Heliyon 5 (2019) e02699

Contents lists available at ScienceDirect

Heliyon

www.elsevier.com/locate/heliyon

Research article

Domain independent redundancy elimination based on flow vectors for static video summarization

Jesna Mohan [a,b,*], Madhu S. Nair [c]

[a] Department of Computer Science, University of Kerala, Kariavattom, Thiruvananthapuram-695581, Kerala, India
[b] Department of Computer Science, Mar Baselios College of Engineering and Technology, Thiruvananthapuram-695015, Kerala, India
[c] Artificial Intelligence & Computer Vision Lab, Department of Computer Science, Cochin University of Science and Technology, Kochi-682022, Kerala, India

ARTICLE INFO

Keywords:
Computer science
Video summarization
Keyframes
SIFT flow
Uniform sampling
Dual Layer Loopy Belief Propagation Network (DLBPN)

ABSTRACT

Video summarization aims to find a compact representation of input videos. The method finds out interesting parts of the video by discarding the remaining parts of the video. The abstracts thus generated enhances browsing and retrieval of video data. The quality of summaries generated by video summarization algorithms can be improved if the redundant frames in the input video are taken care of before summarization. This paper presents a novel domain-independent method for redundancy elimination from input videos before summarization maintaining keyframes in the original video. The frames of input video are first presampled by selecting two frames in one second. The flow vectors between consecutive frames are computed using SIFT Flow algorithm. The magnitude of flow vectors at each pixel position of the frame are summed up to find the displacement magnitude between the consecutive frames. The redundant frames are filtered out based on local averaging of the displacement values. The evaluation of the method is done using two standard datasets namely VSUMM and OVP. The results demonstrate that an average reduction rate of 97.64% is achieved consistently on videos of all categories. The method also gives superior results compared to other state-of-the-art redundancy elimination methods for video summarization

1. Introduction

With the recent advancements in the digital world, there has been an exponential increase in the volume of the video data. This data explosion has put our existing network infrastructure at risk. As a result, any interested user has to browse through substantial video repositories to find the relevant video data. However, these users cannot decide without viewing the entire video content which is time-consuming. Time is of the essence. A summarized video can aid any interested user to make quick decisions. Moreover, a summarized video data can be stored and retrieved efficiently [1].

Video summarization extracts an abstract representation of the original video by selecting the keyframes of the videos and discarding the redundant parts. The keyframes are those frames which contain prime parts of the video. When a set of keyframes are viewed together, it can convey the essential message of the video. The video abstracts can be stored in less space and users can perceive the contents in less time compared to the original video. The abstracts must also be consistent with the human vision so that humans can understand the information conveyed by a video which spans hours long in few minutes. The video abstracts can be still abstracts (static storyboard representation) [2] or moving abstracts (video skims) [3], [4]. The still abstracts convey the essential message of the video as a sequence of frames. The temporal component in the input video is lost in still abstracts. Whereas, video skims represent summarized output as a short video which retains the temporal component. The still abstracts are simple to put into practice than the moving abstracts.

There exists a number of works for automated video summarization. Avila et al. in [5] emphasize on processing videos based on color histogram. Ejaz et al. [6] used aggregation of global features to detect keyframes. But, the global features fail to capture local characteristics of frames which is relevant for detecting content change between frames. Guan et al. in [7] used local SIFT features for detecting the key-frames. Later, Hannane et al. in [8] combined local features with optical flow for generating a good quality summary. Summarization was also done at content level [9] by selecting frames using dynamic programming approach. Zhu et al. [10] emphasized object level processing of frames to extract key parts of the input video.

* Corresponding author at: Department of Computer Science, University of Kerala, Kariavattom, Thiruvananthapuram-695581, Kerala, India.
E-mail address: jesnamohan@gmail.com (J. Mohan).

https://doi.org/10.1016/j.heliyon.2019.e02699
Received 9 July 2019; Received in revised form 22 September 2019; Accepted 16 October 2019

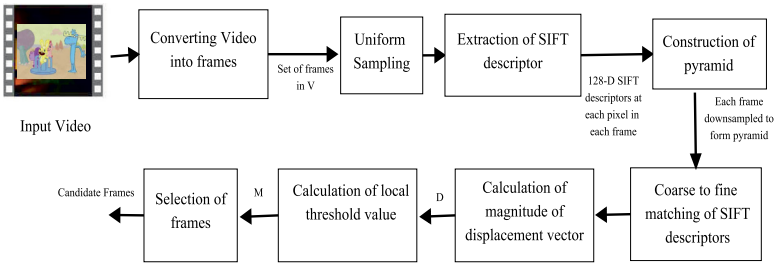

Fig. 1. Overview of proposed approach.

Recently, with the development of high performance computers with GPUs, deep learning based methods have been developed for video summarization. Mahasseni et al. in [11] proposed the long short-term memory network (LSTM) to select the key-frames. The network is trained so that the reconstruction error is minimum. Fei et al. in [12] proposed video summarization framework based on entropy and memorability score. The memorability score is computed using Hybrid-AlexNet. The quality of summaries generated by these methods reveal that deep features can represent the frames more efficiently than hand-crafted features. The sparse dictionary based method for summarization is explored in [13].

