Video Key-Frame Extraction using Unsupervised Clustering and Mutual Comparison

Nitin J. Janwe *nitinj_janwe@yahoo.com*
PhD Student, Department of Information Technology
Yeshwantrao Chavan College of Engineering
Nagpur, 441110, India

Kishor K. Bhoyar *kkbhoyar@yahoo.com*
Professor, Department of Information Technology
Yeshwantrao Chavan College of Engineering
Nagpur, 441110, India

Abstract

Key-frame extraction is one of the important steps in semantic concept based video indexing and retrieval and accuracy of video concept detection highly depends on the effectiveness of key-frame extraction method. Therefore, extracting key-frames efficiently and effectively from video shots is considered to be a very challenging research problem in video retrieval systems. One of many approaches to extract key-frames from a shot is to make use of unsupervised clustering. Depending on the salient content of the shot and results of clustering, key-frames can be extracted. But usually, because of the visual complexity and/or the content of the video shot, we tend to get near duplicate or repetitive key-frames having the same semantic content in the output and hence accuracy of key-frame extraction decreases. In an attempt to improve accuracy, we proposed a novel key-frame extraction method based on unsupervised clustering and mutual comparison where we assigned 70% weightage to color component (HSV histogram) and 30% to texture (GLCM), while computing a combined frame similarity index used for clustering. We suggested a mutual comparison of the key-frames extracted from the output of the clustering where each key-frame is compared with every other to remove near duplicate key-frames. The proposed algorithm is both computationally simple and able to detect non-redundant and unique key-frames for the shot and as a result improving concept detection rate. The efficiency and effectiveness are validated by open database videos.

Keywords: Key-frame Extraction, Semantic Concept Based Video Retrieval, HSV Histogram, GLCM Texture.

1. INTRODUCTION

The advancement in multimedia technology and network technology results into more and more multimedia data being produced and distributed. Of all the media types (text, image, graphic, audio and video), video is the most challenging one, as it combines all the media information into a single stream. Video data contains more intuitive and richer information which is closer to the impression of real world in the human brain. We require efficient methods to retrieve, browse and indexing of the video data [1], as the videos are available in abundance nowadays. However, efficient access to video is a very difficult task due to substantially different nature of video data like video length and unstructured format. We require abstraction and summarization techniques to overcome this problem. Video segmentation also called shot boundary detection and key-frame extraction are the bases of video abstraction and summarization. We find a good amount of research carried out on shot boundary detection and key-frame extraction.

A video shot is an uninterrupted stream of video frames captured by a camera. The purpose of any shot boundary detection method is to divide the video sequence into multiple shots [2]. After videos are segmented into shots, key-frames can be extracted. Key-frame extraction methods convert video processing problem to image processing problem. A key-frame is supposed to be a representative key-frame for a shot and is defined as the frame which best reflects the shot contents. Mostly, the middle frame of a shot is taken as a key-frame, assuming that middle segment of a shot contains key contents, but many more other techniques do exist by which a key-frame is identified. It is not necessary that a shot is always represented by a single frame; in some cases; however, depending on visual complexity of the shot, multiple key-frames can be required to represent a single shot. Key-frames provide a suitable framework for video browsing and retrieval. The basic framework of the key-frame extraction algorithm is shown in figure 1. Key-frames can also be used to find an index for a shot. The use of key-frames significantly reduces the amount of information required in video indexing.

Since effective shot boundary detection algorithms exist in the literature [3-5] and because of the importance, we will focus on key-frame extraction technique. Although progress has been made in this area, but most of the current approaches do not effectively capture the diverse visual content. In this paper we present a clustering based approach which is both effective and efficient.

The reminder of the paper is organized as follows. In section 2, key-frame extraction procedure is reviewed and discussed. The proposed approach based on unsupervised clustering and mutual comparison is presented in section 3. Experimental results over large open data set has been given in section 4 and conclusion is presented in section 5.

2. KEY-FRAME EXTRACTION

2.1 Shot Boundary Detection

Shot boundary detection or video segmentation is the first step of key-frame extraction, which is the main task in many applications like video indexing, video retrieval and video browsing. The video shot is the basic unit of the video stream and is an unbroken string of frames taken by a single camera uninterruptedly. During editing stage, shots are joined together using hard cuts or various gradual transitions like dissolve, fade-in, fade-out, wipe etc. The procedure of detecting the shot transition within a video sequence is known as shot boundary detection or video segmentation. The shot boundary detection methods are categorized into cut boundary detection and gradual transition detection [6].

