Multimed Tools Appl
DOI 10.1007/s11042-016-3569-x

A novel clustering method for static video summarization

Jiaxin Wu[1] · Sheng-hua Zhong[1] · Jianmin Jiang[1] ·
Yunyun Yang[2]

Received: 12 October 2015 / Revised: 17 March 2016 / Accepted: 28 April 2016

Abstract Static video summarization is recognized as an effective way for users to quickly browse and comprehend large numbers of videos. In this paper, we formulate static video summarization as a clustering problem. Inspired by the idea from high density peaks search clustering algorithm, we propose an effective clustering algorithm by integrating important properties of video to gather similar frames into clusters. Finally, all clusters' center will be collected as static video summarization. Compared with existing clustering-based video summarization approaches, our work can detect frames which are highly relevant and generate representative clusters automatically. We evaluate our proposed work by comparing it with several state-of-the-art clustering-based video summarization methods and some classical clustering algorithms. The experimental results evidence that our proposed method has better performance and efficiency.

Keywords Static video summarization · Clustering method · Video representation

1 Introduction

Video summarization, also called as video abstract, is a brief version of the video content. It is usually created by extracting essential and representative information of a video into storyboard or video clip. As a result of the rapid development of the Internet, uploading videos becomes so convenient that a huge quantity of new videos is available online every second. According to the YouTube statistics in 2015 [26], over 300 h of video are published on it every minute. In other word, more than 400 thousand hours of new video

✉ Sheng-hua Zhong
csshzhong@szu.edu.cn

[1] College of Computer Science & Software Engineering, Shenzhen University, Shenzhen, Guang Dong, People's Republic of China

[2] School of Natural Sciences and Humanities, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guang Dong, People's Republic of China

Published online: 19 May 2016

are generated in a day. Facing such huge scale of videos, a good video summarization is useful for web users to browse video quickly and decide whether to watch the whole video or not.

Video summarization has been deeply explored since the 1990s. Generally, the approaches for video summarization can be classified into two categories, static and dynamic video summarization [25]. Static video summarization (usually shown as storyboard) consists of keyframes which mostly represent video content. It takes visual information into account but ignores audio message. Dynamic video summarization is a video clip which combines image, audio and text information together. Compared with dynamic video summarization, static video summarization is easier to be browsed and is helpful to reduce computational complexity for video retrieval and analysis [3, 5]. In this paper, we propose a novel static video summarization method.

Actually, static video summarization is trying to reduce the redundancy of video and select some representative frames to summarize the video content. Thus, formulating static video summarization task as a clustering problem sounds reliable. As a result, many approaches, which are based on clustering algorithm, have been proposed. For example, Zhuang et al. used *k*-means clustering algorithm to separate all the frames of each shot into clusters and then collected each shot clusters center to be static video summarization result [29]. Mundur et al. used a clustering algorithm based on Delaunay Triangulation to divide all keyframes into clusters after selecting these keyframes from input videos [20]. However, existing clustering-based approaches usually used predefined clustering algorithms. These kinds of methods require preset cluster number before clustering. In a more appropriate way, the selected clustering method should be adaptive and has the ability to decide the number of clusters depending on different videos.

In this paper, we formulate static video summarization task as a clustering problem and develop a novel clustering algorithm. Based on the insights from High density peaks search (HDPS) clustering algorithm [23], we propose a video representation based high density peaks search (VRHDPS) clustering algorithm by integrating some important properties of video. Furthermore, VRHDPS is of low computational complexity and doesn't need any iteration to find the cluster center.

The rest of the paper is organized as follows. Section 2 presents related work of static video summarization. Our proposed method is explained in detail in Section 3. In Section 4, we compare the performance of our proposed clustering method with several classical methods. Meanwhile, the comparisons with other clustering-based static video summarization approaches are presented. Finally, Section 5 outlines the conclusion with future work.

2 Related work

Recently, the availability of videos has been growing at a rapid rate. Video has drawn more and more attention. Meanwhile, with the advances of imaging techniques, it has been never easier to access a large amount of video content. As a result, a large number of video applications have shown up, such as: fake views detection in video services [7], video sharing and searching [13], video streaming [8], video caching [27], video quality assessment [6], and video transfer [28], and video object tracking [22]. Video summarization is one of useful video applications to provide users a better video service.

Static video summarization aims to reduce the redundancy of video and selects a collection of representative frames. It has been extensively studied because of its important role in many video-related applications such as video retrieval and analysis. Actually, it is quite similar with clustering algorithms, which gather similar elements together and regard the cluster center as a representative of all elements within the cluster. In fact, a number of clustering-based static video summarization methods have been proposed in the literature.

Zhuang et al. firstly published their clustering-based method in 1998 [29]. They split the video into shots and then *k*-means clustering algorithm was used to group frames within each shot into clusters based on color histogram feature. Finally, cluster centers of each shot were regarded to be static video summarization result. Besides, the number of clusters needed to be preset before clustering. In the same year, Hanjalic et al. described a similar method by splitting video sequence into numbers of clusters, and finding the optimal clustering by cluster-validity analysis [12]. They used an objective model to select a keyframe from each cluster and then generated video summarization result. In 2001, Gong et al. proposed a video summarization method which was able to produce a motion video summary that minimized the visual content redundancy for the input video. The original video sequence was divided into a shot cluster set where any pairs of cluster must have visual variation and all shots belonging to the same cluster must be visually similar to create video summary [10]. In 2006, Mundur et al. proposed a video summarization technique by using Delaunay clusters that generated good quality summaries with fewer frames and less redundancy when compared to other schemes [20]. In 2007, Chang et al. presented the video summarization method with three steps [4]. Firstly, shot boundary detection was executed based on color and edge information. Then the clustering process classified shots according to their similarity of motion type and scene. Finally, the important shots of each cluster were selected in the skimming process by adopting shot-important filter, which determined the importance of each shot by computing the motion energy and color variation. In 2011, Avila et al. improved the performance by eliminating some meaningless frames firstly to get candidate frames [7]. Then all the candidate frames were divided into clusters by using *k*-means. Finally, they filtered some similar frames and the rest were considered as the final video summarization result. Meanwhile, the number of clusters was decided by variation in visual content among adjacent frames.

