Received: April 23, 2019

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Two Level Key Frame Extraction for Action Recognition Using Content Based Adaptive Threshold

Aditi Jahagirdar[1]* **Manoj Nagmode[2]**

[1] *MIT College of Engineering, Pune India*
[2] *Government College of Engineering and Research, Awasari, India*
* Corresponding author's Email: Aditi.jah@gmail.com

Abstract: Extracting keyframes for action recognition purposes is a challenging task, as compression is to be achieved without losing the impression of the action. Insufficient or wrong keyframes can lead to confusion in the type of action. Proposed two-level key frame extraction algorithm uses an adaptive threshold technique to identify most dissimilar frames as keyframes. At the first level, global features based on intensity histogram and at second level local features computed from wavelet decomposition are used as similarity measures. A new performance parameter, Compression Ratio-Normalized Fidelity (CRNF) is introduced for evaluating the keyframe extraction algorithm. Average CRNF of 0.81 is achieved by the proposed algorithm, as compared 0.37 achieved by existing histogram-based method for action recognition dataset. CRNF value of 0.95 is achieved for Open Video project dataset, which is more than existing methods. Qualitative results further prove the effectiveness of the proposed algorithm.

Keywords: Keyframes, Salient frames, Histogram difference, Wavelet decomposition, Correlation, Mutual information.

1. Introduction

With recent developments in multimedia technology, video data processing has become an integral part of data analytics. Automatic video analysis is the current hotbed of research. Finding any relevant information from the videos is a time consuming and tedious job. To tackle the problem of large memory space required for storing the videos, video summarization or keyframe extraction techniques are used [1]. Video summarization is a technique which represents the video with a fewer number of salient frames. Frames carrying maximum information are identified and stored as keyframes. Keyframe extraction finds application in video indexing, video transmission, video annotation, video summarization and video retrieval [2].

Keyframe extraction can reduce the time complexity of automatic human action recognition significantly. In human action recognition videos, the subject present in the video performs an action for a small amount of time, as compared to the complete duration of the video. Keyframe extraction for action recognition is a challenging task, as an inadequate number of keyframes or incorrect keyframes can lead to the wrong classification of the action. For example, a few frames of action 'Punch' can be similar to frames of action 'Handshake' leading the classifier to classify the action of Punching as Handshaking. In case of an automatic surveillance system, this can cause a significant problem. Hence, a technique is needed which can represent action recognition sequence with less number of frames, without losing the essence of the action. In most of the existing work, the emphasis is given to saving the memory space by representing the video in less number of frames, and less attention is given to the ability to reconstruct the video from extracted frames. Most of these existing methods do not perform satisfactorily in this area. Our proposed method overcomes this problem, by capturing necessary and sufficient keyframes required for reconstruction of the video.

In this work, a two-level key frame extraction algorithm based on the adaptive threshold is proposed.

At both levels, the most dissimilar frames are identified as keyframes. The proposed method preserves the chronological order of frames, and makes it possible to reconstruct the video from keyframes. The main contributions of this work are given-

1. Frame difference method is implemented before computing the histogram difference, which reduces the computation cost.

2. The method uses global features at first level and local features at second level. This makes it possible to extract frames having the maximum information, as keyframes.

3. A new performance parameter called CRNF- computed by taking the product of compression ratio and normalized fidelity- is introduced to measure the performance of the system.

4. High performance in terms of CRNF is achieved as compared to existing video summarization methods.

The remaining paper is organized into seven sections. Section 2 gives related work, Section 3 discusses the performance parameters and datasets, and Section 4 explains the existing histogram-based method. The proposed two-level key frame extraction algorithm is discussed in section 5. Section 6 discusses results and paper is concluded in section 7.

2. Related work

Keyframe extraction methods are broadly divided into shot-based methods, feature-based methods, clustering-based methods and motion analysis-based methods [3]. Shot-based detection methods are further classified as pixel-based and histogram-based methods.

In many of the earlier techniques, frames are represented by some local feature and then a clustering algorithm is applied to combine similar frames. One or more frames from each cluster are then identified as the keyframes. Scale Invariant Feature Transform (SIFT) is the most widely used feature for this task because of its property to detect key points accurately.

In previous work [4], Discontinuity values between two consecutive frames are used as a measure to decide keyframes. To find the discontinuities, the ratio of matched SIFT features to the total number of features is used. All the frames in a video are used for computation. As SIFT is based on the histogram of gradients of all pixels in a frame, it is time-consuming. A detailed study of the computational complexity of the SIFT method is given in [5].

In [6], to reduce the computations, SIFT is extracted from selected candidate keyframes. SIFT features are computed for all the candidate frames to find points of interest (POI). If a new candidate frame has 60% or more change in the number of POIs as compared to previously present frame, then the new frame is considered as a keyframe. The method gives a good performance, based on the F1 score measure for tested datasets. The drawback of this method is that for short duration videos, candidate key frames are very less giving a low number of keyframes which results in loss of information.

