Video summarisation using optimum global threshold technique based on genetic algorithm

Amit Phadikar*
Department of Information Technology,
MCKV Institute of Engineering,
Liluah, Howrah-711204, West Bengal, India
Email: amitphadikar@rediffmail.com
*Corresponding author

Nishant Kumar
Department of Computer Science,
Techxell School of Management and Technology,
Jaina More, Bokaro-829301, Jharkhand, India
Email: nishant89.sheo@gmail.com

Baisakhi Sur Phadikar
Cognizant Technology Solutions,
Salt Lake, Kolkata, India
Email: baisakhi.sur@gmail.com

Goutam Kumar Maity
Department of Electronics and Communication Engineering,
MCKV Institute of Engineering,
Liluah, Howrah, WB, India
Email: goutam123_2005@yahoo.co.in

Abstract: Most of the methods for video summarisation rely on complicated clustering algorithms that make them too computationally complex for real time applications. This paper presents an efficient approach for video summary generation that does not rely on complex clustering algorithms and does not require frame length as a parameter. The present scheme combines colour histogram and edge histogram features with optimum global thresholding to detect key frames. The optimum threshold is selected based on genetic algorithm (GA) to increase the performance of the proposed system. For each shot, key frames are extracted and similar key frames are eliminated. Experimental results duly support those claims.

Keywords: YCbCr colour space; colour histogram; edge histogram; genetic algorithm.


Biographical notes: Amit Phadikar received his PhD (Engineering) from Bengal Engineering and Science University, Shibpur, India. Currently, he is working as an Associate Professor in the Department of Information Technology, MCKV Institute of Engineering, Liluah, Howrah, West Bengal, India. He has contributed about 50 research papers and two books.

Nishant Kumar is an Assistant Professor at Department of Computer Science at Techxell School of Management and Technology, Bokaro, Jharkhand, India. His area of research interests are image and video processing.

Baisakhi Sur Phadikar received his BTech in Computer Science and Engineering in 2005 from West Bengal University of Technology, W.B., Kolkata, India. Currently, she is working as a Sr. Associate at Cognizant Technology Solutions, Salt Lake, Kolkata, India.
1 Introduction

Enormous popularity of the internet video repository sites like YouTube or Yahoo video caused increasing amount of video content available over the internet. In such a scenario, it is necessary to have efficient tools that allow fast video browsing. In such application, compact representation of video data is necessary. Such representation provides the user with information about the content of the particular sequence being examined. In other words, those tools should provide concise representation of the video content as a sequence of still or moving pictures, i.e., video summary. Video summary is the abstract of an entire video. It is the essence of the entire video provided in a shorter period of time. The main purpose of video summary is viewing time constraints. Video summarisation plays a major role where the resources like storage, communication bandwidth and power are limited. It has several applications in security, military, data hiding (Iliyasu et al., 2013) and even in entertainment domains (Asadi and Charkari, 2012).

Video summarisation can be carried out in different methods. Each method is suitable in its own domain and can thus give variable results based on a number of parameters. There are two main categories of video summarisation (Truong and Venkatesh, 2007), i.e., static video summary and dynamic video skimming. In static video summary methods, a number of representative frames, often called key frames, are selected from the source video sequence and are presented to the viewer. Dynamic video summary methods generate a new, much shorter video sequence from the source video. Since static video summaries are the most common technique used in practical video browsing applications, we focused our research on static video summarisation.

Most of the existing works on static video summarisation are performed by clustering similar frames and selecting representatives per clusters (Mundur et al., 2006; Hadi et al., 2006; De Avila et al., 2011; Furini et al., 2007; Furini et al., 2010). A variety of clustering algorithms are applied such as: Delaunay triangulation (Mundur et al., 2006), K-medoids (Hadi et al., 2006), K-means (De Avila et al., 2011), furthest point first (Furini et al., 2007) and STIMO (Furini et al., 2010), etc. Although they produce acceptable visual quality, the most of these methods rely on complicated clustering algorithms, applied directly on features extracted from sampled frames. It makes them too computationally complex for real-time applications. Another restriction of these approaches is that they require the number of clusters, i.e., representative frames to be set prior.