However, most of the existing methods consider the whole video for processing. The computational complexity of these methods is very high since a video has many redundant frames. The duplication of similar visual content in the input is unfavorable for generating a brief but comprehensive representation of the original video. The efficiency of video summarization framework can be improved if ambiguous frames are detected and discarded initially so that set of frames with less redundancy can be used for further processing.

Various techniques have been explored in literature to eliminate redundant frames before summarization. Avila et al. in [14] discarded redundant frames using a uniform sampling method by randomly selecting one frame in a second from the input video. Kuanar et al. in [15] identified that the frames corresponding to the significant valley of mutual information curve carry the essential content of the video. These frames must also be added to the presampled set of frames to prevent information loss. Song et al. in [16] selected those frames with minimum Euclidean distance to the average histogram of a shot as candidate frames for summarization. But these methods fail to achieve consistent results on videos of all categories. Thus, a technique that could remove the redundancy from input videos retaining every distinct frame and which generate consistent results for all categories of video is in demand and attracted researchers in this area.

Recent works on video analysis [17], [18] has proved the significance of motion vectors and temporal analysis in capturing the content change in video data. Moreover, the work in [19] suggests motion vectors are more discriminative when it is calculated after reducing the number of redundant frames from the input videos. Thus, an attempt to reduce redundancy based on motion vectors after an initial sampling would be helpful to reduce the computational burden of subsequent summarization step. So, to reduce running time and redundancy, we propose here a domain-independent redundancy elimination method for video summarization based on flow vectors after performing uniform sampling on the set of input frames. The method makes use of SIFT Flow algorithm to find the magnitude of displacement between consecutive frames after uniform sampling. Then, the redundant frames are eliminated using threshold value which is determined based on local averaging of the displacement magnitude. The method is tested on VSUMM and OVP dataset and it achieves high reduction rate of 97.64% and less error compared to other state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 describes the proposed methodology for eliminating redundancy from input video before summarization. Experimental results and discussions are illustrated in Section 3. We conclude the paper in Section 4.

2. Methodology

The proposed system is a redundancy elimination technique using uniform presampling followed by filtering out of ambiguous frames based on flow vectors. The method makes use of SIFT Flow algorithm for the calculation of flow vectors between consecutive frames. The magnitude of flow vectors is then used to find out the abrupt transition in motion and to track the frames where a scene change occurs. The magnitude of displacement is thresholded locally to eliminate redundant frames of input video preserving keyframes. The method works adaptively on videos of all categories. The resultant set of frames can be used for summarization. The algorithm works on a frame by frame basis. Fig. 1 gives an overview of the proposed approach. The detailed steps of the method is as follows.

2.1. Uniform sampling

Suppose, F be the set of frames and fps be the frame rate associated with input video V. Uniform sampling is done by selecting frames in F based on the assumption that most of the frames that constitute video in one second is similar. Most of the existing methods select a single frame in one second to reduce redundancy. But randomly selecting any one frame may lead to keyframe miss especially for low frame rate videos like cartoon videos. Also, the high sampling rate will gain high reduction rate and subsequently reduce the time taken for summarization. But it may lead to loss of relevant information from the input video. A low sampling rate will preserve the information of the original video but will increase the complexity of summarization step. The sampling rate should be chosen carefully so that all unique frames in the video should be present in the output of the sampling step for further processing.

The sampling rate used in the proposed approach is two frames in one second. First, F is divided into N segments $S_1, S_2, S_3, ...S_N$ based on fps. Suppose fps be 30. Frames from F_1 to F_{30} form first segment, F_{31} to F_{60} form second segment and so on. Then, the first frame and middle frame of each segment are selected to form F_s. F_s include F_1, F_{15} of S_1, F_{31}, F_{46} of S_2 and so on.

2.2. Extraction of feature descriptors

Let $F_1{}^*$, $F_2{}^*$, $F_3{}^*$, $.....F_{n_s}{}^*$ be the elements of F_s. The algorithm converts each frame of F_s into SIFT image. A SIFT image is formed by replacing each pixel with 128 dimensional SIFT descriptor [20]. Each input image in F_s is first blurred and downsampled at different scales by convolving with a gaussian filter given by (1).

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{1}$$

where the amount of blur is determined by σ and (x, y) are coordinates of the pixel under consideration. The blurred image $L(x, y, \sigma)$ obtained by convolving the input image $F(x, y)$ with $G(x, y, \sigma)$ is computed as in (2).

$$L(x,y,\sigma) = F(x,y) \star G(x,y,\sigma) \tag{2}$$

The magnitude $m(x,y)$ and orientation $\theta(x,y)$ are computed for each pixel at a particular scale σ in Gaussian-blurred image based on intensity differences computed using (3) and (4).

$$m(x,y) = ((L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2)^{\frac{1}{2}} \quad (3)$$

$$\theta(x,y) = \text{atan2}(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y)) \quad (4)$$

A 16×16 neighborhood is specified around each pixel and orientation histogram of 8 bins is computed for each of 16 (4×4) sub-block of the neighborhood. The histogram of 16 blocks which consists of 8 bins gives a 128-D SIFT feature descriptor. Thus if size of ${F_1}^*$ is $M \times N$, then it is converted into SIFT image of size $M \times N \times 128$.

