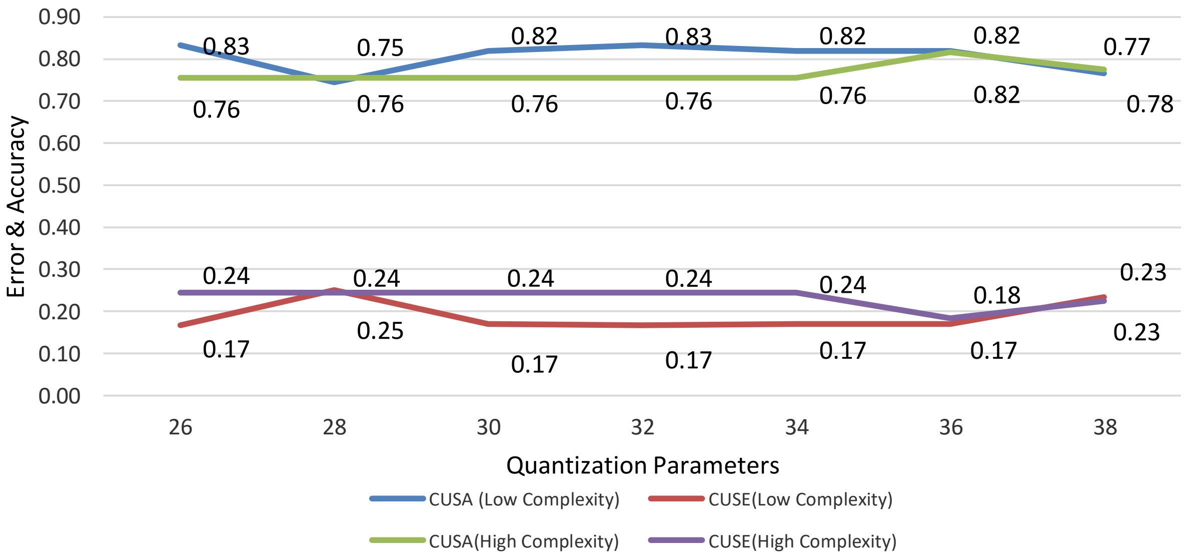


Figure 12. CUS$_A$ and CUS$_E$ variation based on encoding complexity changes in various QP (Lecture).

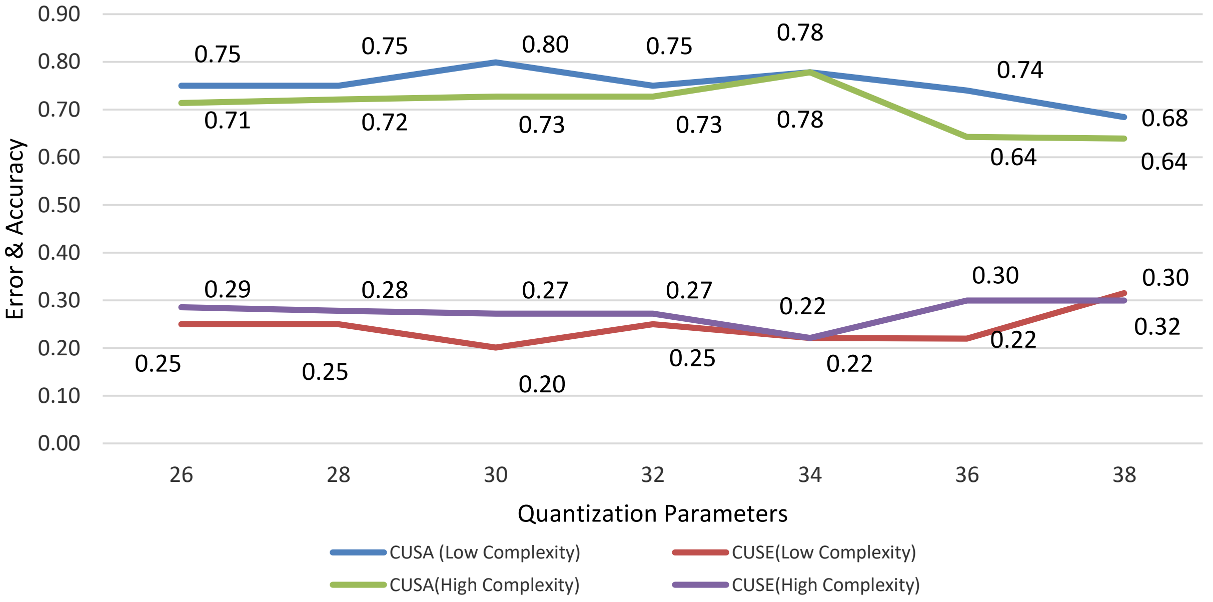


Figure 13. CUS_A and CUS_E variation based on encoding complexity changes in various QP (Documentary).