In [7], two sets of keyframes are extracted. For obtaining the first set of keyframes, frame clustering is done based on low-level semantic features. For obtaining the second set of keyframes, flip invariant SIFT features are combined with six texture features. Keyframes from both the sets are compared using color histogram matching technique, to remove similar frames. The method gives good compression ratio for large scale videos. The disadvantage of this method is that its result depends on empirically found threshold, whose value needs to be changed from video to video for getting an optimal result.

Two-stage clustering based method is proposed for video summarization in [8]. Frames are divided into two sets, the prime number and nonprime number, using Eratosthenes Sieve approach. K-means clustering is implemented to identify keyframes. The number of clusters is optimized using the Davies-Bouldin Index algorithm. The approach shows very good results as compared to previous methods on the basis of the F1 score. The drawback of the method is its high time complexity as the number of steps is higher.

K-means clustering approach is used for identifying key frames for video summarization (VSUMM) in [9]. Clustering is done based on Colour features extracted from the frames. For comparison purpose, the ground truth of video summaries is generated, by taking inputs from five users. The VSUMM method works well on longer duration videos but for videos with smaller duration, important information from the video may get lost as frames are samples at a fixed sampling rate.

Another approach for key frame extraction is a threshold-based technique. Frame averaging and histogram averaging are the two most common methods in this category.

In [10], an adaptive threshold based method is proposed. Histogram difference between two consecutive frames is used as a similarity measure. Moderate value of compression ratio and fidelity is achieved for KTH dataset. The disadvantage of this

method is that it can miss a few keyframes, as histograms of two different frames can be similar.

In [11], two-stage key frame extraction method is proposed. In the first stage, the color histogram difference is used as a similarity measure to extract the candidate keyframes. At the second stage covariance between the frames is used as a similarity measure to identify the most dissimilar frames as keyframes. The method is easy to implement and gives good results on a few videos. The main disadvantage of the method is the fixed value of thresholds at both stages.

A multidimensional curve splitting algorithm is proposed in [12], to identify the linearized curve of perceptually significant key points. Frames corresponding to these perceptually significant points are extracted as keyframes. The disadvantage of this method is that it is unable to identify all the important frames from a video.

In [13], each frame is represented by a color histogram feature. Feature dimensionality is reduced by applying principal component analysis. Delaunay Triangulation algorithm is applied to extracted features to identify the keyframes. DT algorithm extracts less number of keyframes at the cost of low accuracy.

Still and Moving storyboard (STIMO) approach using clustering is proposed in [14]. The pairwise distance of consecutive frames is computed to decide a final number of clusters. In the final stage, similar frames are removed to obtain keyframes. Even if STIMO approach gives good results as compared to k means clustering algorithm in terms of time, it is not able to identify key frames related to all the events occurring in the video.

Even if considerable work is done in the field of key frame extraction, its main focus has been getting high compression. Very less work is done in the area of video summarization for the purpose of human action recognition.

In this work, as the focus is on human action recognition, chronological order of keyframes is very important. Some frames at the beginning and at the end of action might be similar and still be required to be selected as key frames as their significance is different. Both these frames might be important for reconstructing the action sequence from keyframes.

3. Performance parameters and dataset

This section gives brief information about performance parameters used for evaluating the proposed algorithm. Since standard performance parameters or ground truth are not available for key frame extraction methods on action recognition videos, it is a challenging job to compare the results with previously done research work. This section also gives a brief description of the dataset used for evaluating the algorithm.

3.1 Performance parameters

Since the official definition of 'keyframe' is not available in literature, standard performance parameters are not defined for keyframe extraction algorithms. The requirement of properties that keyframes should possess, change as per the application. For video compression, the compression ratio is the most important property while for content-based video retrieval, the information content of keyframes is more important. In this work, the requirement is that reconstruction of the video should be possible with a minimum number of keyframes. To satisfy this requirement, compression ratio and Fidelity are computed as evaluation metric as in [15]. Compression ratio gives compactness while fidelity gives exactness of keyframes. The compression ratio is given as in Eq. (1)

$$Cratio = 1 - (N_{kf}/N_{vf}) \quad (1)$$

where N_{kf} is the number of keyframes and N_{vf} is the total number of frames in a video. High compression ratio implies better compactness of video representation.