In general, those shot-based methods may remain redundant because similar shots may appear several times in video. Also, setting the number of clusters in advance may influence optimal video summarization result generation. In a more reasonable way, the number of clusters should be decided during the clustering procedure depending on different video contents.

3 Clustering-based static video summarization

In this section, we introduce our work in detail. The overview of our proposed method is shown in Fig. 1. Here, the input is a number of video frames, and the output is a storyboard composed of representative frames. The proposed method includes four steps: pre-sampling, video frame representation, clustering, and video summarization result generation. Each step is explained as follows.

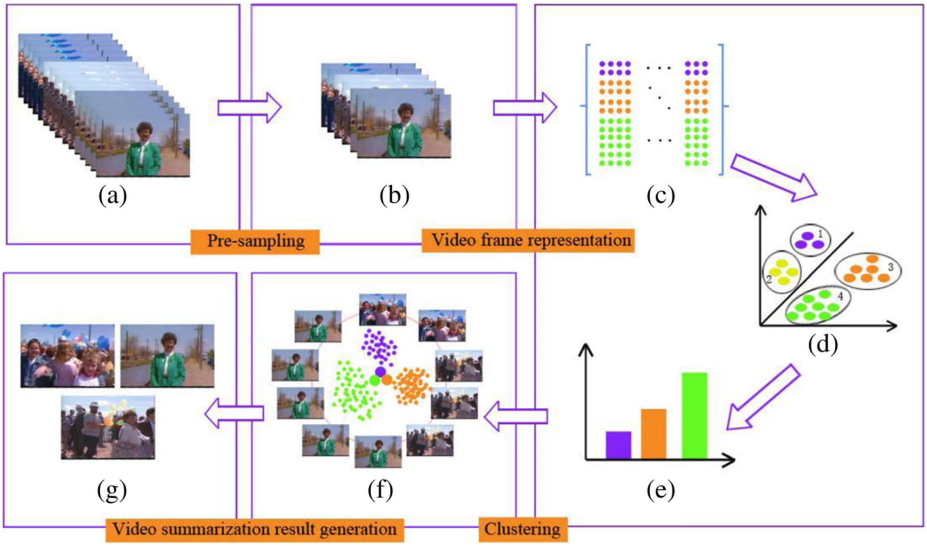

(a)
(b)
(c)
(d)
(e)
(f)
(g)
Pre-sampling
Video frame representation
Video summarization result generation
Clustering
1
2
3
4

Fig. 1 Overview of our proposed method. It includes four steps: pre-sampling, video frame representation, clustering, and video summarization result generation

3.1 Pre-sampling

Video is a media with considerable redundancy and usually dozens of frames represent the same visual information. Thus, many video-related techniques have applied pre-sampling before processing to reduce the number of frames, which can greatly reduce computational complexity and save time. In our approach, video pre-sampling is also performed to get candidate frames $S=[S_1, S_2,\ldots, S_M]$, where M is the total number of candidate frames. In this stage, keyframe extraction is performed for all input frames firstly and then useless frames are removed for further sampling.

3.1.1 Keyframe extraction

A keyframe extraction method based on singular value decomposition (SVD) [1] is utilized in our proposed approach. $\mathbf{H}^t$ presents Hue-Saturation-Value (HSV) color space of the current frame at time t. Three histograms, $\mathbf{h}_H$, $\mathbf{h}_S$ and $\mathbf{h}_V$, of lengths: l_H, l_S and l_V, respectively, are built for the three color channels of $\mathbf{H}^t$. We define a time-varying feature vector $\mathbf{x}^t$ as:

$$\mathbf{x}^t=[\mathbf{h}_H\mathbf{h}_S\mathbf{h}_V] \tag{1}$$

Then, the length of vector $\mathbf{x}^t$ $L=l_H+l_S+l_V$. We establish a $N\times L$ matrix for each frame at time $t>N$ as described below:

$$\mathbf{X}^t=\begin{pmatrix} \mathbf{x}^t \\ \mathbf{x}^{t-1} \\ \vdots \\ \mathbf{x}^{t-N+1} \end{pmatrix} \tag{2}$$

where $t = N, N+1, \ldots, T\text{-}1, T$, N is the window size and T is the number of input frames. $\mathbf{X}^t$ describes the feature matrix during current frame at time t and previous $N\text{-}1$ frames. After that, SVD computation is executed for the matrix $\mathbf{X}^t$ as shown in Eq. (3):

$$\mathbf{X}^t = \mathbf{U}\Sigma\mathbf{V}^T \tag{3}$$

where $\mathbf{U}$ is a matrix of a set of output orthogonal singular vectors, $\mathbf{V}^T$ is a matrix of a set of input orthogonal singular vectors and $\mathbf{\Sigma}$ is a matrix of the singular values which diagonal elements are placed in descending order of significance. For example, if $q_1, q_2, \ldots, q_N$ are diagonal elements of $\mathbf{\Sigma}$, then $q_1$is the biggest singular value. r^t is defined as the rank of $\mathbf{X}^t$ and it can be calculated by the number of singular values which exceed a user-defined threshold τ. If the rank of $\mathbf{X}^t$ is bigger than the rank of its previous matrix $\mathbf{X}^{t-1}$, it means that the visual content of the current frame is visually different from its previous. As a result, the current frame will be selected as a keyframe.

