Pan *et al. EURASIP Journal on Advances in Signal Processing* (2019) 2019:15
https://doi.org/10.1186/s13634-019-0611-y

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

A bottom-up summarization algorithm for videos in the wild

Gang Pan[1], Yaoxian Zheng[1], Rufei Zhang[2], Zhenjun Han[3], Di Sun[4,1] and Xingming Qu[1,5*]

Abstract

Video summarization aims to provide a compact video representation while preserving the essential activities of the original video. Most existing video summarization approaches relay on identifying important frames and optimizing target energy by a global optimum solution. But global optimum may fail to express continuous action or realistically validate how human beings perceive a story. In this paper, we present a bottom-up approach named clip growing for video summarization, which allows users to customize the quality of the video summaries. The proposed approach firstly uses clustering to oversegment video frames into video clips based on their similarity and proximity. Simultaneously, the importance of frames and clips is evaluated from their corresponding dissimilarity and representativeness. Then, video clips and frames are gradually selected according to their energy rank, until reaching the target length. Experimental results on SumMe dataset show that our algorithm can produce promising results compared to existing algorithms. Several video summarizations results are presented in supplementary material.

Keywords: Video summarization, Clip growing, Bottom-up

1 Introduction

Videos in the wild are abundant in personal collections as well as on the web. The processing demand has been increasing rapidly. A number of related work have been proposed over the past decade [1–3]. Such videos mostly have clutter background and abundant human action. And most of these videos remain unedited and contain a large quantity of redundant information. Therefore, several video processing tasks like video summarization need to be performed, which not only present audiences a compact version that captures most informative parts of the video but also benefit companies highly related to video processing and searching.

According to [4], there are two fundamental types of video summarization: unsupervised methods [5–16] and supervised methods [17–25]. However, these tasks are usually treated as independent. Through experiments, we found that these tasks are actually related. The main idea of these tasks is first to measure the significance of video frames and then select the appropriate video frames according to the different needs of users. The previous summarization methods imply a global optimum with input frames under certain criteria, but the ideal conception seldom leads to satisfactory results. One possible reason is that people watch and understand videos from local perspective rather than from global perspective.

In this paper, we propose a clip-based bottom-up approach for video summarization and experiment with both wild video and non-wild video. A video clip represents a spatiotemporally coherent frame sequence, which is initialled with a constant length and could be extended to a shot. The most related work to ours is the work done by Michael Gygli in [26], where superframes defined with a definition of consecutive frames are aligned with positions of a video that are appropriate for a video cut. Inspired by superframe [26], superpixel [27, 28], and video clip growth [29], our algorithm can be summarized as below: clustering is first used to oversegment video into video clips. Intuitively, frames are not isolated and all frames in a short period of time should have a high degree of similarity. Therefore, it is convenient to work with compact video clips when dealing with video processing task. Then, we perform importance measure to assign energy value to each frame, we name it frame's "energy," which

*Correspondence: quxingming@tju.edu.cn
[1]College of Intelligence and Computing, Tianjin University, Yaguan Road, Tianjin, China
[5]Bobby B. Lyle School of Engineering, Southern Methodist University, Boaz Lane, Dallas 75205, USA
Full list of author information is available at the end of the article

consists of two factors: dissimilarity energy and representativeness energy. Based on frame's energy, video clips' average energy also can be calculated. Finally, video clip growing algorithm is used to select the appropriate video clips by their energy to generate video abstract.

Our main contributions are summarized as follows: (1) proposing a bottom-up algorithm for video summarization, which can gradually generate arbitrary length by gradually adding video clips and frames to the output; (2) presenting an energy function to measure each frame's importance in pixel-level; and (3) our method could be easily extended into other video processing applications.

2 Related work

Video summarization is an important topic that potentially enables faster browsing of large video collections and also more efficient content indexing and access.

Video summarization has been surveyed from multiple perspectives. By analysing whether the analyzed information was sourced directly from the video stream, Money and Agius consider video summarization into three categories: internal type, external type, and hybrid type [30]. According to the different form of frames' temporal continuity, Truong and Venkatesh divide video summarization methods into two categories: key frames and video skims [31]. Panda et al. [4] classify video summarization methods into two categories: unsupervised and supervised methods.

2.1 Unsupervised methods

2.1.1 Clustering

The basic idea of clustering methods is to produce the summary by clustering together similar frames or shots and then showing a limited number of frames per cluster. Based on color feature extraction from video frames and k-means clustering algorithm, Avila et al. [32] present a methodology for the production of static video summaries. Almeida et al. [5] present an approach for video summarization that works in the compressed domain and allows user interaction. The proposed method is based on both exploiting visual features extracted from the video stream and using a simple and fast algorithm to summarize the video content. Guan et al. [6] propose a top-down approach consisting of scene identification and scene summarization. The scene summarization is formulated as choosing those frames that best cover a set of local descriptors with minimal redundancy.

2.1.2 Energy minimization

Some work treats video summarization as a process of energy minimization, which is determined by the context in which frames appear in video. Pritch et al. [8] propose to generate a short video that will be a synopsis of an endless video streams, generated by webcams or surveillance cameras. Feng et al. [9] propose a method that adopts an online content-aware approach in a stepwise manner, hence applicable to endless video, with less computational cost.

2.1.3 Sparse optimizations

The problem of finding the representatives could also be formulated as a sparse optimizations problem. Yang et al. [10] formulate video summarization as a top keyframe selection problem using sparsity consistency, and a global optimization algorithm is introduced to solve the keyframe selection model. Vidal et al. [11] propose a framework to detect and reject outliers from the dataset using the solution of the proposed optimization program. Panda et al. [12] develop a diversity-aware sparse optimization method for multi-video summarization by exploring the complementarity within the videos. Panda and Roy-Chowdhury [13] propose an unsupervised framework for summarizing top related videos by exploring complementarity within videos, and a sparse optimization method is developed to extract a diverse summary that is both interesting and representative in describing the video collection. Vidal et al. [11] consider video summarization of finding a few representatives for a dataset and formulate the problem of finding the representatives as a sparse multiple measurement vector problem. Dornaika and Aldine [14] propose a decremental Sparse Modeling Representative Selection (D-SMRS) in which the selection of the representatives is broken down into several nested processes. Meng et al. [15] propose to summarize a video into a few key objects by selecting representative object proposals generated from video frames. Zhao and Xing [16] propose online video highlighting, a principled way of generating short video summarizing the most important and interesting contents of an unedited and unstructured video, costly both time-wise and financially for manual processing.