Fidelity metric is computed using semi-Hausdorff distance. Euclidian Distance is calculated between each keyframe and each frame of the test video. The minimum distance is then selected as the distance between that keyframe and original video. The maximum distance thus obtained is considered as distance between a set of keyframes and a set of original frames and is represented as $Dist(V_{seq}, Key_f)$. The maximum value of distance obtained while computing distance between individual keyframe and original video frames is considered as maximum dissimilarity measure and is represented as Max_{Dist}. Fidelity is then given as in Eq. (2)

$$Fi(V_{seq}, Key_f) = Max_{Dist} - Dist(V_{seq}, Key_f) \quad (2)$$

High fidelity value signifies good representation of the original video in less number of frames. Since high compression ratio, as well as high fidelity, is desired, a new performance parameter called CRNF is introduced, and is given in Eq. (3), where CR represents compression ratio and NF represents Normalized fidelity.

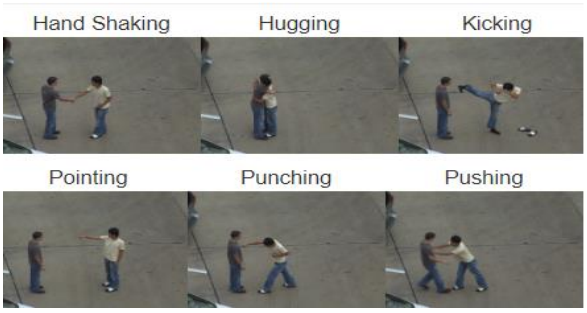

Figure. 1 Sample frames from UT interaction data set

$$CRNF = (CR)\ X\ (NF) \quad (3)$$

3.2 Dataset

As the focus of this work is on human action recognition, the proposed algorithm is evaluated on UT interaction dataset. This data set is widely used for Human interaction detection. UT interaction 1 and 2 datasets have two people interacting with each other, performing actions like Hugging, Handshaking, Pointing a finger, Punching, Pushing and Kicking.

Fig. 1 shows sample frames from UT interaction dataset. In UT interaction 1 dataset, the background is uncluttered and only two people are present in every frame. In the UT interaction 2 data set, the same 6 actions are performed by actors but more than two actors are present in a frame. In this work, videos from UT interaction 1 and 2 datasets are combined action class wise for evaluating the performance.

For comparing the results with existing video summarization methods, Open Video Project (OV) dataset [10] is used for testing. Videos in this dataset are in MPEG-1 format recorded with 30 fps. The videos are of different types like a documentary, educational, ephemeral, historical and lecture. Duration of the videos varies from 1 to 4 minutes. As ground truth, each video has 5 video summaries created by 5 different users manually. Fig. 2 shows sample frames from 'New Horizon Seg 2' video of OV project dataset. All user summaries are available at *http://www.npdi.dcc.ufmg.br/VSUMM*.

Figure. 2 Sample frames from Open Video Project dataset

4. Existing threshold based method using histogram [10]

Existing threshold based key frame extraction method is explained in this section. The difference in intensity histograms of two consecutive frames is used as a similarity measure to identify key frames from a video. Following steps depict the existing algorithm:

1. Read the test video and convert the frames to grey scale.
2. Read $Frame_i$ and $Frame_{i+1}$
3. Find intensity histogram His_i and His_{i+1} for frames read in step 2.
4. Compute the absolute difference between His_i and His_{i+1}
5. Calculate the sum of differences obtained over all the bins of histogram and store as Histogram coefficients.
6. Find the mean $\bar{x}$ and standard deviation σ of histograms coefficients obtained for all the frames.
7. Compute threshold value from mean and standard deviation values obtained in step 6 using Eq. (4)

$$Th = k1 * \bar{x} + k2 * \sigma \quad (4)$$

Where *k1* and *k2* are constants which are found empirically.

8. Compare the Histogram coefficient values of all the frames with the threshold value to identify most dissimilar frames as keyframes.

As the threshold-based technique is most simple but still effective, taking the inspiration from it, in this work, experimentation is done using various similarity measures to identify keyframes. Explanation of all the similarity measures implemented is given in the following section as Single level keyframe extraction techniques. Comparison of techniques implemented is done based on CRNF and execution time required.

4.1 Single level keyframe extraction techniques

As the first stage of work, five techniques using different similarity measures are implemented to find keyframes. The techniques used are given as:

1. Histogram of the difference of three consecutive frames (HDF)
2. Mutual information between two consecutive frames (DMI)

3. Correlation value between two consecutive frames (CD)
4. Correlation between differences of three consecutive frames (CFD)
5. Total Difference between detail coefficients of two consecutive frames (DWC)

In HDF, Histograms of frame differences are computed for three consecutive frames. Histogram difference is then found for these frame differences. Addition of these differences then represents the dissimilarity quotient of the three frames.

