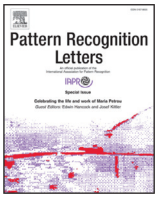
Pattern Recognition Letters 72 (2016) 52–61

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec



Weighted multi-view key-frame extraction☆

Antonis Ioannidis, Vasileios Chasanis*, Aristidis Likas

Department of Computer Science and Engineering, University of Ioannina, GR45110, Greece

ARTICLE INFO

Article history:
Available online 9 February 2016

Keywords:
Key-frame extraction
Spectral clustering
Multi-view clustering

ABSTRACT

The extraction of representative key-frames from video shots is very important in video processing and analysis, since it constitutes the basis for several important tasks such as video shot summarization, browsing and retrieval as well as high-level video segmentation. The extracted key-frames should capture a great percentage of the information of a shot content, while at the same time they should not present similar visual information. Clustering or segmentation methods are usually employed to extract key-frames. A major difficulty is caused by the large variety in the visual content of videos. Thus, using a single image descriptor (color, texture etc) to extract key-frames is not always effective, since there is no single descriptor surpassing the others in all video cases. To tackle this problem, we propose an approach for the weighted fusion of several descriptors that automatically estimates the weight of each descriptor. The weights reflect the relevance of each descriptor for the specific video shot. Moreover, they are used to form a composite similarity matrix as the weighted sum of all the similarity matrices corresponding to the individual descriptors. This matrix is then used as input to a spectral clustering algorithm that partitions shot frames into groups. Finally the medoid frame of each group is selected as key-frame. Numerical experiments using a variety of videos demonstrate that our method is capable of efficiently summarizing video shots regardless of the characteristics of the visual content of a video.

1. Introduction

In recent years there has been a significant increase in the availability of high quality digital video, due to advances in video recorders, broadband services and high-capacity storage media. The extensive use of video in several widely used applications such as distance learning, digital libraries, the internet TV and video on demand, and the steadily increasing rate of movie production, daily adds a huge volume of videos to various repositories. This implies a strong need for techniques and applications that will offer effective indexing, browsing and retrieval of video data.

The first step in this direction is to segment the video into smaller units in order to further proceed with indexing and browsing. The smallest physical segment of a video is the shot which is defined as an unbroken sequence of video frames taken from a single camera. Then, reliable shot summarization, which is one of the most important problems in digital video processing and analysis, should be performed. The most common approach used for shot representation and summarization is the selection of a set of key-frames sufficiently representing the whole shot content. The efficient summarization of a video shot is necessary for two reasons. Firstly, one can rapidly make an assessment about the video content by inspecting the key-frames of the shots of the video. Secondly, having extracted key-frames from video shots, one can define the similarity between two shots based on the similarities of their corresponding key-frames. Shot similarity can then be used for grouping shots into scenes, content-based shot retrieval and video rushes summarization.

In order for a key-frame extraction algorithm to be effective, the extracted key-frames should represent the whole video content without missing important information, while at the same time, these key-frames should not be similar, in terms of video content information, thus containing redundant information. A major category of key-frame extraction algorithms are segmentation-based. Such algorithms detect abrupt changes in terms of similarity between successive frames [1,2]. In [3], three properties (Iso-Content Distance, Iso-Content Error and Iso-Content Distortion) are considered. The selected key-frames are equidistant in the shot content curve with respect to those properties. In [4], a key-frame selection framework based on keypoints is presented. A global pool of unique keypoints extracted from all frames based on SIFT descriptors [5] is generated and those frames that best cover the global keypoint pool are selected as key-frames. This type of key-frame extraction has the disadvantage that it may extract similar key-frames, if the same content reappears during a shot.

☆ "This paper has been recommended for acceptance by Anders Heyden".

* Corresponding author. Tel.: +30-26510 08838; fax: +30 26510 08882.
E-mail address: vchasani@cs.uoi.gr (V. Chasanis).

http://dx.doi.org/10.1016/j.patrec.2016.01.027
0167-8655/© 2016 Elsevier B.V. All rights reserved.

Another category of key-frame extraction algorithms perform clustering of shot frames into groups and select a representative frame of each group as key-frame. In [6], multiple frames are detected using unsupervised clustering based on the visual variations in shots. A variant of this algorithm is presented in [7], where the final number of key-frames depends on a threshold parameter which defines whether two frames are similar. In [8], static video summaries are produced based on color feature extraction from video frames and the k-means clustering algorithm. In [9], spectral clustering on spatio-temporal features is employed to extract key-frames. In [10], the mutual information values of consecutive frames are clustered into groups using a split-merge approach. A different technique for the key-frame selection is described in [11], where the key-frames position in the video is taken into account. In [12], the problem of scalable video summarization is modeled as a problem of scalable graph clustering and is solved using skeleton graph and random walks in the analysis stage.