2.1.4 Leveraging crawled

Several researchers focus on leveraging crawled web images or videos for video summarization recently. Khosla et al. [33] develop a summarization algorithm that uses the web-image based prior information in an unsupervised manner. Kim et al. [34] develop a parallelizable approach for creating not only high-quality video summaries but also novel structural summaries of online images as storyline graphs. Panda and Roy-Chowdhury [35] develop an approach to extract a summary that simultaneously captures both important particularities arising in the given video and generalities identified from the set of videos. Song et al. [36] present TVSum, an unsupervised video summarization framework that uses title-based image search results to find visually important shots.

2.2 Supervised methods

Departing from unsupervised methods, recent work formulates video summarization as a supervised learning problem. Gygli et al. [17] introduce a method that uses a supervised approach in order to learn the importance of global characteristics of a summary and jointly optimizes for multiple objectives. Gong et al. [18] consider video summarization as a supervised subset selection problem and propose the sequential determinantal point process for diverse sequential subset selection. Sharghi et al. [19] develop a probabilistic model, Sequential and Hierarchical Determinantal Point Process (SH-DPP), for query-focused extractive video summarization. Zhang et al. [20] propose a subset selection technique that leverages supervision in the form of human-created summaries to perform automatic keyframe-based video summarization. Xiong et al. [21] propose a storyline representation that expresses an egocentric video as a set of jointly inferred, through MRF inference, story elements comprising of actors, locations, supporting objects, and events, depicted on a timeline. Ghosh et al. [22] introduce egocentric features to train a regressor that predicts important regions. Yao et al. [23] propose a pairwise deep ranking model that employs deep learning techniques to learn the relationship between highlight and non-highlight video segments. Zhang et al. [24] introduce automatically selecting keyframes or key subshots to summarize videos with a long short-term memory supervised learning technique with. Potapov et al. [25] assign importance scores to each video segment with an SVM classifier, and resulting video assembles the sequence of segments with the highest scores.

However, the above approaches assume the availability of large amount of human-created video-summary pairs or importance annotations, which are in practice difficult to obtain in real applications. Our method is designed to fill the above gaps. The novelty of clip growing method is based on the bottom-up strategy. Different from all of the previous techniques, our method grows summarized videos from short to long; therefore, it can obtain extremely accurate summarized video's length frame by frame. Moreover, it does not use any specific information beyond the video content.

2.3 Quantitative evaluation and benchmark

Evaluating the correctness of a video summarization algorithm is not a straightforward task due to the lack of an objective ground-truth. Ideally, in order to compare different algorithms, each one should be tested on the same datasets and measured using the same metrics. Unfortunately, there is no definite quantitative evaluation and benchmark for previous works until now. But some publicly available datasets of user videos allow for a quantitative evaluation of video summarization algorithms these years. Mundur et al. [7] test algorithms on 50 randomly chosen video segments from the Open Video Project and develop an evaluation procedure with significance factor, overlap factor, and compression factor. Avila et al. [32] demonstrate a validity evaluation by testing algorithms on a sample of videos from the Open Video Project. The summaries' quality is evaluated by the accuracy rate and error rate. Panda et al. [12] introduce Tour20 dataset, which contains 140 videos with multiple manually created summaries. Song et al. [36] introduce TVSum50 dataset, which contains 50 videos with their shotlevel importance scores annotated via crowdsourcing. Kim et al. [34] collect a dataset of 20 outdoor activities, consisting of 2.7M Flickr images and 16K YouTube videos, and evaluate algorithms via crowdsourcing using Amazon Mechanical Turk. Gygli et al. [26] contribute SumMe dataset with human scores for video segments, which allows for an automatic evaluation of different methods. In this paper, we use SumMe as the benchmark for quantitative comparison.

3 Algorithm

Figure 1 presents the overview of our algorithm. As pre-processing, our approach employs dimensionality reduction to generate an Eigenspace of low dimension

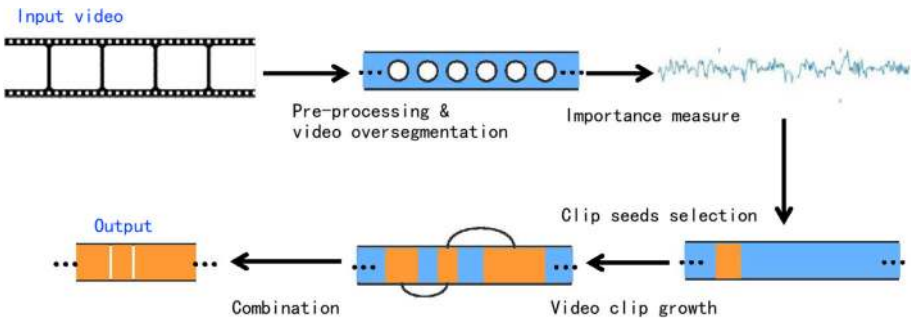

Fig. 1 Flowchart of the proposed algorithm

for each video frame. Firstly, oversegmentation is performed to divide a video into clips. Secondly, each frame's importance, we called Energy, will be measured by two factors: dissimilarity energy and representativeness. Therefore, we can also measure a video clip's energy by summing all the frames in the clip with a weight coefficient. Thirdly, according to each video clip's energy, our algorithm is used to select video clips and frames with higher energy to reach a target length. Finally, video clip merging algorithm is applied to deal with the conflicts in the clip growing process. Three merging cases will be discussed in detail later.