5. Conclusion

In this work, we proposed a new method, AVAIF, to generate automatic video abstracts by H.264/AVC I-frame coding, and employed a compressed domain feature extraction method and fuzzy clustering to determine the key frames. Feature extraction in the compressed domain was used to test the similarity between frames and calculate the number of clusters. Also using the membership degree in fuzzy clustering, we have more precision in the final key frames. The proposed method was compared with other techniques based on the "Comparison of User Summaries" (CUS) mechanism, proposed by Avila *et al.* [10]. The experimental results indicated that by using the proposed approach, we could achieve a higher Accuracy Rate as well as a lower Error Rate compared to other techniques, and generated video abstracts closer to the user summaries. Moreover, we employed an H.264/AVC compressed domain feature vector and fuzzy clustering, which had two significant advantages compared to the previous feature vectors. Firstly, it was in the compressed domain and avoided decompression time and storage of uncompressed video frames for feature extraction. Secondly, the evaluation results indicated a superior performance of the proposed method compared to the other methods. In the performance evaluation experiments, we achieved, on average, 85% accuracy rate and 22% average error rate in the compressed domain video abstraction. Also the analysis of coding parameters indicated that the proposed method had an acceptable robustness to various encoding parameters. As a result, the proposed approach can be considered as a valuable solution for video abstraction in the compressed domain avoiding the time and storage issues using the uncompressed domain techniques.

References

[1] Zhu, X., Wu, X., Fan, J., Elmagarmid, A. K., & Aref, W. G. (2004). Exploring video content structure for hierarchical summarization. Multimedia Systems, vol. 10, no. 2, pp. 98–115. http://dx.doi.org/10.1007/s00530-004-0142-7.

[2] Jeong, D., Yoo, H. J., & Cho, N. I. (2017). Open Access A static video summarization method based on the sparse coding of features and representativeness of frames. EURASIP Journal on Image and Video Processing, pp. 1–14. http://doi.org/10.1186/s13640-016-0122-9.

[3] Guan, G., Wang, Z., Lu, S., Deng, J. Da, & Feng, D. D. (2013). Keypoint-based key frame selection. IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 4, pp.729–734. http://dx.doi.org/10.1109/TCSVT.2012.2214871.

[4] Chinh Dang, & Radha, H. (2015). RPCA-KFE: Key Frame Extraction for Video Using Robust Principal Component Analysis. IEEE Transactions on Image Processing, vol.24, no. 11, pp.3742–3753. http://dx.doi.org/10.1109/TIP.2015.2445572.

[5] Yao, W., Li, Z., & Rahardja, S. (2012). Dynamic threshold-based key frame detection and its application in rate control. IET Image Processing, vol. 6, no. 7, pp. 986-995. http://dx.doi.org/10.1049/iet-ipr.2011.0189.

[6] De Bruyne, S., Van Deursen, D., De Cock, J., De Neve, W., Lambert, P., & Van de Walle, R. (2008). A

compressed-domain approach for shot boundary detection on H.264/AVC bit streams. Signal Processing: Image Communication, vol. 23, no. 7, pp. 473–489. http://dx.doi.org/10.1016/j.image.2008.04.012.

[7] Mohamadi, H., Shahbahrami, A., & Akbari, J. (2013). Image retrieval using the combination of text-based and content-based algorithms. Journal of AI and Data Mining (JAIDM), vol. 1, Issue 1, pp.27-34. http://dx.doi.org/10.22044/JADM.2013.113.

[8] Song, GH., Ji, Q. G., Lu, ZM., Fang, ZD., & Xie, ZH. (2014). A novel video abstraction method based on fast clustering of the regions of interest in key frames. International Journal of Electronics and Communications (AEÜ), vol. 68, pp. 237–243. http://dx.doi.org/10.1016/j.aeue.2014.03.004.

[9] Herranz, L., Martínez, José M. (2009). An Efficient Summarization Algorithm Based on Clustering and bitstream extraction. In Proceedings of International Conference on Multimedia and Expo, 2009. http://dx.doi.org/10.1109/ICME.2009.5202581.

[10] Avila, S. E. D., Lopes, A. P. B., Da Luz, A., Araújo, A.d.A. (2011). VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recognition Letters,vol. 32,no. 1,pp. 56–68. http://dx.doi.org/10.1016/j.patrec.2010.08.004.

[11] Truong, B. T., & Venkatesh, S. (2007). Video abstraction: A Systematic Review and Classification. ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 3, no. 1, pp. 1-37. http://dx.doi.org/10.1145/1198302.1198305.

