J. Vis. Commun. Image R. 23 (2012) 1031–1040

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Adaptive key frame extraction for video summarization using an aggregation mechanism

Naveed Ejaz [a], Tayyab Bin Tariq [b], Sung Wook Baik [a,*]

[a] College of Electronics and Information Engineering, Sejong University, Seoul, Republic of Korea
[b] National University of Computer and Emerging Sciences, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 13 February 2012
Accepted 26 June 2012
Available online 6 July 2012

Keywords:
Key frame extraction
Video summarization
Video abstraction
Static video summary
Storyboard
Video analysis
Evaluation of video summary
Feature aggregation

ABSTRACT

Video summarization is a method to reduce redundancy and generate succinct representation of the video data. One of the mechanisms to generate video summaries is to extract key frames which represent the most important content of the video. In this paper, a new technique for key frame extraction is presented. The scheme uses an aggregation mechanism to combine the visual features extracted from the correlation of RGB color channels, color histogram, and moments of inertia to extract key frames from the video. An adaptive formula is then used to combine the results of the current iteration with those from the previous. The use of the adaptive formula generates a smooth output function and also reduces redundancy. The results are compared to some of the other techniques based on objective criteria. The experimental results show that the proposed technique generates summaries that are closer to the summaries created by humans.

1. Introduction

In the last few years, there has been a drastic increase in creation and storage of video data on the internet. This prompt increase in video data stipulates efficient techniques for indexing, retrieval, and storage of this data. However, these techniques have not progressed at the same pace [1]. This is due to the substantially different nature of video data which is not suited for conventional retrieval, indexing, and storage techniques. Therefore, there is a strong need of research work to solve this problem of video data management. The nature of video itself provides a solution to this problem, as usually every video contains a lot of redundant information which can be removed to make video data more suitable for retrieval, indexing, and storage. This approach of removing redundancies from the video and generating condensed versions of the video is called video summarization [2]. The video summaries contain the most important and relevant content of a video. There must not be any redundancy in the summarized video, but at the same time, the original message of the video must be preserved [3].

The video summaries can be generated in many different forms. However, the two most popular ways to generate summaries are static and dynamic [4]. The static video summary deals with the extraction of the so-called key frames from the video. The key frames are still frames extracted from the video which hold the most important content of the video and thus are representative of the video [5,6]. The dynamic video summary contains small shots that are accumulated in a time ordered sequence. An advantage of a dynamic video summary over static summary is that it preserves the dynamic nature of video content by arranging the semantics based on time. Furthermore, the dynamic summaries may convey more information because of the audio and motion contents [7]. Despite these reasons, the extraction of static summaries or key frame extraction is important, because it provides more flexibility owing to no issues of synchronization. Moreover, key frame extraction can be used as a pre-processing step in video analytics applications [8,9] which suffer from the problem of processing large number of video frames.

This paper addresses the problem of key frame extraction. For key frame extraction, the problem is finding the minimal set of key frames that cover all significant events or maximize the number of key frames while minimizing redundancy of information in these key frames. Although a number of techniques have been presented in the literature for key frame extraction, most of them are computationally expensive [10]. In this paper, we propose a simple and effective technique for key frame extraction based on comparison of frames, called "VSUKFE" (video summarization using key frame extraction). VSUKFE uses inter-frame differences calculated based on the correlation of RGB color channels, color histogram and moments of inertia. The three frame difference measures are then combined using an aggregation mechanism to extract key frames. An adaptive formula is used to make our technique

* Corresponding author. Fax: +82 02 3408 4339.
E-mail addresses: naveed@sju.ac.kr (N. Ejaz), tayyab.tariq@nu.edu.pk (T.B. Tariq), sbaik@sejong.ac.kr (S.W. Baik).

1047-3203/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2012.06.013

partially tolerant to changes in lighting conditions. The adaptive aggregation mechanism effectively combines the three frame different measures such that for the less change in the inter-frame content, the aggregated value is significantly low and vice versa. Moreover, our technique provides a balance between computational complexity and the quality of results.

The problem of key frame extraction is that it is subjective in nature, and thus a quantitative evaluation of results is not easy. A consistent mechanism for evaluation of key frame summaries does not exist in the literature. Generally, the evaluation schemes available in literature are subjective in nature because of involvement of humans in the evaluation. However, we looked for an objective comparison strategy which minimizes human involvement. The proposed scheme (VSUKFE) is evaluated based on the evaluation strategy presented in [3]. VSUKFE exhibits promising results and compares well to the rest of the techniques to which it is compared. In the end, we also compare the asymptotic time complexities of VSUKFE and VSUMM (Video SUMMarization) [3].

The main contributions of this paper include:

- Use of three frame difference measures for representing the features of a frame.
- Use of an adaptive formula to combine frame difference measures of the previous and current iterations, which generates a smooth function and helps in handling gradual changes in lighting conditions.
- Use of an adaptive aggregation mechanism to combine three different frame difference measures which suppresses the frame difference values if content change between frames is minimal and vice versa.

The rest of this paper is organized as follows. In Section 2, a brief overview of related work is presented; in Section 3, the proposed framework for key frame extraction is discussed; the results are discussed in Section 4; and finally, conclusions are drawn in Section 5.

2. Previous work

In this section, some of the major techniques of key frame extraction are discussed. For a detailed review of the existing techniques, we refer readers to the work of [2,6,11].

A popular set of techniques is to compute the frame differences based on some criteria and then discard the frames whose difference with the adjacent frames are less than a certain threshold. Various low level features have been applied for this purpose including color histograms, frame correlations, edge histogram, etc. [12]. For instance, Pal and Leigh [13] used fuzzy geometrical and information measures to develop an algorithm to estimate the difference between two consecutive frames. The similarity between the frames was measured in terms of weighted distance in fuzzy feature space. Hanjalic et al. [14] compared the difference in color histograms of consecutive frames with a threshold to obtain key frames. A famous method for key frame extraction was introduced by DeMenthon et al. [15]. The key frames were extracted by finding discontinuities on a trajectory curve, which represent a video sequence. In the Flexible Rectangles algorithm [16], the frame differences were used to build a "Content Development Curve" from a curve composed of a predefined number of rectangles through the use of an error minimization algorithm. The Adaptive Sampling algorithm [17] extracted key frames by uniformly sampling the y-axis of the curve of cumulative frame differences. The resulting sampling on the x-axis represented the key frames. The Shot Reconstruction Degree Interpolation [18] selected the key frames based on the ability of frames to reconstruct the original video shot using frame interpolation. Ciocca and Schettini [5] extracted key frames by first finding the cumulative frame differences based on certain frame descriptors such as color histogram, histogram of edges and wavelets. Next, a curve of cumulative frame differences was sketched, and then the midpoints of two curvature points on this curve were selected as key frames. A curvature point is a point on the curve where the angle changes are drastic. The frame difference based methods are intuitive and simple in nature. These properties make them suitable for many real-time and/or online applications. However, for extracting a particular key frame, these techniques only consider sufficient content change between the consecutive frames (or between current frame and last key frame). Therefore, a key frame that is extracted by these methods does not fully represent the portion of the video preceding it [6].