The important task in any shot boundary detection method in a video stream consists of detecting frame discontinuities. Here, it is essential to extract visual features such as the features based on color [7], shape, texture [8-9] and motion or their combination that measures the similarity between frames. This measure, *g(i, i+k),* gives the difference or discontinuity between frames *i* and *i+k* where *k ≥1*. To compute *g(i,i+k)* different alternatives exists in a video sequence, the simplest is the absolute difference between frame and is given by following equation:

$$g(i, i+k) = \sum_{x,y} (| I_i(x,y) - I_{i+k}(x,y) |) \tag{1}$$

where *I(x,y)* is the intensity level of the image at *(x,y)* pixel position. The methods based on absolute difference compares the difference with the set threshold to test significant difference in frame sequence. However, *g(i,i+k)*, the measure of discontinuity, is very sensitive to intensity variation or object and camera motion and may result into increased ratio of false detection.

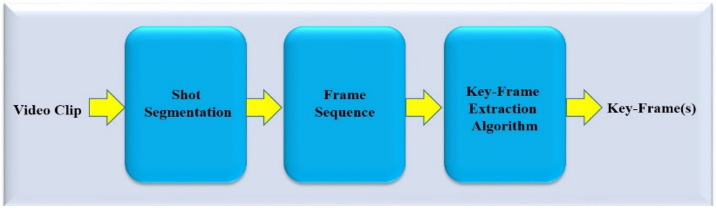

FIGURE 1: The basic framework of the key-frame extraction algorithm.

The major methods that have been used for shot detection are pixel-difference methods, statistical-difference methods, histogram comparisons, edge-differences and motion vector methods. In order to compute the frame difference, the most preferred method is the histogram-based method. Histograms represent the color distribution of a frame. Shot detection efficiency of a method depends on the suitable selection of the similarity measure between successive frames. Color histogram techniques are independent to object motion within a frame sequence. Conventional histogram (RGB, HSV etc.) based techniques are shown to be robust and effective [10]. The color histograms of two images are computed and their similarity is computed using histogram intersection technique. If the similarity between two histograms is below a certain threshold, a shot boundary is assumed. The problem with color histograms is that, images with similar histograms can have different visual appearance.

2.2 Key-Frame Extraction

After the video stream undergoes shot segmentation, the frames in a shot are very much similar to each other; therefore we need to select a key-frame that best reflect the contents of a shot [11]. Current key-frame extraction approaches are categorized into six categories [12]: sequential comparison-based, global comparison-based, reference frame-based, clustering-based, curve-simplification-based, and object/event-based.

1) Sequential comparison-based: In this approach, frames subsequent to previously extracted key-frame are sequentially compared with the key-frame until the much dissimilar frame is obtained and this frame is selected as the next key-frame [13-14]. The advantages of sequential comparison-based algorithms are the simplicity, low computational complexity, and adapting a number of key-frames for a shot. The limitations of these algorithms include 1) The key-frames represent local properties of the shot rather than global 2) The key-frames are irregularly distributed and number of key-frames is variable making the algorithm unsuitable for some applications and 3) There is a chance of redundancy among the key-frames if the content occurs repeatedly.
2) Global comparison-based: The algorithm based on this approach distributes key-frames by minimizing a predefined objective function depending on the application. In general, the objective function has one of the following four forms [12] 1) Even temporal variance 2) Maximum coverage 3) Minimum Correlation 4) Minimum reconstruction error. The merits of the approach are 1) The key-frames reflect the global characteristics 2) The number of key-frames are limited and 3) Redundancy is minimum among the key-frames. The limitation is that it is comparatively more computationally expensive.
3) Reference frame-based: Here a reference frame is generated and then does key-frames extraction by comparing the shot frames with the reference frame [15]. These algorithms are easy to understand and implement but the accuracy of key-frames depends on the accuracy of a reference frame.
4) Clustering: In this approach, shot frames are clustered and then select a frame closest to the cluster center as a key-frame. Yu et al.[16] used fuzzy k-means clustering in the color feature subspace to extract key-frames. The most important advantages of these methods are, the extracted key-frames reflect the global characteristics of a video shot while limitations are the accuracy of extraction are dependent on the accuracy of the clustering results.
5) Curve simplification-based: In these algorithms, each frame in a shot is represented as a point in a feature space and they are linked sequentially to get a trajectory curve. It is then searched to find a group of point that best represent the shape of a curve. The advantage of these algorithms is that the sequential information is maintained during the key-frame extraction process. And limitation is, to get optimization of the best representation of the curve incurs high computational complexity.
6) Objects/Events: In many video processing applications, we might be interested in some objects or events from a shot. These algorithms [11] first detect the object or event we are looking for and then perform key-frame extraction so that the extracted key-frames contain information about required objects or events. The merit of the object/event-based algorithms is that the extracted key-frames contain semantically rich information; the

limitation is that object/event detection strongly relies on heuristic rules specified as per the nature of the application.

