Video Summarization Using Novel Video Decomposition Algorithm

Saumik Bhattacharya, KS Venkatesh and Sumana Gupta

Abstract Because of the large amount of data, storage and processing of video are always a challenging problem. It becomes even more difficult in surveillance videos because of the length of the video data. Thus, it is extremely necessary to design algorithms for faster browsing of the video data with as much compression as possible. In this paper, we propose a novel decomposition algorithm that reduces the redundancy of a video cube by segmenting the motion salient regions using total variation approach. We further use the decomposition algorithm to summarize a video shot for easy interpretation of the event in the video shot. We propose two different methods for the summarization process and demonstrate that the video summarization reduces the storage requirement drastically without sacrificing the understanding of video content.

1 Introduction

Video is one of the most popular medias in today's world. It has been extensively used for surveillance, communication, defense, and entertainment purposes. However, it is difficult to store and browse video data because of its huge volume. Though different video compression algorithms have been designed that save the storage space, they cannot reduce the browsing time of a video. Thus, it is necessary to design new algorithms that can reduce the demand of storage space and browsing time such that the events in a video can be easily interpreted. One of the most popular approaches is the generation of video storyboard or video skimming [1, 5, 6, 10]. However, it is difficult to understand an event from video skimming and often it requires multiple frames to represent a video. Recently, video summarization techniques become popular because of more efficient browsing and file size management [3, 9, 11, 12, 15]. Video summarization algorithms detect the salient events in a video and generate an image that represents the event without disturbing the continuity of

S. Bhattacharya (✉) · K. Venkatesh · S. Gupta
Indian Institute of Technology Kanpur, Kanpur, India
e-mail: saumik@iitk.ac.in

B. B. Chaudhuri et al. (eds.), *Proceedings of 2nd International Conference on Computer Vision & Image Processing*, Advances in Intelligent Systems and Computing 704, https://doi.org/10.1007/978-981-10-7898-9_32

the event. In [11], the authors proposed an interactive video summarization algorithm. In [12], authors extracted the portion of a video where the events are denser than the other parts and used the extracted video for summarization. Sunkavalli et al. proposed a saliency-based algorithm to summarize a video [15]. In [8], authors exploited the spatiotemporal information along with graph-cut method to generate space-time montage from the input video. In [14], authors proposed a multi-scale approach to summarize a video with minimum visual distortion. However, most of these algorithms cannot segment the moving regions precisely and the edges get blurred in the final summarized image [15].

In this paper, we propose a video summarization algorithm based on video decomposition. The video decomposition algorithm extracts the motion salient regions with sharp object boundary. Then, we temporally sample the motion salient regions to generate the final summarized image. We use both uniform sampling and non-uniform sampling to summarize an input video.

Contributions:

- We propose a novel decomposition algorithm that is parallelizable.
- We use both uniform and non-uniform sampling to generate summarized images of input videos with minimum visual distortion.
- As the summarization algorithms are based on the video decomposition technique, the final summarized image has sharp object boundaries.

Rest of the paper is divided as follows. In Sect. 2, the novel video decomposition algorithm is discussed along with the summarization techniques. In Sect. 3, we discuss different aspects of the video decomposition algorithm. We also summarize different input videos using both uniform and non-uniform sampling. Finally, we conclude the paper in Sect. 4 discussing the impact of the work with its future prospects.

2 Proposed Algorithm

Before discussing the detection and restoration process of the artifacts, we briefly discuss the parallelizable video decomposition scheme that we have used in all restoration process. From an input video, the decomposition technique estimates one background video where the frames are visually similar and a residual video that has all the remaining information. The residual video is then used to summarize the motion information and the visually similar video is used to estimate the background. In Fig. 1, we show the basic block diagram of the proposed algorithm.

As the main objective of video decomposition is to decompose the input video, say $\mathbf{V}$ into background video $\mathbf{L}$ and feature video $\mathbf{S}$, we will first estimate the background video from the input video cube and then construct the feature video $\mathbf{S}$ using $\mathbf{V}$ and $\mathbf{L}$.