Some researchers used clustering for extracting key frames by treating video frames as points in the feature space. The core idea behind such techniques is to cluster the frames based on some similarity measure and then select one key frame from each cluster. Yeung and Yeo [19] proposed a method to generate a pictorial summary of a video sequence which consists of a set of video posters, each representing a scene in the sequence. The key frames were extracted using a time-constrained clustering method which takes into account both visual properties and temporal locality of the shots. The video posters were generated by combining key frames based on their dominant scores assigned during the clustering phase. Zhuang et al. [20] presented a technique for key frame extraction based on unsupervised clustering using a color histogram as the visual content. A node is added to a cluster only if the similarity measure between the frame and the cluster centroid is greater than a certain threshold. Doulamis et al. [21] presented a technique for summarizing stereoscopic videos which used clustering of shots to reduce redundancy. The clustering was performed based on the multidimensional fuzzy classification of segment features extracted from stereoscopic frames. A motion based clustering algorithm was introduced by Zhang et al. [22] in which the clustering was done based on the motion compensation error. Mundur et al. [23] used Delaunay triangulation based clustering of color histogram features. Furini et al. [10] proposed a summarization technique called "STIMO" (STIll and MOving Video Storyboard) based on the clustering of HSV color descriptors. Avila et al. [3] presented a method "VSUMM" (Video SUMMarization) which extracted color features from the frames after pre-sampling the frames from video. After removal of useless frames, the rest of the frames are clustered based on the k-means clustering algorithm. The main advantage of clustering based methods is that they generate less redundant summaries as compared to the consecutive frame difference based techniques. The problem with most of the clustering methods (less time constrained clustering) is that temporal information of the frames is not considered. In other words, the key frames are selected regardless of the temporal order of each frame. Therefore, the key frames may not preserve the temporal visual sequence of the video [6].

Some techniques in the literature use the extraction of interesting events and objects in an attempt to find the semantically relevant key frames. For instance, Calic and Thomas [24] selected key frames where objects merge by using techniques of frame segmentation. In the works of Luo et al. [25] and Guironnet et al.[26], the key frames were selected according to the rules defined on sequence and the magnitude of camera motions. The multiple features like automatic scene analysis, camera viewpoint selection, and adaptive streaming for summarizing basketball videos was used by Chen et al. [27]. The camera and motion based techniques may work well for certain experimental settings and specified domain. However, such techniques are dependent on heuristic rules extracted from a rather limited data set. Therefore, such schemes

may fail in situations where videos have complex and irregular motion patterns which were not tested initially [6].

A thorough study of the related work reveals that various visual features have been used for selecting key frames from the videos. Moreover, the techniques are either too complex or too naïve. The simpler techniques heavily compromise the quality of key frame extraction and the more sophisticated techniques are computationally very expensive. Our scheme provides a good tradeoff between complexity and quality of results

3. Framework

There is a given video to be summarized which starts at time "t" and contains n_{NF} frames:

$$V_t = \{ F(t+i) | i = 0, 1, \ldots n_{NF-1} \} \quad (1)$$

In Eq. (1), "F" refers to a single frame of the video. The aim is to extract the set of key frames KF_t from the set V_t. The set KF_t having n_{KF} frames is defined as:

$$KF_t = \{ F_{KF}(t+1), F_{KF}(t+2), \ldots\ldots, F_{KF}(t+n_{KF}) | n_{KF} \leqslant n_{NF} \} \quad (2)$$

The proposed strategy for extraction of key frames is based on a comparison of frames. In general, a single feature is usually not sufficient to estimate all the pictorial details of a frame and the visual complexity of video shots [5,28]. For an effective representation of pictorial contents of a frame, both the color and structural properties must be used. Therefore different features can be assorted to provide an effective representation of a frame. For this reason, three comparison measures are used to capture the representative frame difference: inter-frame correlation of RGB color channels, color histogram, and moments of inertia. The frames are divided into non-overlapping sections (or blocks), and the corresponding sections of the current frame and the last key frame are then compared.

The inter-frame correlation measures the similarity between two frames based on the color contents. Color histograms are used to gauge the frame differences based on the color. Color histograms have been selected because of their simplicity and robustness to small changes in camera motion. The usage of two different color spaces in correlation and histogram frame difference measures, allow the computation of color differences from two different perspectives. If only correlation difference measure is used then even small changes in camera motion may result in a relatively high frame difference value. On the other hand, if only the histogram measure is used, the frames having similar color contents but different visual contents may produce small frame difference values. The moments of inertia are used as additional metric for providing a notion of shape description.

Fig. 1 shows the main steps of VSUKFE for extracting key frames. Each step is now described in detail in subsequent sub-sections.

3.1. Pre-sampling

As a pre-processing step, the frames are pre-sampled from the video by selecting candidate key frames after a specific interval of time. This step reduces the amount of data to be processed and hence significantly improves performance. The pre-sampling step is benefecial in videos having long shots. However, in videos having short shots, important parts of the content may be wasted. For this reason, the sampling rate must be selected carefully to prevent any loss of information. If Skip Factor 'λ' is defined as:

$$\lambda = (\text{Time interval between consecutive candidate key frame}) \times (\text{frame rate}) \quad (3)$$

Then the number of candidate frames 'n_{CF}' and the set of candidate frame 'CF_T' are defined as:

$$n_{CF} = \frac{n_{NF}}{\lambda} \quad (4)$$

$$CF_T = \{ F(t+j) | j = 0, \lambda, 2\lambda \ \ldots\ , j \leqslant n_{NF} \} \quad (5)$$

The next steps are performed only on the set of candidate frames CF_T.

3.2. Correlation frame difference measure

The correlation coefficients have been widely used to capture the similarity between two frames. After dividing the frames into non-overlapping sections, the correlation coefficient is calculated for each color channel (red, green and blue) between each section of the frames to be compared. Let $F(t)$ and $F(t+1)$ be the two frames for the calculation of correlation from the set of candidate frames CF_T. It is assumed that the dimension of each frame is mxn and each frame has been divided into a total of "T_s" sections of pxq each. Then the correlation coefficient for a section "s" for a color channel "c" is given by:

$$r(F(t), F(t+1))_{s,c} = \frac{\sum_{i=1}^{p}\sum_{j=1}^{q}(F_s(t)_{c,i,j} - \overline{F_{c,s}(t)})\,(F_s(t+1)_{c,i,j} - \overline{F_{c,s}(t+1)})}{\sqrt{\sum_{i=1}^{p}\sum_{j=1}^{q}\left(F_s(t)_{c,i,j} - \overline{F_{c,s}(t)}\right)^2 \sum_{i=1}^{p}\sum_{j=1}^{q}\left(F_s(t+1)_{c,i,j} - \overline{F_{c,s}(t+1)}\right)^2}} \quad (6)$$

where $F_s(t)_{c,i,j}$ is the pixel value of "c" color channel of $F(t)$ at row "i" and column "j" in section "s", and $\overline{F_{c,s}(t)}$ and $\overline{F_{c,s}(t+1)}$ are the mean values of the pixel values of color channel "c" in section "s" of frames $F(t)$ and $F(t+1)$. The correlation coefficient is computed for each section ($s = 1 \ldots T_s$) and color channel (c = red, green, blue). The mean of the correlation of all sections is then taken to compute the overall correlation for a color channel:

$$r(F(t), F(t+1))_c = \frac{1}{T_s}\sum_{k=1}^{T_s} r(F(t), F(t+1))_{k,c} \quad (7)$$

Finally, the values from all color channels are combined using the mean function to obtain the result of correlation comparison measure as under:

$$\rho(F(t), F(t+1)) = \frac{r(F(t), F(t+1))_{red} + r(F(t), F(t+1))_{green} + r(F(t), F(t+1))_{blue}}{3} \quad (8)$$

3.3. Histogram frame difference measure

Color histograms have been very popular in extraction of key frames because of their relative simplicity and robustness against small changes in camera viewpoint [3]. They have been widely used for video summarization [3,5,10,20,23].