[12] Money, A. G., & Agius, H. (2008). Video summarization: A conceptual framework and survey of the state of the art. Journal of Visual Communication and Image Representation, vol. 19, no. 2, pp. 121–143. http://dx.doi.org/10.1016/j.jvcir.2007.04.002.

[13] Jiang, R. M., Sadka, A. H., & Crookes, D. (2009). Advances in video summarization and skimming. Studies in Computational Intelligence, vol. 231, pp. 27–50. http://dx.doi.org/10.1007/978-3-642-02900-4_2.

[14] Yan, R., & Hauptmann, A. G. (2007). A review of text and image retrieval approaches for broadcast news video. Information Retrieval, vol. 10, no. (4–5), pp. 445–484. http://dx.doi.org/10.1007/s10791-007-9031-y.

[15] Sun, Z., Jia, K., & Chen, H. (2008). Video key frame extraction based on spatial-temporal color distribution. Proceedings - 2008 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2008, pp. 196–199. http://dx.doi.org/10.1109/IIH-MSP.2008.245.

[16] DeMenthon, D., Kobla, V., Doermann, D. (1998). Video summarization by curve simplification. Proceedings of the ACM International Conference on Multimedia, New York, USA, 1998, pp.211–218. http://dx.doi.org/10.1145/290747.290773.

[17] Kalaiselvi Geetha, M., & Palanivel, S. (2009). Video Classification and Shot Detection for Video Retrieval Applications. International Journal of Computational Intelligence Systems, vol. 2, no.1, pp.39–50. http://doi.org/10.1080/18756891.2009.9727638.

[18] Irtaza, A., Jaffar, M. A., & Aleisa, E. (2013). Correlated Networks for Content Based Image Retrieval. International Journal of Computational Intelligence Systems, vol. 6, no. 6, pp.1189–1205. http://doi.org/10.1080/18756891.2013.823005.

[19] Zhang, Q., Xue, X., Zhou, D., & Wei, X. (2014). Motion Key-frames extraction based on amplitude of distance characteristic curve. International Journal of Computational Intelligence Systems, vol. 7, no. 3, pp.506–514. http://doi.org/10.1080/18756891.2013.859873.

[20] Ejaz, N., Tariq, T. Bin, & Baik, S. W. (2012). Adaptive key frame extraction for video summarization using an aggregation mechanism. Journal of Visual Communication and Image Representation, vol. 23, no. 7, pp. 1031–1040. http://dx.doi.org/10.1016/j.jvcir.2012.06.013.

[21] Zhang, X. D., Liu, T. Y., Lo, K. T., & Feng, J. (2003). Dynamic selection and effective compression of key frames for video abstraction. Pattern Recognition Letters, vol. 24, no. (9–10), pp. 1523–1532. http://dx.doi.org/10.1016/S0167-8655(02)00391-4.

[22] Mundur, P., Rao, Y., Yesha, Y. (2006). Key-frame based video Summarization using Delaunay clustering. International Journal on Digital Libraries, vol. 6, no. 2, pp. 219–232. http://dx.doi.org/10.1007/s00799-005-0129-9.

[23] Chheng, T. (2007). Video Summarization Using Clustering. Department of Computer Science University of California 2007.

[24] Furini, M., Geraci, F., Montangero, M., & Pellegrini, M. (2010). STIMO: STIll and MOving video storyboard for the web scenario. Multimedia Tools and Applications, vol. 46, no. 1, pp. 47–69. http://dx.doi.org/10.1007/s11042-009-0307-7.

[25] Asadi, E., & Charkari, N. M. (2012). Video summarization using fuzzy c-means clustering. 20th Iranian Conference on Electrical Engineering (ICEE2012), 2012, pp. 690–694. http://dx.doi.org/10.1109/IranianCEE.2012.6292442.

[26] Khara, S., Modi, B., Shah, Darshil J., Thakkar, R. (2015). Video Summarization using clustering. International Journal of Innovative Research in Technology, vol. 2, no. 6, pp. 31-36.

[27] Xiang-wei, L., Li-dong, Z., & Kai, Z. (2012). Hierarchical Video Summarization Extraction Algorithm in Compressed Domain. Physics Procedia,

vol. 24, pp. 2360–2366. http://dx.doi.org/10.1016/j.phpro.2012.02.350.

[28] Almeida, J., Leite, N. J., & Torres, R. D. S. (2013). Online video summarization on compressed domain. Journal of Visual Communication and Image Representation, vol. 24, no. 6, pp.729–738. http://dx.doi.org/10.1016/j.jvcir.2012.01.009.