Let us assume that an input video $\mathbf{V}$ has K number of frames with frame resolution $M \times N$. If a pixel $\mathbf{p} = (x, y)$ is in the background of the video, the intensity will not vary at that particular pixel location along the time axis. Thus, if we consider a

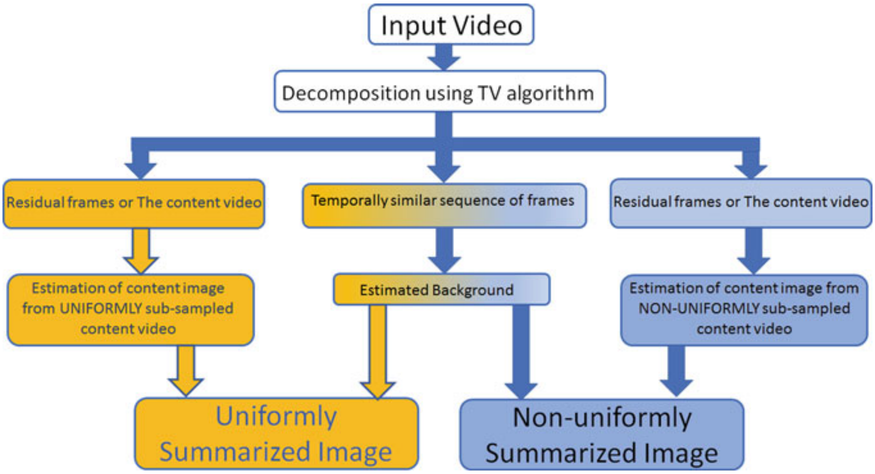
Input Video
Decomposition using TV algorithm
Residual frames or The content video
Temporally similar sequence of frames
Residual frames or The content video
Estimation of content image from UNIFORMLY sub-sampled content video
Estimated Background
Estimation of content image from NON-UNIFORMLY sub-sampled content video
Uniformly Summarized Image
Non-uniformly Summarized Image

Fig. 1 Block diagram of the proposed algorithm

vector $\mathbf{l_p}$ at pixel location $\mathbf{p}$ along time axis, such that l_p^i, ith element of the vector, represents the intensity at pixel location $\mathbf{p}$ in ith frame of input video $\mathbf{V}$, then if we calculate a vector $\mathbf{x_p}$ such that

$$\mathbf{x_p} = [l_p^1 - l_p^2, l_p^2 - l_p^3, \ldots, l_p^{K-1} - l_p^K]^t \tag{1}$$

the vector $\mathbf{x_p}$ will be a sparse vector if $\mathbf{p}$ belongs to the background of the video. We can represent Eq. 1, as $\mathbf{x_p} = \mathbf{Dl_p}$, where $\mathbf{D}$ is the variation matrix. If we take the consecutive element-wise difference, as shown in Eq. 1, then we can express $\mathbf{D}$ as

$$\mathbf{D} = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ 0 & 1 & -1 & \ldots & 0 \\ & & & \ddots & \\ 0 & 0 & \ldots & 1 & -1 \end{bmatrix}_{(K-1)\times K}$$

Using this idea of background pixel, we consider a vector $\mathbf{v_p}$ at any pixel location $\mathbf{p} \in M \times N$, such that v_p^i, the ith element of the vector, represents the intensity at pixel location $\mathbf{p}$ in ith frame of input video $\mathbf{V}$. Then to get the background intensity, we estimate $\mathbf{l_p}$ from $\mathbf{v_p}$ such that $\mathbf{Dl_p}$ is a sparse vector. We define the optimization problem as

$$\begin{aligned} &\underset{\mathbf{l_p}}{\text{minimize}} \quad \{\left\|\mathbf{v_p} - \mathbf{l_p}\right\|_2^2 + \lambda \left\|\mathbf{Dl_p}\right\|_0\} \\ &\text{subject to} \quad \lambda \geq 0 \end{aligned} \tag{2}$$

where $\|.\|_2$ and $\|.\|_0$ denote l_2 norm and l_0 norm of a vector, respectively. The first term of the expression is the data fidelity term and the second term ensures that the estimated vector $\mathbf{l_p}$ is smooth and λ is a non-negative weight that determines the level of smoothness in the final estimate of $\mathbf{l_p}$. As λ increases, estimated $\mathbf{l_p}$ becomes smoother, i.e., $\mathbf{Dl_p}$ becomes sparser.