For using color histograms, the appropriate color space and the quantization of that color space must be chosen. VSUKFE uses HSV color space for histogram computation which has the ability to provide intuitive representation of color closer to human perception. After obtaining a color histogram, a color quantization step is applied to reduce the size of the color histogram. The quantization of the color histogram is set to 16 bins for hue component, and 8 bins for each of the saturation and intensity components. The Hue, saturation, and intensity histograms are then normalized in the range of 0–1 by dividing each value by the maximum value in the respective component. The three histograms are then combined to get a histogram of size 32.

For computing the histogram difference measure between two frames, each frame is first divided into "T_s" number of sections.

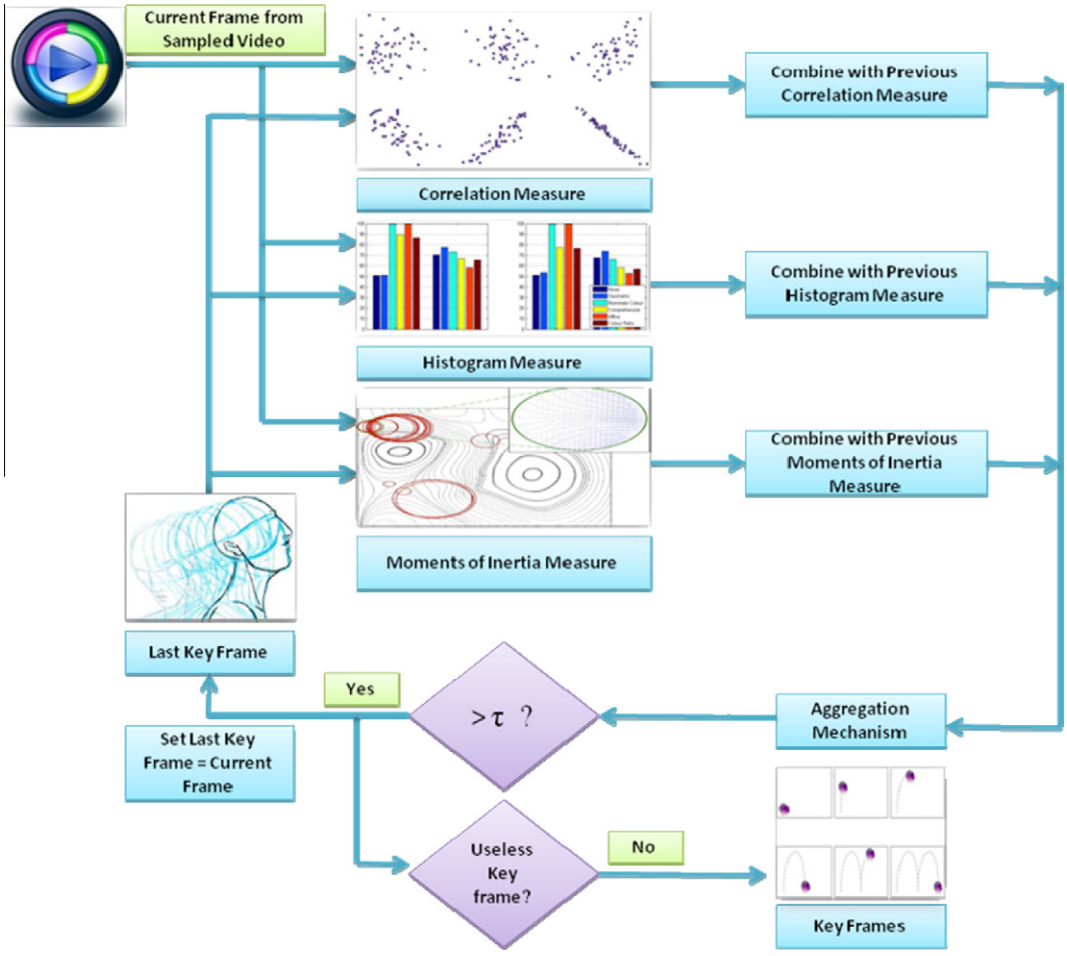

Fig. 1. Framework of VSUKFE.

The histogram is computed for each section of the frames and then the histogram difference between the corresponding sections of the frames is obtained by using a histogram intersection [29]. The mean of the difference values of all sections is then used as the histogram difference measure. The histogram comparison measure between two frames $F(t)$ and $F(t+1)$, divided into T_s number of sections each, is given as:

$$H(F(t),F(t+1)) = \frac{1}{T_s}\sum_{p=1}^{T_s}\left(1-\sum_{l=1}^{32}\min\left(H_{t,p}(l),H_{t+1,p}(l)\right)\right) \quad (9)$$

where $H_{t,p}$ and $H_{t+1,p}$ are the color histograms of the pth section of $F(t)$ and $F(t+1)$ respectively.

3.4. Moments of inertia difference

Moments of inertia are frequently used in image processing as compact image descriptors that can be used to differentiate an image from another image [30]. In this work, the three moments of inertia (mean, variance, skewness) are used to compute the 9 moments from each section of a frame (3 for each color channel). Again, a frame is divided into "T_s" number of sections of size pxq each. For a frame $F(t)$, the mean, variance, and skewness are computed for the three color channels of every section as:

$$\overline{F(t)}_{s,c} = \frac{1}{\text{pxq}}\sum_{i=1}^{p}\sum_{j=1}^{q}F(t)_{i,j} \quad (10)$$

$$\sigma^2(F(t))_{s,c} = \frac{1}{\text{pxq}}\sum_{i=1}^{p}\sum_{j=1}^{q}(F(t)_{i,j}-\overline{F(t)}_{s,c})^2 \quad (11)$$

$$\gamma(F(t))_{s,c} = \frac{1}{\text{pxq}}\frac{\sum_{i=1}^{p}\sum_{j=1}^{q}(F(t)_{i,j}-\overline{F(t)}_{s,c})^3}{(\sigma^2(F(t))_{s,c})^{3/2}} \quad (12)$$

where $\overline{F(t)}_{s,c}$, $\sigma^2(F(t))_{s,c}$ and $\gamma(F(t))_{s,c}$ are the mean, variance, and skewness values of color channel "c" in section "s" respectively. Finally, these values are combined to form a moments of inertia feature vector μ_t of frame $F(t)$. The size of the vector is $9 \times T_s$. The moments of inertia difference measure between two frames $F(t)$ and $F(t+1)$ is computed by using the Euclidean distance between the respective feature vectors.

$$\mu(F(t),F(t+1)) = \sqrt{\sum_{i=1}^{9T_s}(\mu_t(i)-\mu_{t+1}(i))^2} \quad (13)$$

3.5. The adaptive comparison values

The difference of frames, based on three comparison measures described above, is computed between each candidate frame and the selected key frame of the previous iteration. All the values are mapped in the range of 0–1. An adaptive formula is then used to combine the previous and current comparison results to calculate an adaptive comparison value. The previous and current values can be given different weights in generation of combined comparison value. It is to be noted that the first iteration is treated as a special case in which the first frame from the sampled video is taken as a key frame. The combination of current and previous comparison values generates a smooth output function. Moreover, this

function is insensitive to small changes in lighting conditions because of the ability to control the sudden change.