2.3. Construction of pyramid

A pyramid is constructed by using smoothened and downsampled images from each frame of F_s after convolution with a Gaussian filter. The number of levels of pyramid is specified by the user. A four-level pyramid corresponding to ${F_1}^*$ with size $M \times N$ has ${F_1}^*$ at level 0. Let it be d_0. Then, image d_1 of size $M/_2 \times N/_2$ constructed from d_0 form level 1 of the pyramid. Similarly, d_2 of size $M/_4 \times N/_4$ constructed from d_1 form level 2 and finally d_3 of size $M/_8 \times N/_8$ constructed from d_2 form level 3.

2.4. Calculation of displacement vector

The flow vector for all pixels of ${F_1}^*$ is computed from ${F_2}^*$ by matching SIFT descriptors extracted at each pixel of ${F_1}^*$ and ${F_2}^*$. The matching is done from coarse level to fine level. Consider pixel x_1 in d_3 of pyramid constructed from ${F_1}^*$. Define a window of size $M/_8 \times N/_8$ around the pixel at the same location in d_3 of pyramid constructed from ${F_2}^*$. The flow vector for each pixel is initialised to 0. Then, the optimal values for vectors are calculated using the energy function in (5). The energy function is modified version of energy function used in the calculation of optical flow [21], [22] and is the weighted sum of data term, small displacement term and the smoothness term. Suppose, (u,v) denotes flow vectors at each pixel position of ${F_1}^*$ and ${F_2}^*$. 'u' represents the horizontal component and 'v' represents the vertical component. Let, p in ${F_1}^*$ and q in ${F_2}^*$ are the pixels under consideration. The variables 't' and 'd' represent the threshold term for data term and smoothness term respectively.

$$E(w) = \sum_p min(||s_1(p) - s_2(p + w(p)||_1, t) + \sum_p \eta(|u(p)| + |v(p)|) + \sum_{(p,q)\epsilon\varepsilon} min(\alpha|u(p) - u(q)|, d) + min(\alpha|v(p) - v(q)|, d) \quad (5)$$

The first term in (5) is the data term which allows SIFT descriptors to be matched along flow vectors. The second term is the displacement term. The third term is the smoothness term which constrains irregularities at object boundaries. The truncated L1 norm in data and smoothness terms threshold the differences and give a constant cost to the terms above the threshold value. The optimum flow vector is calculated by minimizing energy using dual layer Loopy Belief Propagation Network. The horizontal and vertical components of flow vectors are processed separately by two layers of the network. The values of flow vectors thus obtained are passed to level 2 which is used to initialize the flow vectors of each pixel of d_2 at level 2. Then, d_2 is warped based on these values of flow vectors to get $\hat{d_2}$. Matching is done between d_2 at level 2 of first pyramid and $\hat{d_2}$. The flow vectors are then optimized as in level 3. The flow vector from level 2 is used to find flow vectors in level 1 which in turn is used to find flow vectors in level 0. The vectors at level 0 represent the final optimal flow vector (u,v) of each pixel. The process is illustrated in Fig. 2. The displacement vector is calculated between every consecutive frame in F_s.

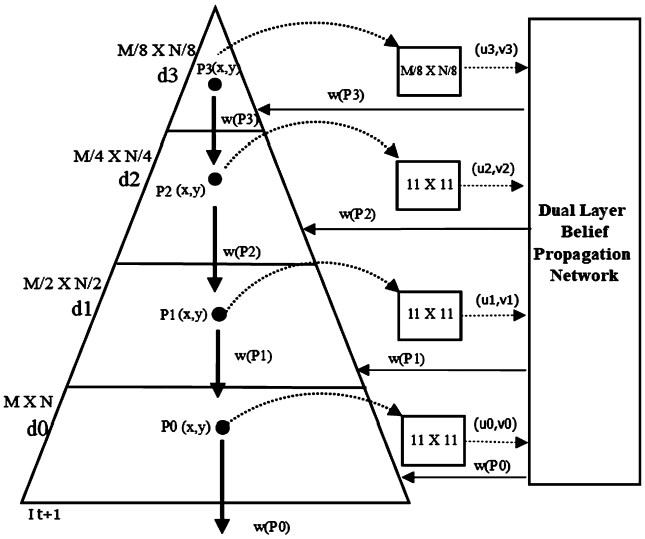

Fig. 2. Coarse to fine matching using BP network.

2.4.1. Belief networks

Belief Networks [23] are graph-based models in which nodes of the graph represent variables and connection between the nodes represent the relationship between the variables. The loopy belief propagation network [24] is similar to belief network except that it includes loops which process neighboring nodes in parallel. Let X be a pixel in an image and L be the set of possible values of flow vectors. The belief propagation network will find a label for X from labels in L based on minimizing energy function in (5). The algorithm works by passing messages between nodes of graph. The nodes are four neighbors of pixel under consideration. Each message has dimension same as number of possible states of flow vector. Suppose ${m_{rs}}^t$ is the message send by node r to the neighboring points. The message is initialized to zero. At each iteration t, the message is updated using (6)

$${m_{rs}}^t = \min_{f_r}(V(f_r, f_s) + D_r(f_r) + \sum_{e\epsilon N(r)\backslash s} {m_{er}}^{t-1}(f_r)) \quad (6)$$

where $V(f_r, f_s)$ and $D_r(f_r)$ denote data term and discontinuity term of energy function and $N(r)\backslash s$ denotes neighbors of r other than s. After T iterations a belief vector is computed for each node. The flow vector that minimizes belief vector is selected. The belief vector is given by (7).