A simple approach to extract key frames from shot is based on frame content change computed by features, such as colour histogram (Zhang et al., 1997) or motion activity (Wolf, 1996), etc. Zhu et al. (2004) proposed a hierarchical video summarisation strategy that explores video content structure to provide the users with a scalable, multilevel video summary. In those above methods, a predefined threshold is required to control the number of key frames. Zeinalpour et al. (2009) proposed a novel method for video summarisation using genetic algorithm (GA) (Sarkar and Chakraborty, 2011) and information theory. The scheme relies on the mutual information for video summarisation. This is due to the fact that mutual information provides better results as it extracts the inter-frame information. Doulamis et al. (2000) proposed a video summarisation scheme based on fuzzy representation of visual content. In particular, a multidimensional fuzzy histogram is constructed for each video frame based on a collection of appropriate features, extracted using video sequence analysis. Then key frames are selected optimally by minimising a cross correlation criterion. Chiu et al. (2000) described a genetic segmentation algorithm for video summarisation. For evaluating segmentations, authors define a similarity adjacency function, which are extremely expensive to optimise by traditional methods. Moreover, the evolutionary nature of GA offers a further advantage by enabling incremental segmentation. Yang and Wei (2011) proposed a GA-based video summarisation scheme for soccer video. The scheme first introduces audio features to improve fitness function which is used to calculate the relative differences among all the selected frames. Then the scheme employs crossover and mutation operators to get the meaningful summary in a video search space. Yang and Wei (2012) proposed a video summarisation method based on GA employing crossover and mutation operators to search for a meaningful summary in a video search space. Authors investigate both binary and decimal GA. It is seen that the binary GA finds the optimal result more quickly and easily than the decimal GA. Sony et al. (2011) use Euclidean distance after clustering to obtain summarised frames. This method is based on the removal of redundant frames from a video and returns the user defined number of unique frames. Visually similar looking frames are clustered into one group using Euclidean distance. After the clusters are formed, the frames that have larger distance metric are retrieved from each group to form a sequence.
Guang-sheng (2008) proposed a novel algorithm for shot boundary detection and key frames extraction based on image segmentation and attention model. Matching difference between two consecutive frames is computed with different weights and shot boundaries are detected based on automatic thresholding technique. Then key frame is extracted using reference frame-based approach. Thakar et al. (2012) proposed a scheme based on \( \chi^2 \) histogram which is effective in detecting abrupt and gradual transitions. Moreover, a new pixel intensity-based method is also proposed for detecting fade transition. Angadi and Naik (2012) proposed a shot boundary detection technique based on local colour moments in YCbCr colour space. By the use of YCbCr colour space, the influence of illumination change and shadows are reduced. Dhgadi and Deshmukh (2012) proposed a new approach for key frame extraction based on block Histogram difference and edge matching rate. The scheme provides global information about the video content. Moreover, the scheme is faster without any performance degradations. Eom and Choe (2005) proposed a fast edge histogram generation technique in discrete cosine transform (DCT) domain based on the properties of AC coefficients. First, they have verified the meaning of two AC coefficients. Second, they have measured the edge orientation using the ratio of the two AC coefficients. Finally, the edge histogram is generated similar to the edge histogram descriptor (EHD) defined by MPEG7 standard. The proposed method is compared with the EHD in complexity, and is found to be slightly superior.

The objective of this paper is to design a fast and effective approach for video summary generation that does not rely on complicated clustering algorithms and also does not require length (number of summary frames) as a parameter. The proposed model is based upon colour and edge histogram in YCbCr colour space with optimum global thresholding to detect key frames. The optimum threshold is selected based on GA (Poli, 1996; Piszcz and Soule, 2007) to increase the performance of the system. Experimental results show that there is an improvement of 2.88% in precision rate and 12.39% in recall rate due to the use of GA in the proposed scheme.