The three adaptive values for correlation (ρ'_n), histogram (H'_n), and moments of inertia (μ'_n) comparison measures at frame 'n' are computed as follows (to simplify the notation we have used ρ, H and μ to indicate the correlation, histogram, and moments difference measures):

$$\rho'_n = \alpha_1 \rho_{n-1} + (1-\alpha_1)\rho_n$$

$$H'_n = \alpha_2 H_{n-1} + (1-\alpha_2) H_n \qquad 0 \leqslant \alpha_i \leqslant 1,\ i = 1,2,3 \tag{14}$$

$$\mu'_n = \alpha_3 \mu_{n-1} + (1-\alpha_3)\mu_n$$

α_1, α_2, and α_3 are the weights assigned to the previous correlation, histogram, and moments of inertia comparison measures respectively. ρ_{n-1}, H_{n-1}, and μ_{n-1} denote the respective comparison values at time $n-1$, and ρ_n, H_n, and μ_n are the comparison values at time n.

3.6. Aggregation mechanism and key frame selection

A new key frame must be selected only if there is a significant change between the current frame and the last key frame. In order to determine the amount of inter-frame change, the three comparison measures are combined using an aggregation mechanism.

The three adaptive frame difference measures ρ'_n, H'_n, and μ'_n are compared with pre-defined threshold τ_ρ, τ_H, and τ_μ respectively. As a result of this comparison, a real number called "contributing value" is obtained for each adaptive frame difference measure. ρ'_n captures the amount of similarity, whereas H'_n and μ'_n captures the amount of difference between the current frame and the last key frame. By comparing with thresholds, the corresponding contributing values of the three measures are generated in such a way that the contributing values are high if there is a high inter-frame difference and vice versa. For instance, a value of ρ'_n that is less than the threshold τ_p indicates significant inter-frame difference and thus results in a positive contributing value. The contributing value will be high if the difference between τ_p and ρ'_n is high and vice versa. On the other hand, if ρ'_n is greater than τ_p, the result is a negative contributing value which signifies low inter-frame difference. Equation (16)–(18) show the process of generation of contributing values for the three adaptive comparison measures.

$$d_\rho = \begin{cases} 1 + |\rho'_n - \tau_\rho| & \text{if} \quad \rho'_n < \tau_\rho \\ -|\rho'_n - \tau_\rho| & \text{otherwise} \end{cases} \tag{15}$$

$$d_H = \begin{cases} 1 + |H'_n - \tau_H| & \text{if} \quad H'_n > \tau_H \\ -|H'_n - \tau_H| & \text{otherwise} \end{cases} \tag{16}$$

$$d_\mu = \begin{cases} 1 + |\mu'_n - \tau_\mu| & \text{if} \quad \mu'_n > \tau_\mu \\ -|\mu'_n - \tau_\mu| & \text{otherwise} \end{cases} \tag{17}$$

d_ρ, d_H, and d_μ represent contributing values of correlation, color histogram and moments frame difference measures respectively. The three contributing values are combined to obtain aggregate frame difference measure "$d_{\rho H \mu}$" as:

$$d_{\rho H \mu} = w_\rho d_\rho + w_H d_H + w_\mu d_\mu \tag{18}$$

w_ρ, w_H, and w_μ denote the weight assigned to the contributing values of correlation, color histogram, and moments of inertia differences respectively. A high aggregate comparison value indicates a significant change between current frame and the last key frame. Therefore, if a frame's aggregate comparison value is higher than τ, it is declared as a key frame. Based on experimentation, the value of τ is selected to be 2. The maximum possible value for frame difference measure is 3 which signify the maximum change. The proposed aggregation mechanism and the integration of previous comparison value exaggerate the frame difference values, if there are significant content differences between frames. Also, the frame difference values are suppressed if the content change between frames is minimal. For too similar frames, the value of adaptive aggregation value is expected to be negative.

3.7. Elimination of useless frames

It has been generally observed that a video usually has some meaningless frames such as totally black frames, totally white frames, faded frames, etc. Some examples of meaningless frames are shown in Fig. 2. Since these frames are generally significantly different from a normal frame, it is quite likely that they get selected as key-frames. Therefore, as a post-processing step such frames are eliminated using a simple step which was used by Furini et al. [10] for the same purpose. After a frame is detected as a key frame, the standard deviation of pixels in the frame is computed. If the standard deviation is very low (close to zero) then that frame is considered as a meaningless frame and is discarded.

Once a key frame is selected, it is then compared with the next candidate frame. The process is repeated for all candidate frames.

After generation of the set of key frames, the redundancy is further reduced by removing those key frames which are very similar to each other. This is achieved by comparing each frame in the set of final key frames with every other key frame, based on the Euclidean distance between color histograms in HSV space. The key frames having more than 50% similarity with any other frames in the key frame set are removed [3].

4. Experiments and results

To evaluate the performance of the proposed technique, three sets of experiments were conducted: (1) the first set of experiments was carried out to demonstrate the benefits of the proposed aggregation mechanism; (2) the second set of experiments was performed to show some tradeoffs of varying system parameters; (3) lastly, the experimental results are shown for the proposed technique and compared with other techniques. The computer used for experimentation was an Intel 2.4 GHz with 2GB of RAM, and running a Windows XP Professional operating system.

4.1. Benefits of adaptive aggregation mechanism

In this sub-section, the benefits of the proposed adaptive aggregation mechanism are presented in comparison with simple non-adaptive aggregation of frame difference measures. In the non-adaptive aggregation scheme, the three frame difference measures, obtained from Eqs. (8)–(10), are aggregated linearly by addition. The results are presented on three sequences of frames selected from the videos downloaded from Open video Project (www.open-video.org). These three frames sequences have; (1) No significant change in inter-frame contents, (2) Intense change in the inter-frame contents, (3) Change in lighting conditions.

The first test sequence is taken from the second shot of the video 'Hurricane Force - A Coastal Perspective, segment 03'. The frames are shown in Fig. 3. The shot is characterized by the minimal changes in the contents, even though there are some small changes because of movement of body parts and camera motion. In this case, the numerical values of inter-frame differences must be very small. Table 1 (first two columns) shows the inter-frame differences yielded by both non-adaptive and adaptive comparison mechanism. It is evident that in non-adaptive scheme, with no

Fig. 2. Some examples of useless frames.

previous value consideration, the frame difference values are relatively high. On the other hand, the proposed adaptive scheme produced negative frame difference values which show non-significant changes in the content. Thus the proposed scheme successfully suppressed the frame difference values when there is minimal change in the inter-frame contents.

The next sequence of frames was also selected from the video 'Hurricane Force - A Coastal Perspective, segment 03' (shown in Fig. 4). The sequence is developed in a way to test the extreme changes in the inter-frame contents. In this case, the resultant frame difference values must be significantly high. Table 1 (last two columns) shows the inter-frame difference values for non-adaptive and adaptive aggregation mechanism for the sequence of frames in Fig. 4. It is evident that the adaptive mechanism yields a relatively high frame difference value than that of non-adaptive aggregation. Thus in case of high difference in the visual contents between frames, the adaptive aggregation mechanism boosts the frame difference values.