As the optimization problem defined in Eq. 2 is a non-convex problem, the estimation of optimal $\mathbf{l_p}$ is NP-hard. To reduce the computational complexity, keeping the concept of sparsity intact, we replace the l_0 with l_1 norm [2, 4] and modify the optimization problem of Eq. 2 as

$$\begin{aligned} &\underset{\mathbf{l_p}}{\text{minimize}} \quad \{\left\|\mathbf{v_p} - \mathbf{l_p}\right\|_2^2 + \lambda \left\|\mathbf{Dl_p}\right\|_1\} \\ &\text{subject to} \quad \lambda \geq 0 \end{aligned} \tag{3}$$

To solve the convex problem defined in Eq. 3, we apply iterative reweighted norm (IRN) approach. IRN uses the concept of iterative reweighted least square (IRLS) method to convert l_p norm of a vector to weighted l_2 norm. This solves the optimization problem in fewer iterations [16] as l_2 norm is differentiable and leads to a closed form solution with an iterative update step of the weight matrix. A simplified form of IRN states that l_p norm minimization of $\mathbf{q} = [q_1, q_2, \ldots q_n]^t$ can be solved using an weighted least square problem as,

$$\|\mathbf{q}\|_p^p = \sum_j |q_j|^p = \left\|\mathbf{R}^{1/2}\mathbf{q}\right\|_2^2 \tag{4}$$

where $\mathbf{R}$ is a diagonal matrix with each diagonal element defined as $\mathbf{R}_{i,i}^{1/2} = (|q_i|^{1-p/2} + \epsilon)^{-1}$, and ϵ is a small positive constant added to avoid division by zero [16].

Using the concept of IRLS, we modify Eq. 3 and define the cost function $C(\mathbf{l_p}^{(k)})$ as

$$C(\mathbf{l_p}^{(k)}) = \frac{1}{2}\left\|\mathbf{v_p} - \mathbf{l_p}^{(k)}\right\|_2^2 + \frac{\lambda}{2}\left\|\mathbf{R}^{(k)^{1/2}}\mathbf{Dl_p}^{(k)}\right\|_2^2 \tag{5}$$

where weighting matrix $\mathbf{R}^{(k)^{1/2}}$ is calculated considering $\mathbf{q}^{(k)} = \mathbf{Dl_p}^{(k-1)}$.

To minimize the cost function, we differentiate right-hand side with respect to $\mathbf{l_p}^{(k)}$ and set that equal to zero. A mathematical simplification gives us

$$\mathbf{l_p}^{(k)} = (\lambda\mathbf{D}^t\mathbf{R}^{(k)}\mathbf{D} + \mathbf{I})^{-1}\mathbf{v_p} \tag{6}$$

$$\mathbf{l_p}^{(k)} = \Psi^{-1}\mathbf{v_p} \tag{7}$$

where $\mathbf{I}$ is the identity matrix of dimension $K \times K$ and $\Psi = \lambda\mathbf{D}^t\mathbf{R}^{(k)}\mathbf{D} + \mathbf{I}$. We may end the iteration process if $\mathcal{Q}(\mathbf{l_p}^{(k)}) - \mathcal{Q}(\mathbf{l_p}^{(k-1)}) = \mathbf{0}$ or if $\text{rank}(\Psi) < K$, where $\mathcal{Q}$ is a quantizer that quantizes element-wise a floating value to its nearest integer and $\mathbf{0}$ is a null vector. It is important to note that Ψ is a symmetric matrix as $\mathbf{R}^{(k)}$ is a diagonal matrix and Ψ has a diagonal loading, i.e., the matrix Ψ is invertible even if $\mathbf{D}^t\mathbf{R}^{(k)}\mathbf{D} = \mathbf{0}$. The linear system $\Psi\mathbf{l_p}^{(k)} = \mathbf{v_p}$ defined in Eq. 7 can be solved for $\mathbf{l_p}^{(k)}$ using Newton's method without performing the matrix inversion explicitly [13].

Finally, we construct the two videos $\mathbf{L}$ and $\mathbf{S}$ where video $\mathbf{L}$ contains the background information of the input video and $\mathbf{S}$ contains the motion information of the input video $\mathbf{V}$ defined as $\mathbf{S} = \mathbf{V} - \mathbf{L}$. The intensity values at pixel location $\mathbf{p}$ in the ith frame are l_p^i and s_p^i for videos $\mathbf{L}$ and $\mathbf{S}$, respectively, where l_p^i is the ith element of the estimated vector $\mathbf{l_p}^{(k)}$.

To summarize the video frames, first we calculate the background image B as

$$b_p = \frac{\sum_{i=1}^{K} l_p^i}{K}$$

where b_p is the intensity at pixel $\mathbf{p}$ in background B.