$$b_s(f_s) = D_s(f_s) + \sum_{r\epsilon N(r)} {m_{rs}}^T(f_s) \quad (7)$$

2.5. Selection of frames

The redundant frames are eliminated from F_s based on the magnitude of flow vectors. Let (u_1, v_1) be the flow vector of a pixel in ${F_1}^*$. The magnitude of displacement is calculated from (u_1, v_1) as

$$mag = \sqrt{{u_1}^2 + {v_1}^2} \quad (8)$$

The sum of the magnitude of displacement for all the pixels in ${F_1}^*$ gives the magnitude of displacement of entire $\text{frame}(d_1)$. The magnitude of displacement is computed between each pair of consecutive frames in F_s. Let D represent the set consisting of displacement magnitude of consecutive frames in F_s. Suppose $d_1, d_2, d_3 \ldots\ldots d_{n_s-1}$ be the values of D.

Algorithm 1: Redundancy elimination from videos.

Input: Original Video V
Output: A subset of frames with keyframes, V_s

Calculate Frame rate of input video, fps
Find the total number of frames, n
Convert the input video into frames $F = F_1, F_2, ...F_n$
Divide F into segments $S_1, S_2, S_3, ...S_N$ of length $= fps$.
Presample F by selecting the frames at position $1, fps/2$ and fps of each segment to form F_s
$i \leftarrow 1$
while $i < length(F_s)$ **do**
    Read frames F_i *and* F_{i+1} from F_s
    Extract SIFT image corresponding to each frame.
    Construct pyramids using Gaussian filter from F_i and F_{i+1} with 4 levels.
    At level 4, Find labeling of flow vectors that corresponds to minimum energy as optimum flow vector $({u_1}^*, {v_1}^*)$ as in Section 2
    The process is repeated in subsequent levels to get final flow vector (u_1, v_1) and magnitude of flowvector is calculated as $d \leftarrow \sqrt{{u_1}^2 + {v_1}^2}$
end
Define a sliding window size of 3.
Consider array D which contain the magnitude of displacement of flow vectors of all the frames of F.
Process the elements of D within the window at a time and store it in temp.
Find mean m of those elements within the window.
Find frames within the window with displacement greater than m and store in $Newtemp$.
$n = \max(Newtemp)$
if $n == \emptyset$ **then**
    Add frame with frame number $(j + length(temp) + 1)$ *to* V_s
else
    Add frame with frame number $(j + n - 1)$ *to* V_s
    $Index \leftarrow Index + n$
Output V_s

To eliminate the redundant frames, a local threshold value is chosen based on frames within a sliding window. The sliding window size specifies the number of frames to be considered for finding local threshold value. Here, sliding window size of 3 is chosen so that the displacement magnitude of four consecutive frames is used to find local threshold value. The local threshold value for the first four consecutive frames of F_s is determined as the mean of d_1, d_2 and d_3. The mean value is chosen since it is simple to compute. Based on the analysis done, the mean value of the magnitude of displacement perform best to filter out redundant frames. Then, the frames having displacement greater than the local threshold value M is added to the reduced set of frames V_s. Similarly, thresholding and selection of frames is done for each sliding window over the entire set D. The sequence of steps for the proposed approach is given in Algorithm 1.

3. Analysis

The performance of the proposed method has been evaluated on VSUMM and OVP dataset. VSUMM consists of cartoon, sports and news videos which span 1 to 4 minutes duration. The OVP dataset has 50 documentary videos. Datasets also include ground truth created with user summaries of 5 different users for each video.

All implementation is done in MATLAB 2016a on Windows 10 Pro with an Intel(R) Core(TM) i7-3770 CPU at 3.40 GHz with 4.00 GB RAM running 64-bit operating system.

3.1. Performance metrics

The commonly used performance metrics for evaluating summaries are precision, recall and F-score. Since the proposed method is redundancy elimination step before summarization, three other metrics Reduction Rate (RR), error factor on comparison with user summaries (CUS_E) in [14] and Miss Rate (MR) are used to prove its efficiency. RR measures the percentage of redundant frames eliminated from videos, CUS_E measures the fraction of redundant frames in output compared to frames in user summary and MR measures the ratio of number of misses to the number of frames in ground truth. These metrics are calculated using (9), (10) and (11).

$$RR = \frac{N_{input} - N_{output}}{N_{input}} * 100 \tag{9}$$

Table 1
RR, CUS_E and MR of proposed method.

Dataset	RR	CUS_E	MR
VSUMM	97.13	9.49	0
OVP	97.8	8.5	0

$$CUS_E = \frac{N_{output} - N_{match}}{N_{US}} \tag{10}$$

$$MR = \frac{N_{miss}}{N_{US}} * 100 \tag{11}$$

where N_{input} represents total number of frames in input video V, N_{output} represents total number of frames in output, N_{miss} represents number of frames in user summary that is not present in the output, N_{match} represents number of frames in ground truth that are present in output V_s and N_{US} represents number of frames in user summary.