Finally, the efficacy of proposed system is tested on a sequence of frames with gradual change in lighting conditions. Fig. 5 displays frame number 3413–3428 of the video "Ocean floor Legacy, segment 04". It can be seen that there is no overall content change except the gradual change in lighting conditions. An effective frame difference calculation scheme must result in low difference values as the contents change is limited. However, because of change in lighting conditions, a simple aggregation will always result in relatively high values as can be seen in Table 2. The adaptive frame difference mechanism, on the other hand, results in low frame difference values. The change in gradual lighting conditions is effectively handled because of the inclusion of previous comparison value in the generation of current adaptive measure which leads to the generation of a smooth output function.

4.2. Tradeoffs in varying system parameters

To determine the tradeoffs in varying the values of thresholds, the technique was tested on 15 different videos containing a total of 18,846 frames. Each video was tested for a combination of values for number of sections (T_s), threshold values (τ_ρ,τ_H,τ_μ), and weight of previous comparison values (α_1,α_2,α_3). The thresholds and weights can be modified to control the quality and detail level of video summary.

Fig. 6(a) shows the change in fraction key frames (Number of key frames/Total frames) against different values of weights of previous comparison values. In this graph, the weights of previous comparison values α_1, α_2, and α_3 are assumed to be the same. It was observed that assigning equal weight to previous and current results gives the optimal value for the compression ratio. Increasing the value of alpha gives too much weight to the previous result, causing significant events to be missed. It can also be observed that if $\alpha = 0$, the fraction key frames value is higher which indicates redundancy in generation of key frames.

The graph in Fig. 6(b) shows the relation between fraction key frames and the threshold values. In this graph, the threshold values of τ_ρ, τ_H, and τ_μ are assumed to be the same. As expected, a more succinct summary is obtained as the threshold is made smaller. However, decreasing the threshold beyond a certain limit will start affecting the quality of the summary. The graphs in Fig. 6(c) show time taken (y-axis) against number of frames in videos for different frame divisions. The time taken is the greatest for 1×1 (no frame division), because the time taken by moments and correlation methods are not linear in number of pixels to process. Also the time taken increases rapidly with the increase in the size of individual sections.

4.3. Comparison with other techniques

We evaluated our technique based on the "Comparison of User Summaries" (CUS) mechanism proposed by Avila et al. [3]. In this evaluation scheme, the manually created user summaries are taken as reference to compare with the summaries generated by automated methods. The manually created summaries are the set of frames that are selected by humans as key frames after watching the video. Each key frame from the automatic summary (output of summarization algorithm) is compared with the frames of users' summaries. The comparison is done based on the Manhattan distance between the color histograms (in HSV color space) of the two frames. The two key frames are matched if the Manhattan distance between them is less than a certain threshold. The threshold value is taken as 0.5 as suggested in [3]. As per this mechanism, the quality of an automatically generated summary is determined by two metrics called Accuracy Rate (CUS_A) and Error Rate (CUS_E) which are defined as follows:

$$CUS_A = \frac{n_{mAS}}{n_{US}} \tag{19}$$

$$CUS_E = \frac{n_{m'AS}}{n_{US}} \tag{20}$$

Fig. 3. Sequence of frames with low inter-frame content changes.

Table 1
Frame difference values for frame sequences of Figs. 3 and 4.

Frame	Frame difference values for frames in Fig. 3		Frame difference values for frames in Fig. 4	
	Non adaptive	Adaptive	Non adaptive	Adaptive
1–2	1.18	−1.28	2.63	2.71
2–3	1.27	−1.21	1.91	2.44
3–4	1.47	−0.08	2.13	2.49
4–5	1.16	−0.21	2.39	2.76
5–6	1.06	−1.29	2.61	2.94
6–7	1.23	−1.26	2.26	2.81

Fig. 4. Sequence of frames with high inter-frame content change.

Fig. 5. Frames 3413–3428 for video "Ocean floor legacy, segment 04".

Table 2
Frame difference values for sequences of Fig. 5.

Frame no.	Non-adaptive difference value	Adaptive difference value
3413–3414	1.47	−1.32
3414–3415	1.44	−1.34
3415–3416	1.46	−1.34
3416–3417	1.53	−1.28
3417–3418	1.49	−1.30
3418–3419	1.65	−1.18
3419–3420	1.65	−0.15
3420–3421	1.50	−1.24
3421–3422	1.67	−0.14
3422–3423	1.72	−0.08
3423–3424	1.69	−0.08
3424–3425	1.70	−0.07
3425–3426	1.73	−0.04
3426–3427	1.78	0.01
3427–3428	1.82	0.06

wheren_{mAS} = number of matching key frames from automatic summary (AS)$n_{m'AS}$ = number of non-matching key frames from automatic summaryn_{US} = number of key frames from user summary

The value of CUS_A ranges from 0 to 1. The value 0 is the worst case where none of the key frames from automated summary matches with any of the user key frames, and the value 1 is the best case which means that all the key frames of automatic summary matches with the user key frames. The value of CUS_E ranges from 0 to n_{AS} /n_{US} (n_{AS} is the number of frames in automatic summary). The value 0 is the best value for CUS_E where no mismatch occurs between *AS* and user key frames, whereas n_{AS} /n_{US} is the worst value means none of the key frames are matched. The highest summary quality is achieved when $CUS_A = 1$ and $CUS_E = 0$. The summary having high CUS_A does not always mean a high quality summary until its CUS_E is sufficiently low. For instance, if a technique selects too many key frames from the video, then it is likely to have high CUS_A but low CUS_E.

The experiments were carried on two data sets that are made publically available by Avila et al. [3]. The user summaries, for each video in these data sets, are also available. The evaluation on a common data set helps in proper evaluation of the framework and makes possible a comparison with other techniques. The first data set consisted of 50 videos chosen from the Open Video Project (www.open-video.org). The second data set was comprised of videos of different genres collected from different web sites. Based on these data sets, we compared our technique with OV [15], DT [23], STIMO [10], and VSUMM [3]. A sampling rate of 1 frame/s was selected as was used by Avila et al. [3] for the same data set. Moreover, the values of α_1, α_2, and α_3 were selected to be 0.5 and the section size as 4×4. We have used the threshold values 0.45, 0.5, and 0.55 for τ_ρ, τ_H, and τ_μ respectively. Moreover weights w_ρ, w_H, and w_μ were selected as 1, 0.8, and 0.8 respectively.