The rest of the paper is outlined as: in Section 2, colour space and edge histogram have been discussed. Section 3 discusses the proposed work. Performance evaluation is discussed in Section 4. Finally, Section 5 discusses the conclusion and scope of future works.

## 2 Colour space and edge histogram

Frame feature extraction is a crucial part of any key frame extraction algorithm which directly affects the performances of the algorithm. The visual feature which is optimal for video processing applications should satisfy several main requirements. Those are robustness, discriminability, compactness, and low complexity (Cvetkovic et al., 2013). The present scheme uses colour histogram and edge histogram descriptors as those features are characterised by low computational complexity, very compact representation and invariance to resolution changes.

### 2.1 YCbCr colour space

The colour histograms have been commonly used for key frame extraction in frame difference-based techniques and also used for image retrieval (Liu et al., 2013). This is because the colour is one of the most important visual features to describe an image (Liu et al., 2013). Colour histograms are easy to compute and are robust in case of small camera motions (Rajendra and Keshaveni, 2014). It has been observed in the literature that YCbCr colour space always yields better result as compared to other colour space in case of key frame detection (Mishra and Subban, 2014). That is why, the present scheme uses YCbCr colour space. Moreover, by the use of YCbCr colour space, the influence of illumination change and shadows are also to be reduced (Angadi and Naik 2012). The difference between YCbCr and RGB is that YCbCr represents colour as brightness and two colour difference signals, while RGB represents colour as red, green and blue. In YCbCr, the \( Y \) is the brightness (luma), \( C_b \) is blue minus luma (B-Y) and \( C_r \) is red minus luma (R-Y). This colour space exploits the properties of the human eye. The eye is more sensitive to light intensity changes and less sensitive to hue changes. When the amount of information is to be minimised, the intensity component can be stored with higher accuracy than the \( C_b \) and \( C_r \) components. The joint photographers engineering group (JPEG) file format makes use of this colour space to throw away unimportant information (Kekre et al., 2012). In this paper, \( Y \) component is used for edge histogram feature and the colour component, i.e., \( C_b \) and \( C_r \) are used for colour histogram feature. RGB images can be converted to YCbCr colour space using equation (1).

\[
\begin{bmatrix}
Y \\
C_b \\
C_r
\end{bmatrix} =
\begin{bmatrix}
0.2989 & 0.5866 & 0.1145 \\
-0.1688 & -0.3312 & 0.5000 \\
0.5000 & -0.4184 & -0.0816
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]

### 2.2 Edge histogram

Edge detection is one of the most commonly used operations in image analysis. Edges define the boundaries between regions in an image, which helps in segmentation (Seixas et al., 2009) and object recognition. The edge histogram is used to match the edges of adjacent frames to eliminate redundant frames (Rajendra and Keshaveni, 2014). Edge detection operators that are commonly used are viz. Roberts operator, Cannny operator, Sobel operator, Prewitt operator and the Laplace operator (Dhgadi and Deshmukh, 2012), etc. To find the edge histogram, the image \( f(x, y) \) is first divided into \( (4 \times 4) \) subimages. The present scheme uses Cannny edge detector. It finds edges based on the local maxima of the gradient of image \( f(x, y) \). The gradient is calculated using the derivative of the Gaussian filter. The image is smoothed using a Gaussian filter with a
specified standard deviation, to reduce noise. To generate the histogram, edges in the sub-images are categorised into five types; vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional edges. Since there are 16 subimages, a total of \((16 \times 5) = 80\) histogram bins are required (Chang et al., 2001; Manjunath et al., 2002).

**Figure 1** Flowchart of the proposed scheme
Algorithm 1  Colour histogram

Input: Extracted frames of input video.
Output: Euclidean distances (EDC_b and EDC_r) between two consecutive frames for each component C_b and C_r.