3.2. Results and discussion

The RR, CUS_E and MR of our proposed approach on VSUMM and OVP dataset is given in Table 1. The high reduction rate of our method shows that it can eliminate 97% of redundant frames from the frames remaining after uniform sampling. The MR of 0% shows that in spite of eliminating redundant frames, our method still preserve all the keyframes in the ground truth. So, the set of frames available for keyframe extraction after redundancy elimination step is only 3% of input frames preserving keyframes.

3.2.1. Trade off in varying the size of sliding window

The window size in our method specifies the number of frames to be considered at a time to find local threshold value. Here, a window size of 3 is chosen based on the analysis done on the results of proposed approach using videos of different categories. The comparison of evaluation metrics on results for three window sizes in cartoon videos of VSUMM dataset is presented in Fig. 3. It illustrates that the reduction rate is more and the error factor is less for each video with window size 4.

Results also show that a miss rate is 0 for window size 2 and 3. When window size 4 is chosen, some of the frames in the ground truth are missed from the set of output frames. This is because the frames selected

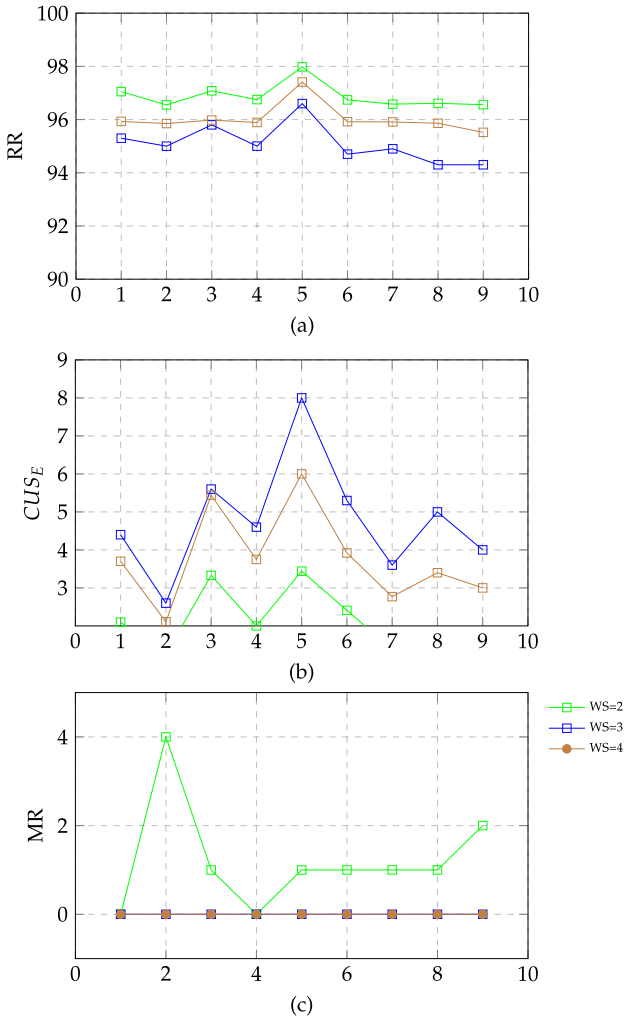

Fig. 3. Comparison of RR, CUS_E and MR at window size = 2, 3 and 4.

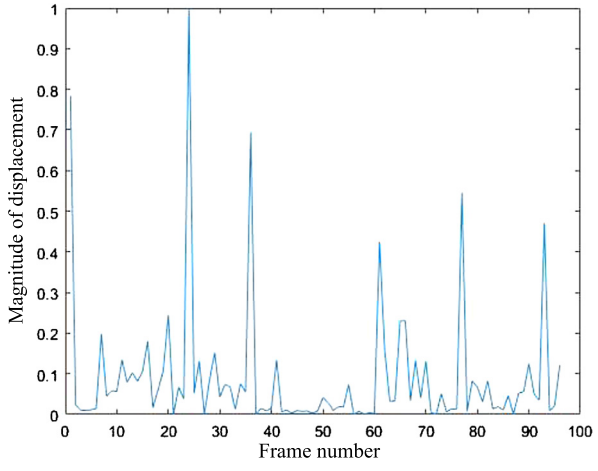

Fig. 4. Magnitude of displacement between consecutive frames of cartoon video $v_{11}.flv$ in VSUMM dataset.

for local averaging must be such that it belongs to same scene of a video. As shown in Fig. 4 displacement of consecutive frames varies abruptly and choice of window size including the frames after the abrupt change of displacement values leads to keyframe miss since those frames belong to a different scene. The redundancy elimination algorithm for video summarization should never miss a single keyframe since this affect the

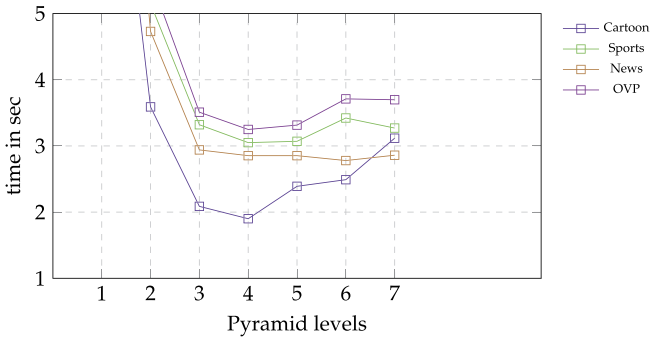

Fig. 5. Comparison of time for different pyramid levels.

efficiency of video abstracts. So, a window size of 3 is chosen to be optimum though window size of 4 is better in terms of RR and CUS_E.