Next, we uniformly sub-sample the feature video $\mathbf{S}$ with sampling rate z to construct a video $\mathbf{S}_u$ such that

$$S_u^j = S^{zj}; \quad j = 1, 2, 3 \ldots \lfloor K/z \rfloor \tag{8}$$

where S_u^i and S^i are the ith frames of videos $\mathbf{S}_u$ and $\mathbf{S}$, respectively.

As $\mathbf{V} = \mathbf{L} + \mathbf{S}$, $\mathbf{S}_u$ or $\mathbf{S}$ does not contain the actual intensity values of a moving object. We apply adaptive thresholding on video $\mathbf{S}_u$ to extract the moving object. If $\mathbf{F}$ is uniformly sampled motion segmented video, then,

$$f_p^j = \begin{cases} v_p^j & \text{if } |s_{u_p}^j| \geq \tau_j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $s_{u_p}^j$ is the intensity at pixel location $\mathbf{p}$ in the jth frame of video $\mathbf{S}_u$ and τ_j is an adaptive constant calculated as $\tau_j = \mu_j + \sigma_j$, where μ_j and σ_j are the mean and standard deviation of frame S_u^j, respectively, and $j \in J_u$ where $J_u = \{1, 2, 3, \ldots \lfloor K/z \rfloor\}$.

We generate the uniformly summarized image I_u as

$$i_{u_p} = \begin{cases} {f_p}^j & \text{if } {f_p}^j \neq 0; \text{ for any } j \in J_u \\ b_p & \text{otherwise} \end{cases} \tag{10}$$

where i_{u_p} is the intensity value at pixel location $\mathbf{p}$ in I_u.

Though uniformly summarized image generates satisfactory results in simple videos, it performs poorly if the scene has non-uniform motion, acceleration, or multiple moving objects. Thus, we define another approach to summarize an input video non-uniformly.

To do so, we first segment the motion information present in input video **V** using the information present in **S**. If **U** is the final motion segmented video, then,

$$u_p^i = \begin{cases} v_p^i & \text{if } |s_p^i| \geq \tau_i \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where u_p^i and s_p^i are the intensities at pixel location **p** in the ith frame of videos **U** and **S**, respectively, and τ_i is an adaptive constant calculated as $\tau_i = \mu_i + \sigma_i$ where μ_i and σ_i are the mean and standard deviation of frame S^i, respectively.

Next, we select the indices of the frames such that

$$J_n = \{i : d(U^i) \cap d(U^h) = \phi,\ \ i, h \in K, i \neq h\} \tag{12}$$

where $d(.)$ is dilation operation performed on a frame, ϕ is an null matrix and $\cap$ computes spatial intersection of nonzero elements in two images. Suppose, we get a sampled video $\mathbf{F}_n$ such that

$$F_n^i = U^i;\ \ i \in J_n \tag{13}$$

where F_n^i and U^i are the ith frames of videos $\mathbf{F}_n$ and **U**, respectively. We construct the final non-uniformly summarized image I_n as

$$i_{n_p} = \begin{cases} f_{n_p}^i & \text{if } f_{n_p}^i \neq 0;\ \text{for any } i \in J_n \\ b_p & \text{otherwise} \end{cases} \tag{14}$$

where $f_{n_p}^i$ is the intensity at pixel location **p** in the ith frame of video $\mathbf{F}_n$.

3 Experimental Results

To validate the decomposition algorithm and the summarization algorithms, we test them on different input videos. Figure 2a shows frame from a typical input video. Figure 2c shows the respective frames from feature video **S**. In Fig. 3a, we show the estimated $\mathbf{l_p}$ for different λ values. The input vector $\mathbf{v_p}$ is the change in intensity at the center of the red circle shown in Fig. 2a. The change in rank of the video **L** is shown in Fig. 3b. The rank of the video **L** is calculated as described in [7]. It is important to inform that in our previous work [2], we reported a parallelizable decomposition method based on majorization-minimization algorithm. However, the algorithm in [2] takes large number of iterations (∼500) to complete the decomposition. As shown in Fig. 3b, the proposed decomposition minimizes the rank in much smaller number of steps (∼60) without increasing the complexity of the algorithm. Thus, the proposed decomposition algorithm is faster than the algorithm mentioned in [2]. In Fig. 3c and d, we compare the execution times of these decomposition algorithms on different datasets and further validate the claim. As the decomposition algorithms are pixel based, the algorithms are parallelizable and increase in number of cores in the processor reduces the execution times in both the cases.