3.1 Pre-processing

In order to reduce the amount of computation, a pre-processing step is presented in this section to convert frames into feature vectors.

3.1.1 Frame representation

Operating directly on frames makes the computational complexity extremely high, which makes the operation hard to handle. Therefore, Singular Value Decomposition (SVD) is performed for dimensionality reduction:

$$F = U\Sigma V^T \tag{1}$$

where F is a frame. In this paper, we use 120×160 grayscale image. U and V^T are the real left and right singular vector matrices respectively, and Σ is the real $\lambda \times \lambda$ diagonal matrix. Next, the first λ left singular vectors will be picked out and reshaped to a column vector x, which can be used as the feature of a frame. In this paper, x is a 720D vector. After pre-processing, a consecutive sequence of frames can be converted to be a sequence of vectors.

3.2 Video oversegmentation

Because frames are not isolated, all frames in a short period of time should have a high degree of similarity. Therefore, it is convenient to work with video clips which are compact, local, and representative. The processing steps are as follows:

3.2.1 Frames distance measure

Measuring the distance between frames is based on their similarity and proximity. The computation is done in $[xt]$ space, where x is the feature of a frame and t is the frame sequence number.

$$ds(i,j) = \|x_i - x_j\| \tag{2}$$

$$dp(i,j) = t_i - t_j \tag{3}$$

where ds, dp is the similarity and proximity between two frame vectors. Since the maximum possible distance between two frame vectors is limited and the temporal distance in the t axis depends on the video length, normalizing need to be done before combining similarity and proximity. Thus, min-max normalization is performed.

After that a variable is used to balance the effect of them. A distance measure D is defined as follows:

$$D = ds + \gamma dp \tag{4}$$

where D is the sum of the similarity distance and temporal distance normalized by a variable γ. The greater the value of γ is, the more significance the temporal proximity counts. In this paper, we use $\gamma = 0.5$. The experimental results show that frames in each video clips have high similarity.

3.2.2 Video oversegmentation

Input a considerable number K, oversegmentation algorithm will divide the video into K video clips. Considering a video with N frames, the length of each video clips L is about N/K.

To begin with, the video is divided into K equal video clips and the center of each video clips is assigned as cluster center. Since frames far from the cluster center frame generally do not belong to this cluster, we can safely assume frames that belong to this cluster center lie within a $2L$ area around the center on the t axis.

Next, for each cluster center C_k, search $2L$ area around the center and assign the nearest frames to this cluster until all the frames are classified. Then, calculate the average frame vector of K video clips to get K new cluster center. Iteratively repeat the above process until the cluster center convergence.

Finally, some margin frames of video clips might be mislabeled after clustering. This is because frames at boundaries are often at similar distances from two adjacent clusters center. Despite just a few frames have been mislabeled, it still affects later operations. Therefore, we re-label these frames by using a voting window with length L_v. Specifically, in the window of length L_v centered at x_t, the label with the highest occurrences is used as the label of x_t.

Besides, it is worth discussing the selection of K, which needs some trade-offs. For example, a small K will cause video clips become too long so that some details may be missed. And a large K will produce too many video clips which makes the computational complexity increase. Therefore, we empirically set $K = \text{Length(video)}/30$, which means the initial length of video clips is about 1 s.

3.3 Importance measure

After oversegmentation, K video clips is generated. The next step is to measure the importance of each video clip. In this section, we introduce each frame's importance, we called Energy, which can be evaluated from both dissimilarity and representativeness. Intuitively, if there is a great

difference between a frame and its neighbor frames, this frame tends to have higher dissimilarity Energy, vice versa. And if there are many similarities between the content before and after a frame, this frame tends to have higher representativeness Energy, vice versa.

Dissimilarity energy can be directly obtained by calculating how many pixels in two frames have changed. If a pixel changes more than a threshold, we call it an active pixel. Active pixels ratio is used as the dissimilarity energy.

$$Ed(t) = \frac{\sum I(F(a,b,t) - F(a,b,t+1) > \sigma)}{a \times b} \tag{5}$$

where $Ed(t)$ is the dissimilarity energy of t^{th} frame. F represents the original frame (a, b represent the pixel location) and I is the indicator function. $I = 1$ when $F(a,b,t) - F(a,b,t+1) > \sigma$; otherwise, $I = 0$. We use $\sigma = 3$ in this paper.

To compute the representativeness energy of a frame, a sliding window with length L_w is created to collect vectors before and after this frame. L_w can be adjusted according to different types of video. Generally, if the content of the video changes drastically, small L_w should be used, vice versa.

$$\bar{x_t} = \frac{1}{L_w - 1} \sum_{j \in (t-l, t+l), j \neq t} x_j \tag{6}$$

where $\bar{x_i}$ is the average vector of frames in sliding window. And $l = L_w/2$. The representativeness energy can be calculate by :

$$Er(t) = \frac{1}{\|x_t - \bar{x_t}\|}. \tag{7}$$

Before combining these two energy, we need to make the two items on the same magnitude.

$$\text{ratio} = \frac{\sum Ed(t)}{\sum Er(t)} \tag{8}$$

$$E(t) = \alpha \times Ed(t) + \beta \times \text{ratio} \times Er(t) \tag{9}$$

where α, β are hyper-parameters controlling the importance of the two parts, respectively.

After importance measure, each frame has its own energy so that we can obtain video clips' energy by simply computing the average energy for each video clip.

3.4 Video clip growing

The energy of a video clip indicates the significance of the video over a period of time. Higher energy video clips and frames tend to be selected. The video clip growing method takes a video clip candidate set $C = \{c_1, c_2 \cdots c_n\}$ from Section 3.2 as input, where c_i represents a video clip with length L_{ci}. The left and right adjacent frames of c_i are called "neighbor." It is worth mentioning that each c_i has its own "neighbor."