It is noted that, there is no uniform evaluation method available for key-frame extraction because of the subjectivity of key-frame definition.

3. CLUSTERING BASED APPROACH AND PROPOSED METHOD

Figure 2 shows the proposed key-frame extraction method based on unsupervised clustering and mutual comparison. Clustering is a very effective technique used in many areas like pattern recognition and information retrieval. Clustering can be categorized into two types namely supervised and unsupervised. Supervised clustering is useful when it is priory known the number of clusters to be formed and when number is uncertain unsupervised clustering is useful. An approach was introduced in [12] to extract key-frames from a shot boundary using unsupervised clustering. In supervised clustering, given a video shot $S = \{ f_1,f_2,.....f_N \}$ obtained from a shot boundary detection algorithm [13]. We cluster the N frames into M clusters, say, σ_1, σ_2,....,σ_M. The salient content of any object or a frame is defined as the visual content of that object or a frame which could be color, texture or shape of the object or a frame. The similarity between two frames is determined by computing the similarity of their visual content. In this paper, we select the weighted combination of the color and texture components of a frame to represent visual content. The color feature we used is global level histogram in the HSV color space and texture feature is GLCM. In GLCM, the actual features used are 1. Contrast 2. Correlation 3. Energy and 4. Homogeneity. After computing these features, next step is to find the similarity index between the frames i and j for HSV histogram and GLCM texture features respectively. The histogram similarity can be computed by histogram intersection method using equation (2) as summation of *min* values of color bins. The histogram similarity index between frames i and j is thus defined as:

$$Simi_{hsv} = \sum_{i=1}^{C} min\left(h_x(i), h_y(i)\right) \tag{2}$$

where h_x and h_y are HSV histograms for frames x and y respectively and C is the number of color bins in the histogram. $Simi_{hsv}$ gives us the total pixel count common in both the frames. The frame similarity index using GLCM texture features can be computed using Euclidean distance method

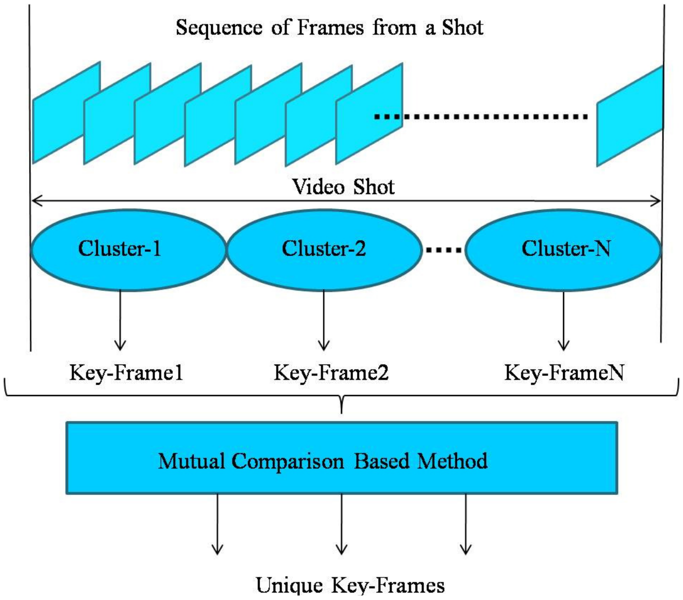

FIGURE 2: Proposed key-frame extraction method.

using equation (3) as follows:

$$Simi_{glcm} = \sqrt{\sum\left(glcm_{x-}glcm_{y}\right)^{2}} \tag{3}$$

where *glcm*$_x$ and *glcm*$_y$ are the GLCM texture features for the frames *x* and *y* respectively. When we select features, all are not equally effective to represent the salient visual content of a video frame. Therefore we need to properly assign weights to such features depending on their importance. We have assigned 70% weightage to color histogram and 30% to GLCM texture with an understanding that color component represents major salient content than texture. Our next step is to merge these features and find the combined frame similarity index, *Combine*$_{Simi}$, between the frames *x* and *y*, using equation (4) as follows:

$$Combine_{Simi} = \left(Simi_{hsv} * 0.7 + Simi_{glcm} * 0.3\right) \tag{4}$$

Using the frame similarity index, *Combine*$_{Simi}$, clustering operation is carried out. Any clustering algorithm has a threshold parameter *δ*, which controls the density of clustering. The higher the *δ*, the more the number of clusters. In human learning and recognition system we also have this threshold. For example, if the threshold is low, we will classify cars, wagons, mini-vans as vehicles; however, if the threshold is high, we will classify them into different categories. The threshold parameter provides us a control over the density of classification. Before a new frame is classified into a certain cluster, the similarity between this node and the centroid of the cluster is computed. If this value is less than *δ*, it means this node is not close enough to be added into the cluster.