In Fig. 4, we show the outputs of both the summarization algorithms for different input videos. In Fig. 4a–d, we show frames of the input videos. All the videos in the dataset contain complex motions like acceleration, multiple objects, nonlinear motion, etc. Figure 4e shows the respective summarized images using the uniform

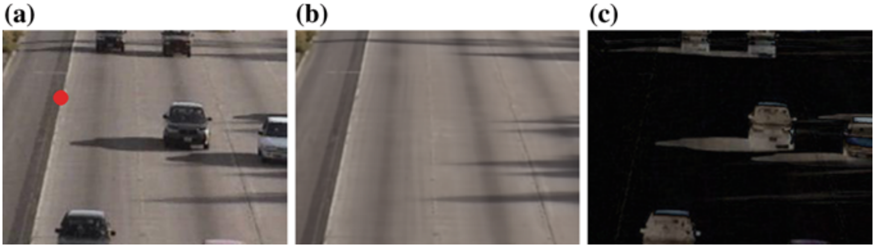

Fig. 2 **a** Estimation of $\mathbf{l_p}$ for different values of λ; **b** rank of **L** versus iteration for $\lambda = 100$; **c** execution time of [2] with different number of cores for different dataset; **d** execution time of the proposed algorithm

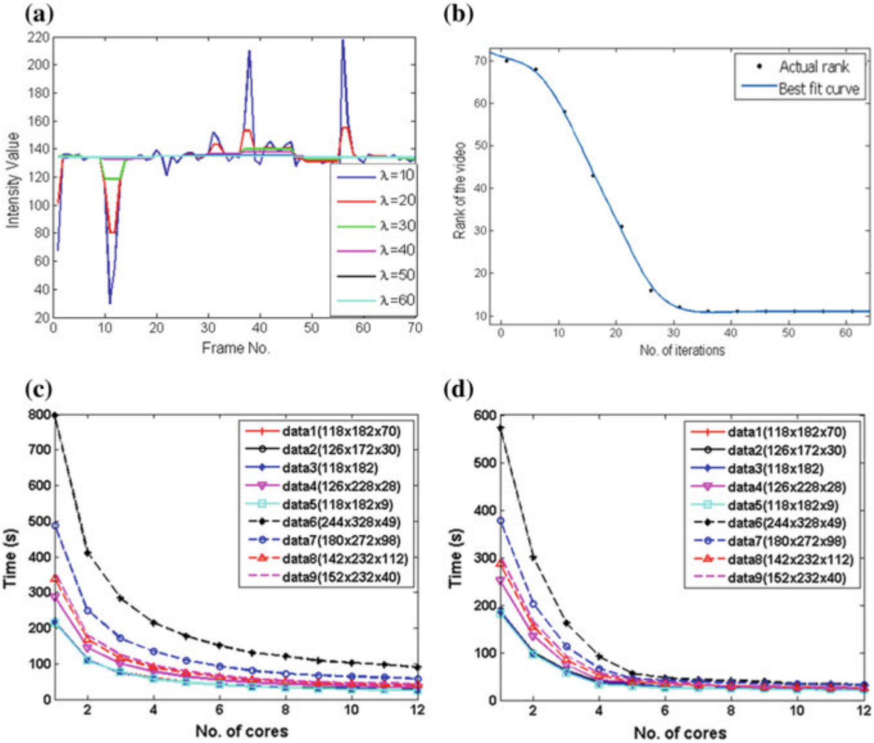

Fig. 3 **a** Frame of an input video; **b** estimated background **c** respective frame from **S** video