The idea of our proposed video clip growing method is to pick higher energy c_i to form a video clip selected set S. Then, by constantly growing each c in S through adding their "neighbor" frames, the total length T_L, of all c in S, can be reached, where T_L is defined by user. To obtain output video S, firstly, we sort all c in C by their average energy in descending order and select the first c as the initial S. Secondly, pick out c from S whose neighbor frames have the highest energy E_n and re-calculate the average energy E_{ave} of this c. If E_n is less than the E_{ave} and the current length C_L of all c in S plus the length of next c from C is less than T_L, add the next c from C into S (Fig. 2 (3)); otherwise, add the highest energy neighbor frame into this c (Fig. 2 (1)). Repeatedly add new frames or add new video clips into S until T_L is reached.

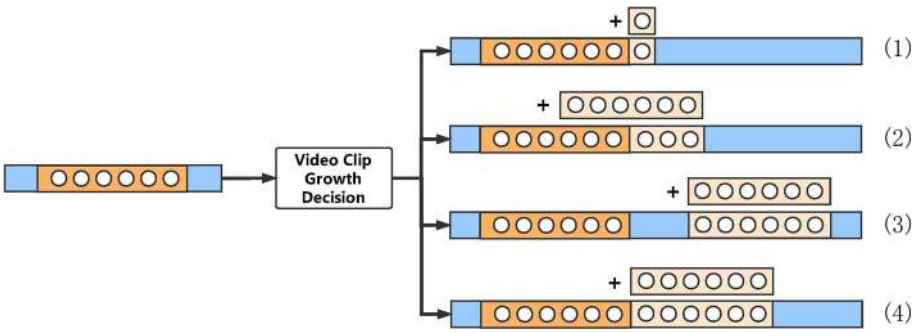

Fig. 2 Video clip growing vitalization and merging cases. From top to bottom: **1** Adding a new frame to the selected video clips. **2** Adding a new clip to the selected clips with overlap, we just preserve the new non-overlapping frames. **3** Adding a new clip without overlap. **4** Adding a new clip which is adjacent with one previous. We merge this two video clips into one

3.4.1 Weight coefficient

Since T_L can be achieved by combining many shorter video clips or several longer video clips, we introduce a weight coefficient $\Theta(c)$ to determine the number and lengths of video clips according to the users' preferences. It is worth mentioning that each clip c has its own $\Theta(c)$.

Since the neighbor frame with the highest E_n has been picked out as indicated in the previous section, now we multiply a Θ to E_n and then redo the comparison between E_n and E_{ave}. Each time we add a new neighbor frame, Θ needs to be updated. For example, if users prefer more shorter video clips, the Θ can be assigned a number less than 1. Every time we add a neighbor frame into S, Θ becomes smaller. Thus, the product of the E_n and its Θ is more likely to be smaller than E_{ave} and then more video clips tend to be selected. Similarly, we can continue increasing Θ when users prefer more longer video clips. In this paper, the weight coefficient is defined as a strictly decreasing function:

$$\Theta(E_n) = e^{-\gamma \text{length}(E_n)} \tag{10}$$

where length(E_n) is the current length of a video clip E_n; γ is a constant, and we set $\gamma = 0.1$ in our experiments.

3.4.2 Merge overlapping video clips

When constantly growing video clips overlap, checking and merging need to be performed. Simply merge a video clip into its neighbor video clips when overlap happened (if a video clip overlaps with two other video clips at the same time, merge it into its left neighbor). Then, check if this new video clip overlaps with other video clips. Continue merging steps until there is no overlap. There are three cases which need to perform merging (Fig. 2).

Case.1 A new video clip overlaps with one pervious video clip (Fig. 2 (2)).

Case.2 Similar to *Case*.1, a new video clip overlaps with two pervious video clips.

Case.3 A new video clip is adjacent to one pervious video clip (Fig. 2 (4)).

The pseudo code of the video clip growing method is shown in Algorithm 1.

Algorithm 1: Video clip growing method

Input: Candidate set C
Output: non-overlapping video clip selected set S
Initialize Target video length T_L;
Sort all c in C by their average energy value in descending order ;
add the first c from C into S ;
while $C_L < T_L$ **do**
  pick out c whose E_n is highest ;
  calculate the average energy E_{ave} of this c. ;
  if $(E_n * \Theta < E_{ave})\&\&(C_L + L_{next\_c} < T_L)$ **then**
    add next c from C into S;
  else
    add neighbor fame with E_n and into its c ;
  update Θ;
  check and merge overlapping c in S;
return S;

4 Experimental results

4.1 Dateset

Experiments have been conducted using a publicly available dataset "SumMe" given in [26] to verify the effectiveness of our algorithm. "SumMe" dataset consists of 25 videos covering holidays, events, and sports. Detailed description of these datasets is available in this page[1]. To provide a quantitative comparison, we use the f-measure and human consistency defined in [26]. We compare our method with several existing methods in [26], including random, uniform, clustering and visual attention [37] baseline.

4.2 Parameters selection

We provide some range of parameter choices. The over-segmentation number K can be in the range [10, 300]. The voting window can be in the range [3, 25]. α and β, the hyper-parameters controlling the importance of Ed and Er, can be adjusted according to different videos. Generally, if the video is intense, α should be greater than β; if the video is relatively calm, the situation is opposite. In our experiments, we set $\alpha = 0.65$, $\beta = 0.35$, $K = 260$, and voting window is 15.

4.3 Results

In order to be consistent with [26], f-measures at 15% summary length is used for our method. As Table 1 shows, in 21 of the total 24 test videos, our method performs best or second best. Our method achieves an average performance of 57% relative to the upper bound, which is 5% higher than the method in [26]. Our results have exceeded mean human on eight test videos if we compare to the human consistency. Furthermore, the results on some other videos are actually very close to mean human.