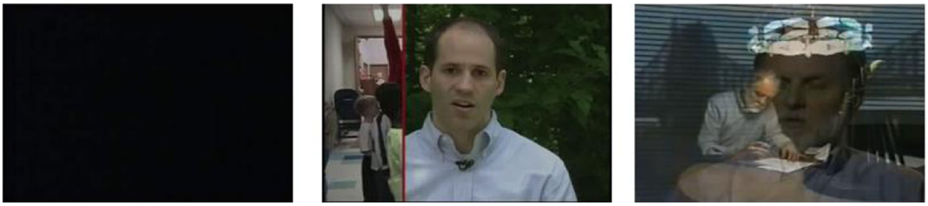
3.1.2 Useless frames removing

Video usually has some useless frames. Figure 2 shows a sample of useless frames, where (a) is a black frame, (b) and (c) are shot boundaries. Actually, (b) is abrupt transition and (c) is gradual transition. As the video summarization aiming to capture the essence of video content, we define black frames and shot boundaries as useless frames in this paper. It can be observed that the content of these useless frames are not supposed to be video summarization result. Therefore, we will remove them in the pre-sampling stage.

3.2 Video frame representation

After pre-sampling, a number of candidate frames $S = [S_1, S_2, \ldots, S_M]$ are selected. Then bag of word (BoW) model [24] is applied to represent each candidate frame. Basically, BoW modeling has three steps: feature extraction, codebook generation, and histograms representation. As local feature has been proved to be more representative recently, we apply one of classical local features, Scale Invariant Feature Transform (SIFT) [15] as the descriptor. We provide an overview of generating video frame representation in Algorithm 1. The algorithm is fed with a number of candidate frames and outputs the representation for each candidate frame.

Algorithm 1 describes the procedure of video frame representation. Firstly, we extract SIFT features on every candidate frame and then we have a large number of SIFT descriptors. Secondly, numbers of representative features are selected among them as the codebook by using k-means algorithm. After that, we can generate a histogram for each candidate frame



(a) Black frame (b) Abrupt transition (c) Gradual transition

Fig. 2 Some useless frames examples. (**a**) is black frame, (**b**) is abrupt transition, and (**c**) is gradual transition

according to their feature distribution of the codebook. Finally, every video frame is denoted as a histogram.

```
Algorithm 1: Generating Video Frame Representation
Input: $S = [S_1, S_2, ..., S_M]$: $M$ candidate frames of the target video, $L$: codebook size.
Output: $\mathbf{z}^q = [z_1, z_2, ..., z_L]$: the representation of the $q^{th}$ candidate frame.
1. Extract interesting points from each candidate frame $S_i$ and collect all the local descriptors to construct a matrix $\mathbf{F}=\{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N\}$.
2. Generate a codebook $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_L]$ by performing $k$-means clustering algorithm over $\mathbf{F}$.
3. Obtain each candidate frame representation $\mathbf{z}^q$ according to its local features distribution in the codebook:
    - For each candidate frame $S_i$, compute the similarity between its all extracted descriptors and each codeword $\mathbf{c}_i$.
    - For each codeword $\mathbf{c}_i$, calculate the total number of descriptors which similar with it as $z_i$.
```

3.3 Clustering

With all candidate frames being represented by BoW model, the next step of our proposed method is separating all candidate frames into clusters. In this paper, we propose VRHDPS based on HDPS [23], in which the number of clusters doesn't need to be specified, as the clustering method. HDPS was relied on the idea that cluster centers were characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. However, some characteristics of video summarization have not been considered in HDPS. Therefore, we propose VRHDPS clustering algorithm, which is more capable to dealing with video summarization task, for video summarization task.

In the clustering procedure, VRHDPS needs to calculate two quantities for each point: its local density and its distance from points with higher density. The local density is defined as

$$\rho_i = \sum_j \chi\left(d_{ij} - d_c\right) \tag{4}$$

Where

$$\chi(x) = \begin{cases} 1, x < 0 \\ 0, x \geq 0 \end{cases} \tag{5}$$

ρ_i denotes the density of the i^{th} point, d_{ij} is the distance between point i and point j, and d_c is a cutoff distance.

Basically, ρ_i is the number of points which are close than d_c to point i. δ_{i} is measured by computing the minimum distance between point i and any other points with higher density:

$$\delta_i = \min_{j:\rho_j > \rho_i} \left(d_{ij}\right) \tag{6}$$

For the point with highest density, we generally identify

$$\delta_i = \max_{j:\rho_j > \rho_i} \left(d_{ij}\right) \tag{7}$$

Finally, cluster centers should be those points with high density and anomalously large δ_{i}. After all cluster centers have been recognized, the rest of the points are assigned to the same cluster of their nearest neighbor with high density. As the assignments for all points are completed, clustering procedure is finished. The proposed clustering method VRHDPS only needs to calculate two quantities based on distance, therefore it has low computational complex. Furthermore, this clustering algorithm is stable. Once the input distances of data points are confirmed, the clustering results will not change anymore. In the experimental part, the results evidence these advantages clearly.