3.2.2. *Finding the number of levels of the pyramid*

The number of levels in the pyramid is determined based on the analysis of time taken for the proposed method using different levels as in Fig. 5. It shows the average time taken by each category of video in VSUMM dataset and time taken by OVP dataset in seven pyramid levels. It shows the time is less at level 4 and it increases when the level is less than or greater than 4. This is because as the number of level increases, time decreases since the flow vectors computed at the coarse levels are used to initialize flow vectors at finer levels. But when the number of levels is 5, the size of the image at coarse level is too small and SIFT descriptors cannot be extracted from the image and its size is resized by padding which increases the time. So, pyramid level of 4 is chosen as best choice in the proposed method.

3.2.3. *Determination of sampling rate*

The sampling rate for the uniform sampling step is chosen to be two frames per second. It is chosen based on the analysis done on different sampling rates. A trade-off between sampling rate and loss of information is very essential in case of summarization. Since this is only a preliminary stage we aim to preserve each distinct frame in the input video by reducing the redundancy. Most of the approaches in [25], [14], [15] selects one frame per second. This may lead to missing of distinct frames when low frame rate videos are used. If the frame rate is high the changes between consecutive frames is less and sampling rate of one frame per second yield good results. The uniform sampling step in the proposed approach achieved average RR of 91% on OVP and VSUMM database with no keyframe miss. The remaining 9% of frames are processed to find the magnitude of displacement between the consecutive frames.

3.2.4. *Parameter setting of the energy function*

The value parameters η, α and d of the energy function given in (1) is fixed in our experiments and taken as in [26] for object tracking in videos such that $\alpha = 300$, $\eta = 0.5$ and $d = 3$.

3.2.5. *Comparison with other techniques*

We compared the performance of proposed method with other redundancy elimination methods Uniform Sampling (US) [27], Mutual Information (MI) [15], Singular Value Decomposition (SVD) [28]. Fig. 6 shows the comparison of performance of proposed method with other methods. We achieved the highest reduction rate with less error on VSUMM and OVP dataset though SVD and US achieved 0%MR. Detailed results on category basis are summarized in Table 2. It shows that method attain RR of 97% consistently on videos of all categories with 0%MR. But, for sports videos CUS_E is more which shows that a small fraction of redundant frames are eliminated. This is so because in sports videos scene change is gradual compared to other two categories of videos which shows an abrupt scene change. As a result, in sports videos

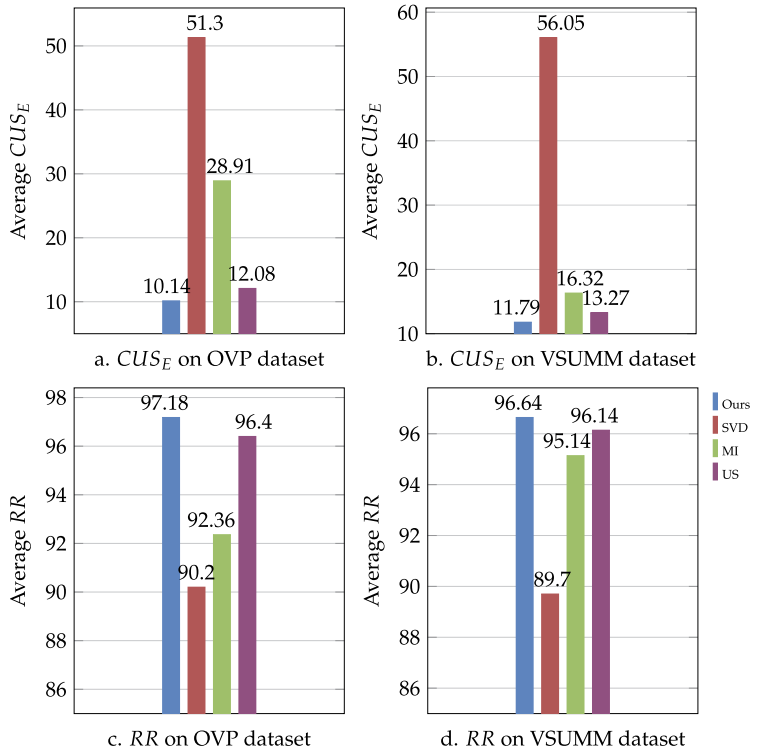

Fig. 6. Comparison of results of redundancy elimination step of the proposed method with other state-of-the-art methods.

Table 2
Results of various categories of videos in VSUMM dataset.

