Int J Multimed Info Retr (2016) 5:89–104
DOI 10.1007/s13735-016-0095-6

REGULAR PAPER

An efficient method for video shot boundary detection and keyframe extraction using SIFT-point distribution histogram

Rachida Hannane[1] · **Abdessamad Elboushaki**[1] · **Karim Afdel**[1] · **P. Naghabhushan**[2] · **Mohammed Javed**[2]

Received: 19 November 2015 / Revised: 17 February 2016 / Accepted: 28 February 2016 / Published online: 16 March 2016

Abstract In today's digital era, there are large volumes of long-duration videos resulting from movies, documentaries, sports and surveillance cameras floating over internet and video databases (YouTube). Since manual processing of these videos are difficult, time-consuming and expensive, an automatic technique of abstracting these long-duration videos are very much desirable. In this backdrop, this paper presents a novel and efficient approach of video shot boundary detection and keyframe extraction, which subsequently leads to a summarized and compact video. The proposed method detects video shot boundaries by extracting the SIFT-point distribution histogram (SIFT-PDH) from the frames as a combination of local and global features. In the subsequent step, using the distance of SIFT-PDH of consecutive frames and an adaptive threshold video shot boundaries are detected. Further, the keyframes representing the salient content of each segmented shot are extracted using entropy-based singular values measure. Thus, the summarized video is then generated by combining the extracted keyframes. The experimental results show that our method can efficiently detect shot boundaries under both abrupt and gradual transitions, and even under different levels of illumination, motion effects and camera operations (zoom in, zoom out and camera rotation). With the proposed method, the computational complexity is comparatively less and video summarization is very compact.

✉ Rachida Hannane
rachida.hanane08@gmail.com

[1] Laboratory of Computer Systems and Vision, Faculty of Science, Ibn Zohr University, Agadir 80000, Morocco

[2] Department of Studies in Computer Science, Mysore University, Mysore 570006, India

Keywords Video summarization · Video segmentation · keyframe extraction · Scale invariant feature transform (SIFT) · Point distribution histogram

1 Introduction

The advances in multimedia and communication technologies have made vast amounts of videos data available (example of long-duration movies, continuously generated videos from surveillance cameras and so on). Owing to the complexity in manipulating these large video data, limited memory size and also required time for watching the entire video to know its contents, video abstraction and summarization are needed to overcome these difficulties. The meaning of video summarization refers to the process of recapitulating and summarizing a video [1]. This process is achieved by producing an abstract of the salient keyframes that could cover the overall content of the video so that the viewer could quickly process and browse the video by viewing only the collage of this few highlighted frames and without wasting memory to store redundant data of the video. However, an efficient video summarization requires an efficient video segmentation and keyframes extraction; these two mechanisms represent the bases of video summarization. The goal of video segmentation is to segment or partition video into a set of meaningful and manageable segments, called shots. A shot refers to a sequence of frames captured from a unique and continuous record from a camera. Two transition types are distinguished while moving through two consecutive shots [2]. The first transition which is simplest one is called abrupt or cut transition, known also as a camera break, in which a transition between different shots is made over a single frame. The second type is called gradual transition including fade-in, fade-out, dissolve and so on. This type of transition is

more sophisticated and involves much more gradual changes between consecutive frames than an abrupt transition does. The entire shot can be mapped into a single representative frame, called keyframe. Uses of keyframes reduce the amount of storage data and also the time browsing required for a video summarization.

In this paper, we suggest a novel, fast and precise video segmentation and keyframe extraction approach for an efficient video summarization. The proposed method for video segmentation is based on the combination of the local and global features of the video frame. The local features are represented by a set of key locations of stable points extracted using difference of Gaussian [3] so that the segmentation can proceed successfully despite changes in scale (zoom in, zoom out), illumination, noise and distortions. The video frame is globally represented by a histogram, named SIFT-point distribution histogram (SIFT-PDH). This histogram describes the distribution of the extracted stable keypoints within the frame under polar coordinates. The difference between each two consecutive frames of the video, we call it distance comparison, is computed by comparing their SIFT-PDHs. This difference is based on bin to bin comparison (block to block comparison) instead of pixel to pixel comparison so that object motion will not affect the efficiency of the shot segmentation process. Therefore, the distance comparison between extracted SIFT-PDH of the consecutive frames that is above an adaptive threshold is considered as a shot boundary. However, the keyframe representing the salient information of the segmented shot is extracted by selecting the frame holding more information which is measured by the entropy based on the singular values of the frame.

The rest of the paper is organized as follows: In Sect. 2, we review some related works in shot boundary detection and keyframe extraction. Section 3 describes the proposed model for video summarization. Section 4 mentions the validation process used for keyframe extraction. Section 5 reports the experimental results of our proposed model. The last Sect. 6 concludes the paper.

2 Related work

A wide number of research efforts have been made generally in video summarization and specifically in the areas of video segmentation and keyframes extraction.

Basically, for video segmentation, the simplest way to test whether two frames are notably and meaningfully different is by comparing directly the pixels; if the number of different pixels in the consecutive frames is large enough, then these two frames belong to different shots. In [4], the video was segmented into shots according to video content by employing histogram-based approach with the use of histogram intersection and nonuniform partitioning and weighting, whilst [5] proposed a shot boundary detection based on color space by analyzing brightness and calculating frame difference with improved histogram weighted partition method. Janwe and Bhoyar [6] employed just noticeable difference JND color histogram model and computed the degree of similarity where cuts and gradual transitions were detected using an adaptive threshold based on sliding window. Gunal et al. [7] proposed a method for detecting gradual shot changes using fractal dimension information of gray scale video frames. In [8], shot boundary detection was performed by analyzing the color histogram differences with an adaptive threshold based on sliding window to detect cut transition; for gradual transition, a preprocessing and an automatic threshold with reference to the variation of histogram differences was selected to quantify local histogram value into binary value. SVD can be also utilized to detect shot boundaries as in [9] where they performed SVD on the frame feature matrices that were formed from the HSV color histogram extracted for all frames in each candidate segment; then a pattern matching method based on a similarity measurement was used to identify cut and gradual transition. Shot boundary was also detected using k-means clustering method by [10], where the features of color were extracted and the video frames were divided into several different sub-clusters through performing k-means clustering and the cut and gradual shot were detected by the adaptive double threshold of different sub-clusters.Moreover, by using local keypoints matching as in [11], both abrupt and gradual transition were detected without modeling different kinds of transaction. In [12], the shot boundaries were detected on-the-fly in video sequence using a sliding window mechanism where an automatic video frames clustering was performed using a graph based technique called “modularity”. Unfortunately, these above mentioned methods for video shot boundary detection are either computationally expensive or extremely sensitive to local motion, camera motion, scale variance (zoom in/out), noise sensitivity and illumination changes since they capture all details of the frame.

There are a few attempts reported in the literature related to keyframe extraction. According to [13] keyframe based video summarization can be achieved in three different ways: the first method of keyframe extraction based on sampling chooses keyframes uniformly or randomly under sampling without considering the video content. The second method of keyframe extraction based on scene segmentation extracts keyframes using scenes detection; the scene includes all parts with a semantic link in the video or in the same space or in the same time. The third method of keyframe extraction based on shot segmentation extracts adapted keyframes to video content; they extract either the first image as shot keyframe or the first and the last frames of the shot. An additional method for keyframe extraction was proposed by [4], where within each segmented shot the keyframes were determined with the cal-

culation of image entropy of every frame in HSV color space, whilst [14] select the first and last frame as keyframes and then the other keyframes are extracted using motion attention model, These keyframes are then clustered and a priority value is computed by estimating motion energy and color variation of shots. Another approach for keyframe extraction in video summarization is proposed by [15] where the complexity of the sequences in terms of changes in visual content are expressed by different frame descriptors and keyframes are extracted by detecting curvature points within the curve of the cumulative frame differences. The approach proposed in [16] is based on Faber Shauder discrete wavelet transform (FSDWT) and singular value decomposition (SVD). Their method extracts the block dominant image features of each video frame and constructs a 2D feature matrix, and then it factorizes the matrix using SVD. Finally keyframes are extracted based on the traced rank. Unfortunately, these methods may provide on the one hand a video summary holding some redundancy of keyframes with similar content which will lead to inaccurate video abstraction; on the other hand, it will not take into account the temporal position of the frames. Another drawback of these methods is that the produced summary will not provide an adequate representation of shot with strong movements.