A third category of algorithms transform key-frame extraction into a sparse dictionary selection problem. In [13] a scalable video summarization based on sparse dictionary selection is proposed and a relaxed constraint based on $L_{2,1}$ norm is imposed to ensure sparsity. In [14] the minimum number of key-frames are selected to reconstruct the entire video as accurate as possible by adopting a real sparse constraint based on L_0 norm.

Most of the existing key-frame extraction algorithms employ a single frame descriptor (e.g. color histograms, texture descriptors, visual words) to capture the content of shot frames. However, due to the large variety in visual content a single image descriptor does not suffice for the efficient summarization of several videos. Therefore, methods capable of fusing several descriptors constitute a promising solution to tackle this problem. Typical fusion methods (also called multi-view methods) are based either on the concatenation of the individual descriptor vectors [13] or on the combination of the solutions obtained using each descriptor separately. However, a significant drawback of those fusion approaches is that all descriptors are considered to be equally important. Therefore, the existence of irrelevant descriptors might lead to performance deterioration.

In this work, we propose an approach for the *weighted fusion of several descriptors* that automatically estimates a weight of each descriptor. The weight reflects the relevance of each descriptor and therefore adjusts its contribution to the final solution obtained. In this way, low (near zero) weights are assigned to irrelevant descriptors, thus their presence does not affect the solution which is determined by the relevant descriptors which are assigned higher weight values. The fusion weights are estimated automatically using a weighted multi-view clustering algorithm [15]. Then, a single similarity matrix is computed as the weighted sum of all individual similarity matrices of the image descriptors (views). This matrix serves as input to a spectral clustering algorithm that clusters shot frames to groups. Finally, the medoid of each group is selected as key-frame.

This is an extended version of the paper presented in [16] with improved presentation of motivation and related work and a more detailed description of the weighted multi-view clustering method which is the main computational tool of our method. Moreover, it includes additional experimental results using more than two image descriptors, experiments on two new datasets and it also contains a new subsection where visual examples are presented aiming to provide a better understanding of the advantages of the proposed method.

The rest of the paper is organized as follows. The key-frame extraction algorithm is presented in Section 2. This method consists of two processing stages: In the first stage the weighted multi-view clustering algorithm is applied in order to compute the weights assigned to the different descriptors (views). In the second stage the estimated weights are exploited to construct a composite similarity matrix computed as the weighted sum of the individual view kernel matrices. This similarity matrix is used to perform spectral clustering of set of video frames and then extract key-frames as cluster medoids. In Section 3 we describe the image descriptors (views) employed in the herein approach, while in Section 4 we provide numerical experiments and present three visual examples. Finally, in Section 5 we summarize our method and provide suggestions for future work.

2. Weighted multi-view key-frame extraction

A large variety of existing image descriptors (color, texture, visual words etc) can be used to represent the visual content of a video sequence. Due to large variations observed in the visual content of videos, a single descriptor cannot efficiently describe the content of several videos. To tackle this problem we propose the combination of two or more descriptors that we call *views*. For each view, we compute a kernel matrix that provides the similarity between each pair of frames of a video shot in terms of the corresponding descriptor. All view kernel matrices are then combined (through their weighted sum) to form a final similarity matrix that serves as input to a spectral clustering algorithm providing the extracted key-frames. The weights that reflect the quality of each view are estimated using a technique for training weighted multi-view Convex Mixture Models [15] that is described below.

2.1. View weight estimation based on multi-view Convex Mixture Models

Suppose we are given a dataset of N data points $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$. *Convex Mixture Models* (CMMs) [17] are mixture models of special type aiming to assign data points into clusters by extracting representative exemplars from the data set. In CMMs the number of mixture components is equal to the number of data points, thus for each data point x_i we consider a separate component $f_i(x)$ ‘centered’ at this data point with prior (mixing weight) q_i. This prior probability denotes the probability that the corresponding data point will become a cluster representative. The CMM distribution is given by:

$$Q(x) = \sum_{j=1}^{N} q_j f_j(x) = C_\phi(x) \sum_{j=1}^{N} q_j e^{-\beta d_\phi(x, x_j)}, \tag{1}$$

where $q_j \geq 0$ is the prior probability of the jth component, satisfying the constraint $\sum_{j=1}^{N} q_j = 1$, $f_j(x)$ is an exponential family distribution (see Eq. (7)) with d_ϕ being the Bregman divergence corresponding to the components distribution, $C_\phi(x)$ is independent of x_j, and β is a constant affecting the obtained number of clusters [17]. Appropriate β values can be found in the range of an empirically defined β_0 value:

$$\beta_0 = N^2 log N / \sum_{i,j=1}^{N} d_\phi(x_i, x_j). \tag{2}$$

Initially all data priors q_i are set equal, thus all data points are equally considered as possible cluster representatives. The CMM model is trained by maximizing the data log-likelihood with respect to the priors q_i and, after training, the data points with highest prior are selected as exemplars (cluster representatives) and the rest points are grouped based on their most similar exemplar. Since the likelihood maximization problem is convex, a unique global optimum solution is easily obtained through a simple iterative update procedure.

Suppose now that we are given a *multi-view dataset* of N instances and V views, $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ where x_i is the representation of the ith instance across the views, i.e., $x_i = \{x_i^1, x_i^2, \ldots, x_i^V\}, x_i^v \in \Re^{d^v}$. We assume that the mixture distribution of each view v, is a CMM:

$$\begin{aligned} Q^v(x^v) &= \sum_{j=1}^{N} q_j f_j^v(x^v) \\ &= C_{\phi_v}(x^v) \sum_{j=1}^{N} q_j e^{-\beta^v d_{\phi_v}(x^v, x_j^v)}, x^v \in \Re^{d^v}, \end{aligned} \tag{3}$$

where $q_j \geq 0$ is the prior probability of the jth component, satisfying the constraint $\sum_{j=1}^{N} q_j = 1$. Appropriate β^v values can be found in the range of an empirically defined β_0^v value:

$$\beta_0^v = N^2 logN / \sum_{i,j=1}^{N} d_{\phi_v}(x_i^v, x_j^v). \tag{4}$$

In the case where a $N \times N$ kernel matrix K^v is available for each view, with $K^v(i, j)$ reflecting the similarity between x_i^v and x_j^v, then it holds that:

$$d_{\phi_v}(x_i^v, x_j^v) = K^v(i, i) + K^v(j, j) - 2K^v(i, j) \tag{5}$$

In *weighted multi-view CMM* [15] each view vector is assumed to be generated from a corresponding CMM, thus it can be considered as a *mixture of CMMs* with mixing weights π^v indicating the relevance of each view v. In this way it is possible to locate exemplars in the dataset by allowing all views to contribute to the objective function with different weights, which are learned automatically. The weighted multi-view CMM is defined as follows:

$$\begin{aligned} F(x = \{x^1, x^2, \ldots, x^V\}) &= \sum_{v=1}^{V} \pi^v Q^v(x^v) \\ &= \sum_{v=1}^{V} \pi^v \sum_{j=1}^{N} q_j f_j^v(x^v), x^v \in \Re^{d^v}, \end{aligned} \tag{6}$$

where

$$\begin{aligned} & f_j^v(x^v) = C_{\phi_v}(x^v) e^{-\beta^v d_{\phi_v}(x^v, s_j^v)}, \\ & \quad \pi^v \geq 0, \sum_{v=1}^{V} \pi^v = 1, q_j \geq 0, \sum_{j=1}^{N} q_j = 1. \end{aligned} \tag{7}$$

In the above equations $F(x)$ is a mixture model whose number of components is equal to the number of the views and each component is a CMM $Q^v(x^v)$, corresponding to the vth view. Each CMM is associated with a weight π^v which represents the contribution of each view in the mixture model.

All instances are considered as possible cluster representatives, since a CMM is used for each view. It is important to note that the priors q_j are the same across all views, to allow the extraction of representative exemplars based on every view. Therefore, if after training an instance has a high q_j value, then probably it is a good exemplar for all the available views. Moreover, if after training, a low weight π^v is assigned to view v, this is an indication that this view is irrelevant for the specific dataset.

Since $F(x)$ is considered as a mixture model, in order to perform model training the log-likelihood of the dataset must be maximized with respect to q_j and π^v. The log-likelihood is defined as follows:

$$\begin{aligned} L(\mathcal{X}; \{\pi^v\}_{v=1}^{V}, \{q_j\}_{j=1}^{N}) &= \sum_{i=1}^{N} \log \sum_{v=1}^{V} \pi^v Q^v(x_i^v) \\ &= \sum_{i=1}^{N} \log(\sum_{v=1}^{V} \pi^v \sum_{j=1}^{N} q_j f_j^v(x_i^v)). \end{aligned} \tag{8}$$