The unsupervised clustering algorithm is summarized as follows:

1. Initialization: $f_1 \rightarrow \sigma_1$, $f_1 \rightarrow$ the centroid of σ_1 (denoted as c_{σ_1}), 1 → *numCluster* ;
2. Get the next frame f_i. If the frame pool is empty, goto 6;
3. Compute the similarities between f_i and existing clusters σ_k (k = 1,2, …….,*numCluster*): *simi*(f_i,σ_k), based on equation (4);
4. Determine which cluster is the closest to f_i by calculating *Maxsimi*. Let
   Maxsimi = $max_{k=0}^{numCluster}$ *Simi*(f_i,σ_k).
   If *Maxsimi* < δ, it means that f_i is not close enough to be put in any of the clusters, goto 5; otherwise, put f_i into the cluster which has *Maxsimi*, and goto 6.
5. *numCluster* = *numCluster* + 1. A new cluster is formed: $f_i \rightarrow \sigma_{numCluster}$.
6. Adjust the cluster centroid: Suppose the cluster σ_k's old centroid is c'_{σ_k}, *D* is the number of frames in it, the new centroid is c_{σ_k}, thus $c_{\sigma_k} = D/(D+1)c'_{\sigma_k} + 1 = (D+1)\ f_i$. goto 2.

After the clusters are formed, the next step is to select key-frame(s). Here, we select only those clusters which are big enough and are considered as key clusters, a representative frame is extracted from this cluster as the key-frame. In this paper, we say a cluster is big enough if its frame count is greater than, *min_clust_size* = 10% of total frames in a shot where *min_clust_size* is the minimum size of a cluster. The key-frame for each corresponding key cluster is the one which is closest to the cluster centroid, and it is supposed to capture the salient visual content of the key cluster of the underlying shot. If we decrease *min_clust_size*, number of clusters will be increased and over-segmentation may result and if it is increased, under-segmentation may result.

Once we extract key-frames from the clustering, we find that, there are some near duplicate key-frames which are very similar in appearance to each other, which might be a result of over-segmentation. If we try to decrease *min_clust_size* to remove over-segmentation, it usually results into under-segmentation. Therefore to handle such situation, we need some mechanism whereby these near duplicate key-frames in the same shot can be removed. Our strategy is mutual comparison, where we compare each key-frame with every other and find the similarity. If the similarity value is greater than certain threshold, key-frame is considered as duplicate key-frame and is removed.

4. EXPERIMENTAL RESULTS AND ANALYSIS

Various test videos are downloaded from the standard video library dataset Open Video to investigate the performance of the proposed approach. Ten different shots from three video clips with different characteristics are selected for experimentation.

The first three shots with shot numbers 1, 3 and 4 are taken from video, *Anni003.mpg.* Shot number 1 and 3 are of video characteristics *little change* and *object motion* respectively while shot number 4 is of *object motion & high variation in brightness* characteristics.

The next four shots are taken from the video clip *Indi009.mpg*, numbered as 1, 2, 3 and 4. The shot number 1 belongs to *fast camera movement* where the car moves forward rapidly crossing two cars coming in opposite direction and background scenery also moves in opposite direction, shot number 2 belongs to *camera and object motion* giving a feel of stationary object, shot number 3 is with *fast camera motion* and shot number 4 belongs to *object motion* as the car coming from one direction crosses and moves in opposite direction.

Last three shots are taken from a video clip named *Enviro.mp4*, with shot numbers 1, 31 and 40. In shot number 1, the frame has been divided into 4 parts, and as it begins; it starts with picture appearing in top left corner of the screen, then gradually in the top right corner then to the bottom left corner and finally bottom right corner. This shot belongs to *special effects*. Shot number 3 displays a moving whale and shot number 40 consists of a scene of a meeting where four persons are discussing. When camera moves from one person to next, there is an *abrupt change* in switching over the persons.