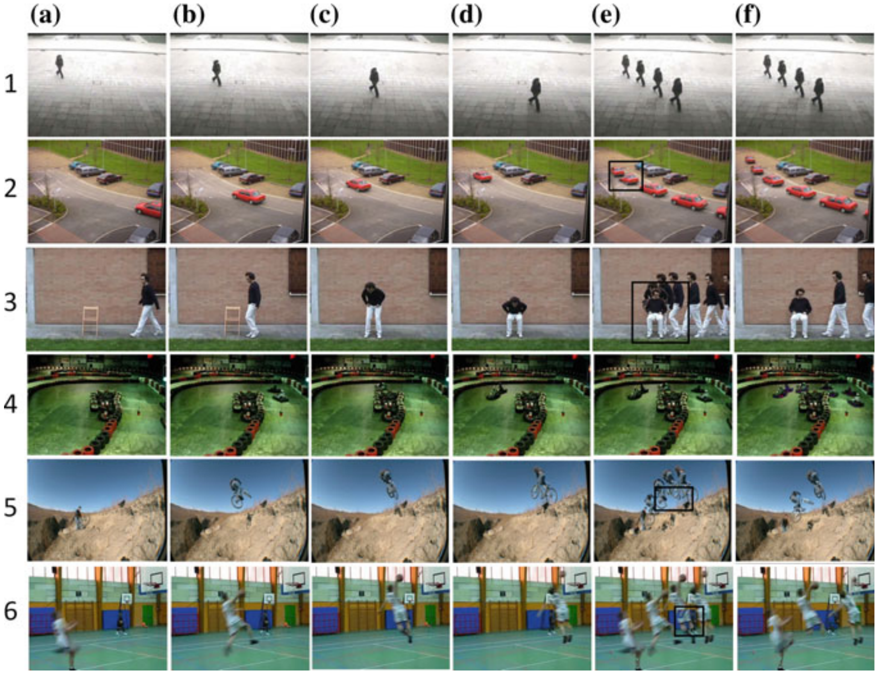

Fig. 4 **a**–**d** Frames of input videos; **e** uniformly summarized images I_u; **f** non-uniformly summarized image I_n

summarization method, and Fig. 4f shows the summarized images using non-uniform summarization method. However, as mentioned in Sect. 2, the uniformly summarized image I_u may contain distortion due to overlapping regions. In Fig. 4e, we show the overlapping regions within the black rectangles. As shown in Fig. 4f, the non-uniformly summarized images I_n are free from such artifacts. An interesting property of non-uniform summarization algorithm is that the summarized image I_n may differ for the same input video depending on the frame to initialize the summarization process. This is further depicted in Fig. 5. For the same input videos, Fig. 5a and b show the final non-uniformly summarized images initialized from the first frame and the last frame, respectively.

Though it is easier to interpret an event in a summarized video, video summarization algorithm drastically reduces the file size as it generates a single image as the final output. In Table 1, the sizes of the input videos and the summarized images I_u and I_n are shown. The execution times of both the proposed summarization algorithms are also tabulated in Table 1. It is evident that the space-time requirements of both the algorithms are comparable. All the evaluations are done in MATLAB 2013 using 4 cores running on an Intel(R) core(TM) i7-4770 3.90 GHz processor with 8 GB RAM.

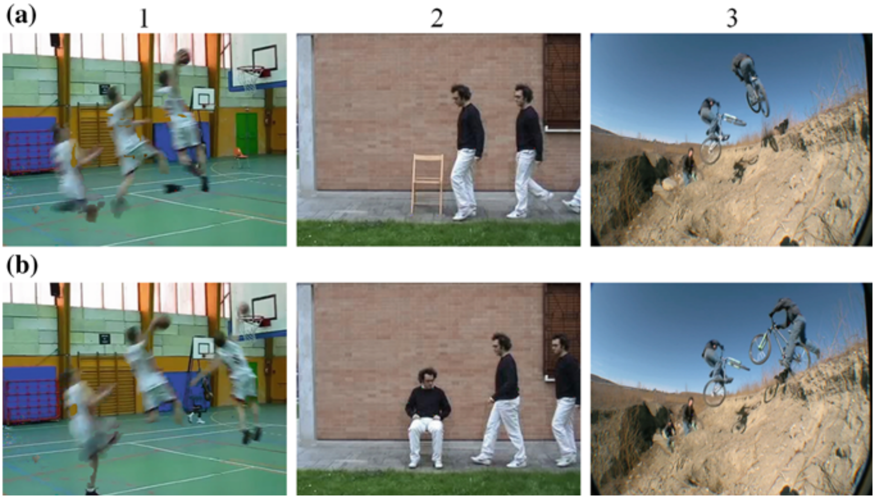

Fig. 5 **a** Non-uniformly summarized image starting from the first frame; **b** non-uniformly summarized image starting from the last frame