When we use the VRHDPS clustering algorithm to cluster candidate frames, it can be detailedly separated into three steps. These three steps are described as follows:

Step 1 Calculate Euclidean distances between each two candidate frames.
Step 2 Compute local density according to Eq. (4) for each candidate frame.
Step 3 Obtain minimum distance according to Eq. (6) for each candidate frame.

3.4 Video summarization result generation

In the above steps, we use BoW model to represent the video frame and utilize the clustering algorithm to select essential frames and remove redundancy. In VRHDPS, we have suggested a new strategy to generate video summarization result which makes the clustering algorithm more capable of capturing the essence of the video content.

Actually, some features of video summarization are important. For example, in video summarization, we should pay attention to the isolated points because they represent unique frames. In the HDPS clustering approach [23], Alex et al. advised that cluster center can be chosen depending on $\gamma = \rho * \delta$. Ploting γ sorted in decreasing order, cluster centers are those points higher than the point when the graph starts growing anomalously. However, $\gamma = \rho * \delta$ considers that two parameters are equally important. In an extreme situation, points with a high density and a small minimum distance will have the same γ as points with a low density and a big minimum. Actually, in video summarization, points with a lower density and a relatively large distance are more important, which are more supposed to be video summarization result. We have suggested a new strategy $\gamma = \alpha * (\rho * \delta) + (1 - \alpha) * \delta$ for generating the cluster centers which let the frames with fewer similar but a relatively large distance tend to be regarded as video summarization result. Also, α ranges from 0 to 0.5. Actually, we have compared the performances of VRHDPS with HDPS methods when dealing with video summarization task (shown in Section 4.2) and the result shows our proposed VRHDPS is more capable of dealing with video summarization task than HDPS generally.

4 Experimental results and discussion

Several comparative experimental results are presented to validate the effectiveness of our proposed method in this section. Experimental setting is introduced firstly, and then the performance comparison with other clustering algorithms is provided. After that, the comparison results with several video summarization methods are illustrated in section 4.3. Moreover, we demonstrate the proposed clustering method VRHDPS by comparing it with original HDPS. In the end, we have a discussion about the advantages and disadvantages of the proposed method.

4.1 Experimental setting

4.1.1 Database

We perform our experiments on two benchmark databases. The first one is VSUMM (VSUMM: A simple and efficient approach for automatic video summarization) database [2]. It is composed of 50 videos selected from open video project (OVP) [21], which are distributed among several genres (e.g., documentary, educational, ephemeral, historical, lecture). All videos are in MPEG-1 format (30 fps, 352×240 pixels). The duration of these videos varies from 1 to 4 min and approximately 75 min of video in total. VSUMM is a benchmark database which has been utilized by many video summarization methods such as [2, 19, 20]. The second database is collected from video web sites such as YouTube. In this paper, we call it VYT database. It is provided by [2], which contains 50 videos covering several genres (cartoons, news, sports, commercials, TV-shows and home videos). Their durations vary from 1 to 10 min. This database has been used in [2, 11, 19]. In these two databases, static ground truth summaries of each video were labeled by the publishers. Figure 3 shows a sample video clip of VSUMM database.

4.1.2 Evaluation

In this paper, we apply precision, recall and F-score as the evaluation metrics.

$$Precision = \frac{n_{match}}{n_{AS}} \tag{8}$$

where n_{match} is the number of correct matches between the ground truth and automatic video summarization result generated by different methods. n_{AS} is the total number of automatic video summarization result.

$$Recall = \frac{n_{match}}{n_{GT}} \tag{9}$$

where n_{GT} is the total number of the ground truth. Also, we employ F-score to aggregate the precision and recall. This evaluation metrics are also used in [16, 19] to demonstrate their work.

$$F - score = \frac{2 \times Precision \times Recall}{Precision} \tag{10}$$

Fig. 3 A sample video clip of VSUMM database

4.1.3 Parameter setting

In our experiments, we set the size of codebook to be 1024, which is the classical setting in [16]. In addition, we use 0.5 as the threshold which is recommended by Lowe in [15] to judge if two SIFT descriptors are similar.

4.2 Comparison with several clustering algorithms

In order to evaluate the effectiveness and efficiency of our proposed clustering method, we compare it with several clustering algorithms such as *k*-means [18], spectral clustering (SC) [17], affinity propagation (AP) [9]. In the comparison, we replace the proposed clustering algorithm with these three clustering methods to generate video summarization results. We repeat the experiment five times and take the mean of them as reported results. Performances of video summarization methods based on *k*-means (KVS), SC (SCVS), AP (APVS) and our proposed clustering approach video representation clustering based video summarization (VRCVS) on the VSUMM database are shown in Table 1 and efficiency results are illustrated in Table 2. The extracted examples are displayed in the Fig. 5. The frames marked with green borders are the correct matches between the summary and the ground truth. Furthermore, we also compare our proposed method with *k*-means on VYT database. Figure 4 illustrates the comparison result of KVS and our proposed method VRCVS.