In order to visualize the results, we draw the energy curve for all the videos and represent the selected video clips with a green rectangle. Besides, "energy curve" from users is defined to make good comparison: In [26], 15 different people were asked to produce a summary with summary length about 15% of the total length. At the beginning, we initialize all frames' energy equal to 0; if one frame is selected by a person, the energy of this

Table 1 Quantitative results

Video name	Dataset [26]		Humans [26]			Computational methods [26, 37]				
	Ran	Max	Worst	Mean	Best	Uniform	Cluster	Att.	Superframe	Ours
Base jumping	0.144	0.398	0.113	0.257	0.396	0.168	0.109	**0.194**	0.121	*0.218*
Bike Polo	0.134	0.503	0.190	0.322	0.436	0.058	0.130	0.076	*0.356*	**0.296**
Scuba	0.138	0.387	0.109	0.217	0.302	0.162	0.135	**0.200**	0.184	*0.261*
Valparaiso Downhill	0.142	0.427	0.148	0.272	0.400	0.154	0.154	0.231	**0.242**	*0.306*
Bearpark climbing	0.147	0.330	0.129	0.208	0.267	0.152	0.158	*0.227*	0.118	**0.218**
Bus in rock tunnel	0.135	0.359	0.126	0.198	0.270	0.124	0.102	0.112	**0.135**	*0.205*
Car railcrossing	0.140	0.515	0.245	0.357	0.454	**0.146**	0.146	0.064	*0.362*	0.132
Cockpit landing	0.136	0.443	0.110	0.279	0.366	0.129	0.156	0.116	**0.172**	*0.298*
Cooking	0.145	0.528	0.273	0.379	0.496	0.171	0.139	0.118	*0.321*	**0.293**
Eiffel Tower	0.130	0.467	0.233	0.312	0.426	0.166	0.179	0.136	*0.295*	**0.205**
Excavators river	0.144	0.411	0.108	0.303	0.397	**0.131**	0.163	0.041	*0.189*	0.123
Jumps	0.149	0.611	0.214	0.483	0.569	0.052	0.298	0.243	*0.427*	**0.309**
Kids playing in leaves	0.139	0.394	0.141	0.289	0.416	*0.209*	0.165	0.084	0.089	**0.182**
Playing on water slide	0.134	0.340	0.139	0.195	0.284	**0.186**	0.141	0.124	*0.200*	0.179
Saving dolphins	0.144	0.313	0.095	0.188	0.242	0.165	*0.214*	0.154	0.145	**0.169**
St Maarten Landing	0.143	0.624	0.365	0.496	0.606	0.092	0.096	**0.419**	0.313	*0.513*
Statue of Liberty	0.122	0.332	0.096	0.184	0.280	0.143	0.125	0.083	*0.192*	**0.153**
Uncut evening flight	0.131	0.506	0.206	0.350	0.421	0.122	0.098	**0.299**	0.271	*0.346*
Paluma jump	0.139	0.662	0.346	0.509	0.642	0.132	0.072	0.028	**0.181**	*0.214*
Playing ball	0.145	0.403	0.190	0.271	0.364	*0.179*	0.176	0.140	**0.174**	0.137
Notre Dame	0.137	0.360	0.179	0.231	0.287	0.124	0.141	0.138	*0.235*	**0.205**
Air Force One	0.144	0.490	0.185	0.332	0.457	0.161	0.143	0.215	**0.318**	*0.407*
Fire domino	0.145	0.514	0.170	0.394	0.517	0.233	*0.349*	0.252	0.130	**0.311**
Car over camera	0.134	0.490	0.214	0.346	0.418	0.099	0.296	0.201	*0.372*	**0.366**
Paintball	0.127	0.550	0.145	0.399	0.503	0.109	0.198	0.281	**0.320**	*0.374*
Mean	0.139	0.454	0.179	0.311	0.409	0.143	0.163	0.167	**0.234**	*0.257*
Relative to max	31%	100%	39%	68%	90%	31%	36%	37%	**52%**	*57%*
Relative to mean	45%	146%	58%	100%	131%	46%	53%	54%	**75%**	*83%*

We show f-measures at 15% summary length for our method, the baselines, and the human selections. We highlight the best (italics) and the second best (bold) computational methods. "Ran" represents random sample. "Uniform" and "Cluster" are computational methods from [26]. "Att." is the visual attention from [37]. "Superframe" is the method from [26]

frame is increased by a constant. Therefore, user- produced "energy curve" could be plotted as shown in Fig. 3. As can be seen, our energy curves are very similar to users' curves. The peaks of high energy are exactly found, and video clips subsequently grow in peak's vicinity.

4.4 Discussion

For static camera videos (Air Force One, Paintball, Car over camera, Fire Domino), our result outperforms all baselines. The reason behind this is actually very obvious: only violent movements of objects which attract user most cause *Ed* to rise and no camera movement causes *Er* to decrease. Thus, relatively accurate *Er* and *Ed* are obtained.

In addition to static camera videos, most of other results have exceeded baseline and perform well on other two video types. Our worst result is made on video "Excavators river crossing," which contains a lot of shots zoom in and zoom out. Besides, the content of the video is repeated. As can be seen in Fig. 3j: users' attention are only concerned on the first time that people cross the river. However, our algorithm objectively considers all crossing the river as equivalent. So, the selected video clips are very uniform.