From the above discussed methods, it is clear that several techniques have been proposed in the area of video summarization; most of these techniques have been achieved to a certain extent based on a certain assumptions and limitations. In spite of that success, it is still a big challenging problem to segment video and extract keyframes for an efficient video summarization in the midst of complication caused by camera operations, object motions, illumination changes and also high computational cost. Therefore, we proposed in this paper a novel approach to detect video shot boundaries using not the entire information of the frames, but only the relevant keypoints extracted using difference of Gaussian. We measure the entropy based on the singular values of each frame in the segmented shot to extract the frame holding more information as a keyframe of the shot.

3 Proposed model

An overview of our proposed system to segment and extract keyframes for a video summarization is shown in Fig. 1. Generally, the main challenge in this area is to develop a robust technique against illumination changes, object motion, camera operation (zoom in, zoom out, camera rotation), distortion and addition of noise, which often cause false detection on shot transition and emersion of redundant or missed keyframes. Therefore, the features based on the video segmentation should be robust against the aforementioned problems. For this reason, we have used difference of Gaussian by SIFT algorithm [3] to extract stable keypoints of the video frames. Moreover, our proposed model is based on two main stages. In the first stage, video stream is segmented into a set of shots by detecting the boundary (start frame, end frame) of each shot. In the second stage, keyframes are extracted using entropy-based singular values.

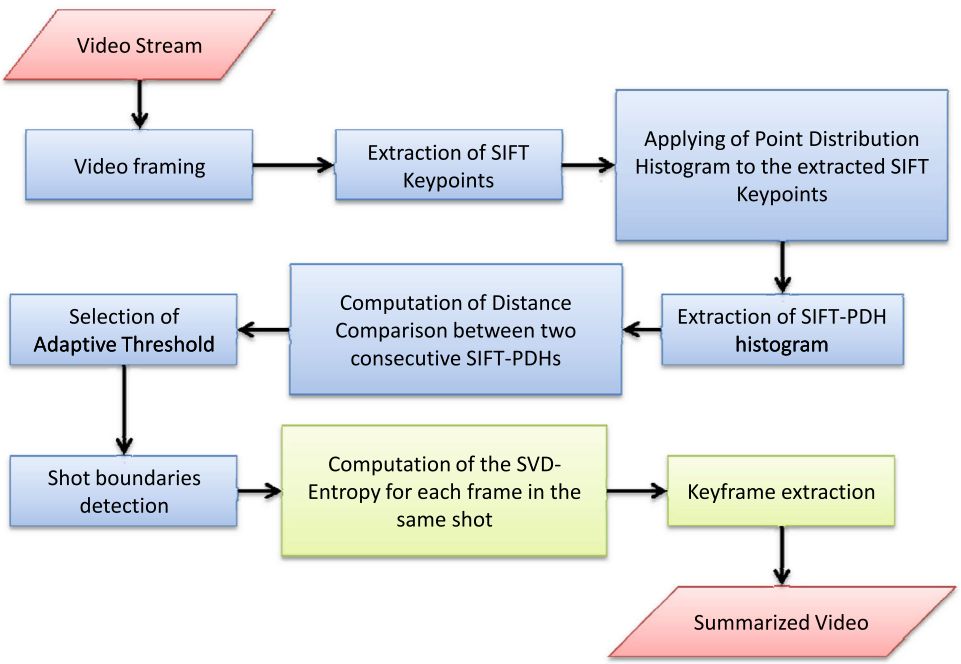


Fig. 1 Proposed system for video summarization by detecting shot boundaries and extracting keyframes

3.1 Video shot boundary detection

The main steps involved in our work to detect shot boundaries are presented as follows:

3.1.1 SIFT and edge-SIFT keypoints extraction

The video shot boundary detection approach presented in this paper is based on the extraction of SIFT keypoints and edge-SIFT keypoints from the video frame. The SIFT algorithm [3,17,18] starts by constructing a scale space. Key locations are defined as maxima and minima of the result of difference of Gaussian function applied in scale space to a series of smoothed and resampled images, namely each pixel in each scale is compared to its 26 neighbors in 3×3 regions at the current and adjacent scales (marked maxima or minima). Scale space extrema detection produces too many keypoint candidates, some of which are unstable. The next step in the algorithm is to perform a detailed fit to the nearby data for accurate location and scale. To accomplish this, all keypoints that have low contrast and are sensitive to the noise will be discarded using the approach of Taylor expansion. Figure 2 illustrates an example applied on a movie frame from dataset-2 in which the unstable keypoints are rejected using Taylor Expansion. For the experiments in [17], all peaks with a value less than 0.03 will be rejected.

SIFT keypoints located on edges of the frame's objects are significantly important in the video information; and due to its robustness to the illumination and scale variances, we will keep these keypoints as edge-SIFT keypoints using 2×2 Hessian matrix H to compute the principal curvatures which will be large across the edges.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}, \tag{1}$$

where D is the difference of Gaussian function.

To check that the ratio of principal curvatures is above some threshold r, we only need to check

$$\frac{\mathrm{Tr}(H)^2}{\mathrm{Det}(H)} \geq \frac{(r+1)^2}{r}, \tag{2}$$

where $\mathrm{Tr}(H)$ is the trace of H and $\mathrm{Det}(H)$ is the determinant of H.

In our case, we keep both the keypoints that have the ratio less than $r = 10$ as SIFT keypoints and the others having the ratio greater than $r = 10$ as edge-SIFT keypoints. Figure 3b shows an example of the extracted SIFT keypoints which are presented on blue color and edge-SIFT keypoints that are presented by red color. Regarding different parameters of SIFT algorithm, a large number of octaves or scale levels increase the number of SIFT keypoints, but makes the boundaries between shots less distinct. Therefore, the empirical data used in this work can be summarized as number of octaves $o = 5$, number of scale levels $s = 5$ and parameter of Gaussian function $\sigma = 1.6$.

3.1.2 SIFT-point distribution histogram extraction

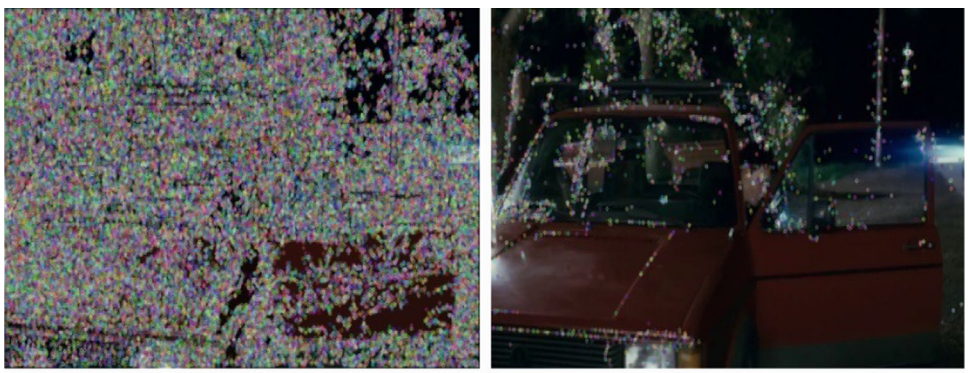
After detecting SIFT and edge-SIFT keypoints from the video frame, a histogram, named SIFT-point distribution histogram (SIFT-PDH), is extracted. The SIFT-PDH describes the distribution of SIFT and edge-SIFT keypoints within the video frame under a polar coordinate. The utilization of polar coordinate in this work will reduce the effect of camera rotation. The SIFT-PDH algorithm uses as inputs the coordinates of both of the extracted SIFT and edge-SIFT keypoints according to x-axis and y-axis.

$$P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \; (x_i, y_i) \in R^2 \tag{3}$$

Next the centroid $C = (x_c, y_c)$ of the frame F is computed using the following equation:

$$x_c = \frac{F_{\text{width}}}{2} \quad \text{and} \quad y_c = \frac{F_{\text{height}}}{2} \tag{4}$$

After setting the centroid as the origin, we then translate P into polar coordinates



(a) **(b)**

Fig. 2 Sample of a movie frame from dataset-2 where the unstable SIFT and edge-SIFT keypoints are rejected. **a** Before rejecting unstable keypoints (frame with 71,508 keypoints). **b** After rejecting unstable keypoints (frame with 3430 keypoints)