From Table 1, it is obviously that our proposed clustering method is more effective in capturing the representative frames than others. Our work VRCVS has a higher recall than KVS and SCVS, which means that the proposed method is capable of generating more representative frames as video summarization result. In Fig. 5, it illustrates that our proposed method can capture more correct matches than KVS with low redundancy. Although SCVS has the same correct matches as ours in this sample, its unmatched frame (the second frame) is similar with the first frame in its video summarization result. In contrast, our unmatched frame is visually different from the other three and more likely to match the third frame of the ground truth. Thus, our work VRCVS tends to have higher recall than SCVS. Also, our proposed method with a precision of 0.68 clearly outperforms than KVS with 0.58, SCVS with 0.43 and APVS with a precision of 0.41, which declares that our work can capture representative frames more accurately. APVS has the highest recall, however it has low precision because of selecting many frames as video summarization result. Having more frames in summary is likely to have more correct matches to reach higher recall but it sometimes will sacrifice the precision, too. As shown in Fig. 5,

Table 1 Performance comparison of several clustering algorithms

	Recall	Precision	F-score	STD
KVS	0.56	0.58	0.54	0.72
SCVS	0.42	0.43	0.41	0.6
APVS	**0.84**	0.41	0.51	**0**
VRCVS	0.63	**0.68**	**0.63**	**0**

Bold entries show the best results of each evaluation method

Table 2 Efficiency comparison of several clustering algorithms

	Running time(s)	STD
KVS	0.169	0.014
SCVS	0.509	0.086
APVS	0.375	0.028
VRCVS	**0.014**	**0.003**

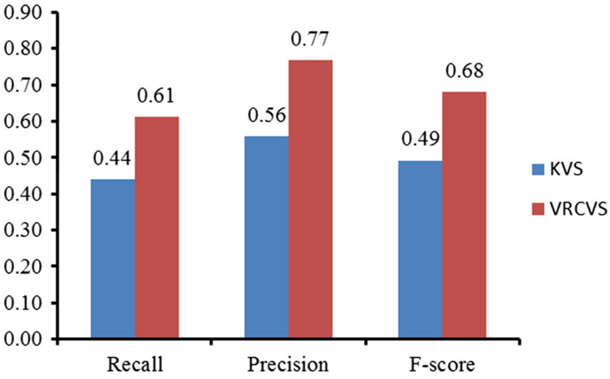
Bold entries show the best results of each evaluation method

APVS captures the most representative frames but it has much more frames than others, which decreases its precision. In general, our proposed clustering method outperforms the compared methods with a highest F-score of 0.63, and it demonstrates that our work is more capable to deal with video summarization problem. Furthermore, our proposed method and APVS are more stable with a standard deviation of zero while KVS with 0.72 and SCVS with 0.60. In other word, the summaries of a video generated by VRCVS or APVS vary every time. However, the summaries of a video captured by KVS or SCVS are steadily. Furthermore, our proposed method is more efficient than other with average runtime of 0.014 s while KVS with 0.169 s, SCVS with 0.509 s and APVS with 0.375 s. The results show that our work is more efficient than other in the clustering procedure. For the VYT database, it also can be observed from Fig. 4 that our proposed method VRCVS is able to achieve superior performance in capturing the essence of the video content.

4.3 Comparison with a number of clustering-based static video summarization methods

To further demonstrate the efficiency of our proposed (VRCVS) method, we compare our work with a number of video summarization approaches. We firstly compare our proposed method with several clustering-based static video summarization methods on VSUMM database. The comparative systems are open video project (OVP) storyboard [21], Delaunay Triangulation (DT) [20] and VSUMM [2]. Four summarization results are shown in Figs. 7 and 6 illustrates the performances of the comparison results. Besides, we also compare our work with several state-of-the-art methods on VYT database: VSUMM proposed in [2] and

Fig. 4 Evaluation in terms of Precision, Recall and F-score of several clustering algorithms on VYT database



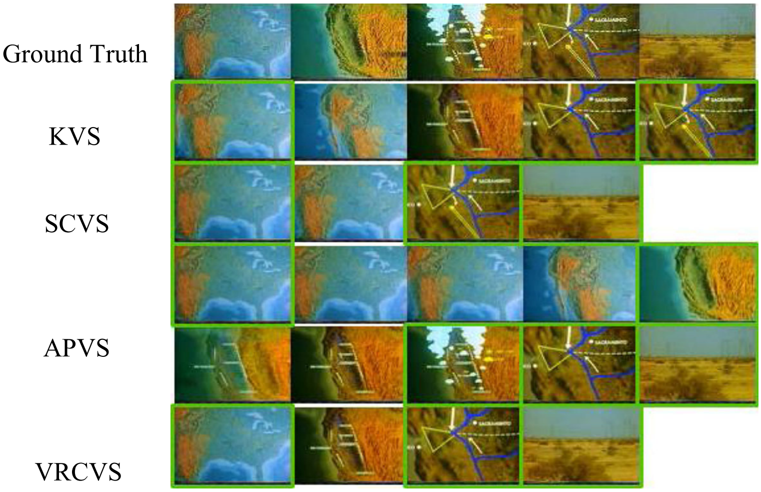


Fig. 5 Comparison results of several clustering methods, where the first row is the ground truth, the second is the summary obtained by KVS, the third is the summary captured by SCVS, the fourth and fifth row are the summary obtained by APVS and the sixth row is generated by our proposed method VRCVS. The frames marked with green borders are the correct matches between the summary and the ground truth

Keypoint base keyframe selection (KFVS) proposed in [11]. Table 3 shows the comparison results and the extracted examples are displayed in Fig. 8.