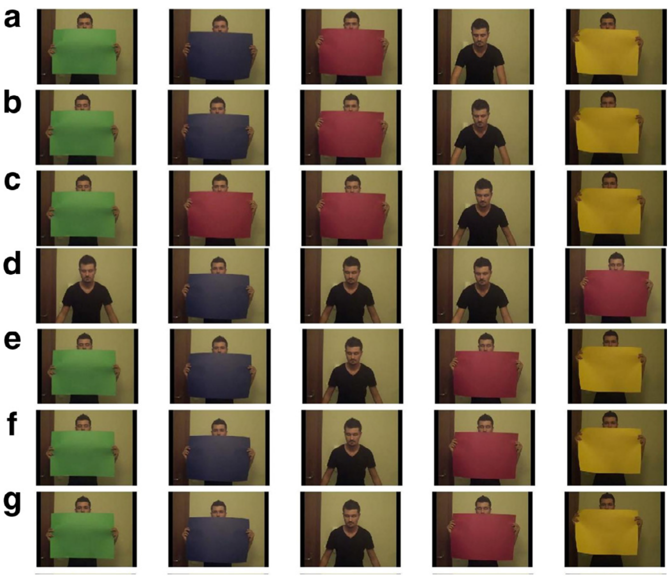


Fig. 1. Example shot sequence (better seen in color). (a) Ground truth. Solutions using: (b) HSV, (c) Centrist (CEN), (d) Wavelets (WAV), (e) HSV - CEN, (f) HSV - WAV, (g) CEN - WAV. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Weights assigned to the descriptors by the weighted multi-view CMM.

Descriptors	HSV	CEN	WAV
HSV - CEN	0.9880	0.0120	–
HSV - WAV	0.9449	–	0.0551
CEN - WAV	–	0.3286	0.6714

This maximization problem is not convex, due to the addition of the weights π^v. Local maxima can be found by applying an EM algorithm [18]. The EM algorithm starts with some initial parameter values and iteratively adjusts them in order to increase the likelihood until a local maximum is reached. Since the only parameters of the weighted multi-view CMM are the prior probabilities π^v and q_j, by initializing those values uniformly ($\pi^{v(0)} = 1/V, q_j^{v(0)} = 1/N$), multiple executions can be avoided. Note that the weights π^v might be initialized accordingly if there exists prior knowledge concerning the quality of the available views. More information about the EM for the Weighted Multi-view CMMs can be found in [15]. After the completion of EM, in order to extract k key-frames, the k exemplars with the higher q_j values are selected.

In Fig. 1, we give an example of the key-frames extracted from a video shot using the weighted multi-view CMM algorithm with HSV ("HSV3D"), Centrist and Wavelet descriptors. The same kernel (9) was used for all descriptors. In each experiment we employ only two of the three views. The first row depicts the ground truth of the video sequence. The ground-truth contains four key-frames where the same person holds square placards of different color and one key-frame representing all the frames where the person changes placards. It is obvious that only the color descriptor is relevant for this video sequence, providing the best single view clustering solution. When the HSV descriptor is combined with each of the other two descriptors, the clustering solution is still optimal. Surprisingly, even when centrist and wavelet descriptors are combined, the clustering solution of the weighted multi-view CMM algorithm is also optimal, contrary to the corresponding single-view clustering solutions. In Table 1, the view weights estimated by the weighted multi-view CMM algorithm are presented. The weight assigned to HSV is very high correctly indicating that this is the best view.

2.2. Key-frame selection using spectral clustering

Although the weighted multi-view CMM algorithm provides both the view weights $\pi^v, v = 1, \ldots, V$ and the representative exemplars (key-frames), we have empirically found that better results are obtained if we use only the view weights provided by this algorithm in order to build a final kernel as a weighted sum of the individual view kernels K^v: $S = \sum_{v=1}^{V} \pi^v K^v$: The composite matrix S is then used as input similarity to a spectral clustering algorithm.

Spectral methods [19] are well-known approaches to clustering. Assume we are given n data objects $X = [x_1 x_2 \ldots x_n]$ and a similarity matrix $\mathrm{S} \in \mathbb{R}^{n \times n}$, where $S_{ij} \geq 0$ reflects the similarity between x_i and x_j. Spectral clustering computes the top k eigenvectors of the similarity matrix to partition the objects of set X into k clusters. In our method the video frames of a shot are clustered into groups using an improved spectral clustering algorithm [20], that employs the global k-means algorithm [21] in the clustering stage after the eigenvector computation. Then, the medoid of each group, defined as the frame of a group whose average similarity to all other frames of this group is maximum, is characterized as a key-frame. The number of clusters in spectral is set equal to the number of key-frames in the ground-truth for each video.