Begin
1  For 1st frame do
2    Convert the frame from RGB to YC_bCr colour space using equation (1).
3    Calculate the normalised histogram of each component, i.e., C_b and C_r using equation (2).
   \[ P_i = \text{imhist}(f, b) / \text{numel}(f) \]  
where, the symbol ‘i’ is the frame number, ‘f’ is the colour difference components, i.e., ‘C_b’ and ‘C_r’, ‘b’ is the number of bins used in forming the histogram (b = 256 for 8-bit grey scale image), and numel(f) is the number of components in array ‘f’ (i.e., the number of pixels in the frame). Finally, the function \text{imhist()} calculate the histogram of frame ‘f’ with ‘b’ bins.
4  End for
5  For second frame onwards do
6    Steps 2 and 3.
7    Euclidean distance (EDC_b and EDC_r) between two normalised histogram using equation (3) for C_b and equation (4) for C_r, respectively, for two consecutive frames.
   \[ EDC_b = \sum ((P_{oldframe,C_b} - P_{newframe,C_b})^2)^{1/2} \]  
   \[ EDC_r = \sum ((P_{oldframe,C_r} - P_{newframe,C_r})^2)^{1/2} \]  
The symbol EDC_b stores Euclidean distances between two consecutive frames of ‘C_b’ feature. EDC_r stores Euclidean distances between two consecutive frames of ‘C_r’ feature. The symbol P_{oldframe} represents the first frame and P_{newframe} represents the consecutive second frame.
8  Assign the value of P_{newframe} in P_{oldframe}, i.e.,
   \[ P_{oldframe} = P_{newframe} \]
9  Store the values of EDC_b and EDC_r.
10  End for

End

3 Overview of the proposed method

We propose an approach which is based on several efficient video processing procedures. At first, video frames are sampled in order to reduce further computational burden. Then, colour and edge histogram features are extracted on pre-sampled video frames and Euclidean distance measure is used to evaluate the similarity between the frames. Those features are deployed for key frames detection using a thresholding approach. In the present scheme, the optimum threshold is selected based on GA and key frame is said to be detected at places where the frame difference is maximal and larger than the global threshold. Then, representative key frames are extracted and similar key frames are eliminated. Figure 1 depicts the flow chart of the proposed scheme. In the rest, detail description of the proposed method is presented.

Step 1  Frame extraction: depending on properties of video, it may have 24–30 frames per second. First, frames are extracted from the video. Extracted frames generally contain redundant frames (information).

Step 2  YC_bC_r colour space: frames extracted in Step 1 are converted in YC_bC_r colour space using equation (1).

Step 3  Frame feature extraction: frame feature extraction is a crucial part of any key frame extraction algorithm. It directly affects the performances of the proposed scheme. In this paper, we have used two features, i.e., colour and edge. This is due to the fact that several methods for retrieving images on the basis of colour feature have been described in the literature. Colour feature is the easy and simple to compute. The colour histogram is one of the most commonly used features for video summarisation as it is invariant to scaling and rotation. Colour histogram of frames in the C_b (chrominance of blue), and C_r (chrominance of red) colour space are calculated. Colour histogram is very effective, for classification of frames based on colour. Algorithm 1 shows the steps to find the colour histogram and Algorithm 2 shows the steps to find the edge histogram.
Algorithm 2  Edge histogram

**Input:** Extracted frames of input video.

**Output:** Euclidean distances (EDY) between two consecutive frames for component Y.

**Begin**

1. **For** 1st frame **do**
2. Convert the frame from RGB to YC_{b}C_{r} using equation (1).
3. Store the component(Y), i.e., luminance information.
4. Split the image into (4 × 4) non-overlapping rectangular region.
5. In each region, a (1 × 5) edge histogram is computed (horizontal, vertical, 2 diagonal and 1 non-directional)
   Say, variable E1 contains 80 histogram bins.
6. **End for**