Category	Metric	Ours	SVD	MI	US
Cartoon	RR	96.88	81.45	94.11	96.0
	CUS_E	2.97	22.24	6.28	3.85
	MR	0	20	0	0
Sports	RR	97.15	93.53	95.59	96.04
	CUS_E	17.29	119.03	22.84	19.47
	MR	0	65.17	0	0
News	RR	97.3	94.34	95.73	96.39
	CUS_E	15.11	26.88	19.86	16.49
	MRNews	0	43.2	0	0

displacement of pixels between consecutive frames is less. However, our method when compared to existing redundancy elimination methods gained less CUS_E for sports videos. The uniform sampling is computationally simple. The uniform sampling method achieved 0%MR and RR and CUS_E close to our method. But the analysis shows that rather than using random uniform selection, eliminating similar frames based on flow vectors achieved better results.

3.2.6. Impact of redundancy elimination step

To evaluate effectiveness of our redundancy elimination step, we compared the results of the methods presented in VISCOM [27], VRHDPS [28], VSUMM [14] after including proposed redundancy elimination step before summarization. Fig. 7 (a)-(f) shows the results of comparison. Results show that the methods achieved better accuracy when summarization is done using set of frames from which redundant frames are discarded. This is so because the features between consecutive frames are more discriminative when the redundant frames between two distinctive frames are eliminated.

3.2.7. Limitations and future work

Though our method attained better results, it has some limitations. We were able to achieve an error factor of only 17.29 for sports videos. Even though this value is low compared to the error factor of the existing approaches, it still reveals the presence of some redundant frames in the output. This is mainly due to the presence of multiple redundant frames between two distinct frames and the displacement between these frames are too low compared to other categories of videos used for evaluation. The motion vectors used by the method failed to make these displacements more evident. The small displacements can be captured using other features which can represent the high-level characteristics of the video. The major concern in including such a step prior to summarization is that it raises the overall complexity of the summarization algorithm. Hence, the appropriate method should be chosen in such a way that the quality of the summaries generated are not compromised while maintaining the computational complexity in limited range. In our future work we plan to address this issue.

4. Conclusion

We proposed a novel method for redundancy elimination from input videos using uniform sampling followed by SIFT Flow algorithm so that none of the keyframes is missed. This reduced set of frames which maintain content of the original video can be used for summarization. The approach achieved high reduction rate with less error factor and 0% Miss Rate. The method when tested on videos of different categories achieved promising results. The proposed method of redundancy elimination is applied on existing summarization methods to prove the impact of the step in the final results.

Declarations

Author contribution statement

Jesna Mohan, Madhu S. Nair: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

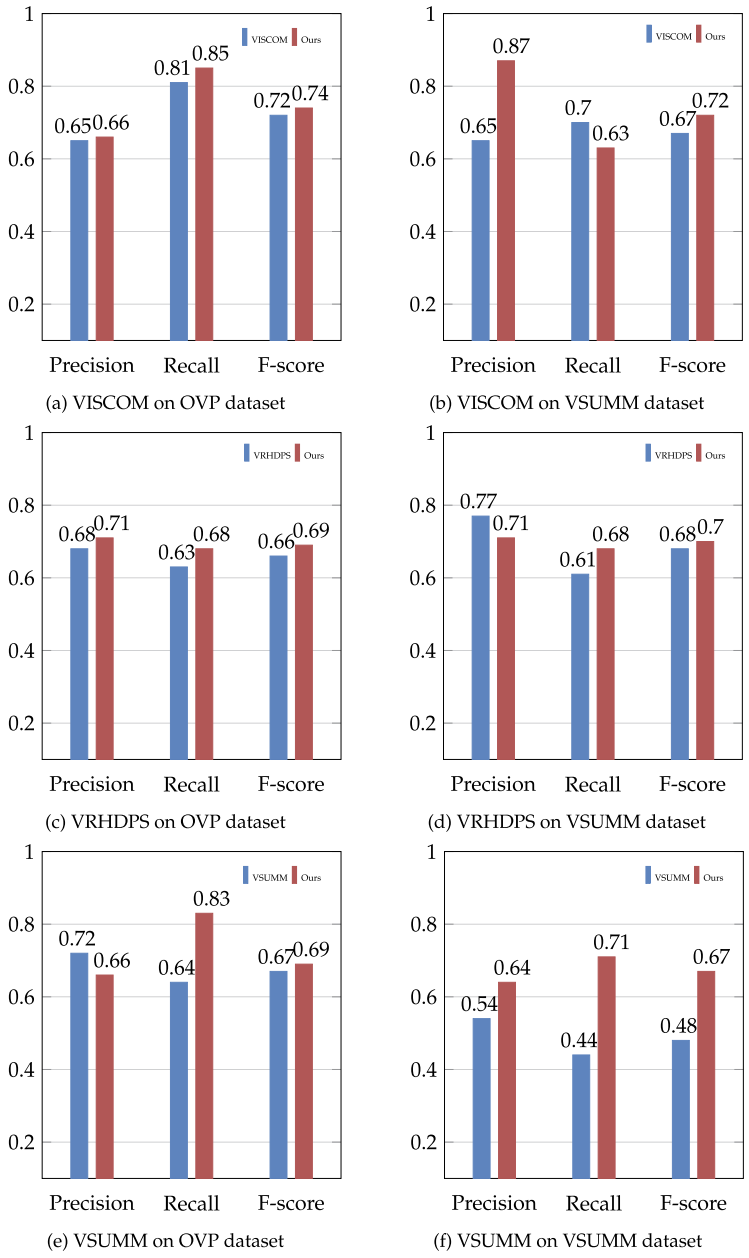

Fig. 7. Comparison with other methods showing the impact of redundancy elimination step on final results.