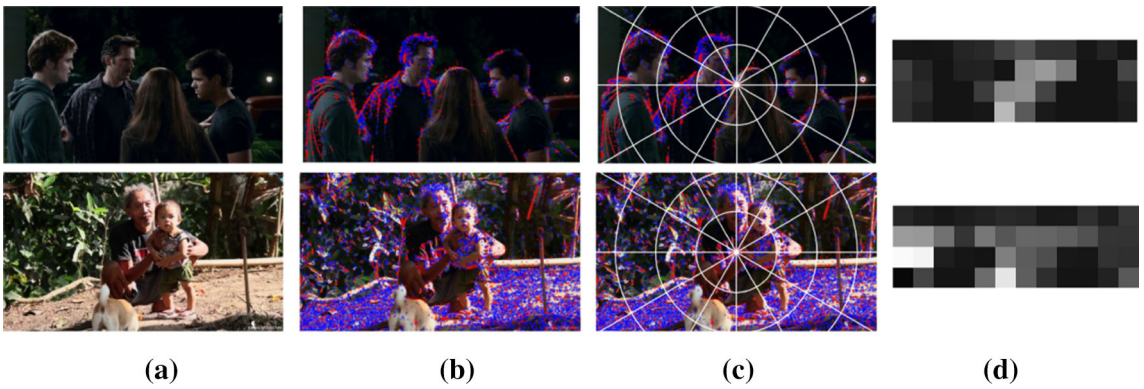


Fig. 3 Extraction process of SIFT-PDH histogram. **a** Example of two frames from different environments with different light conditions where the top row contains a frame extracted from movie video and the bottom row contains a frame extracted from documentary video both from dataset-2. **b** Extraction of SIFT keypoints (represented in *blue color*) and edge-SIFT keypoints (represented in *red color*). **c** Applying of point distribution histogram to the extracted SIFT and edge-SIFT keypoints (example of 4×12 bins). **d** Extraction of SIFT-PDH histogram

$$P = \{(r_1, \theta_1), (r_2, \theta_2), \ldots, (r_n, \theta_n)\} \quad (r_i, \theta_i) \in R^2, \tag{5}$$

where

$$r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{6}$$

is the distance between SIFT/edge-SIFT keypoint (x_i, y_i) and the centroid (x_c, y_c).

$$\theta_i = \arctan\left(\frac{(y_i - y_c)}{(x_i - x_c)}\right) \tag{7}$$

represents the orientation of the keypoint (x_i, y_i) relative to the centroid (x_c, y_c). Thereafter, we get the minimum circumscribed circle C with the centroid (x_c, y_c) and the radius $\rho_{\max}$, where

$$\rho_{\max} = \sqrt{\left(\frac{F_{\text{width}}}{2}\right)^2 + \left(\frac{F_{\text{height}}}{2}\right)^2} \tag{8}$$

We partition the area C into $u \times v$ bins where u indicates the number of circles and v denotes the number of angles starting from x-axis ($\varphi = 0$). Finally, the SIFT-PDH histogram is constructed by counting the number of SIFT keypoints and edge-SIFT keypoints which are located in every bin $n \times m$ using the following formula:

$$\text{SIFTPDH}(n.m)_{\substack{0 \le n < u \\ 0 \le m < v}} = \left\| \left\{ \begin{array}{c} (r_i, \theta_i) | n.\left(\frac{\rho_{\max}}{u}\right) \le r_i < (n+1).\left(\frac{\rho_{\max}}{u}\right) \\ \text{and} \\ \varphi + m.\left(\frac{2\pi}{v}\right) \le \theta_i < \varphi + (m+1).\left(\frac{2\pi}{v}\right) \end{array} \right\} \right\| \tag{9}$$

In this paper, 4×12 , 5×12 , 4×16 and 5×16 SIFT-PDH histograms are extracted, and the experimental results are compared. Figure 3 shows an example of 4×12 SIFT-PDH histogram process extracted from two frames belonging to different videos from dataset-2 .

3.1.3 Distance comparison histogram (DCH) of SIFT-PDH

The efficient way to detect a quantitative change between a pair of frames is to compare the salient features of these two frames, which can overcome effects of environmental factors like scale variance, illumination change, distortion and video motion. Therefore, the SIFT-PDH histogram is used as a feature to achieve this efficient comparison. The objective behind this comparison is that each two consecutive frames belonging to the same shots, having unchanging background and unchanging objects, will produce a little difference in SIFT and edge-SIFT keypoints locations, resulting in a slight difference of their SIFT-PDH histograms. Figure 4 illustrates the difference between extracted SIFT-PDH histograms from frames of two consecutive shots of a movie video in dataset-2, namely all frames of shot_i as well as shot_{i+1} having, respectively, almost similar SIFT-PDH histograms and unlike the frames belonging to different shots, (example of frame-6 and frame-7 which are consecutive) having dissimilar SIFT-PDH histograms.

Formally, let F_k and F_{k+1} be the SIFT-PDH histograms of two consecutive frames. The distance comparison between F_k and F_{k+1} denoted $\text{Dist}(F_k, F_{k+1})$ is computed using the following formula:

$$\text{Dist}(F_k, F_{k+1}) = \sqrt{\sum_{i=1}^{u}\sum_{j=1}^{v}(F_k(i,j) - F_{k+1}(i,j))^2}, \tag{10}$$

Fig. 4 A set of 12 consecutive frames from movie video in dataset-2 where each frame is represented by its SIFT-PDH, a significant variation while comparing each two consecutive frames is observed during the transition from frame-6 to frame-7 which is marked as end shot$_i$ and start shot$_{i+1}$

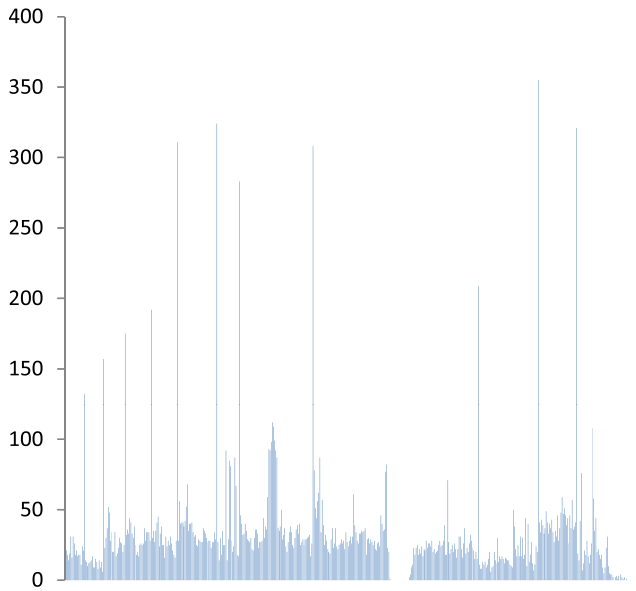


Fig. 5 The distance comparison histogram extracted for an advertising video from dataset-2

where u is the number of bins for $\rho_{\max}$ and v is the number of bins for $\theta = 2\pi$ ($\rho_{\max}$ and θ are the parameters of the SIFT-PDH histogram).

The distance comparison of each two consecutive frames of the entire video stream is computed as shown in Fig. 5, where it is clearly observed that the distance comparison between two consecutive frames belonging to the different shots produces high peaks. However, the low peaks refer to the consecutive frames having approximately similar content and belonging to the same shot. Therefore, an adaptive threshold is required to identify these high peaks as video shot boundaries.

3.1.4 Selection of adaptive threshold

In order to achieve high accuracy in video partitioning, we need to find adaptive and appropriate threshold values which present a key issue in video segmentation. When there is no camera shot change or high speedy camera movement in a video sequence, the frame to frame distance comparison value can only be due to the noise. Therefore, the distribution of distance comparison of two consecutive frames can be decomposed into a sum of two parts: the Gaussian noises and the differences produced by camera shot change and speedy camera movement.

Formally, let σ be the standard deviation and μ the mean of the DCH. The distance comparison of two consecutive frames where there is no transition will fall in the range of 0 to T_{DCH}, where T_{DCH} can be expressed as follows:

$$T_{\mathrm{DCH}} = \mu + \alpha\sigma \tag{11}$$

Hence, the distance comparison values that are beyond this threshold can be considered as an indicator of shot boundaries. From our experiments, the value α should be chosen

between 0.5 and 1.0. Figure 6 shows a sequence of the distance comparison values from an advertising video of dataset-2, in which shot boundaries are clearly observed as the peaks above the threshold T_{DCH}.