Table 1 shows the key-frame extraction results for *Anni003.mpg* for threshold *δ*=0.60, *δ*=0.65, δ=0.70 and *δ*=0.80, Table 2 presents the results for *Enviro.mp4* when *δ*=0.60, *δ*=0.65, *δ*=0.70 and *δ*=0.80. The final results are taken using *δ*=0.80, thereafter if *δ* is increased, over-segmentation results and near duplicate key-frames will get increased. Table 3 gives detailed results we obtained i.e. number of key-frames extracted from each sample shot when (unsupervised) clustering is applied, we can observe the redundancy amongst the key-frames post clustering and when these key-frames are mutually compared with each other, the near duplicate key-frames, if any, from a shot get removed or minimized and unique key-frames are obtained.

Shot-ID	Shot Activity	δ=0.60	δ=0.65	δ=0.70	δ=0.80
		K-frames	K-frames	K-frames	K-frames
1 (1-70)	Low	2	2	2	37, 64
3 (215-257)	High	216	216	216, 230	218, 224, 230, 252
4 (258-528)	High	339	339	339	286, 323, 349, 387, 424, 475

TABLE 1: Example from Anni003.mpg.

Shot-ID	Shot Activity	δ=0.60	δ=0.65	δ=0.70	δ=0.80
		K-frames	K-frames	K-frames	K-frames
1(1-248)	Low	127	161, 238	42, 228	41, 178, 186
31(1624-1647)	High	1646	1646	1629,1645	1627,1629, 1635,1641
40(2139-2207)	High	2141, 2175	2141, 2196, 2205	2141, 2181, 2205	2141, 2175, 2181, 2205

TABLE 2: Example from Enviro.mp4.

Video Name	Shot No.	Frame Count in a shot	Number of Key-Frames Extracted using Clustering	Number of Key-Frames Extracted using Proposed Method	Video characteristics
Anni003.mpg	1	70	2	1	Little change
	3	43	4	4	Object Motion
	4	271	6	2	Object Motion & high variation in brightness
Indi009.mpg	1	171	9	2	Fast Camera Motion
	2	147	3	1	Camera & Equal Object Motion (moving but seems stationary)
	3	95	3	1	Fast camera Motion
	4	91	5	2	Object Motion
Enviro.mp4	1	248	3	3	Special Effects
	31	24	4	3	Object Motion
	40	69	4	3	Abrupt change

TABLE 3: Comparison of the key-frames extracted from ten sample shots of three video sequences using unsupervised clustering algorithm and using proposed method with mutual comparison.

For low activity shots, the number of clusters formed will be less and hence extracted key-frames will be less or even a single key-frame, while for high activity shots; the clusters will be more for a shot having more visual complexity and extract multiple key-frames depending on the clusters (Table 1 and 2). Figure 3 and figure 5 shows the key-frame extraction results when unsupervised clustering is applied and figure 4 and figure 6 shows the extracted key-frames when proposed method using mutual comparison is applied. It is observed that, in figure 3, shot number 4, the near duplicate key-frame nos. 286, 323, 349, 387, 424, 475 are a result of over-segmentation and this over-segmentation is a result of high variation of brightness in a shot. After applying mutual comparison as shown in figure 4, only two most prominent key-frames remained, rest of the frames have been filtered out. Likewise, in figure 5, for shot number 40, the duplicate key-frames 2175 and 2181 are a result of text scrolling which has been removed and shown in figure 6. It is observed that the near duplicate key-frames have been removed or minimized from the output.

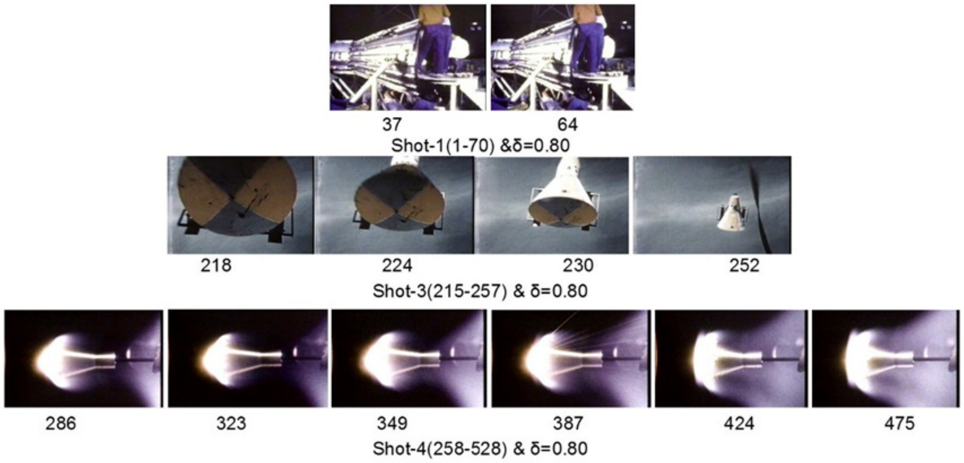

FIGURE 3: The key-frame extraction results from video sequence of Anni003.mpg using unsupervised clustering.