7. **For** second frame onwards **do**
8. Repeat Steps 2, 3 and 4.
9. In each region, a (1 × 5) edge histogram is computed (horizontal, vertical, 2-diagonal and 1 non-directional).
   Say, variable E2 contains 80 histogram bins.
10. Calculate the Euclidean distance (EDY) between the two edge histogram using equation (5).
    \[
    EDY = \sqrt{\sum ((E1 - E2)^2)}
    \] (5)
    The symbol EDY represents the Euclidean distance of Y component between two consecutive frames.
11. **Assign** the value of E2 in E1, i.e., E1 = E2.
12. Store the value of EDY.
13. **End for**

**End**

**Step 4  Optimised threshold selection using GA:** Key frames are identified based on the visual content change. Therefore, the most critical activity in the key frame detection process is the selection of the thresholds. In this paper, global thresholding is used for detecting key frames using GA. We have used precision (P) and recall (R) to develop the objective function, as both are ranging from 0 (zero) to 100 and one does not bias the other. The objective is to maximise the fitness function. The normalised fitness function ‘f_{n}’ is then defined as:
    \[
    f_{n} = \frac{P + R}{2}
    \] (6)
    The precision (P) is defined as the ratio of correctly detected key frame to the sum of correctly detected and falsely detected key frame of a video data. The recall (R) is defined as the ratio of detected key frame to the sum of detected and undetected key frame.
    \[
    P = \frac{\text{Number of relevant frames retrieved}}{\text{Total number of frames retrieved}}
    \] (7)
    \[
    R = \frac{\text{Number of relevant frames retrieved}}{\text{Total number of frames retrieved}}
    \] (8)
    Algorithm 3 shows the steps for threshold selection using GA.

**Step 5  Detection of key frames:** The proposed model is based on colour histogram and edge histogram feature. Given a video which contains many frames, the colour histogram and edge histogram for each frame is computed and the Euclidean distance measure is used to measure the dissimilarities between the frames, based on the threshold that is selected by GA. A key frame is said to be detected if the dissimilarity between the frames is higher than the threshold value. Algorithm 4 shows the steps for key frames detection.

**4 Performance evaluation**

This section presents the results of the experiments conducted to confirm the success of the proposed model. The experimentation is conducted on set of YouTube-videos and the open video project video. The detail descriptions of these videos are provided in Table 1. They vary in duration from 901 frames to 2,938 frames. In this experiment, all the test videos are abrupt transition videos and Figure 2 shows the frames of one of the test video. The experiments are conducted in Pentium IV, 2.80 GHz processor with 512 MB RAM using MATLAB 7.
Algorithm 3  Genetic algorithm

**Input:** Euclidean distances (EDCb, EDCr and EDY) between two consecutive frames for components Cb, Cr and Y from Algorithm 1 and Algorithm 2, respectively.

**Output:** Global thresholds (TY, TCb and TCr) for each components, i.e., Y, Cb and Cr.

**Begin**

1. Formulate initial population randomly. Initial population contains information regarding initial value of TY, TCb and TCr that are randomly selected and within the range [0 1].
2. **Repeat**
3. Evaluate the objective function (f_n) as defined in equation (6).
4. Apply genetic operators
   5. Selection
   6. Crossover
   7. Mutation
5. **Until** stopping criteria (i.e., optimum solution is found).
6. Optimum solution set, i.e., TY, TCb and TCr are obtained, where, TY is the threshold for Y component; TCb is the threshold for Cb component; TCr is the threshold for Cr component.

**End**

Algorithm 4  Key frame detection

**Input:** Euclidean distances (EDY, EDCb and EDCr) between two consecutive frames for each components Y, Cb and Cr from Algorithm 2 and Algorithm 1, respectively, and Global thresholds (TY, TCb and TCr) of each components Y, Cb and Cr from Algorithm 3.