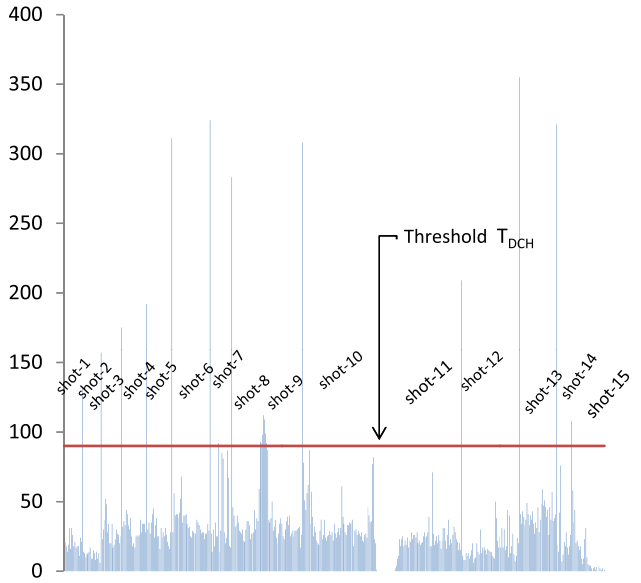


Fig. 6 Selection of adaptive threshold for the previous DCH of Fig. 5 has clearly detected all abrupt transition and some of gradual transition

3.1.5 Gradual transition detection

Gradual transitions are generally more difficult to be detected than abrupt transition, due to camera and object motion while moving from shot to shot. However, using SIFT-PDH approach which combines both local and global features of the frames, we overcome this difficulty and then detect the different types of the gradual transition.

A dissolve in video sequence is a shot transition where the first shot gradually disappears (fades out) while the second shot gradually appears (fades in). Typically, these fade out and fade in begin at the same time and overlap two shots across a sequence of frames starting by clearly appearing end-frame of the first shot and ending by clearly appearing start-frame of the second shot, Fig. 7 shows an example of the dissolve transition. Furthermore, this sequence of frames F will hold overlapping information of both consecutive shots, which will lead to an increasing number of SIFT keypoints in the frames and then enrolling a set of peaks S_p that is above the threshold T_{DCH} in the DCH, Fig. 8 shows a zoom on the portion of the DCH that is illustrated in Fig. 6 where the dissolve appears. Generally, this set of peaks S_p exceeds the threshold T_{DCH} in an increasing order followed by a decreasing order of the SIFT keypoints in the histogram which results in a concave curve; the highest peak $S_{\max}$ in this case represents the beginning of the curve bending. The main idea of this method is that the first peak on the set S_p presents the frame in which the dissolve starts (D_s) and the last peak on

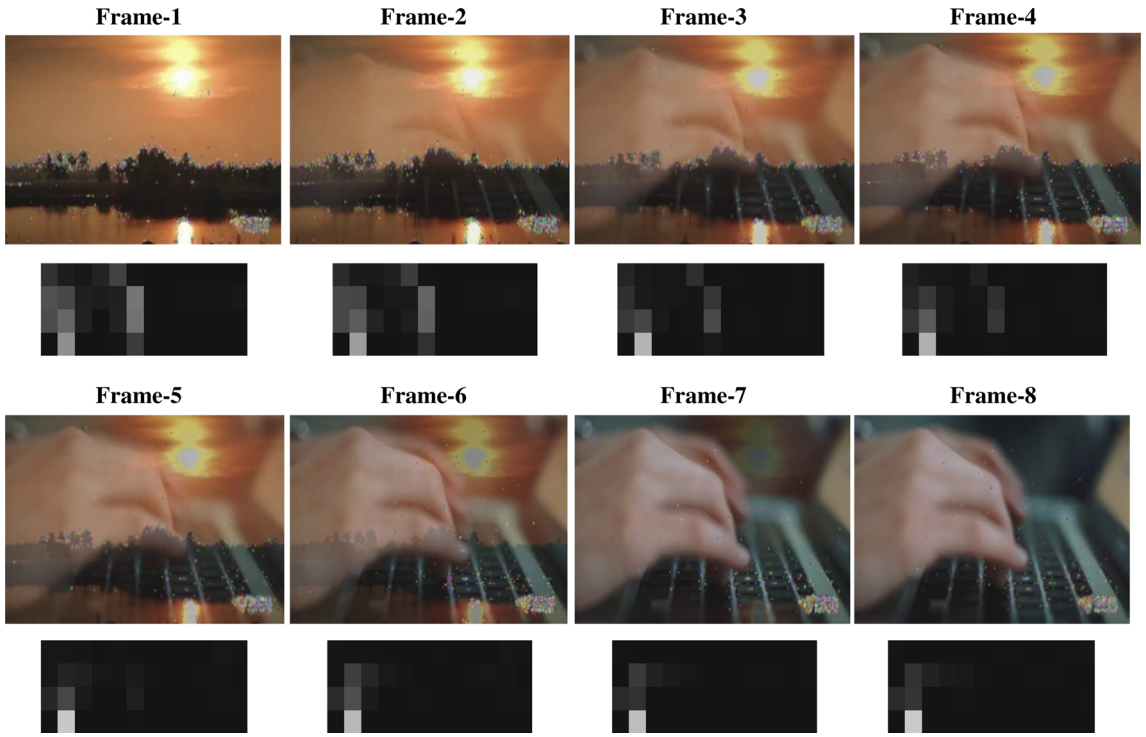


Fig. 7 A set of consecutive frames of the dissolve transition extracted from documentary video in dataset-2

the set S_p presents end of dissolve (D_e). Therefore, the first peak on the left side of start dissolve (D_s) presents end shot$_i$ and the right peak of end dissolve (D_e) presents start shot$_{i+1}$

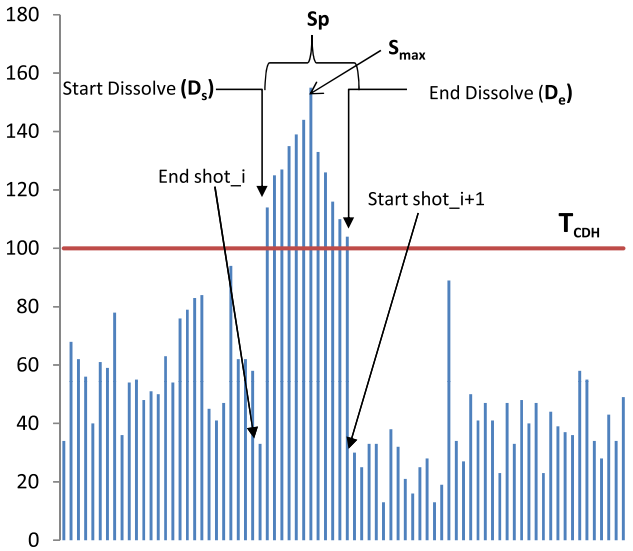


Fig. 8 A subpart of the DCH of the Fig. 6 where the dissolve transition occurs. The end of shot$_i$ is considered just before the dissolve starts and start of shot$_{i+1}$ is considered after end of dissolve occurs

as illustrated in Fig. 8. Hence, every subpart of DCH that satisfies the conditions below will be considered as indicator of dissolve.

$$\text{Dissolve} \leftrightarrow \begin{cases} S'_p(F) > 0 \;/\; D_s < F < S_{\max} & \text{and } S_p(F) > T_{\text{DCH}} \\ S'_p(F) < 0 \;/\; S_{\max} < F < D_e & \text{and } S_p(F) > T_{\text{DCH}} \end{cases} \tag{12}$$

Fading denoted F_d in the video sequence is either the progressive darkening of a shot until the last frame becomes black (fade-out) or the gradual transition from a black frame to a fully illuminated one (fade-in). During a fade, images have their intensity multiplied by some value α where α increases from 0 to 1 in fade-in and decreases from 1 to 0 in fade-out. Due to the usage of SIFT-PDH, fades can be easily distinguished in the Distance Comparison Histogram (DCH) by a set of null values of the number of SIFT keypoints, Fig. 9 shows an example of fade-out and fade-in transition. Significantly, in fade-in, α increases from 0 to 1 which imply that the number of SIFT keypoints of the frame F where α is approximately near to 0 will be zero as illustrated in Fig. 10. Therefore, end of the first shot is presented by the last non zero peak in the histogram and the start of consecutive shot is presented by the first non zero peak coming directly after