Table 1 File size comparison after summarization for different videos

Video name	Frame dimension	Video size	Summarized (linear)	Summarized (nonlinear)	Run time (linear)	Run time (nonlinear)
Walk	384 × 288	4.3 MB	68.1 KB	67.9 KB	10.31 s	12.72 s
Cars	768 × 567	4.74 MB	48.4 KB	47.5 KB	13.96 s	15.91 s
Dunk	480 × 360	2.88 MB	197 KB	191 KB	12.18 s	15.27 s
Gokart	640 × 360	9.1 MB	315 KB	312 KB	11.76 s	13.46 s
Flip	960 × 540	16.4 MB	672 KB	644 KB	8.1 s	12.62 s
Man	320 × 240	23.2 MB	123 KB	107 KB	12.38 s	14.81 s

4 Conclusion

Storage and interpretation of videos require large amount of resources. It is crucial to develop algorithms which can represent an input video consuming as minimum resource as possible without disturbing the flow of the events. In this paper, we proposed two algorithms to summarize an input video to an image using uniform and non-uniform sampling of the video frames. The methods consume little amount of disk space and can be executed in small amount of time as the entire algorithms are pixel based and can be executed using parallel processing. Though the uniformly summarized image may contain some distortion depending upon the content of the input video, the non-uniformly summarized image is always distortion-free. However, the non-uniformly summarized image requires slightly more resources.

Though the proposed summarization algorithms work for static cameras, it is necessary to design such algorithm in future for videos with camera motions.

References

1. Bhattacharya, S., Gupta, S., Venkatesh, K.: Video shot detection & story board generation using video decomposition. In: Proceedings of the Sixth International Conference on Computer and Communication Technology 2015. pp. 223–227. ACM (2015)
2. Bhattacharya, S., Venkatsh, K., Gupta, S.: Background estimation and motion saliency detection using total variation-based video decomposition. Signal, Image and Video Processing 11(1), 113–121 (2017)
3. Brunelli, R., Mich, O., Modena, C.M.: A survey on the automatic indexing of video data. Journal of visual communication and image representation 10(2), 78–112 (1999)
4. Donoho, D.L.: Compressed sensing. Information Theory, IEEE Transactions on 52(4), 1289–1306 (2006)
5. Furini, M., Geraci, F., Montangero, M., Pellegrini, M.: Stimo: Still and moving video storyboard for the web scenario. Multimedia Tools and Applications 46(1), 47 (2010)
6. Goldman, D.B., Curless, B., Salesin, D., Seitz, S.M.: Schematic storyboarding for video visualization and editing. In: ACM Transactions on Graphics (TOG). vol. 25, pp. 862–871. ACM (2006)
7. J., C.E., Li, X., Ma, Y., Wright., J.: Robust principal component analysis? J. ACM 58 no.3 (2011)
8. Kang, H.W., Chen, X.Q.: Space-time video montage. In: computer vision and pattern recognition, 2006 IEEE computer society conference on. vol. 2, pp. 1331–1338. IEEE (2006)
9. Money, A.G., Agius, H.: Video summarisation: A conceptual framework and survey of the state of the art. Journal of Visual Communication and Image Representation 19(2), 121–143 (2008)
10. Mundur, P., Rao, Y., Yesha, Y.: Keyframe-based video summarization using delaunay clustering. International Journal on Digital Libraries 6(2), 219–232 (2006)
11. Pritch, Y., Rav-Acha, A., Peleg, S.: Nonchronological video synopsis and indexing. IEEE transactions on pattern analysis and machine intelligence 30(11), 1971–1984 (2008)
12. Rav-Acha, A., Pritch, Y., Peleg, S.: Making a long video short: Dynamic video synopsis. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. vol. 1, pp. 435–441. IEEE (2006)
13. Rodríguez, P., Wohlberg, B.: Efficient minimization method for a generalized total variation functional. IEEE Transactions on Image Processing 18(2), 322–332 (2009)
14. Simakov, D., Caspi, Y., Shechtman, E., Irani, M.: Summarizing visual data using bidirectional similarity. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
15. Sunkavalli, K., Joshi, N., Kang, S.B., Cohen, M.F., Pfister, H.: Video snapshots: Creating high-quality images from video clips. IEEE transactions on visualization and computer graphics 18(11), 1868–1879 (2012)
16. Uruma, K., Konishi, K., Takahashi, T., Furukawa, T.: Image colorization based on the mixed l 0/l 1 norm minimization. In: 2012 19th IEEE International Conference on Image Processing. pp. 2113–2116. IEEE (2012)