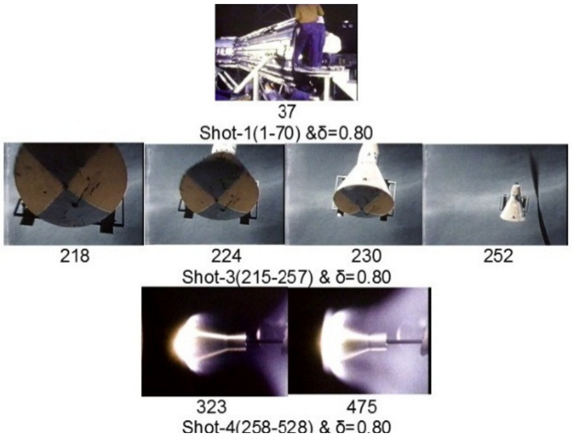

FIGURE 4: The key-frames remained after applying the mutual comparison step for video sequence Anni003.mpg.

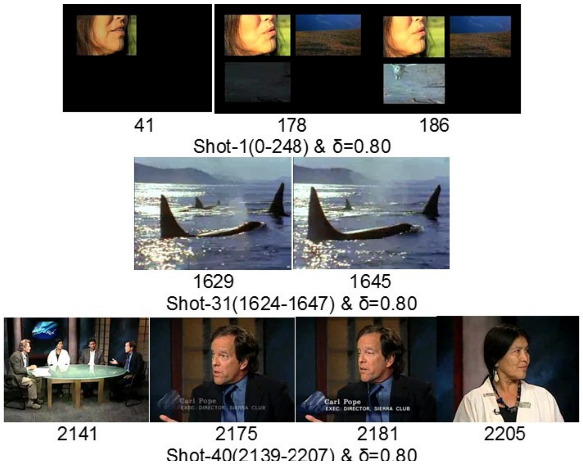

FIGURE 5: The key-frame extraction results from video sequence of Enviro.mp4 using clustering.

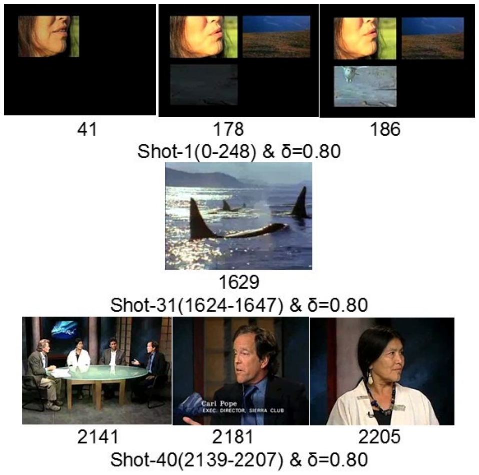

FIGURE 6: The key-frames remained after applying the mutual comparison step for video sequence Enviro.mp4.

From above simulation of our proposed algorithm, it is observed that, the number of resulting key-frames are less for low activity shots as compared to high activity shots or shots having more visual complexity. It is also noticed that the resulting near duplicate key frames are successfully removed in the mutual comparison phase and as a result the effectiveness of the video abstraction increases.

Because of the absence of well-defined objective criteria, some subjective evaluation criteria are specified to check perception of users towards video summary [18, 19]. There are no benchmarking or ground truth results for key-frame extraction algorithm so far. And therefore, we do not perform any direct results comparison between our proposed algorithm and existing algorithms.

But, on the basis of following characteristics, we have compared our algorithm with those of six other key-frame extraction algorithms compared by G. Ciocca1 et al. [20] and also with the algorithm by M. Mentzelopoulos [21]. Table 4 presents comparison of our proposed method with Adaptive Temporal Sampling (ATS) algorithm of Hoon et al. [22], the Flexible Rectangles (FR) algorithm of Hanjalic et al. [23], the Shot Reconstruction Degree Interpolation (SRDI) algorithm of H. Chang et al. [24], the Perceived Motion Energy (PME) algorithm of T. Liu et al. [25], a simple Mid-Point (MP) algorithm, Curvature Points (CP) algorithm of G. Ciocca1 et al. [20], and Entropy Distance (ED) algorithm of M. Mentzelopoulos et al. [21] and by S. Algur [26].