**Output:** Key frames

**Begin**

1. If EDCb>TCb and EDCr>TCr and EDY>TY then
2. Select the frame as key frame.
3. **End if**

**End**

### Table 1  Description of video used for evaluation

<table>
<thead>
<tr>
<th>Video name</th>
<th>No. of key frames</th>
<th>Key frame numbers</th>
<th>Total number of frames</th>
<th>Indoor/outdoor</th>
<th>Camera motion</th>
<th>Persp. changes</th>
<th>Bright. changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video 1</td>
<td>8</td>
<td>1, 126, 214, 519, 539, 625, 738</td>
<td>901</td>
<td>Outdoor</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Video 2</td>
<td>20</td>
<td>1, 132, 189, 254, 276, 299, 456, 491, 534, 778, 989, 1,243, 1,439, 1,587, 1,745, 1,952, 2,136, 2,467, 2,654, 2,879</td>
<td>2,908</td>
<td>Outdoor</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Video 3</td>
<td>18</td>
<td>1, 89, 187, 376, 573, 623, 811, 934, 1,089, 1,365, 1,524, 1,679, 1,821, 1,956, 2,298, 2,445, 2,786, 2,911</td>
<td>2,938</td>
<td>Outdoor</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

4.1  The gold standard

The results of the proposed method are compared with the ground truth agreed by multiple human judges. The goals of creating the ground truth are to:

1. create a reference database of video
2. identify a foundation by which automated algorithms can be used for comparison.

To establish the ground truth, human judges were asked to independently surf the video and provide the key frames. The key frames estimated by the judges were reviewed in a group meeting with a final judge to derive final key frames for each of the video. In practical scenario where the ground truth is not available, the number of key frames will be defined by users.
Figure 3 represents the component wise (component of YCbCr colour space, i.e., Y, C_b and C_r) Euclidean distance between the successive frames. Horizontal axis represents the number of frames where as vertical axis represents the normalised frame distance. Frames at high distance are tagged as key frames. Figure 4(a) shows the best fitness at each generation. Figure 4(b) shows the average distance between individuals at each generation, which is a good measure of the diversity of a population in GA based threshold detection.

Figure 5 shows the key frames extracted by our proposed algorithm. Figures 6(a) and 6(b) shows the summarised video for Video 2 and Video 3, respectively. From Figures 5 and 6 it is seen that the scheme can effectively select the key frames which ultimately returns good results of video summarisation. The results indicate that our approach is among the best with respect to the ground truth included in the video summary. Table 2 shows the performance of the proposed scheme in term of precision and recall. It is seen that the precision and recall values are quite large for the proposed scheme. This is due to the use of GA to find the optimum threshold of the proposed scheme. Table 3 shows the performance if only colour feature and un-optimised global threshold are used for video summarisation. It is seen that there is an improvement of 2.88% in precision rate and 12.39% in recall rate due to the combined use of colour and edge histogram and GA in the proposed scheme.

In Table 4, we have compared the proposed work with the existing works found in the literature. Table 4 shows that the scheme offers good results than the others. This is due to the use of GA to find optimum value of threshold for key frame selection. The schemes described in Thakar et al. (2012), Angadi and Naik (2012) and Dhagdi and Deshmukh (2012) use only one type of frame feature like $\chi^2$ histogram, colour moments and edge matching information, respectively for key frame detection. Those single features alone are not sufficient to maintain all the requirements as described earlier, i.e., robustness, discriminability, compactness, and low complexity. So those schemes return poor performance in term of R and P. Moreover, none of the schemes use GA to select the optimum threshold for identifying key frames. This is another reason for the poor value of P and R than the proposed scheme.