As shown is Fig. 6, although the proposed method doesn't have the highest value in recall, our precision is much higher than others. It means that our method VRCVS is more likely to generate video summarization result with much higher accuracy. In Fig. 7, all extracted frames in our video summarization result are visually different. Actually, we have six correct matches, one unmatched frame and one likely matched frame with a precision of 0.75. Besides, on the VSUMM database our work can capture essential frames more accurately with a precision of 0.73, while OVP is 0.6,

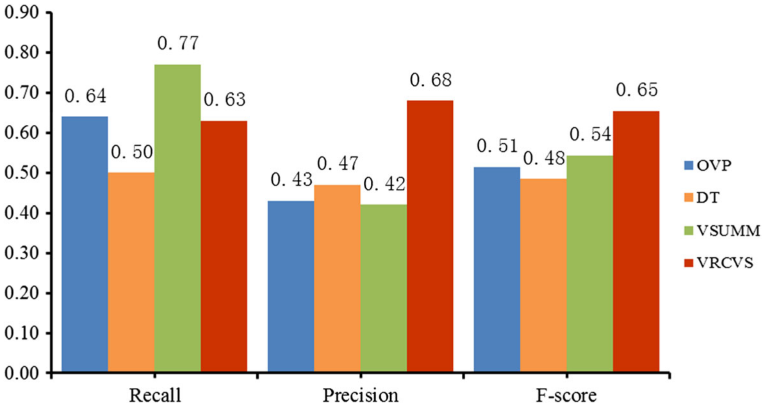


Fig. 6 Evaluation in terms of Precision, Recall and F-score of several clustering-based video summarization methods

Table 3 Performance comparison of several clustering-based video summarization methods on VYT database

	Recall	Precision	F-score
VSUMM	0.44	0.54	0.48
KFVS	0.37	0.60	0.37
VRCVS	**0.61**	**0.77**	**0.68**

Bold entries show the best results of each evaluation method

DT is 0.5 and VSUMM is 0.6. OVP can get rid of redundancy but it tends to have much more uncorrected matches than VRCVS. As the result shown in Fig. 7, almost half of them are unmatched frames. DT has selected fewer frames than others to reduce redundancy but it also has fewer correct matches. As the summary shown in Fig. 6, DT has fewer frames than others but it is likely to have fewer correct matches than others, too. In contrast, VSUMM likes to capture more frames as the video summarization result, which tends to match more frames but it sometimes will sacrifice the precision. As shown in Fig. 6, it has the most correct frames, but selecting many frames also leads to many unmatched frames. Generally, our proposed VRCVS clearly outperforms other compared methods with the highest F-score. It demonstrates that our work VRCVS is able to achieve better performance of capturing the essential content of video accurately.

For the VYT database, as shown in Table 3, our proposed method VRCVS clearly outperforms several state-of-the-art methods. The extracted examples can be seen from Fig. 8. The first two rows are ground truth frames chosen by the publishers, the third and fourth rows are summary generated by VSUMM, the fifth and sixth rows are the summary obtained by KFVS and the seventh and eighth rows are summary obtained by our proposed method VRCVS. The frames marked with green border are the correct matches between the summary and the ground truth. The frames marked with yellow border are the frames with discriminative content despite not being chosen by the publisher. The comparison demonstrates that our proposed method can capture more representative frames with great accuracy.

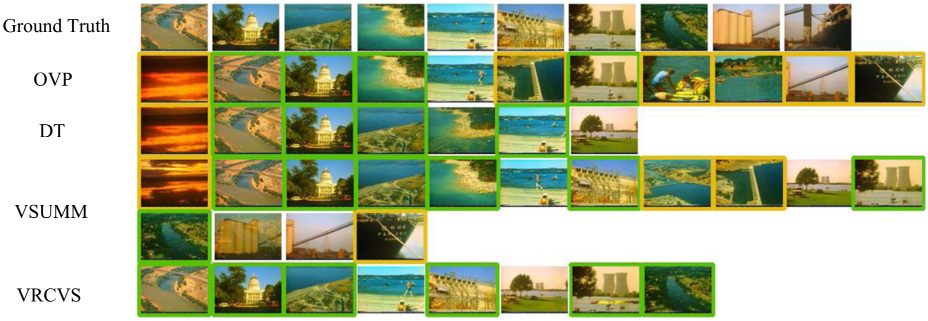


Fig. 7 Sample video summarization results of all the methods for the 23*th* video of VSUMM database, where the first row is ground truth, the second row is the summary obtained by OVP [21], the third row is summary generated by DT [20], the fourth and fifth row are summary captured by VSUMM [2] and the sixth row is summary obtained by our proposed method VRCVS. The frames marked with green borders are the correct matches between the summary and the frames marked with yellow borders are unmatched frames