4.5 The flexibility

Since our algorithm produces video summary by adding frames, it is flexible to produce different lengths of video

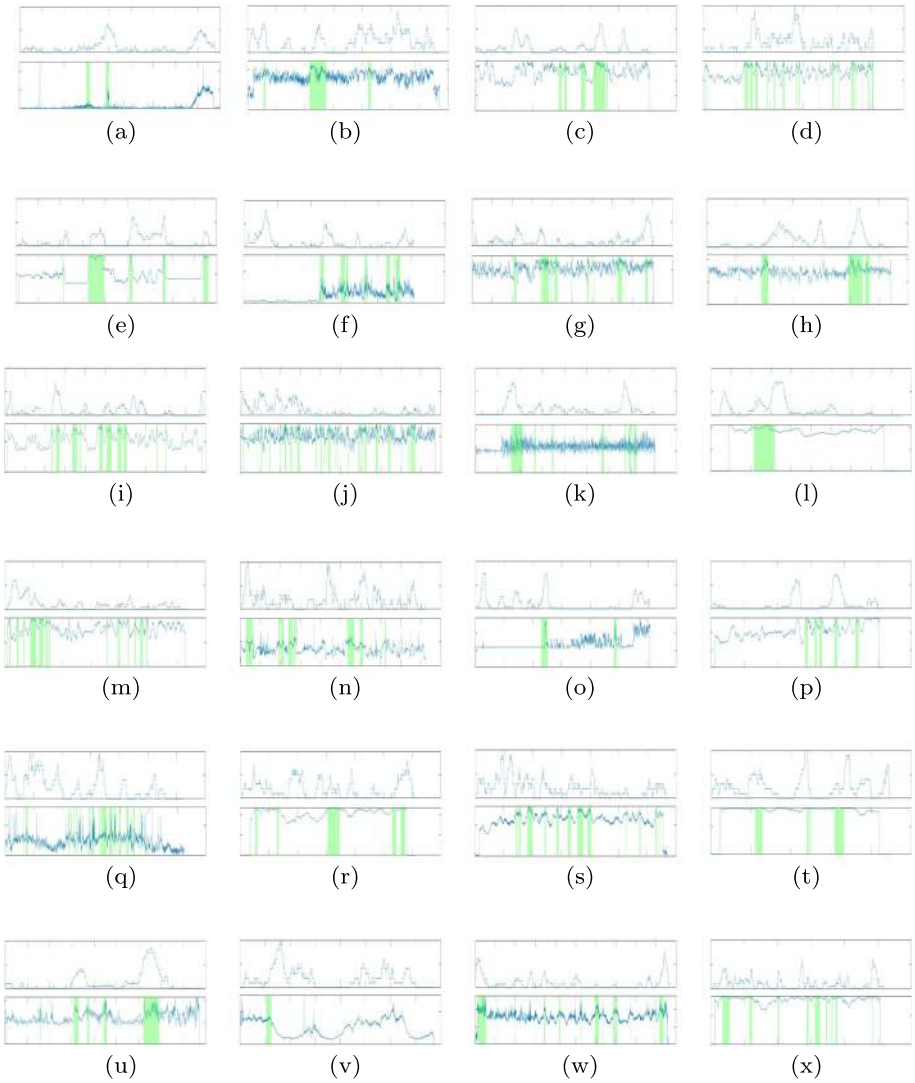

Fig. 3 Results illustration from videos in SumMe dataset with lengths 15%. For each video, the first graph shows Ground Truth (User annotated importance scores); the second graph shows our summarization results, where green bars indicate frames or clips selected by our method. **a** Air Force One. **b** Bearpark climbing. **c** Bike Polo. **d** Bus in rock tunnel. **e** Car over camera. **f** Car railcrossing. **g** Cockpit landing. **h** Cooking. **i** Eiffel Tower. **j** Excavators river crossing. **k** Fire domino. **l** Jumps. **m** Kids playing in leaves. **n** Notre Dame. **o** Paintball. **p** Paluma jump. **q** Playing ball. **r** Playing on water slide. **s** Saving dolphins. **t** Scuba. **u** St Maarten Landing. **v** Statue of Liberty. **w** Uncut evening flight. **x** Valparaiso downhill

summary. Figure 4 shows the results of "Base jumping" from SumMe dataset on different length. High-energy peaks will not be missed, and low-energy frames will not be added.

4.6 Supplementary material

Video summaries of video "Base jumping," "Cooking," and "Valparaiso Downhill" are made and will be shown in Additional files 1, 2, 3, 4, 5, and 6.

Figure 5 shows visual examples of the video summarization by the proposed method on "Cockpit Landing" video from SumMe dataset. It shows that the produced summaries can capture both repeated visual contents that reflect the global commonness and local contents that are representative of the video.

4.7 Other applications

By adjusting parameters to control the length of output video clips or frames, the proposed method could be applied to other video processing applications, e.g., key-frame extraction.

5 Limitation

The limitation of the proposed algorithm lies in the fact that it chooses frames based on their energy. Therefore,