Table 4 summarizes some of the important characteristics of the algorithms compared. The first row regards the most important property of any key-frame extraction algorithm which automatically does key-frame selection. Next characteristic for any good key-frame extraction algorithm is to select variable number of key-frames depending on the visual complexity or semantic visual content of the given video sequence. Third characteristic is on-the-fly processing which means the ability of the algorithm to determine key-frames without having to process all the frames in a shot. Real time processing is the ability of the algorithm to extract key-frames from the incoming raw video frames of the capturing camera. Some algorithms require mpeg videos which are in encoded and compressed format and embed motion vectors into it. Next parameter tells us whether the key-frame extraction algorithm requires any optimization algorithm. If the number of key-frames is dependent on the shot length then it is called shot length sensitive algorithm.

	ATS	**FR**	**SRDI**	**MP**	**PME**	**CP**	**ED**	**Proposed Approach**
Automatic key frames selection	N	N	N	Y	Y	Y	Y	Y
Variable number of key frames	Y	Y	Y	N	Y	Y	Y	Y
On-the-fly processing	N	N	N	Y	N	Y	Y	Y
Real time processing	Y	Y	N	Y	N	Y	Y	Y
Requires motion vectors	N	N	Y	N	Y	N	N	N
Uses an optimization algorithm	N	Y	N	N	N	N	N	N
Shot length sensitive	?	Y	N	N	Y	N	N	N
Reference	[22]	[23]	[24]	-	[25]		[21]	

TABLE 4: Comparison of the seven key-frame extraction algorithms and our proposed algorithm for some important characteristics.

From Table 4, it is observed that, our proposed algorithm is better than ATS, FR, SRDI, MP and PME algorithms and at par with CP and ED algorithms.

5. CONCLUSION AND FUTURE WORK

This paper proposes a novel key-frame extraction method based on unsupervised clustering and mutual comparison, in which an attempt has been made to remove near duplicate or redundant key-frames with the similar semantic content, if any, arising due to over segmentation because of the visual complexity and/or nature of the video shot and hence to improve video summarization.

The proposed key-frame extraction method performs well in terms of removing repetitive key-frames and getting non-redundant and unique key-frames as shown in Table 3 and it is noticed that, the number of resulting key-frames are less for low activity shots as compared to key-frames for high activity shots or shots having more visual complexity. As the number of extracted key-frames are well balanced, it gives best video summarization and also improves video concept detection and video retrieval accuracy.

The efficiency and effectiveness of the proposed approach is as follows:

- *Efficiency*: Easy to implement and fast to compute. We need to compute HSV histograms and GLCM texture features and requires computations to find out the combined frame similarity index by assigning 70% weightage to HSV histogram frame similarity value and 30% to GLCM texture frame similarity value.

- *Effectiveness*: The visual features used for unsupervised clustering are so effective that the proposed approach is able to capture the salient visual content of the key clusters and that of the underlying shot. The selection of the key-frames is based on the frame count i.e. the size of the cluster and hence inherently depends on the visual complexity of the shot. In the proposed method while clustering (for eligibility to be a cluster) it must contain minimum set number of frames. No location based like first frame [17] or the middle frame is used to extract key-frame rather complex shot results into multiple key-frames (high activity).

- Real-time video processing: Since this approach uses current and incoming frame, it can be easily implemented for real-time video processing applications.

In future work, we suggest deep convolution features to be used for unsupervised clustering for frames in a video shot which would replace the features used in this work (HSV histogram and GLCM texture). Rest of the process i.e. mutual comparison remains same.

6. REFERENCES

[1] J. Son, H. Lee, and H. Oh. “PVR: a novel PVR scheme for content protection.” IEEE Transactions on Consumer Electronics, vol. 57, no. 1, pp. 173-177, 2011.

[2] M. Naphade, A. Ferman, and et al. “A high performance algorithm for shot boundary detection using multiple cues.” In Proc. ICIP, Chicago, Oct. 1998, pp. 884-887, vol.1.

[3] J. Boreczky and L.Rowe. “Comparison of video shot boundary detection techniques.” Journal of Electronic Imaging, 5(2), pp. 122-128, Apr. 1996.

[4] H. Zhang, J. Wang, and Y. Altunbasak. “Content-based video retrieval and compression: A unified solution.” in Proc. IEEE Int. Conf. on Image Proc., 1997, pp. 13-16, vol.1.