In DMI, Mutual Information (MI) is used as a dissimilarity measure. MI is calculated for consecutive images using Eq. (5).

$$MI(F_i, F_{i+1}) = E_n(F_i) + E_n(F_{i+1}) - E_n(F_i, F_{i+1}) \quad (5)$$

Where $E_n(F_i)$ and $E_n(F_{i+1})$ represent entropies of two consecutive frames and $E_n(F_i, F_{i+1})$ represents joint entropy between them. Joint entropy is computed using the joint histogram method. For similar frames, the histogram is focused while for dissimilar frames, the histogram is dispersed. Joint entropy can be defined as a measure of dispersion in the histogram. Higher the joint entropy less distinct are the frames. Joint entropy can be used for selecting keyframes but the advantage of using MI over joint entropy as a measure is that MI takes into account entropies of individual frames. Frames having low MI are selected as keyframes.

In CD, the correlation between two consecutive frames is used to find dissimilar frames. In CFD, variation in technique three is implemented. In this method, the correlation between frame differences is computed and used as a dissimilarity measure. Less the correlation value, more the dissimilarity between frames.

In DWC, wavelet decomposition is used for finding average and detail coefficients of two consecutive frames. Addition of difference between vertical, horizontal and diagonal detail coefficients of two frames is then computed and used as a dissimilarity measure. More the value of coefficients, more dissimilar are the frames.

Techniques 1 to 5 are implemented and evaluated on UT interaction dataset. CRNF and execution time are used as performance measures for comparing the five techniques.

4.2 Results: Single level key frame extraction

Results obtained by implementing single level key frame extraction are discussed in this section. An efficient key frame extraction algorithm should have high CRNF and low execution time. To identify the technique that can satisfy this criterion, average CRNF and average execution time over all the action classes are computed for each technique discussed in the previous section. Fig. 3 shows a comparison of average CRNF and normalized execution time.

Even if the highest CRNF is achieved by DMI technique, its execution time is more than double the execution time of other methods. Since less execution time is desirable, the DMI technique is not used in the two-level key frame extraction algorithm. DWC, CDF, CD and HDF techniques give moderate CRNF in low execution time.

5. Two level keyframe extraction algorithm

In single-level key frame extraction algorithms, CRNF in the range of 0.6 to 0.7 is achieved. To improve the performance of keyframe extraction method, the two-level technique is proposed. In the first level, salient frames are extracted using global information of the frames by means of intensity histogram. Salient frames encompass all the frames from the start of the action to the end of the action. First level extraction removes all the redundant frames at the start and at the end of the action and identifies salient frames. In the second level, final keyframes are extracted from the salient frames using local characteristics of frames by means of the wavelet decomposition. Chronological order of frames is maintained at both levels.

By considering the results obtained in single level key frame extraction, Histogram of Difference Frames (HDF) is selected as first level algorithm, while Wavelet coefficient based algorithm (DWC) is used as the secondary level algorithm. Fig. 4 shows a basic flow diagram of the proposed two-level key frame extraction algorithm.

The following sub-sections explain the proposed method in detail.