5 Conclusions

In this paper, we have proposed an efficient method for video summary generation based on GA. Experimental results on standard YouTube video and on the open video project video show that proposed scheme offers satisfactory performance for key frame detection in term of recall (R) and precision rate (P). However, the proposed scheme is only concentrated on the abrupt transitions of videos and not for gradual shot. Abrupt shot boundaries are created by simply attaching a shot to another. On the other hand, gradual transition results from the editing effects applied to the shots during attachment operation. Depending on the different editing effect gradual transitions can be further divided into different sub-types, i.e., dissolve, fade (fade in, fade out) and wipes. The limitation of our proposed method is that it does not support gradual shot of video. But most of the videos contain gradual transition.
**Figure 3** Normalised frame distances between successive frames (Video 1) (see online version for colours)
Figure 4  Best fitness and distance plot (Video 1)

![Best fitness and distance plot for Video 1](image1)

(a) Best fitness

(b) Average Distance between Individuals

Figure 5  Summarised video frames (Video 1) (see online version for colours)

Note: The algorithm shows eight frames, i.e., frame 1, 126, 214, 354, 519, 539, 625, 738.

Table 2  Performance of the proposed scheme with colour histogram, edge histogram and GA

<table>
<thead>
<tr>
<th>Size</th>
<th>No. of frames tested</th>
<th>Performance of proposed work (%)</th>
<th>( \text{Precision} )</th>
<th>( \text{Recall} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video 1</td>
<td>25 MB</td>
<td>901</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Video 2</td>
<td>4.83 MB</td>
<td>2,908</td>
<td>90</td>
<td>95</td>
</tr>
<tr>
<td>Video 3</td>
<td>14.26 MB</td>
<td>2,938</td>
<td>95</td>
<td>96</td>
</tr>
</tbody>
</table>

Table 3  Performance measure with only colour histogram and without GA

<table>
<thead>
<tr>
<th>Size</th>
<th>No. of frames tested</th>
<th>Performance measure (%)</th>
<th>( \text{Precision} )</th>
<th>( \text{Recall} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video 1</td>
<td>25 MB</td>
<td>901</td>
<td>95.72</td>
<td>80.00</td>
</tr>
<tr>
<td>Video 2</td>
<td>4.83 MB</td>
<td>2,908</td>
<td>89.90</td>
<td>87.10</td>
</tr>
<tr>
<td>Video 3</td>
<td>14.26 MB</td>
<td>2,938</td>
<td>91.40</td>
<td>91.80</td>
</tr>
</tbody>
</table>

Table 4  Comparison of proposed and existing work

<table>
<thead>
<tr>
<th></th>
<th>( \text{R (in %)} )</th>
<th>( \text{P (in %)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>97</td>
<td>95</td>
</tr>
<tr>
<td>( B )</td>
<td>90</td>
<td>92</td>
</tr>
<tr>
<td>( C )</td>
<td>88</td>
<td>75</td>
</tr>
<tr>
<td>( D )</td>
<td>95</td>
<td>91</td>
</tr>
<tr>
<td>( E )</td>
<td>93</td>
<td>92</td>
</tr>
</tbody>
</table>

Source:
A: Proposed scheme,  
B: Guang-sheng (2008),  
C: Thakar et al. (2012),  
D: Angadi and Naik (2012),  
E: Dhagdi and Deshmukh (2012)

Future work can be concentrated for further performance improvement of the proposed scheme by selecting adaptive threshold based on GA and combined use of motion, edge and colour to increase the efficiency of key frame detection and also dealing with gradual transitions of videos such as dissolve, fades (fade in, fade out), and wipes.
Video summarisation using optimum global threshold technique based on genetic algorithm

Figure 6
Preview of generated summaries, (a) Video 2 (the algorithm shows 20 frames, i.e., frame 1, 132, 191*, 254, 276, 299, 456, 491, 534, 778, 989, 1,243, 1,439, 1,587, 1,745, 1,952, 2,136, 2,467, 2,654, 2,879) (b) Video 3 (the algorithm shows 18 frames, i.e., frame 1, 89, 187, 376, 573, 623, 811, 934, 1,089, 1,365, 1,524, 1,679, 1,821, 1,956, 2,299*, 2,445, 2,786, 2,911) (see online version for colours)

Note: The symbol * indicates the disparity in frame numbers between the proposed algorithm and the ground truth.

References