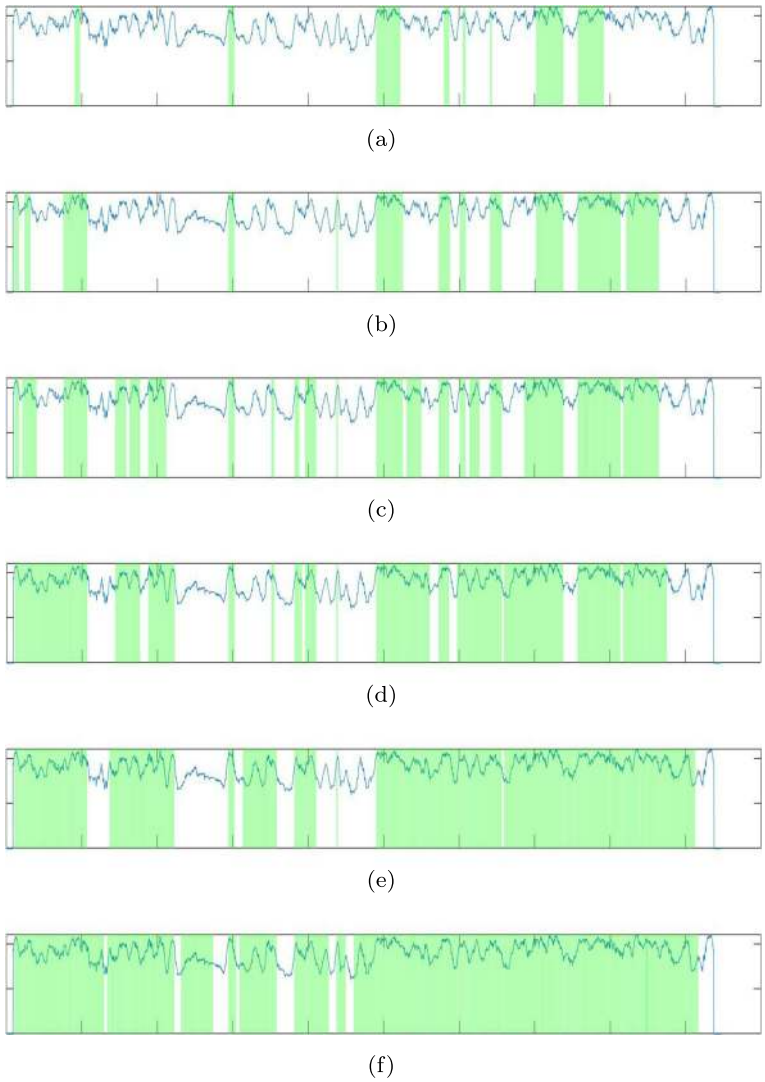

Fig. 4 Results of "Base jumping" from SumMe dataset for different summarization ratios. The selected frames are presented with a green bar at time axis. The energy curves and the selected clips are shown for lengths **a**15%, **b**30%, **c**45%, **d**60%, **e**75%, and **f**90%

whether the computed energy curve is similar to the user's energy curve is very significant. However, this task is quite subjective. The definition of the importance of frames varies in different people. To calculate *Ed*, our algorithm believes that dramatic changes in content lead to higher energy while users might think it is less attractive because dramatic changes lead to chaos. To calculate *Er*, our algorithm believes a frame which is similar to its adjacent frames has higher energy while users might think it is boring because there is not much change before and after this frame. Therefore, parameters of the algorithm need to be adjusted depending on different videos, which requires a lot of experiments.

6 Conclusion and future work

In this paper, we have presented a greedy video clip growing algorithm for video summarization. Clustering is performed to oversegment videos into video clips. Then, we propose the frames' energy which is used as the standard of selecting video clips and adding frames. Clip growing allows users to customize the quality of the video summaries, which is important because different users often vary in needs. By adjusting the parameters, our algorithm can adapt different types of video as well.

Rigorous experiments have been performed on SumMe dataset. Our results show that it is able to create good

Fig. 5 Results for sample video summarization. We select 29 frames with fixed steps from "Cockpit Landing" video, while the automatically selected frames are highlighted by black rectangles

video summarizations in most cases, which are comparable to mean human performance.

In the future, new energy function will be developed to measure the importance of frames. We would like to explore a method to automatically estimate the number of clusters when performing oversegementation. Besides, the pre-processing of frame representation still has a lot of room for improvement.

Endnote

[1] https://people.ee.ethz.ch/~gyglim/vsum/#benchmark

Additional files

Additional file 1: Base jumping (first-person camera) input. (MP4 8947 kb)
Additional file 2: Base jumping (first-person camera) output. (MP4 1363 kb)
Additional file 3: Cooking (indoor) input. (MP4 4833 kb)
Additional file 4: Cooking (indoor) output. (MP4 712 kb)
Additional file 5: Valparaiso Downhill (outdoor) input. (MP4 1451 kb)
Additional file 6: Valparaiso Downhill (outdoor) output. More results could be supplied in an appropriate way. (MP4 9654 kb)

Abbreviations

SVD: Singular value decomposition

Acknowledgements

This work acknowledged the Editor, anonymous Reviewers and Mr. Wang Meng and Mr. Guo Song for hardware and technical support.

Funding

This work was supported by Tianjin Philosophy and Social Science Planning Program under grant TJSR15-008, China National Social Science Foundation under grant 15XMZ057.

Availability of data and materials

Data and source code are available from the corresponding author upon request.

Authors' contributions

GP conceived the method and developed the algorithm. XQ conceived the method, oversaw the project, and wrote the first draft of the manuscript. YZ assembled the formulations and drafted the manuscript. DS analyzed the results and improved the draft. RZ and ZH analyzed the results and edited the manuscript. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

All the data (including individual details, images or videos) in the paper are all from open data sets.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]College of Intelligence and Computing, Tianjin University, Yaguan Road, Tianjin, China. [2]Beijing Institute of Control and Electronics Technology, Muxidi North Street, Beijing 100038, China. [3]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Science, Yuquan Road, Beijing 100049, China. [4]School of Computer Science and Information Engineering, Tianjin University of Science and Technology, Dagu South Road, Tianjin 300222, China. [5]Bobby B. Lyle School of Engineering, Southern Methodist University, Boaz Lane, Dallas 75205, USA.

Received: 26 October 2018 Accepted: 30 January 2019
Published online: 26 February 2019