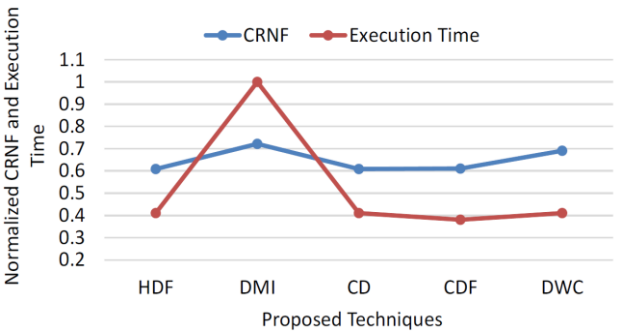

Figure. 3 Comparison of Average CRNF and execution time

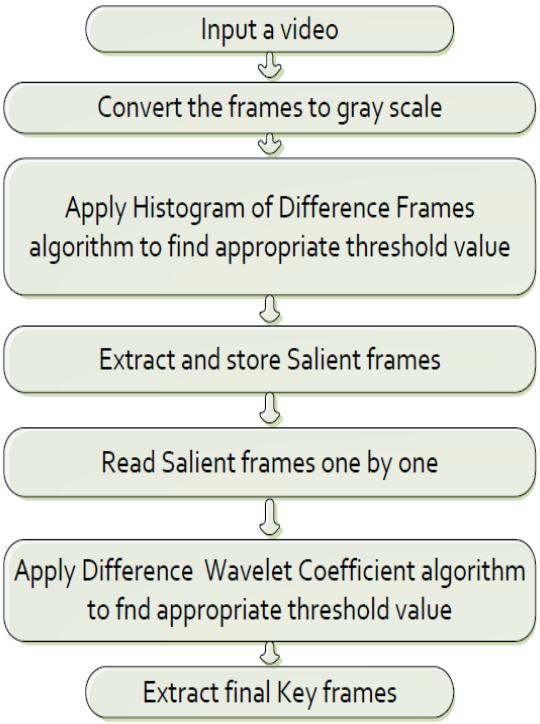

Figure. 4 Flow diagram of proposed Two level Key frame extraction technique

5.1 First level: Salient frame extraction

Salient frame extraction at primary level is done using histogram based, HDF technique. The ability of HDF technique to identify shot boundaries makes it the right candidate for the first level of algorithm. Graphs of distribution of histogram coefficients obtained with HDF technique plotted with respect to frames is shown in Fig. 5. The graph shows two distinct peaks-the first peak marks the start of action or event while the second peak marks the end of it. Algorithm 1 depicts steps of Histogram of frame difference technique.

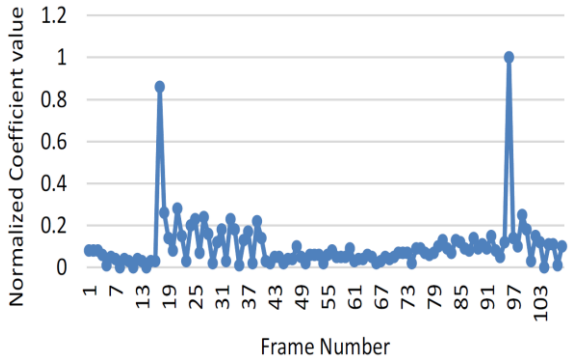

Figure. 5 Distribution of coefficients for HDF technique

Algorithm 1:HDF

$Input = test\ video$
$Output = \{Salient\ frames\}$
$Begin$
$Read\_Video$
$T = Total\ Number\ of\ Frames$
$For\ i = 1\ To\ T-1$
$Diff_i = Image_{diff}(Frame_i, Frame_{i+1})$
$Histf_i = Find_{hist}(Diff_i)$
End
$For\ i = 1\ To\ T-2$
$HistDiff_i = Diff\_hist\ (Histf_i, Histf_{i+1})$
$HistCoeff = Sum(HistDiff_i)$
End
$Mean_{coeff} = Find_{mean}(HistCoeff)$
$STD_{Coeff} = Find_{STD}(HistCoeff)$
$Th_{value} = k1 * Mean_{coeff} + k2 * STD_{Coeff}$
$For\ j = 1\ To\ T-2$
$If\ \ HistCoeff(j) > Th_{value}$
$Salient_{frames}\{\} = Frame\ (j+1)$
End
End

1. Read a test video and convert the frames to grey scale
2. Find frame difference $Diff_i$ and $Diff_{i+1}$ by taking the absolute difference between $Frame_i, Frame_{i+1}$ and $Frame_{i+1}, Frame_{i+2}$
3. Compute intensity histograms $Histf_i$ and $Histf_{i+1}$ for difference frames $Diff_i$ and $Diff_{i+1}$ respectively.
4. Find histogram difference $HistDiff_i$ by taking the difference of $Histf_i$ and $Histf_{i+1}$
5. Add the differences over all the bins to obtain $HistCoeff$
6. Find the mean and standard deviation of $HistCoeff$ to compute threshold value, Th_{value}
7. Compare the values of $HistCoeff$ with Th_{value} to identify salient frames.

For frames having high similarity, the absolute difference between the frames is almost zero. Histogram of such a difference frame is focused at zero grey level. Such frames are generally present before and after the action sequence. For the frames in which motion is present, difference frames have pixels with varying values. Histogram of these difference frames is more dispersed. Histogram coefficients thus obtained are able to discriminate between frames and are able to identify most dissimilar frames as keyframes. While doing so, chronological order of frames is maintained which is important for recognition of action.

Once the salient frames are identified they are stored in a folder to be given as input to algorithm 2 for final key frame extraction.

5.2 Secondary level: Keyframe extraction

To extract the final key frames from the salient frames, wavelet decomposition technique i.e. DWC is used because of its high CRNF and less execution time. Algorithm 2 shows detail steps for extracting key frames at the second level.

Algorithm 2: DWC

$Input = \{Salient\ frames\}$
$Output = \{Key\ frames\}$
$Begin$
$T = Total\ Number\ of\ Salient\ Frames$
$For\ i = 1\ To\ T - 1$
$Coeff_i\{\mathrm{Hi, Vi, Di}\} = \mathrm{Wavelet_{Decomp}}(Frame_i)$
$C_{HOR} = \{\mathrm{H_{i+1}} - \mathrm{H_i}\}$
$C_{VER} = \{\mathrm{V_{i+1}} - \mathrm{V_i}\}$
$C_{DIG} = \{\mathrm{D_{i+1}} - \mathrm{D_i}\}$
$Coeff_{final} = \{C_{HOR} + C_{VER} + C_{DIG}\}$
End
$Mean_{coeff} = Find_{mean}(Coeff_{final})$
$Th_{value} = k * Mean_{coeff}$
$For\ j = 1\ To\ T - 1$
$If\ Coeff_{final}(j) > Th_{value}$
$Key_{frames}\{\} = Salient_{frame}\ (j + 1)$
End
End