Fig. 9 A set of ten consecutive frames of a fade out/in transition extracted from advertising video in dataset-2 where the SIFT-PDH of each frame shows clearly the shot boundaries (frame-4 presents end of the current shot$_i$ and frame-8 presents start of the consecutive shot$_{i+1}$)

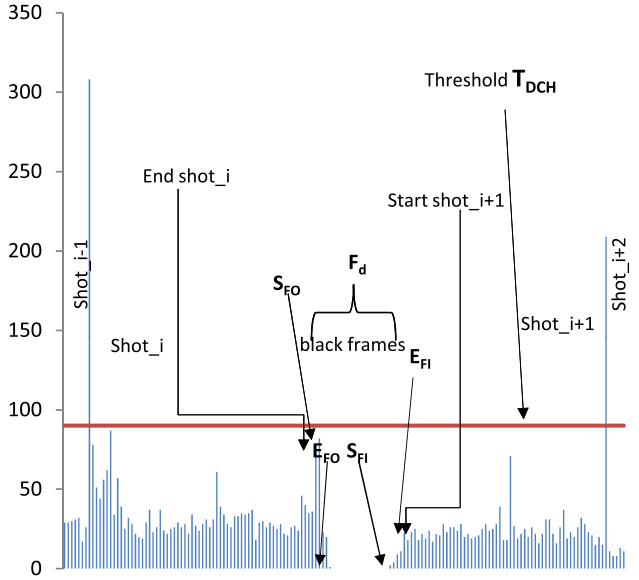


Fig. 10 A subpart of DCH of Fig. 6 where the fade out/in transition appears. End shot$_i$ is considered exactly before the set of null values in the histogram and start shot$_{i+1}$ is considered just after the last null values in the histogram

the set of zero elements in the histogram (Fig. 10). A subpart F_d of the DCH can be considered as fading if F_d satisfy the conditions below, where S_{FI} denotes the start fade-in, E_{FI} denotes the end fade-in, S_{FO} denotes the start fade-out and E_{FO} represents the end fade-out.

$$\text{Fading} \leftrightarrow \begin{cases} F'_d(F) < 0 \,/\, S_{FO} < F < E_{FO} \text{ and } F_d(F) < T_{\text{DCH}} \\ F'_d(F) = 0 \,/\, E_{FO} < F < S_{FI} \text{ and } F_d(F) < T_{\text{DCH}} \\ F'_d(F) > 0 \,/\, S_{FI} < F < E_{FI} \text{ and } F_d(F) < T_{\text{DCH}} \end{cases} \tag{13}$$

3.2 Keyframe extraction

After video shot boundary detection, extraction of keyframes from the segmented shots is required for video summarization. A keyframe is the key image which always reflects the salient content of the shot; this keyframe will reduce greatly the data size of video index and will provide an organized structure for the video stream. In order to extract the keyframes, we first perform the singular value decomposition (SVD) [19] to each frame of the segmented shot which will result in a vector of singular values. Next, the entropy measure is calculated from this vector. Finally, the frame having highest measure of entropy-based singular values is selected as a keyframe of the corresponding shot. The methodological techniques used throughout this section are presented as follows:

3.2.1 Singular value decomposition (SVD)

Any matrix $F(m \times n)$ can be decomposed using the singular value decomposition as follows:

$$F = \text{USV}^T \tag{14}$$

with U an $m \times k$ orthogonal matrix and V an $n \times k$ orthogonal matrix. S is a diagonal matrix defined by

$$S = \text{diag}(\lambda_1, \ldots, \lambda_k), \tag{15}$$

where $k = \min(m, n)$. The entries λ_k of the matrix S are called singular values; they are non negative and ordered from the biggest to the lowest elements.

The main idea behind the exclusive usage of singular values of the frame is that we only keep the essence of the frame (usefulness information) which is basically captured in a few singular values, instead of using the total frame's information. Therefore, representing the frame by a few singular values will mostly reduce the complexity while processing the frame.

3.2.2 Entropy-based singular values

Entropy is a measure of information obtained by observing a data source; it is merely a statistical average of information in the image. As mentioned in the SVD section, we already extract the singular values of the frames as useful features. Hence, we can now construct an entropy measure based on these singular values λ_k.

We first normalize the singular values λ_k using the following formula:

$$\lambda_k = \frac{\lambda_k}{\sum \lambda_k} \tag{16}$$

Then, a measure of entropy can be derived similarly to the [20] formula using

$$\text{Ent} = \sum \lambda_k \log(1/\lambda_k), \tag{17}$$

where the term $\log 1/\lambda_k$ indicates the amount of information gained by observing the singular values of the frame. One should not confuse the standard definition of entropy, based on probabilities [20], with the one used here, which is based on the distribution of singular values.

The computed entropy-based singular values of the frame vary between 0 and 1. The maximum entropy value among the computed entropies in the same shot corresponds to the frame holding more information. Therefore, we select this essence frame as a keyframe of the corresponding shot. Figure 11 illustrates the entropy-based singular values resulting

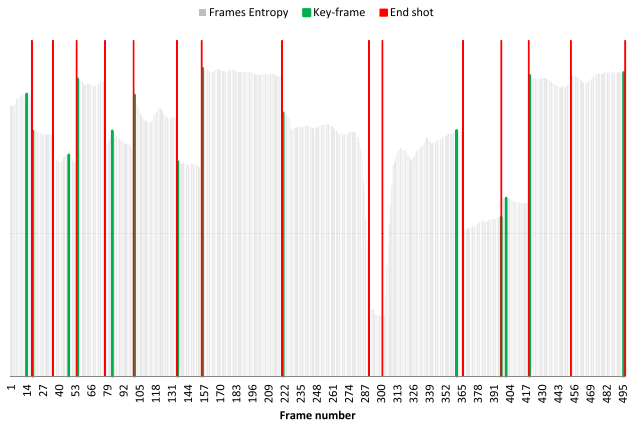


Fig. 11 Example of the extracted keyframes from the computed entropy-based singular values of each frame of the segmented shot for an advertising video sequence in the dataset-2

from a segmented shot where the highest entropy value is selected as a keyframe of the corresponding shot. The x-axis holds the number of frames, y-axis represents frame's entropy-based singular values, the red bins denotes end shots of the video and the green bins represent the extracted keyframe of each segmented shot.

4 Keyframes verification approach

The keyframe basically represents the salient content of the entire shot; this means that the information holding by this keyframe should cover the maximum information expressed through a shot. The verification technique proposed in this work to make sure that the extracted keyframe deserves to present the entire shot is based on the total of difference added information while transiting from frame to the consecutive frame. This amount of added information is extracted by calculating the difference between the entropy of two consecutive frames. The summation of this difference should be tenuous compared to the computed entropy of the extracted keyframe; this means that our keyframe is good as much as it covers all the added information through the shot. Therefore, the extracted keyframe is declared as good keyframe if we satisfy the condition below:

$$\sum_{i=1}^{\mathrm{end}(S)-1} |\mathrm{Ent}(F_i) - \mathrm{Ent}(F_{i+1})| \ll \mathrm{Ent}(K_f), \tag{18}$$

where $\mathrm{Ent}(F_i)$ denotes the entropy measured for frame F_i of the shot S, $\mathrm{Ent}(F_{i+1})$ denotes the entropy measured for the consecutive frame F_{i+1} of the shot S and $\mathrm{Ent}(K_f)$ denotes the entropy measured for the extracted keyframe of the shot S.

5 Experimental results

The performance of the proposed system was evaluated using three different datasets containing various numbers and types of video sequences. The first dataset (dataset-1) was provided by [4] holding five videos including advertising material, music and preview videos. In order to perform more accurate analysis, we collected another dataset (dataset-2) holding more challenging videos including documentaries, educational, sport news and movies with different light conditions, varieties of camera operations (zoom in, zoom out, camera rotation and camera motion) and having both abrupt and gradual transitions. This dataset-2 has 13 videos with multiple resolutions and a total of 37,293 frames digitizing with an average frame rate of 30 fps. The proposed shot boundaries detection method was also evaluated and compared with other recently existing methods [11,12,15] using eight complex video sequences taken from TRECVID 2001 dataset [21]. These video sequences were selected because the ground truth information regarding scene change is available.