References

1. J. Liu, J. Luo, M. Shah, in *Computer Vision and Pattern Recognition*. CVPR 2009. IEEE Conference On (IEEE, 2009), pp. 1996–2003. https://ieeexplore.ieee.org/document/4960392
2. H. Kaya, F. Gürpinar, A. A. Salah, Video-based emotion recognition in the wild using deep transfer learning and score fusion. Image Vis. Comput. **65**, 66–75 (2017)
3. Z. Huang, R. Wang, S. Shan, X. Chen, Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning. Pattern Recog. **48**(10), 3113–3124 (2015)
4. R. Panda, A. Das, Z. Wu, J. Ernst, A. K. Roychowdhury, Weakly supervised summarization of web videos. IEEE Int. Conf. Comput. Vis., 3677–3686 (2017)
5. J. Almeida, N. J. Leite, R. D. S. Torres, Vison: video summarization for online applications. Pattern Recogn. Lett. **33**(4), 397–409 (2012)
6. G. Guan, Z. Wang, S. Mei, M. Ott, M. He, D. D. Feng, A top-down approach for video summarization. ACM Trans. Multimed. Comput. Commun. Appl. **11**(1), 4 (2014)
7. P. Mundur, Y. Rao, Y. Yesha, Keyframe-based video summarization using Delaunay clustering. Int. J. Digit. Libr. **6**(2), 219–232 (2006)
8. Y. Pritch, A. Rav-Acha, A. Gutman, S. Peleg, Webcam synopsis: peeking around the world. IEEE Int. Conf. Comput. Vis., 1–8 (2007)
9. S. Feng, Z. Lei, D. Yi, S. Z. Li, Online content-aware video condensation. Comput. Vis. Pattern Recognit., 2082–2087 (2012)
10. C. Yang, J. Yuan, J. Luo, Towards scalable summarization of consumer videos via sparse dictionary selection. IEEE Trans. Multimedia. **14**(1), 66–75 (2012)
11. R. Vidal, G. Sapiro, E. Elhamifar, See all by looking at a few: sparse modeling for finding representative objects. IEEE Conf. Comput. Vis. Pattern Recognit., 1600–1607 (2012)
12. R. Panda, N. C. Mithun, A. Roychowdhury, Diversity-aware multi-video summarization. IEEE Trans. Image Process. Publ. IEEE Signal Proc. Soc. **26**(10), 4712–4724 (2017)
13. R. Panda, A. K. Roy-Chowdhury, Sparse modeling for topic-oriented video summarization. IEEE Int. Conf. Acoust. Speech Signal Process., 1388–1392 (2017)
14. F. Dornaika, I. K. Aldine, Decremental sparse modeling representative selection for prototype selection. Pattern Recog. **48**(11), 3714–3727 (2015)
15. J. Meng, H. Wang, J. Yuan, Y. P. Tan, From keyframes to key objects: video summarization by representative object proposal selection. Comput. Vis. Pattern Recognit., 1039–1048 (2016)
16. B. Zhao, E. P. Xing, Quasi real-time summarization for consumer videos. Comput. Vis. Pattern Recognit., 2513–2520 (2014)
17. M. Gygli, H. Grabner, L. V. Gool, Video summarization by learning submodular mixtures of objectives. Comput. Vis. Pattern Recognit., 3090–3098 (2015)
18. B. Gong, W. L. Chao, K. Grauman, F. Sha, Diverse sequential subset selection for supervised video summarization. Int. Conf. Neural Inf. Process. Syst., 2069–2077 (2014)
19. A. Sharghi, B. Gong, M. Shah, Query-focused extractive video summarization, 3–19 (2016). https://link.springer.com/chapter/10.1007/978-3-319-46484-8_1
20. K. Zhang, W. L. Chao, F. Sha, K. Grauman, Summary transfer: exemplar-based subset selection for video summarization. Comput. Vis. Pattern Recognit., 1059–1067 (2016)
21. B. Xiong, G. Kim, L. Sigal, Storyline representation of egocentric videos with an applications to story-based search. IEEE Int. Conf. Comput. Vis., 4525–4533 (2015)
22. J. Ghosh, J. L. Yong, K. Grauman, *Discovering important people and objects for egocentric video summarization, Computer Vision and Pattern Recognition*, (2012), pp. 1346–1353
23. T. Yao, T. Mei, Y. Rui, *Highlight detection with pairwise deep ranking for first-person video summarization, Computer Vision and Pattern Recognition*, (2016), pp. 982–990
24. K. Zhang, W. Chao, F. Sha, K. Grauman, Video summarization with long short-term memory, 766–782 (2016). https://link.springer.com/chapter/10.1007%2F978-3-319-46478-7_47
25. D. Potapov, M. Douze, Z. Harchaoui, C. Schmid, Category-specific video summarization. *European Conference on Computer Vision*, 540–555 (2014)
26. M. Gygli, H. Grabner, H. Riemenschneider, L. V. Gool, *Creating summaries from user videos, European Conference on Computer Vision*, (2014), pp. 505–520
27. X. Ren, J. Malik, Learning a classification model for segmentation. Proc. Int. Conf. Comput. Vis. **1**, 10–171 (2003)
28. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern. Anal. Mach. Intell. **34**(11), 2274 (2012)
29. G. Pan, X. Qu, L. Lv, S. Guo, D. Sun, in *19th Pacific-Rim Conference on Multimedia*. Video clip growth: a general algorithm for multi-view video summarization, (2018), pp. 112–122
30. A.G. Money, H. Agius, Video summarisation: a conceptual framework and survey of the state of the art. J. Vis. Commun. Image Represent. **19**(2), 121–143 (2008)
31. B.T. Truong, S. Venkatesh, Video abstraction: a systematic review and classification. ACM Trans. Multimed. Comput. Commun. Appl. **3**(1), 3 (2007)
32. S.E.F.D. Avila, VSUMM: a mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recogn. Lett. **32**(1), 56–68 (2011)
33. A. Khosla, R. Hamid, C. J. Lin, N. Sundaresan, Large-scale video summarization using web-image priors. Comput. Vis. Pattern Recognit., 2698–2705 (2013)
34. G. Kim, L. Sigal, E. P. Xing, Joint summarization of large-scale collections of web images and videos for storyline reconstruction. IEEE Conf. Comput. Vis. Pattern Recognit., 4225–4232 (2014)
35. R. Panda, A. K. Roy-Chowdhury, Collaborative summarization of topic-related videos. IEEE Conf. Comput. Vis. Pattern Recognit., 4274–4283 (2017)
36. Y. Song, J. Vallmitjana, A. Stent, A. Jaimes, TVSum: Summarizing web videos using titles. Comput. Vis. Pattern Recognit., 5179–5187 (2015)
37. N. Ejaz, I. Mehmood, S. W. Baik, Efficient visual attention based framework for extracting key frames from videos. Signal Proc. Image Commun. **28**(1), 34–44 (2013)