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] S.S. Thomas, S. Gupta, K. Venkatesh, Perceptual synoptic view-based video retrieval using metadata, Signal Image Video Process. 11 (2017) 549–555.
[2] M. Furini, F. Geraci, M. Montangero, M. Pellegrini, STIMO: STIll and MOving video storyboard for the web scenario, Multimed. Tools Appl. 46 (2010) 47–69.
[3] L.I. Kuncheva, P. Yousefi, J. Almeida, Edited nearest neighbour for selecting keyframe summaries of egocentric videos, J. Vis. Commun. Image Represent. 52 (2018) 118–130.
[4] Z. Gao, G. Lu, P. Yan, L. Wang, Retrospective analysis of time series for frame selection in surveillance video summarization, Signal Image Video Process. 11 (2017) 581–588.
[5] S.E. de Avila, A. da Luz Jr, A.d.A Araújo, M. Cord, VSUMM: An approach for automatic video summarization and quantitative evaluation, in: IEEE XXI Brazilian Symposium on Computer Graphics and Image Processing, 2008, SIBGRAPI'08, 2008, pp. 103–110.
[6] N. Ejaz, T.B. Tariq, S.W. Baik, Adaptive key frame extraction for video summarization using an aggregation mechanism, J. Vis. Commun. Image Represent. 23 (2012) 1031–1040.
[7] G. Guan, Z. Wang, S. Lu, J. Da Deng, D.D. Feng, Keypoint-based keyframe selection, IEEE Trans. Circuits Syst. Video Technol. 23 (2013) 729–734.
[8] R. Hannane, A. Elboushaki, K. Afdel, P. Naghabhushan, M. Javed, An efficient method for video shot boundary detection and keyframe extraction using sift-point distribution histogram, Int. J. Multimed. Inf. Retr. 5 (2016) 89–104.
[9] H.C. Chang, C.K. Yang, Fast content-aware video length reduction, Signal Image Video Process. 8 (2014) 1383–1397.
[10] D. Zhu, Y. Luo, L. Dai, X. Shao, Q. Zhou, L. Itti, J. Lu, Salient object detection via a local and global method based on deep residual network, J. Vis. Commun. Image Represent. 54 (2018) 1–9.
[11] B. Mahasseni, M. Lam, S. Todorovic, Unsupervised video summarization with adversarial LSTM networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017.
[12] M. Fei, W. Jiang, W. Mao, Memorable and rich video summarization, J. Vis. Commun. Image Represent. 42 (2017) 207–217.

[13] M. Ma, S. Mei, S. Wan, Z. Wang, D. Feng, Video summarization via nonlinear sparse dictionary selection, IEEE Access 7 (2019) 11763–11774.
[14] S.E.F. De Avila, A.P. Brandão Lopes, A. da Luz, A. de Albuquerque Araújo, VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method, Pattern Recognit. Lett. 32 (2011) 56–68.
[15] S.K. Kuanar, R. Panda, A.S. Chowdhury, Video key frame extraction through dynamic Delaunay clustering with a structural constraint, J. Vis. Commun. Image Represent. 24 (2013) 1212–1227.
[16] G.H. Song, Q.G. Ji, Z.M. Lu, Z.D. Fang, Z.H. Xie, A novel video abstraction method based on fast clustering of the regions of interest in key frames, AEÜ, Int. J. Electron. Commun. 68 (2014) 783–794.
[17] C. Chattopadhyay, S. Das, Use of trajectory and spatiotemporal features for retrieval of videos with a prominent moving foreground object, Signal Image Video Process. 10 (2016) 319–326.
[18] D.M. Tsai, W.Y. Chiu, M.H. Lee, Optical flow-motion history image (OF-MHI) for action recognition, Signal Image Video Process. 9 (2015) 1897–1906.
[19] K. Taşdemir, A.E. Cetin, Content-based video copy detection based on motion vectors estimated using a lower frame rate, Signal Image Video Process. 8 (2014) 1049–1057.
[20] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.
[21] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: European Conference on Computer Vision, Springer, 2004, pp. 25–36.
[22] D. Sun, S. Roth, M.J. Black, Secrets of optical flow estimation and their principles, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2010, pp. 2432–2439.
[23] J. Pearl, Fusion, propagation, and structuring in belief networks, Artif. Intell. 29 (1986) 241–288.
[24] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient belief propagation for early vision, Int. J. Comput. Vis. 70 (2006) 41–54.
[25] F.F. Chamasemani, L.S. Affendey, N. Mustapha, F. Khalid, Video abstraction using density-based clustering algorithm, Vis. Comput. (2017) 1–16.
[26] H. Zhang, Y. Wang, L. Luo, X. Lu, M. Zhang, Sift flow for abrupt motion tracking via adaptive samples selection with sparse representation, Neurocomputing 249 (2017) 253–265.
[27] M.V.M. Cirne, H. Pedrini, VISCOM: A robust video summarization approach using color co-occurrence matrices, Multimed. Tools Appl. (2017) 1–19.
[28] J. Wu, S.h. Zhong, J. Jiang, Y. Yang, A novel clustering method for static video summarization, Multimed. Tools Appl. 76 (2016) 1–17.