1. Read salient frames and convert to grey scale
2. Find horizontal, vertical and diagonal detail coefficients $Coeff_i\{\mathrm{Hi, Vi, Di}\}$ for $Frame_i$ and $Frame_{i+1}$
3. Compute the difference C_{HOR}, C_{VER} and C_{DIG} by subtracting horizontal, vertical and diagonal coefficients of two consecutive frames.
4. Add the differences obtained in step 3 to obtain final coefficients $Coeff_{final}$
5. Find mean of final coefficients and compute threshold Th_{value}
6. Compare the values of $Coeff_{final}$ with Th_{value} to identify keyframes.

6. Results

Quantitative and qualitative performance analysis is carried out for the evaluation of the proposed algorithm. Quantitative analysis is done using performance measures explained in the previous section. For qualitative analysis, video is reconstructed from keyframes and experts are asked to identify the action.

6.1 Quantitative results

Compression ratio and fidelity are computed for all the videos in UT interaction dataset 1 and 2. Average of the parameters is then computed for each action class.

Fig. 6 shows compression ratio obtained at the first level and at the second level. It is seen that the compression ratio increases considerably for all the action classes. The maximum increase of 42.37 % is achieved for action class ‘Handshake' for which CR has increased from 0.59 to 0.84. The lowest increase of 24.32% is achieved for action class ‘Punch', for which CR is increased from 0.74 to 0.92. Average increase of 33.69% is achieved in CR overall action classes.

Fig. 7 shows a graph of the comparison of fidelity achieved at first and second level of extraction. For action class ‘Pointing a finger’, fidelity remains constant at both levels. For action classes ‘Punch’ and ‘Push’, fidelity increases by 0.04 and 0.02, respectively. For remaining action classes, fidelity value reduces by a negligible amount. Maximum reduction occurs for action class ‘Kick', for which fidelity is reduced from 1 to 0.9. An average reduction of 1.28% happens in fidelity at the second level of keyframe extraction.

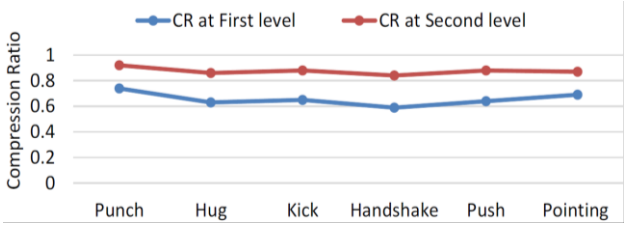

Figure. 6 Comparison of compression ratios at two levels

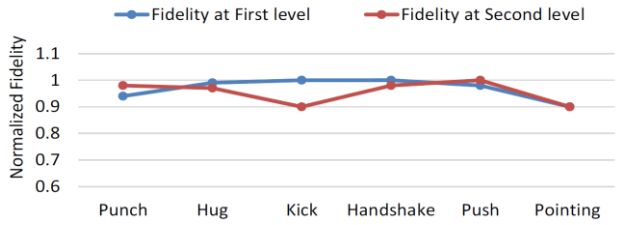

Figure. 7 Comparison of fidelity at two levels

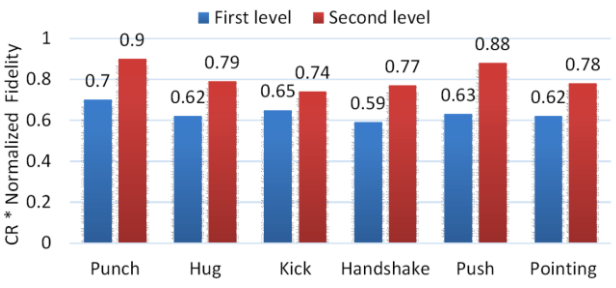

Figure. 8 Comparison of CRNF achieved at two levels

Fig. 8 shows a graph of CRNF acquired at the first and second level for all action classes.

It is seen that, for all the action classes CRNF increases when second level key frame extraction is applied. The maximum increase of 0.25 is achieved for action class ‘Push' while the minimum increase of 0.14 is achieved for action class ‘Kick'.

Since ground truth for key frame extraction on UT interaction data set is not available, it is difficult to compare the results obtained by the proposed technique with techniques presented in the literature. To solve this problem, the absolute difference of histogram of consecutive frames algorithm [10], explained in section 4 is implemented and evaluated on UT interaction dataset. Table 1 shows a comparison of the proposed method with the existing histogram-based method.