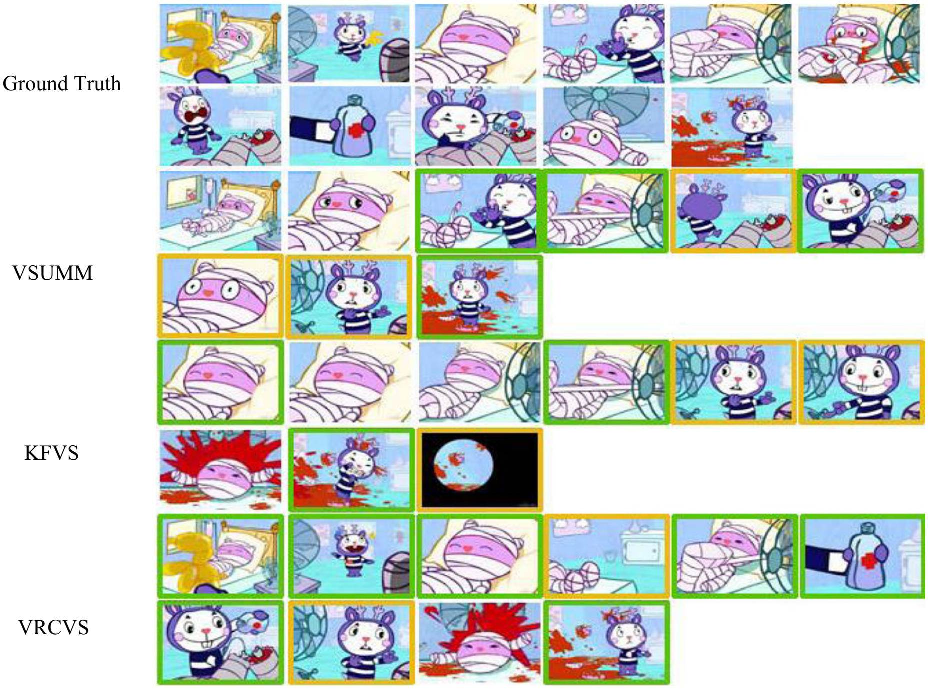


Fig. 8 Sample video summarization results of all the methods for the 20*th* video of VYT database, where the first and second row are ground truth, the third and fourth row are the summary obtained by VSUMM [2], the fifth and sixth row are summary captured by KFVS [11] and the seventh and eighth row are summary obtained by our proposed method VRCVS. The frames marked with green borders are the correct matches between the summary and the frames marked with yellow borders are unmatched frames

4.4 Comparison with our proposed VRHDPS and HDPS clustering method

In this section, we compare the performances of our proposed clustering method VRHDPS with HDPS [23] when dealing with video summarization task. The comparison is executed on VSUMM database and the results are shown in Table 4.

In Table 4, HDPS has a higher precision than our proposed method, but our work VRHDPS outperforms it in recall and F-score. The result means that our proposed method can generate more representative frames than HDPS does and has a great overall performance. In fact, the recall of HDPS is 0.40 which declares that not a half of ground truth is captured by HDPS. After analyzing the video summarization result of HDPS, we discover that it selects little

Table 4 Comparative performance of HDPS and our proposed clustering method VRHDPS

	Recall	Precision	F-score
HDPS	0.40	**0.79**	0.48
VRHDPS	**0.63**	0.68	**0.63**

Bold entries show the best results of each evaluation method

(a) The frame has three detected SIFT descriptors (b) The frame has zero detected SIFT descriptor

Fig. 9 Two frames with little detected SIFT descriptors. (**a**) is the frame has three SIFT descriptors, and (**b**) is the frame has zero SIFT descriptor

frames as result to reach a high precision. Generally, our proposed clustering method VRHDPS can do better in summarizing the video than HDPS.

4.5 Discussion

In the above sections, the comparisons with several classical clustering algorithms, a number of clustering-based video summarization models and the HDPS clustering algorithm are presented. The experimental results illustrate that the overall performance of our proposed method is better than other compared approaches and evidences that our proposed method is capable of accomplishing the task of video summarization. In addition, the results show that the proposed method is more robust and stable than several clustering-based static video summarization approaches. Moreover, because VRHDPS does not require any iteration in clustering process, our work realizes more efficiency than some classical or state-of-the-art clustering algorithms. However, when analyzing in detail, we also find a weakness of the proposed work. The frames with few SIFT descriptors have similar BoW representation. In other way, they will be considered as same in the clustering process although they are visually different. For example, in Fig. 9, two frames with little SIFT descriptors are likely to be included into one cluster due to the similarity of their representation, which is possible to have a bad effect on performance. Dense-SIFT [14] will be involved in our future work to solve this problem.

5 Conclusion and future work

Static video summarization is seen as an effective tool for user to deal with massive videos. In this paper, we formulate static video summarization as a clustering problem and propose a novel clustering-based method for this task. Based on the insights from HDPS approach, we propose VRHDPS method by integrating some important properties of video summarization into our model. Firstly, we utilize pre-sampling to reduce redundancy of the given video and get candidate frames. Then we use BoW model to present the visual content of candidate frames. Finally, the proposed VRHDPS clustering approach is used to gather candidate frames into clusters and a considerate strategy is also adopted to generate static video summarization result. In empirical evaluation, we compare the proposed method with several classical clustering algorithms and the experiment results illustrate that our work is more effective, efficient and robust. Meanwhile, the comparative results with several state-of-the-art clustering methods also indicate better performance of the proposed method. In future, we will explore to apply our proposed method for the task of dynamic video summarization. In addition, we will investigate how to improve the representation of video frames. For example, global features or dense-SIFT may be added to represent the frame.

Acknowledgments This work was supported by the National Natural Science Foundation of China (No. 61502311), Natural Science Foundation of Guangdong Province (No. 2016A030310053), Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase), the Science and Technology Innovation Commission of Shenzhen under Grant (No. JCYJ20150324141711640), and the Shenzhen University research funding (201535).