Table 3 shows the mean values of Accuracy Rate (CUS_A) and Error Rate (CUS_E) for all techniques under consideration for the first data set. The results indicate that VSUMM achieved the highest Accuracy Rate and DT achieved the lowest Error Rate. VSUMM offer a high Accuracy Rate but has a relatively high Error Rate. DT generates much shorter summaries than the summaries produced by human users and thus DT summaries have a low Error Rate. However, this low Error Rate is achieved at the cost of low

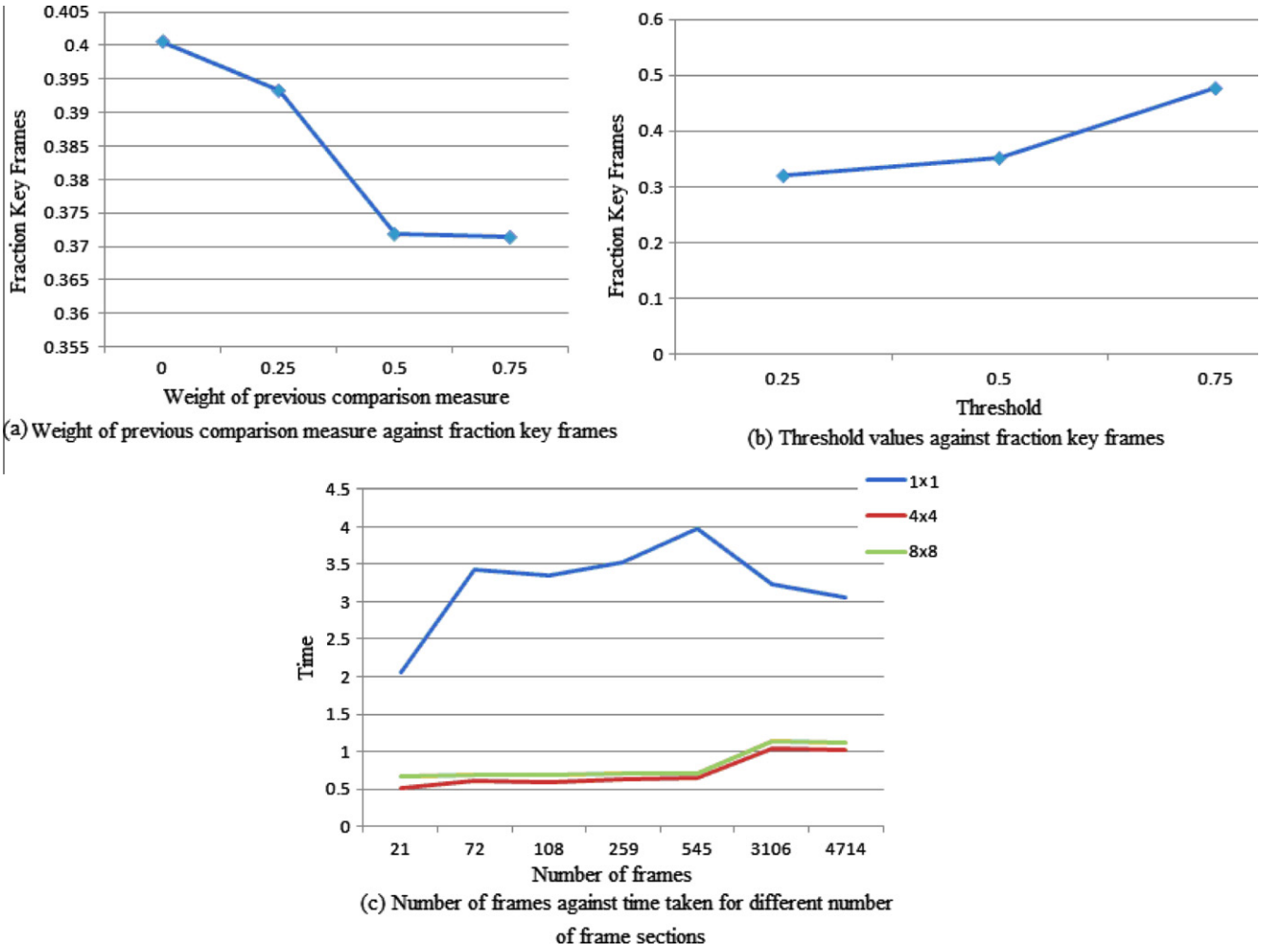

Fig. 6. Results for the computation of alpha, threshold, and number of sections.

Accuracy Rate. Our technique provides a balance between Accuracy and Error Rates by providing sufficiently high value of Accuracy Rate and low value for the Error Rate. Therefore, VSUKFE provides better results relative to the other techniques to which it was compared.

Table 4 shows the mean values of Accuracy Rate (CUS_A) and Error Rate (CUS_E) for VSUMM and VSUKFE for the second data set. This web data set contains videos of various genres (cartoons, news, sports, commercials, TV shows and home videos). According to these results, it can be seen that our technique has yielded a low Error Rate and its Accuracy Rate is comparable with VSUMM. Specifically, for sports videos, the Error Rate of VSUMM is fairly high as compared to VSUKFE. Both VSUMM and VSUKFE aim to generate concise summaries whereas the human users chose entire sequences of the moves for sports videos, thus leading to a higher Error Rate compared to other genres.

Fig. 7 shows the user summaries of the video "The Voyage of the Lee, segment 05" created by 5 human users. Fig. 8 shows the video summaries generated by various techniques under consideration.

Table 3
Comparison of results for first data set.

	OV	DT	STIMO	VSUMM	VSUKFE
CUS_A	0.70	0.53	0.72	0.85	0.80
CUS_E	0.57	0.29	0.58	0.38	0.32

By looking at the key frames visually, it can be observed that the summary generated by our technique is very close to that of the user summaries. For this video, it is observed that the highest value for Accuracy Rate is achieved by VSUMM and VSUKFE ($CUS_A = 0.9$) but the Error Rate for VSUMM ($CUS_E = 0.45$) is quite high compared to the Error Rate of VSUKFE ($CUS_E = 0.24$). For the rest of the techniques, the lowest Error Rate is achieved by STIMO ($CUS_E = 0.32$), but its Accuracy Rate is quite low ($CUS_A = 0.55$). Therefore, VSUKFE yields the best summary for this video based on its high Accuracy Rate and low Error Rate. This can also be observed based on the visual comparison of all the techniques.

From Table. 3 and 4, it is quite evident that the results of VSUKFE compare well with the VSUMM technique. Therefore, we compare the asymptotic time complexity of both the algorithms.

Table 4
Comparison of results for web data set.

Video genre	No. of videos	VSUMM		VSUKFE	
		CUS_A	CUS_E	CUS_A	CUS_E
Cartoons	10	0.87	0.22	0.83	0.2
News	15	0.88	0.32	0.85	0.25
Sports	17	0.76	0.65	0.75	0.48
Commercials	2	0.93	0.06	0. 9	0.1
TV-shows	5	0.91	0.33	0.83	0.3
Home	1	0.85	0.23	0.82	0.22
Weighted average	50	0.84	0.4	0.81	0.31