We evaluated the proposed method for various partitions of SIFT-PDH histogram, while 4×12, 5×12 , 4×16 and 5×16 SIFT-PDH histogram features were extracted and the experimental results compared. Furthermore, the most successful case occurs while utilizing 4×12 SIFT-PDH feature vector (see Fig. 12).

Usually, the performance of the shot boundary detection algorithm is measured with terms of recall and precision. The recall and precision are defined as follows:

$$\mathrm{Recall} = \frac{C_d}{C_d + M_d} \times 100\,\% \tag{19}$$

$$\mathrm{Precision} = \frac{C_d}{C_d + F_d} \times 100\,\%, \tag{20}$$

where C_d is the number of correct detections, M_d is the number of missed detections and F_d is the number of false detections.

A good shot transition detector should have both high precision and high recall. Therefore, the performance of the proposed method was evaluated based on the comparison of the results of our method with regard to the other traditional methods [4,5,7,10] using dataset-1 . The details are reported in Table 1.

Figure 13 shows a comparative study between our method and other approaches working on the dataset-1. It is evident that our proposed shot detection algorithm outperforms the compared approaches, where our average recall is 99.94 % and the average precision is 99.78 %. Since the compared methods [4,5,7,10] capture any details of the frame while analyzing pixel by pixel, they are on the one hand computationally expensive compared to our method that uses

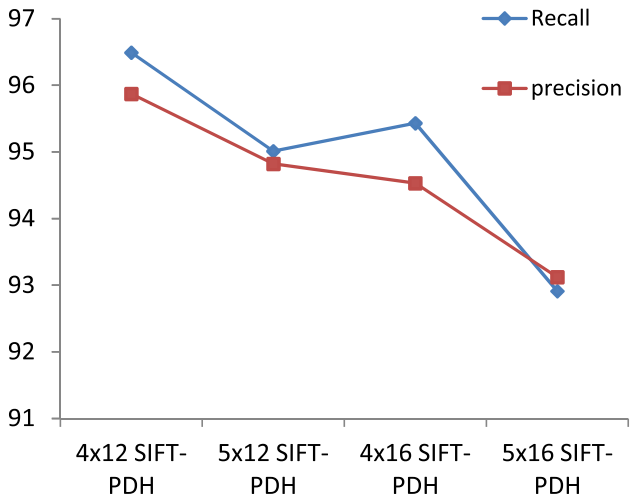


Fig. 12 Recall and precision for various partitions of SIFT-PDH applied on the dataset-2

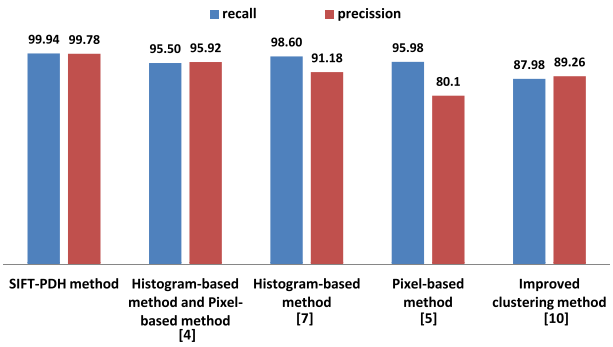


Fig. 13 Performance comparison between our proposed method with the other approaches [4,5,7,10] using dataset-1

only the SIFT keypoints of the frame and not the entire pixels of the frame; on the other hand, those methods are extremely sensitive to the noise, local motion and camera motion; also they are sensible to the illumination changes as they used color space. However, our proposed method segments the video based on the salient keypoints of the frame, the extracted SIFT-PDH feature vector resumes the combination between local and global features of the frame and gives a better performance in shot boundary detection; and

Table 1 Comparative study in terms of recall and precision between our proposed method using 4 × 12 SIFT-PDH and other approaches [4,5,7,10] using dataset-1

Video name	Video shot detection methods	Shot	Detected shot	False drop	Recall	Precision
[CM] Beelzebub ED	SIFT-PDH method	13	13	0	**100 %**	**100 %**
	Histogram-based method and pixel-based method [4]	13	13	1	92.9 %	92.9 %
	Histogram-based method [7]	13	18	5	100 %	78.3 %
	Pixel-based method [5]	13	31	19	96.9 %	62.0 %
	Improved clustering method [10]	13	18	6	94.7 %	75.0 %
[CM]Innisfree cm	SIFT-PDH method	13	13	0	**100 %**	**100 %**
	Histogram-based method and pixel-based method [4]	13	13	0	100 %	100 %
	Histogram-based method [7]	13	13	0	100 %	100 %
	Pixel-based method [5]	13	13	0	100 %	100 %
	Improved clustering method [10]	13	8	0	61.5 %	100 %
[News] Cctv_news	SIFT-PDH method	13	14	1	**100 %**	**98.9 %**
	Histogram-based method and pixel-based method [4]	13	15	2	100 %	86.7 %
	Histogram-based method [7]	13	15	2	100 %	86.7 %
	Pixel-based method [5]	13	18	5	100 %	72.2 %
	Improved clustering method [10]	13	16	3	100 %	81.3 %
[Preview]Anime 10th anniversary	SIFT-PDH method	4	4	0	**100 %**	**100 %**
	Histogram-based method and pixel-based method [4]	4	4	0	100 %	100 %
	Histogram-based method [7]	4	4	0	100 %	100 %
	Pixel-based method [5]	4	5	1	100 %	83.3 %
	Improved clustering method [10]	4	4	0	100 %	100 %
[MV]Taiyou no Uta_clip	SIFT-PDH method	39	39	0	**99.7 %**	**100 %**
	Histogram-based method and pixel-based method [4]	39	33	0	84.6 %	100 %
	Histogram-based method [7]	39	43	4	93.0 %	90.9 %
	Pixel-based method [5]	39	47	8	83.0 %	83.0 %
	Improved clustering method [10]	39	43	4	83.7 %	90.0 %

Table 2 The detailed information about each video in dataset-2 with the obtained results in terms of recall and precision based on our proposed method using 4 × 12 SIFT-PDH

Video name	Total frames	Length (in s)	Total real shots	# of cuts	# of dissolve	# of fades	Total output of detected shots	Added shots (false shots)	Missed shots	Recall (%)	Precision (%)	Execution time in (s)
Sport-news	5254	175	47	41	6	0	48	1	0	100	97.91	5.93
Advert-1	497	19	13	12	0	1	12	1	2	84.61	91.67	2.15
Educ-1	1900	76	57	57	0	0	59	2	0	100	96.61	3.86
Movie-1	896	37	12	11	0	1	14	3	1	91.67	78.57	2.92
Doc-1	1125	45	37	37	0	0	37	0	0	100	100	3.53
Movie-2	753	30	25	24	1	0	25	1	1	96	96	2.81
Movie-3	8672	346	78	76	0	2	89	0	0	100	100	7.78
Educ-3	1249	49	28	27	1	0	29	1	0	100	96.55	3.12
Doc-2	3100	124	50	46	1	3	49	1	2	96	97.95	5.29
Doc-3	3097	129	74	69	2	3	74	4	4	94.59	94.59	5.43
Advert-2	748	30	16	15	0	1	16	0	0	100	100	2.79
Sport-1	748	87	34	33	0	1	32	0	2	94.11	100	3.95
Educ-2	9254	308	112	99	7	6	116	4	3	97.39	96.55	8.56
Total	37,293	1455	583	547	18	18	600	18	15	96.49	95.87	58.12

the missed detection in case of gradual transition compared to these methods are also improved.

The database-2 is more challenging since it includes a large number of videos containing both abrupt and gradual transition and it covered almost all video types, including documentaries,movies, sports, news shows and so on to well evaluate the performance of the proposed method. Through our experiments, the total of real shots is determined by expert human eye observation. We have observed that our method generally detects shot transition either for hard as well as smooth shot breaks, and we have reported the results of the various video categories in Table 2 where supplementary information for each video category is cited.