References

1. Almageed A, Online W (2008) Simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing. In: Proceedings of IEEE International Conference of Image Processing pp. 3200–3203
2. Avila SEF, Lopes APB, Luz A Jr, Araújo AA (2011) Vsumm: a mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recogn Lett 32:56–68
3. Ballan I, Bertini M, DelBimbo A, Seidenari L, Serra G (2011) Event detection and recognition for semantic annotation of video. Multimed Tools Appl 51:279–302
4. Chang IC, Chen KY (2007) Content-selection based video summarization. In: Proceeding of the Int Conf on Consumer Electronics pp. 1–2
5. Chang HS, Sull S, Lee SU (1999) Efficient video indexing scheme for content-based retrieval. IEEE T Circ Syst Vid 9:1269–1279
6. Chen YJ, Wu KS, Zhang Q (2015) From QoS to QoE: a tutorial on video quality assessment. IEEE Commun Surv Tutor 17:1126–1165
7. Chen L, Zhou YP, Chiu DM (2015) Analysis and detection of fake views in online video services. ACM T Multim Comput 23:1163–1175
8. Chen L, Zhou YP, Chiu DM (2015) Smart streaming for online video services. IEEE T Multimedia 17:485–497
9. Frey JB, Dueck D (2007) Clustering by passing messages between data points. Science 315:972–976
10. Gong Y, Liu X (2011) Video summarization with minimal visual content redundancies. In: Proceeding of the IEEE Int Conf on Image Processing pp. 362–365
11. Guan G, Wang Z, Lu S, Deng JD, Feng D (2013) Keypoint base keyframe selection. IEEE T Circ Syst Vid 23:729–734
12. Hanjalic A, Lagendijk RL, Biemond J (1998) A New Method for Key Frame based Video Content Representation. In: Proceeding of Image Databases and Multimedia Search
13. Liu JK, Au MH, Susilo W, Liang K, Lu RX, Srinivasan B (2015) Secure sharing and searching for real-time video data in mobile cloud. IEEE Netw 29:46–50
14. Liu C, Yuen J, Torralba A (2011) SIFT flow: dense correspondence across scenes and its applications. IEEE T Pattern Anal 33:978–994
15. Lowe DG (2004) Distinctive image features from scale-invariant key-points. Int J Comput Vision 60:91–110
16. Lu SY, Wang ZY, Mei T, Guan GL, Feng DD (2014) A bag-of-importance model with locality-constrained coding based feature learning for video summarization. IEEE T Multimedia 16:1497–1509
17. Luxburg UV (2007) A tutorial on spectral clustering. Stat Comput 4:395–416
18. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. Proc Fifth Berkeley Symp Math Statistics Probability 1:281–297
19. Mei S, Guan G, Wang Z, Wan S, He M, Feng D (2015) Video summarization via minimum sparse reconstruction. Pattern Recogn 48:289–612
20. Mundur P, Rao Y, Yesha Y (2006) Keyframe-based video summarization using Delaunay clustering. Int J Digit Libr 6:219–232
21. Open video project. https://www.open-video.org
22. Ren TW, Qiu ZY, Liu Y, Yu T, Bei J (2015) Soft-assigned bag of features for object tracking. Multimedia Systems (MMSJ) 21(2):189–205
23. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344:1492–1496
24. Sivic J, Zisserman A (2009) Efficient visual search of videos cast as text retrieval. IEEE T Pattern Anal 31:591–605
25. Truong BT, Venkatesh S (2007) Video abstraction: a systematic review and classification. ACM T Multim Comput 3:1–37
26. YouTube statistics (2015). https://www.youtube.com/yt/press/statistics.html

27. Zhou YP, Chen L, Yang CF, Chiu DM (2015) Video popularity dynamics and its implication for replication. IEEE T Multimedia 17:1273–1285
28. Zhou YP, Fu TZJ, Chiu DM (2015) A unifying model and analysis of P2P VoD replication and scheduling. IEEE ACM T Network 23:1163–1175
29. Zhuang Y, Rui Y, Huang TS, Mehrotra S (1998) Adaptive keyframe extraction using unsupervised clustering. In: Proceedings of the International Conference on Image Processing pp. 866–870

Jiaxin Wu received her B.Sc. in College of Computer Science and Software Engineering from Shenzhen University in 2015. She is currently a post-graduate student in the Department of Computer Science, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her current research interests include video content analysis and deep learning.

Sheng-hua Zhong received her B.Sc. in Optical Information Science and Technology from Nanjing University of Posts and Telecommunication in 2005 and M.S. in Signal and Information Processing from Shenzhen University in 2007. She got her Ph.D. from Department Computing, The Hong Kong Polytechnic University in 2013. She worked as a Postdoctoral Research Associate in Department of Psychological & Brain Sciences at The Johns Hopkins University from 2013 to 2014. Currently, she is an Assistant Professor in College of Computer Science & Software Engineering at Shen Zhen University in Shen Zhen. Her research interests include multimedia content analysis, cognitive science, psychological and brain science, and machine learning.

Jianmin Jiang received the Ph.D. degree from the University of Nottingham, Nottingham, U.K., in 1994. He joined Loughborough University, Loughborough, U.K, as a Lecturer in computer science. From 1997 to 2001, he was a Full Professor of Computing with the University of Glamorgan, Wales, U.K. In 2002, he joined the University of Bradford, Bradford, U.K, as a Chair Professor of Digital Media, and Director of Digital Media and Systems Research Institute. In 2014, he moved to Shenzhen University, Shenzhen, China, to carry on holding the same professorship. He is also an Adjunct Professor with the University of Surrey, Guildford, U.K. His current research interests include image/video processing in compressed domain, computerized video content understanding, stereo image coding, medical imaging, computer graphics, machine learning, and AI applications in digital media processing, retrieval, and analysis. He has published over 400 refereed research papers. Prof. Jiang is a Chartered Engineer, a member of EPSRC College, and EU FP−6/7 evaluation expert. In 2010, he was elected as a scholar of One-Thousand-Talent-Scheme funded by the Chinese Government.

Yunyun Yang obtained Ph.D. degree from Department of Mathematics in Harbin Institute of Technology in July 2012. Currently, she is an Assistant Professor in School of Natural Sciences and Humanities, Harbin Institute of Technology Shenzhen Graduate School. Her research interests include image processing, image segmentation, pattern recognition and applied mathematics. She worked as a visiting scholar in Department of Mathematics at the Ohio State University in USA from Sept. 2009 to Sept. 2010 and from Mar. 2014 to Apr. 2014.