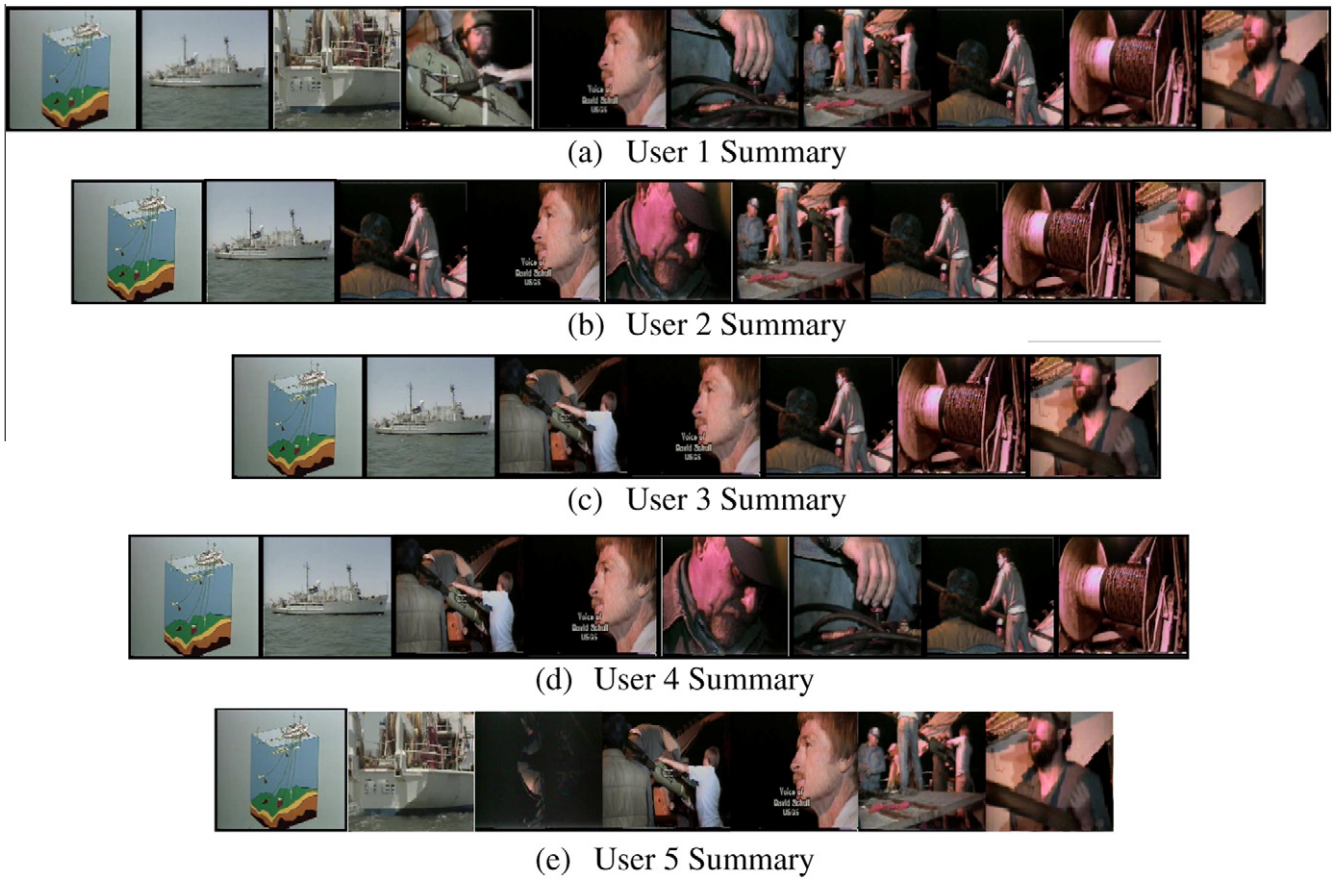

Fig. 7. User summaries for the video "The voyage of the lee, segment 05".

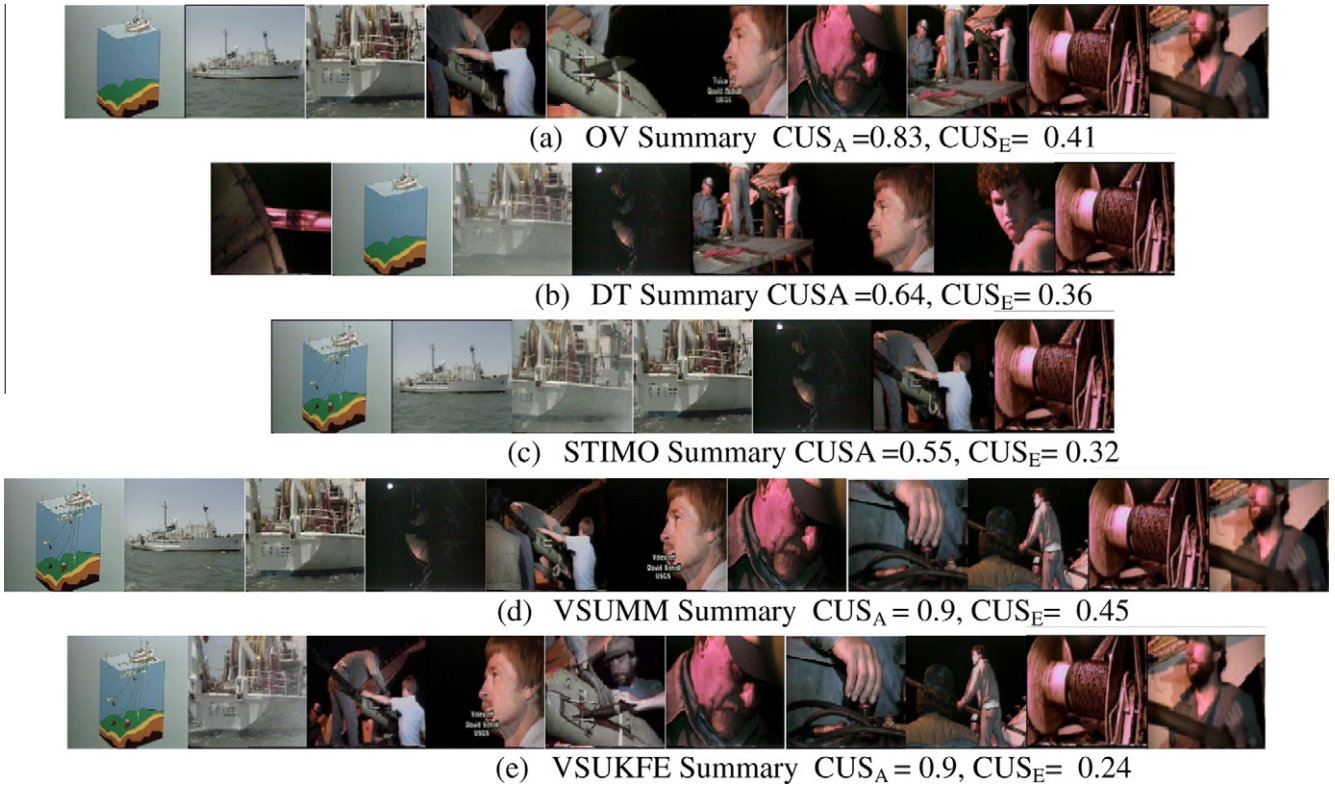

Fig. 8. Summaries generated by various techniques for the video "The voyage of the lee, segment 05".

We assume that total frames in the video are n_{NF}, total frames left after sampling are n_{CF}, and the number of key frames are n_{KF}. For simplicity, we assume that each frame has a dimension of mxm. The dominant step in VSUMM technique is k-means clustering whose time complexity is O(kOI) where k is the number of clusters, O is the number of objects to cluster and I is the number of iterations that k-means clustering takes [31]. Vattani [31] asserts that the lower bound on number of iterations of k-means clustering is $2^{\Omega(k)}$ which is exponential in k. Therefore the time complexity of k-means clustering is $O(kn_{CF}\ 2^{\Omega(k)})$ in VSUMM. The other main steps of VSUMM like color feature extraction, elimination of useless frames and computation of "k" have asymptotic time complexities of $O(n_{CF}m^2)$ each. Thus, the overall time complexity of VSUMM is $O(n_{CF}m^{2\ +}kn_{CF}\ 2^{\Omega(k)}))$ which is exponential in nature. The majors steps of VSUKFE includes the computation of correlation, histogram and moments, each of which takes $O(n_{CF}m^2)$. Therefore, according to the sum rule of computing asymptotic time complexities, the overall time complexity is $O(n_{CF}m^2)$. Thus, the quadratic time VSUKFE is thus more efficient than the exponential time VSUMM. However, if the number of iterations in k-means clustering is fixed, the time complexities of both algorithms are then comparable.