From the statistics of the Table 2, we observe that the average recall for the tested videos is 96.49 % and the average precision is 95.87 %. It is also clearly observed that almost all of the shots through the various videos are correctly detected. However, as shown in Table 2, there are some added shots that are falsely segmented due to the extremely high motion and/or to the low resolution of the video, resulting in a jumbling of the objects in the frame. Figure 14 illustrates an example extracted from sport news video in dataset-2 where the jumbled frames are detected as new shots (frame-3 and frame-8 are detected as new shots).

Figure 12 illustrates the recall and precision results for a various SIFT-PDH histograms on the dataset-2. Note that the recall and precision is higher when 4 × 12 SIFT-PDH partition is considered for features extraction of the frame. The reason is that, when the number of SIFT-PDH bins increases, the area of point distribution histogram blocks in the frame becomes small. Therefore, the process of video segmentation will be affected by the factor of object motion.

Figure 15 Illustrates an example of illumination variance where all the representative frames belong to the same shot as correctly detected by our method, whilst in other methods it is detected as two different shots.

In order to make our results more significant and comparative with the benchmarking approaches [11,12,15], we have evaluated our method using some video sequences of the TRECVID-2001 dataset where the description of these test videos with the obtained results in terms of recall and precision for both cut and gradual transitions is listed in Table 3.

As per Table 3, it is critical that even by using some sequences taken from TRECVID-2001 dataset, our proposed method for shot boundary detection still achieves good results in terms of recall and precision while detecting both cut and gradual transitions and also outperforms the state-of-the-art methods [11,12,15] since our method use more representative keypoints which consider scale variance and illumination changes in addition to the employment of the adaptive threshold.

Notice that the shot boundary detection is almost assured for cut transition. However, one of the failure cases for dissolve transition occurs where some undesirable missed shot changes are found, which is because of the very long and slow effect of dissolve during the transition from shot to shot. In addition, the proposed shot boundary detection will face also some false detection (added shots that are falsely detected) while the motion on the video is largely noticeable (problem of very high motion detection). Furthermore, some more false detection will appear when the processed

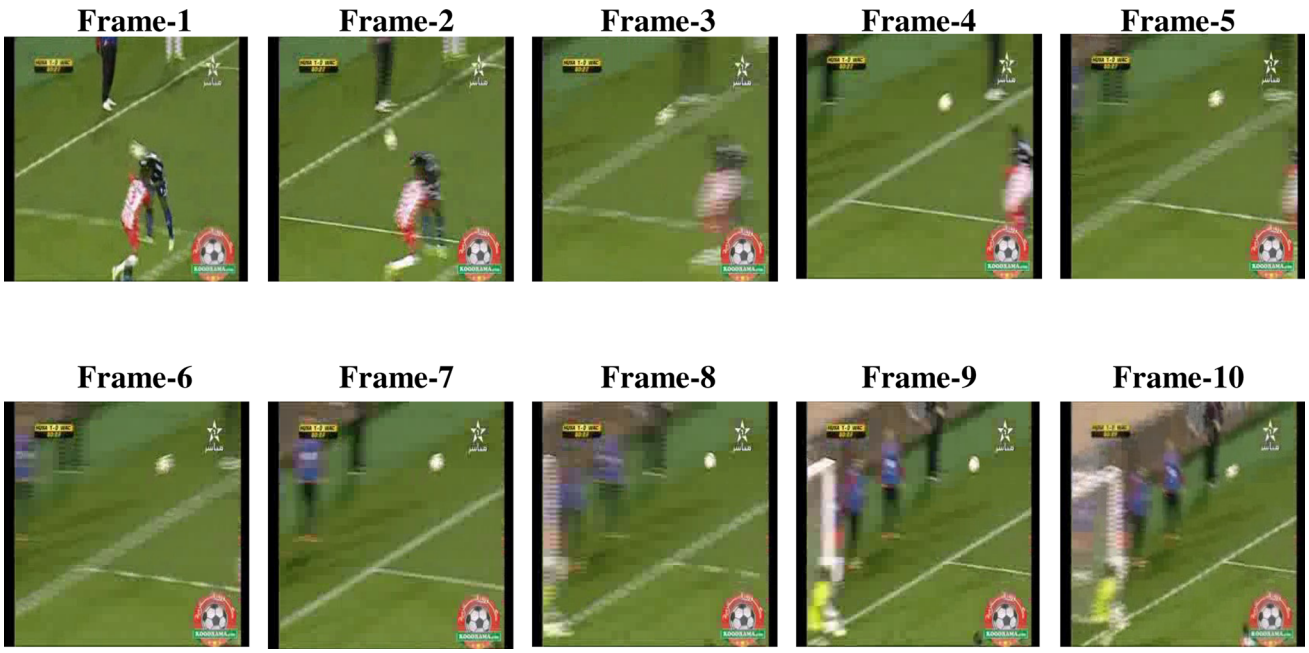


Fig. 14 Example of jumbled frames from sport video in dataset-2 caused by the low quality of the video where false detection occurs (frame-3 and frame-8 is detected as new shot

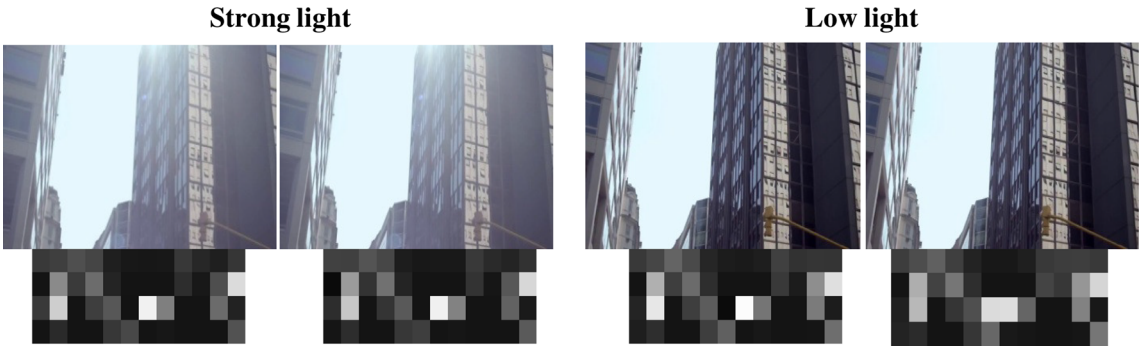


Fig. 15 Example of consecutive frames from advertising video in dataset-2 belonging to the same shot with different light conditions where the extracted SIFT-PDHs of the strong light frames (the two frames in the *left side*) and the low light frames (the two frames on the *right side*) are almost similar

video has low resolution so that the objects on the frames are jumbled as illustrated it in Fig. 14.

The idea behind combining both SIFT and edge-SIFT keypoints through our process is crucial as mentioned in Sect. 3.1.1. Therefore, to evaluate and demonstrate the improvements imported by combining SIFT and edge-SIFT keypoints compared to the use of single SIFT keypoints only, we carry out experiments on the videos of our collected dataset-2 and we compare the obtained results using single SIFT keypoints only and both SIFT and edge-SIFT keypoints. The recall and precision rates are shown in Fig. 16.

As seen in Fig. 16, the performance of our method in terms of recall and precision increased significantly while using both SIFT and edge-SIFT keypoints, and it gives good results compared to the one with only single SIFT keypoints, which is due to the importance of the SIFT keypoints that are located on the edges of the frame objects as we have already mentioned.