[29] Wu, J., Zhong, S. hua, Ma, Z., Heinen, S. J., & Jiang, J. (2018). Foveated convolutional neural networks for video summarization. Multimedia Tools and Applications, vol. 77, no. 22, pp. 29245–29267. http://doi.org/10.1007/s11042-018-5953-1.

[30] Rochan, M., Ye, L., & Wang, Y. (2018). Video summarization using fully convolutional sequence networks. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11216, LNCS, pp. 358–374. http://doi.org/10.1007/978-3-030-01258-8_22.

[31] Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J., & Yokoya, N. (2017). Video summarization using deep semantic features. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10115 LNCS, pp. 361–377. http://doi.org/10.1007/978-3-319-54193-8_23.

[32] Richardson, I. (2011). The H. 264 advanced video compression standard. Second Edition, Wiley Publishing.

[33] Zargari, F., Mehrabi, M., & Ghanbari, M. (2010). Compressed domain texture based visual information retrieval method for I-frame coded pictures. IEEE Transactions on Consumer Electronics, vol. 56, no. 2, pp. 728–736. http://dx.doi.org/10.1109/TCE.2010.5505994.

[34] Yamaghani, M., & Zargari, F. (2017). Classification and retrieval of radiology images in H.264/AVC compressed domain. Signal, Image and Video Processing, vol. 11, no. 3, pp. 573–580. http://doi.org/10.1007/s11760-016-0996-0.

[35] Hanjalic, A., Zhang, H., (1999). An Integrated Scheme for Automated Video abstraction based on unsupervised cluster-validity analysis. IEEE Transactions on Circuits and Systems, vol. 9, no. 8, pp.1280–1289. http://dx.doi.org/10.1109/76.80916.

IAIDM

نشریه هوش مصنوعی و داده کاوی

دانشگاه صنعتی شاهرود

چکیده سازی ویدئو در حوزه فشرده استاندارد کد گذاری H.264

علی رضا یمقانی[۱] و فرزاد زرگری[۲،*]

[۱] گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد علوم و تحقیقات، تهران، ایران.

[۲] دانشیار، پژوهشگاه ارتباطات و فناوری اطلاعات، تهران، ایران.

ارسال ۲۰۱۸/۱۲/۲۴؛ بازنگری ۲۰۱۹/۰۳/۰۸؛ پذیرش ۲۰۱۹/۰۴/۱۰

چکیده:

چکیده سازی ویدئو، امکان جستجو، کاوش و مشاهده یک ویدئو را تنها با دسترسی به محتویات موثر آن، فراهم می‌آورد. بیشتر تحقیقات در این زمینه مبتنی بر ویدئو های در حوزه پیکسل بوده که نیاز به پردازش کد گشایی دارند ودر نتیجه زمان، حافظه و پردازش بیشتری را طلب می‌کنند. در این مقاله، روش چکیده سازی جدیدی مبتنی بر حوزه فشرده H.264 ارائه شده است. این روش ، مبتنی بر هیستوگرام نرمال حالت‌های پیش بینی فریم‌های I از ویدئوی فشرده در استانداردH.264 ، می‌باشد. فاصله هر دو فریم، اشتراک هیستوگرام آن دو، به عنوان معیار شباهت، در نظر گرفته می‌شود. در ادامه با استفاده از خوشه بندی فازی، فریم‌های مشابه در کلاس‌های مشخصی تعیین شده و مجموعه نماینده‌های این کلاس‌ها به عنوان فریم‌های چکیده، در نظر گرفته می‌شوند. به منظور افزایش کیفیت چکیده نهایی، فریم‌های افزونه در آن با استفاده از معیار شباهت گفته شده در الگو ریتم، شناسایی شده و حذف می‌گردند. نتایج مقایسه این روش با سایر روش‌های چکیده سازی، نشان می‌دهد که روش پیشنهادی ضمن آن که در حوزه فشرده انجام می‌شود و چکیده سازی را بدون نیاز به کد گشایی کامل ویدئو انجام می‌دهد، به طور میانگین به ۸۵٪ نرخ صحت و ۲۲٪ نرخ خطا در چکیده سازی ویدئو های حوزه فشرده رسیده که مقادیر فوق از نتایج حاصل از روش‌های دیگر بهتر بوده و در نتیجه به چکیده های تولید شده کاربران نزدیک‌تر می‌باشد.

کلمات کلیدی: چکیده سازی ویدئو، خوشه بندی، هیستوگرام حالت‌های پیش بینی، ویدئو ی فشرده، استخراج فریم‌های کلیدی، بردار ویژگی حوزه فشرده.