A significant increase is achieved in compression ratio for all the action classes. Fidelity achieved is comparable with that achieved with the existing method. For four action classes, fidelity is reduced. For one action class, it is the same and for one action class, it is increasing. Average CRNF of 0.37 is achieved over all classes by the existing method, while 0.81 is achieved by the proposed method.

For comparing the time complexity of the algorithm, existing SIFT-based and histogram-based methods were evaluated. Comparison of SIFT and SURF algorithms given in [16] shows that SIFT does not perform well on the basis of time complexity for key frame extraction. For further evaluation, when the SIFT algorithm was implemented and tested on action recognition video, it took an average time of 20 seconds to extract SIFT features from one frame, of size 312x428. Since the time required using the SIFT method is very high, it is not used for further comparison. Table 2 gives a comparison of the time required for extracting the keyframes for all the classes using the existing histogram-based method and the proposed method.

Table 1. Comparison of the proposed method with the existing method

	CRNF value	
Action Class	Existing Histogram based Method [10]	Proposed Method
Punch	0.34	0.9
Hug	0.37	0.79
Kick	0.38	0.74
Hand Shake	0.4	0.77
Push	0.34	0.88
Point Finger	0.43	0.78

Table 2. Comparison of time complexity

	Execution Time(sec)	
Action Class	Existing Histogram based Method [10]	Proposed Method
Punch	20.93	20.67
Hug	51.74	52.77
Kick	31.57	31.68
Hand Shake	58.30	58.02
Push	23.10	23.67
Point Finger	14.79	15.10

It is observed that the average time required to extract keyframes by both methods is almost the same. Depending on the number of frames present in each video, the time required changes. For the proposed technique, the execution time of the second level depends on the number of frames extracted in the first level of extraction.

6.2 Qualitative results

To evaluate the qualitative performance of the proposed algorithm, keyframes extracted at the second level were assessed manually by five experts. All the experts were able to identify the action class from the keyframes correctly. Fig 9 show keyframes extracted for sample video of ‘Kick' action class. Kick action class is selected here as it is has minimum CRNF as compared to other action classes. It can be seen that action ‘Kick' can be easily recognized from keyframes.

Figure. 9 Keyframes extracted at the second level from ‘Kick’ action class

To take the qualitative performance evaluation one step further, videos were reconstructed from the extracted key frames and evaluated from experts. All the experts were able to recognize the action from reconstructed videos satisfactorily.

6.2 Comparison of results on Open Video Project dataset

For reviewing the use of the proposed algorithm for video summarization, it was tested on an open video project (OV) dataset. The aim of video summarization is to represent a video having multiple events in a concise way. Results of five different algorithms along with the keyframes selected by five users are available at the site of VSUMM project and are used for comparison purposes. (*https://sites.google.com/site/vsummsite/download*)

For comparison, CRNF is used as a performance metric. Fidelity is calculated by comparing keyframes extracted by each method with keyframes selected by users. The compression ratio is calculated by considering the number of keyframes extracted and the number of frames in the original video. Fig. 10 shows keyframes extracted by the proposed method for video 'New Horizons seg 2' from OV project dataset.

The proposed method is able to extract at least one frame related to each event in the video. For New Horizon Seg 2 video, at first level 148 frames are extracted from 1797 original frames. At second level 16 frames are extracted as final keyframes. High fidelity and compression ratio are achieved for all the tested videos. Table 3 shows the quantitative results obtained for OV project dataset. It is seen that the proposed method outperforms existing video summarization methods based on CRNF measure.

Figure. 10 Keyframes extracted from sample video 'New Horizon Seg 2' from OV Project dataset

Table 3. CRNF values obtained for OV project dataset

Method Used	CRNF
VSSUM1 [9]	0.896958
VSUMM2 [9]	0.897122
DT [13]	0.32435
VISTO [14]	0.50703
OV [12]	0.761552
Proposed method	0.951096

7. Conclusion

In this work, two-level key frame extraction is proposed. The approach is based on the use of adaptive threshold for identifying most dissimilar frames. The algorithm is evaluated on UT interaction dataset and Open video project dataset. A new performance parameter, CRNF is introduced in this work which takes into account compression ratio and fidelity parameter.

HDF technique uses the difference in intensity histogram as a similarity measure. Since it is a global feature, it is able to distinctly identify the start of the action and the end of the action. At the second level of extraction, DWC technique is used, which uses the wavelet-based feature as a similarity measure. Since this is a local feature, it is able to identify the similarity between the frames efficiently. Most dissimilar frames are retained as keyframes. This combination of global and local features used in the proposed method makes it possible to outperform existing methods where only one type of feature is used. The time complexity of the proposed algorithm is same as of the existing histogram-based method.