The approach used in this paper to extract keyframe from the segmented video is sufficient enough to represent the original shot. Moreover, the obtained frames from the extracted keyframes built a summarized video which is able to represent the original video in a short and concise manner. Therefore, the size of our summarized video (output video) will be the more reduced one in such a way that the number of frames forming the summarized video will be equal to the number of the segmented shots on the original video. The average size of the summarized video represents 2.36 % from the entire video stream; Fig. 17 shows the whole size of the original video from dataset-2 compared with its

Table 3 Comparative study in terms of recall (Rec) and precision (Pre) between our work (using 4×12 SIFT-PDH) and the approaches in [11,12,15] using some video sequences from TRECVID-2001 dataset

Video sequences	Video size	Run time	Total frames	Transition types	# transitions	Ours		[12]		[11]		[15]	
						Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre
Anni005	66.9	6:19	11364	Cut	38	100	90.4	100	82.6	94.6	100	86.8	94.3
				Gradual	27	93.1	90.0	88.8	82.8	75.9	100	0.0	0.0
Anni009	72.4	6:50	12307	Cut	38	100	84.5	100	76.0	_	_	94.7	85.7
				Gradual	65	95.5	92.8	90.7	88.1	_	_	4.6	75.0
Nad31	260.1	29:08	52405	Cut	187	97.9	89.4	96.3	81.8	_	_	4.3	100
				Gradual	55	88.7	85.9	81.8	78.9	_	_	0.0	0.0
Nad33	247.1	27:40	49768	Cut	189	96.9	91.3	94.7	86.1	_	_	91.0	95.0
				Gradual	26	96.2	89.6	92.3	80.0	_	_	23.1	85.7
Nad53	128.0	14:20	25783	Cut	82	98.8	88.1	98.8	80.2	_	_	84.2	92.0
				Gradual	75	97.4	94.9	97.0	91.3	_	_	4.0	75.0
Nad57	63.4	7:06	12781	Cut	44	100	95.6	97.7	91.5	_	_	90.9	90.9
				Gradual	23	95.8	92.0	95.6	84.6	_	_	8.7	100
Bor03	240.5	26:56	48451	Cut	231	98.7	95.4	97.8	90.4	_	_	62.7	92.3
				Gradual	11	91.6	84.6	100	68.8	_	_	18.2	100
Bor08	251.0	28:07	50569	Cut	380	97.6	95.7	93.4	86.8	_	_	49.7	99.0
				Gradual	151	97.4	96.1	96.0	91.3	_	_	7.0	50.0

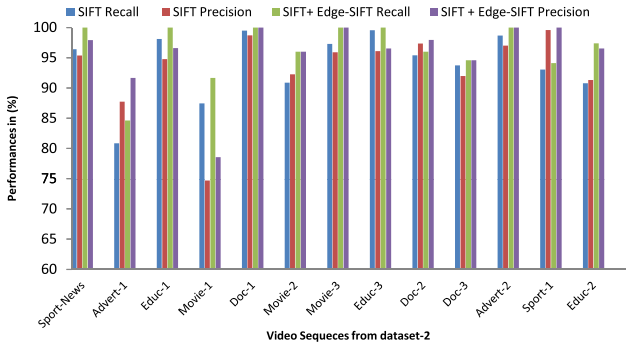


Fig. 16 Performance comparison in terms of recall and precision of our method using only SIFT keypoints and combining both SIFT and edge-SIFT keypoints through the 13 videos of dataset-2

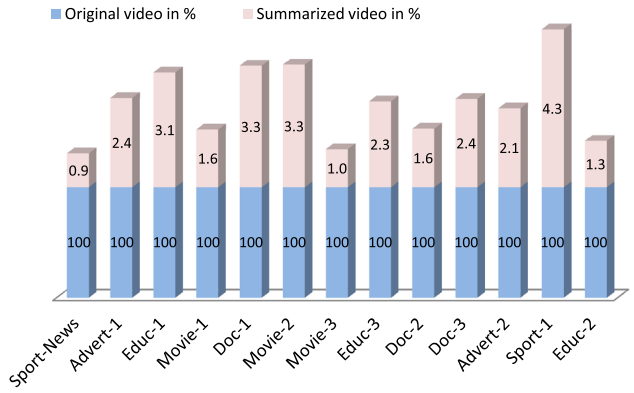


Fig. 17 The corresponding ratio of the summarizing video compared to its original video

summarized video obtained from the extracted keyframes. The correctness verification of the keyframes resulted are entirely based on the visual observation of an expert group on the domain of video summarization. Figure 18 shows the extracted keyframes from a segmented video of dataset-2 where it is clearly observed that the extracted keyframes are sufficient enough to represent and summarize the whole original video in a concise manner.

The proposed approach can process a video frame in 2.9 ms for an average speed of 344 frames per second. The extraction of the SIFT-PDH histogram takes about 1 ms. The computation of the distance comparison values takes about 0.6 ms and the remaining computation time of 1.3 ms was for calculation of entropy-based singular values for keyframes extraction.

6 Conclusion

In this research paper, we have described and discussed a novel and fast method for video summarization. Our system detects video shot boundaries by extracting the SIFT and edge-SIFT keypoints for each frame in the video. SIFT-point distribution histogram (SIFT-PDH) is extracted as a global feature of the frame. The distance comparison of SIFT-PDH between each two consecutive frames is computed, and an Adaptive Threshold is employed to detect video shot boundaries. Keyframes are extracted using entropy-based singular value metric. The video summary of the original video stream is generated by combining the extracted keyframes. The experimental results show that our method can efficiently

Fig. 18 Keyframes extraction from the segmented shots of a documentary video in dataset-2. **a** Four segmented shots from the video data. **b** Four extracted keyframes from segmented shots

detect shot boundaries for different types of videos, even under different levels of illumination, motion effects and camera operations (zoom in, zoom out, camera rotation), as it can also summarize the original video in a concise manner with minimum size and less computational complexity. The proposed system is implemented in OPENCV-C++ using 10 GHz and 8 GB RAM system and validated using three different datasets. The first dataset is provided by [4] containing five videos of different types. The second dataset is collected from the internet containing 13 challenging videos with multiple resolution and different transitions. The last used dataset is relatively complex and is composed of eight videos taken from TRECVID-2001.

References

1. Ajmal M, Husnain Ashraf M, Shakir M, Abbas Y, Shah FA (2012) Video summarization: techniques and classification. In: Computer vision and graphics, lecture notes in computer science, vol 13, no 1, pp 7594
2. Koprinska I, Carrato S (2001) Temporal video segmentation: a survey. Signal Process Image Commun Elsev 16(5):477–500
3. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
4. Zhong Q, Lidan L, Tengfei G, Yongkun W (2013) An improved keyframe extraction method based on HSV colour space. J Softw 8(7):1751–1758
5. Hong-cai F, Xiao-juan Y, Wei M, Cao Y (2010) A shot boundary detection method based on colour space. In: International conference on E-business and E-government, pp 1647–1650
6. Janwe N, Bhoyar K (2013) Video shot boundary detection based on JND color histogram. In: Proceedings of the international conference on ICIIP, Shimla, pp 476–480
7. Gunal ES, Canbek S, Adar N (2009) Gradual shot change detection in soccer videos via fractals. In: International conference on electrical and electronics engineering, pp 88–92
8. Zhang H, Hu R, Song L (2011) A shot boundary detection method based on color feature. In: International conference on computer science and network technology (ICCSNT), vol 4, pp 2541–2544
9. Lu Z, Shi Y (2013) Fast video shot boundary detection based on SVD and pattern matching. IEEE Trans Image Process 22(12):5136–5145
10. Wenzhu X, Lihong X (2010) A novel shot detection algorithm based on clustering. In: 2nd international conference on education technology and computer, vol 1, pp 570–572
11. Huang CR, Lee HP, Chen CS (2008) Shot change detection via local keypoint matching. IEEE Trans Multimed 10(6):1097–1108
12. Choudhury A, Medioni G (2012) A framework for robust online video contrast enhancement using modularity optimization. IEEE Trans Circ Syst Video Technol 22(9):1266–1279
13. Sabbar W, Chergui A, Bekkhoucha A (2012) Video summarization using shot segmentation and local motion estimation. In: Sec-

ond international conference on innovative computing technology (INTECH), pp 18–20
14. Sujatha C, Mudenagudi U (2011) A study on keyframe extraction methods for video summary. In: International conference on computational intelligence and communication networks (CICN), vol 73, no 77, pp 7–9
15. Gianluigi C, Raimondo S (2006) An innovative algorithm for key frame extraction in video summarization. J Real-Time Image Process 1:69—88
16. Azeroual A, Afdel K, El Hajji M, Douzi H (2014) Video shot detection and key-frames extraction using Faber Shauder DWT and SVD. Int J Comput Control Quant Inf Eng 8(12):2003–2006
17. Lowe DG (1999) Object recognition from local scale-invariant features. In: International conference on computer vision, pp 1150–1157
18. Brown M, Lowe D (2002) Invariant features from interest point groups. In: Proceedings of the british machine conference, pp 23.1–23.10
19. Klema V, Laub AJ (1980) The singular value decomposition: its computation and some applications. IEEE Trans Autom Control 25(2):164–176
20. Shannon CE (1948) A mathematical theory of communication. Bell Syst Techn J 27:379–423
21. Interaction Design Laboratory, The Open Video Project. http://www.open-video.org/