[5] N. Janwe and K. Bhoyar. “Video Shot Boundary Detection based on JND Histogram.” In Proc. of ICIP, Image Information Processing, Second IEEE conference on, 2013, pp. 476-480.

[6] I. Koprinska, S. Carrato. “Temporal video segmentation: A Survey.” Signal processing: Image Communication, vol. 16, no. 5, pp. 477-500, 2001.

[7] J. Mas and G. Fernandez. “Video shot boundary detection based on color histogram.” Digital Television Center (CeTVD) La Salle School of Engineering, Ramon Llull Univ., Barcelona, Spain, In TREC2003 Video Track.

[8] R. Haralick, K.Shanmugam, I. Dinstein. "Textural features for image classification." IEEE Trans. Systems Man Cybernet. SMC-3, pp. 610 – 621, 1973.

[9] G. LaxmiPriya, S.Domnic. "Transition detection using Hilbert transform and texture features." American Journal of Signal Processing, vol. 2 (2), pp. 35-40, 2012.

[10] J. Boreczky and L. Rowe. "Comparison of video shot boundary detection techniques." In Storage and Retrieval for Image and Video Databases (SPIE), vol. 2670, pp.170-179, 1996.

[11] K. Sze, K. Lam, and G. Qiu. "A new key frame representation for video segment retrieval." IEEE Trans. Circuits Syst. Video Technol., vol.15, no.9, pp.1148–1155, Sep.2005.

[12] B. Truong and S. Venkatesh. "Video abstraction: A systematic review and classification." ACM Trans. Multimedia Comput., Commun. Appl., vol.3, no.1, art. 3, pp.1–37, Feb.2007.

[13] H. Zhang, J. Wu, D. Zhong, and S. Smoliar. "An integrated system for content-based video retrieval and browsing." Pattern Recognit., vol.30, no.4, pp.643–658, 1997.

[14] X. Zhang, T. Liu, K. Lo, and J. Feng. "Dynamic selection and effective compression of key frames for video abstraction." Pattern Recognit. Lett., vol.24, no.9–10, pp.1523–1532, Jun. 2003.

[15] A. Ferman and A. Tekalp. "Two-stage hierarchical video summary extraction to match low-level user browsing preferences." IEEE Trans. Multimedia, vol.5, no.2, pp.244–256, Jun. 2003.

[16] X. Yu, L. Wang, Q. Tian, and P. Xue. "Multilevel video representation with application to keyframe extraction." In Proc. Int. Multimedia Modelling Conf., 2004, pp.117–123.

[17] A. Nagasaka and Y. Tanaka. "Automatic video indexing and full-video search for object appearances." In Visual Database Systems II, 1992.

[18] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini. "STIMO: STIll and MOving video storyboard for the web scenario." Multimedia Tools and Applications, vol.46, no.1, pp.47–69, 2010.

[19] N. Ejaz, I. Mehmood, and S.Baik. "Efficient visual attention based framework for extracting key frames from videos." Signal Processing: Image Communication, vol.28, pp.34–44, 2013.

[20] G. Ciocca and R. Schettini. "An innovative algorithm for key frame extraction in video summarization." Journal of Real-Time Image Processing, Mar. 2006, vol. 1, issue 1, pp. 69-88.

[21] M. Mentzelopoulos, and Alexandra Psarrou. "KeyFrame Extraction Algorithm using Entropy Difference." ACM MIR, pp. 39-45, Oct. 2004.

[22] S. Hoon, K. Yoon, and I. Kweon. "A new Technique for Shot Detection and Key Frames Selection in Histogram Space." Proc. 12th Workshop on Image Processing and Image Understanding, 2000, pp. 475-479.

[23] A. Hanjalic, R. Lagendijk, J. Biemond. "A new Method for Key Frame Based Video Content Representation." In Image Databases and Multimedia Search, World Scientific Singapore, 1998.

[24] H. Chang, S. Sull, L. Sang. "Efficient Video Indexing Scheme for Content-Based Retrieval." IEEE Trans. on Circuits and Systems for Video Technology, 1999, 9(8), pp. 1269-1279.

[25] T. Liu., H. Zhang, F. Qi. “A novel video key-frame-extraction algorithm based on perceived motion energy model.” IEEE Trans. Circuits and Systems for Video Technology, 2003, 13(10), pp. 1006-1013.

[26] S. Algur and R. Vivek,” Video Key Frame Extraction using Entropy value as Global and Local Feature.” arXiv:1605.08857 [cs.CV], 2016.