The quantitative results obtained for action recognition videos show that the proposed two-level method for keyframe extraction outdoes the existing histogram-based method. Average CRNF of 0.81 is achieved by the proposed algorithm as compared to 0.37 achieved by existing histogram-based method. Significant increase in CR values, and inconsequential changes in fidelity values prove that second level keyframe extraction is able to remove the redundant frames from a set of salient frames. The method is able to preserve frames with most of the information.

The results obtained on OV project dataset prove the ability of the proposed algorithm for video summarization task. Average CRNF value of 0.95 is achieved on selected videos of Open video project Even if the number of keyframes extracted by the proposed method is more than that by existing VSUMM method, the proposed method outperforms existing methods because of high fidelity value. High fidelity value indicates the closeness of a set of keyframes to a set of frames selected by users. This

proves that the proposed method is closer to human perception of keyframes.

Qualitative results obtained from experts prove the performance of the proposed algorithm further. Successful reconstruction of videos from extracted keyframes proves that video summaries created using the proposed algorithm will be useful in applications where further tasks like action recognition or content-based video retrieval are to be performed. It is seen that actions can be recognized without any ambiguity from reconstructed videos. In the future, work will be done on classifying human actions using keyframes.

References

[1] J. Rodriguez, P. Yao, and W. Wan, "Selection of Key Frames Through the Analysis and Calculation of the Absolute Difference of Histograms", In: *Proc. of IEEE International Conf. on Audio, Language and Image Processing*, pp. 423-429, 2018.

[2] A. Ali, M. Al-Mufraji, and T. Saeed, "Improved Key Frame Extraction Using Discrete Wavelet Transform with Modified Threshold Factor", *Telecommunication Computing Electronics and Control*, Vol.16, No.2, pp. 567-572, 2018.

[3] K. Sahu and S. Verma, "Key Frame Extraction From Video Sequence: A Survey", *International Research Journal of Engineering and Technology*, Vol. 4 No. 5, pp. 1346-1350, 2017.

[4] G. Liu, X. Wen, W. Zheng, and P. He, "Shot Boundary Detection and Keyframe Extraction based on Scale Invariant Feature Transform", In: *Proc of, International Conference on Computer and Information Science*, pp. 1126-1130, 2009.

[5] B. De-Souza, T. Tessarolli, and R. Goularte, "KS-SIFT: a keyframe extraction method based on local features", In: *Proc of IEEE International Symposium on Multimedia*, pp. 13-17, 2014.

[6] R. Ranjan and A. Agrawal, "Video Summary Based on F-Sift, Tamura Textural and Middle level Semantic Feature", In: *Proc of Twelfth International Multi-Conference on Information* pp. 870-876, 2016.

[7] P. Drews, R. de Bem, and A. de Melo, "Analyzing and exploring feature detectors in images", In: *Proc of the 9th IEEE International Conference on Industrial Informatics*, pp. 305-310, 2011.

[8] K. Kumar, D. Shrimankar, and N. Singh, "Eratosthenes sieve based key-frame extraction technique for event summarization in videos", *Multimedia Tools and Applications*, Vol. 77, No. 6, pp. 7383-7404.

[9] S. de Avila, A. Lopes, A. da Luz, and A. de Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method", *Pattern Recognit. Letters*, Vol. 32, No. 1, pp. 56–68, 2011.

[10] C. Sheena and N. Narayanan, "Key-frame extraction by analysis of histograms of video frames using statistical methods", *Procedia Computer Science*, Vol. 70, pp. 36-40, 2015.

[11] Y. Shi, H. Yang, M. Gong, X. Liu, and Y. Xia, "A fast and robust key frame extraction method for video copyright protection", *Journal of Electrical and Computer Engineering*, Vol. 2017, Article ID 12317942017.

[12] D. DeMenthon and V. Kobla, "Video summarization by curve simplification", In: *Proc. of ACM International. Conf. on Multimedia*, pp. 211– 218, 1998.

[13] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using Delaunay clustering", *International Journal on Digital Libraries*. Vol. 6, No. 2, pp. 219–232.

[14] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini, "STIMO: STIll and Moving video storyboard for the web scenario", *Multimedia Tools Applications*, Vol. 46, No, pp. 47–69, 2010.

[15] D. Papadopoulos, V. Kalogeiton, S. Chatzichristofis, and N. Papamarkos, "Automatic summarization and annotation of videos with lack of metadata information", *Expert Systems with Applications*, Vol. 40, No. 14, pp. 5765–5778, 2013.

[16] S. Athani and C. Tejeshwar, "Performance analysis of Key Frame Extraction using SIFT and SURF algorithms", *International Journal of Computer Science and Information technologies*, Vol 7, No. 4, pp.2136-2139, 2016.