5. Conclusions

In this paper we presented a technique, VSUKFE, for the automated generation of video summaries by extracting key frames from the video. A single technique for frame difference measure is usually not enough to capture all visual features of a video frame. Our technique uses correlation of RGB color channels, color histogram comparison, and moments of inertia and adaptively combines them by using an aggregation mechanism. Our technique also works well in videos with multiple shots. The usage of adaptive aggregation mechanism reduces redundancy and is also tolerant to small changes in lighting conditions. We evaluated our technique on an objective evaluation criteria developed by Avila et al. [3]. Our technique works well by providing low Error Rate and, at the same time, it is able to achieve reasonably higher Accuracy Rate. The presented technique is also computationally efficient as compared to VSUMM [3]. In the future, we intend to add more visual features along with the three features which we already used.

Acknowledgment

This work was supported by the Industrial Strategic Technology Development Program (10041772, the Development of an Adaptive Mixed-Reality Space based on Interactive Architecture) funded by the Ministry of Knowledge Economy (MKE, Korea).

References

[1] A.F. Smeaton, Techniques used and open challenges to the analysis, indexing and retrieval of digital video, Information Systems 32 (4) (2007) 545–559.
[2] A.G. Money, H. Agius, Video summarisation: a conceptual framework and survey of the state of the art, Journal of Visual Communication and Image Representation 19 (2) (2008) 121–143.
[3] S.E.D. Avila, A.B.P. Lopes, L.J. Antonio, A.d.A. Araújo, VSUMM: a mechanism designed to produce static video summaries and a novel evaluation method, Pattern Recognition Letters 32 (1) (2011) 56–68.
[4] Y. Li, T. Zhang, D. Tretter, An overview of video abstraction techniques, Technical Report HP Laboratory, 2001, HP-2001-191.
[5] G. Ciocca, R. Schettini, Innovative algorithm for key frame extraction in video summarization, Journal of Real-Time Image Processing 1 (1) (2006) 69–88.
[6] B.T. Truong, S. Venkatesh, Video abstraction: a systematic review and classification, ACM Transactions Multimedia Computing, Communications and Applications. 3 (1) (2007).
[7] H.H. Kim, Y.H. Kim, Toward a conceptual framework of key-frame extraction and storyboard display for video summarization, Journal of the American Society for Information Science and Technology 61 (5) (2010) 927–939.
[8] N. Ejaz, U. Manzoor, S. Nefti, S.W. Baik, A collaborative multi-agent framework for abnormal activity detection in crowded areas, International Journal of Innovative Computing, Information and Control 8 (6) (2012).
[9] P.L. Venetianer, H. Deng, Performance evaluation of an intelligent video surveillance system- a case study, Computer Vision and Image Understanding 114 (11) (2010) 1292–1302.
[10] M. Furini, F. Geraci, M. Montangero, M. Pellegrini, STIMO: STIll and moving video storyboard for the web scenario, Multimedia Tools and Applications 46 (1) (2010) 47–69.
[11] Y. Li, S.-H. Lee, C.-H. Yeh, C.-C. Kuo, Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques, IEEE Signal Processing Magazine 23 (2) (2006) 79–89.
[12] R.M. Jiang, A.H. Sadka, D. Crooks (Eds.), Advances in video summarization and skimmingM. Grgic, K. Delac, M. Ghanbari (Eds.), Recent Advances in Multimedia Signal Processing and Communications, 231, Springer Berlin, Heidelberg, 2009, pp. 27–50.
[13] S.K. Pal, A.B. Leigh, Motion frame analysis and scene abstraction: discrimination ability of fuzziness measures, Journal of Intelligent and Fuzzy Systems 3 (1995) 247–256.
[14] A. Hanjalic, R.L. Langendijk, J. Biemond, A new key-frame allocation method for representing stored video-streams, in: 1st International Workshop on Image Databases and Multimedia, Search, 1996, pp. 67–74.
[15] D. DeMenthon, V. Kobla, D. Doermann, Video summarization by curve simplification, in: Proceedings of the ACM Internaternational Conference on Multimedia, NewYork, USA, 1998, pp. 211–218.
[16] A. Hanjalic, R.L. Langendijk, J. Biemond, A new method for key frame based video content representation, Image Databases and Multimedia Search, World Scientific, Singapore, 1998.
[17] S.H. Hoon, K. Yoon, I. Kweon, A new technique for shot detection and key frames selection in histogram space, in: 12th Workshop on Image Processing and Image Understanding, 2000, 217–220.
[18] L.M. Tieyan, X. Zhang, J. Feng, K.T. Lo, Shot reconstruction degree: a novel criterion for key frame selection, Pattern Recognition Letters 25 (12) (2004) 1451–1457.
[19] M.M. Yeung, B.-L. Yeo, Video visualization for compact presentation and fast browsing of pictorial content, IEEE Transactions on Circuits and Systems for Video Technology 7 (5) (1997) 771–785.
[20] Y. Zhuang, Y. Rui, T.S. Huang, S. Mehrotra, Adaptive key frame extraction using unsupervised clustering, in: International Conference on Image Processing, 1998, 866-870.
[21] N.D. Doulamis, A.D. Doulamis, Y.S. Avrithis, K.S. Ntalianis, S.D. Kollias, Efficient summarization of stereoscopic video sequences, IEEE Transactions on Circuits and Systems for Video Technology 10 (4) (2011) 501–517.
[22] X.D. Zhang, T.Y. Liu, K.T. Lo, J. Feng, Dynamic selection and effective compression of key frames for video abstraction, Pattern Recognition Letters 24 (9–10) (2003) 1523–1532.
[23] P. Mundur, Y. Rao, Y. Yesha, Keyframe-based video summarization using delaunay clustering, International Journal on Digital Libraries 6 (2) (2006) 219–232.
[24] J. Calic, B. Thomas, Spatial analysis in key-frame extraction using video segmentation, in: Proceedings of Workshop on Image Analysis, Multimedia Interactive Services, Portugal, 2004.
[25] J. Luo, C. Papin, K. Costello, Towards extracting semantically meaningful key frames from personal video clips: from humans to computers, IEEE Transactions on Circuits and Systems for Video Technology 19 (2) (2009) 289–301.
[26] M. Guironnet, D. Pellerin, N. Guyader, P. Ladret, Video summarization based on camera motion and a subjective evaluation method, EURASIP Journal on Image and Video Processing, 12 2007.
[27] F. Chen, D. Delannay, C. Vleeschouwer, An autonomous framework to produce and distribute personalized team-sport video summaries: a basketball case study, IEEE Transactions on Multimedia 13 (6) (2011) 1381–1394.
[28] N. Ejaz, S.W. Baik, Weighting low level frame difference features for key frame extraction using fuzzy comprehensive evaluation and indirect feedback relevance mechanism, International Journal of the Physical Sciences 6 (14) (2011) 3377–3388.
[29] M.J. Swain, D.H. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.
[30] J. Flusser, T. Suk, B. Zitove, Moments and Moment Invariants in Pattern Recognition, Wiley & Sons Ltd, 2009.
[31] A. Vattani, K-means requires exponentially many iterations even in the plane, in: Symposium on, Computational Geometry, 2009, 324–33.