3. Visual descriptors

Several image descriptors have been employed in the herein approach to describe the content of shot frames from different aspects.

- HSV Color Histograms: Two normalized HSV color histograms have been considered. The first one, denoted as “HSV1D”, results from the concatenation of 64 bins for hue and 16 bins for each of saturation and value. The second one, denoted as “HSV3D”, is obtained from the 3-dimensional HSV histogram and it is constructed using 8 bins for hue and 4 bins for each of saturation and value, resulting into a 128 (8 × 4 × 4) dimension feature vector. Although color descriptors are easy to compute and perform relatively well for key-frame extraction, they have certain disadvantages such that they cannot represent shape and texture and they are also sensitive to noise, eg. due to lighting changes.
- Wavelets: 9 Haar wavelet sub-bands are used on 3 × 3 grids to form a 81-d feature vector [22]. They are suitable for texture representation.
- Scale Invariant Feature Transform (SIFT): A very popular descriptor that represents local features in images [5], based on storing the weighted edge orientation histograms of salient corners of an image. It is more expensive to compute compared to color descriptors.
- Census Transform Histogram (CENTRIST): The Centrist descriptor (a 254-d feature vector) [23] encodes the structural properties of an image and is considered to be superior to SIFT in place and scene recognition tasks. It is easier and faster to compute compared to SIFT.

The bag of visual words representation [24] is employed to represent shot frames when SIFT descriptors were used. More specifically, all the descriptors extracted from shot frames are clustered into 20 or 50 clusters thus forming visual vocabulary with 20 or 50 visual words, respectively. For each frame, its corresponding set of descriptors is mapped into these 20 or 50 visual words, each descriptor is mapped to its nearest visual word, resulting into a vector containing the normalized count of each visual word in the frame (denoted as “SIFT20” or “SIFT50”, respectively).

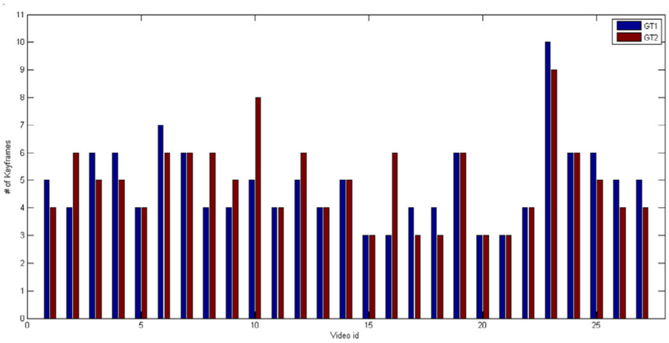


Fig. 2. Number of key-frames per video sequence for each ground truth assessment.

3.1. Kernel function

The kernel function used in our experiments to build the kernel matrix for each view, is the Chi-Square kernel, due to its simplicity and effectiveness, especially as a measure of histogram similarity in computer vision tasks. Given two image descriptor vectors x, y, our kernel function is computed as:

$$K(x, y) = 1 - \sum_i \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}. \tag{9}$$

Note that it is possible to use a different kernel function for each descriptor, however in our experiments the same kernel function was used for all frame descriptors (views).

4. Experimental evaluation

The empirical evaluation of key-frame extraction methods is a rather difficult task, since there do not exist any widely accepted performance indicators that could be computed automatically. Thus, apart from the necessary manual annotation of each video sequence for the specification of its ground-truth key-frames, we also have to resort to visual assessment in order to determine whether the key-frames provided by a method match to the ground-truth key-frames of each video sequence.

4.1. Datasets and ground truth

In our experiments we have considered three different datasets. The first dataset contains 27 shot sequences of various visual content including car motion, construction demolition, car accidents, changing traffic lights, indoor and outdoor movement. Ground-truth key-frames were visually extracted by two different persons (video editing specialists) so as to represent adequately the content of each shot sequence. Each of the two persons also provided for each sequence a visual assessment of whether the key-frames extracted by each compared method match to the ground-truth set. These two assessments are denoted as “GT1” and “GT2” in the rest of the paper. In Figs. 2 and 3, we present the number of key-frames per video sequence for each ground truth assessment and selected representative frames from the video sequences of this dataset, respectively. The second dataset contains 50 videos taken from Open video project[1]. These videos are distributed among several genres (documentary, educational, ephemeral, historical, lecture), their duration varies from 1 to 4 min and there exist approximately 75 min of video in total. Finally, the third dataset[2] contains 50 videos from websites like YouTube. These videos are distributed among several genres (cartoons,

[1] http://www.open-video.org.

[2] https://sites.google.com/site/vsummsite/download.

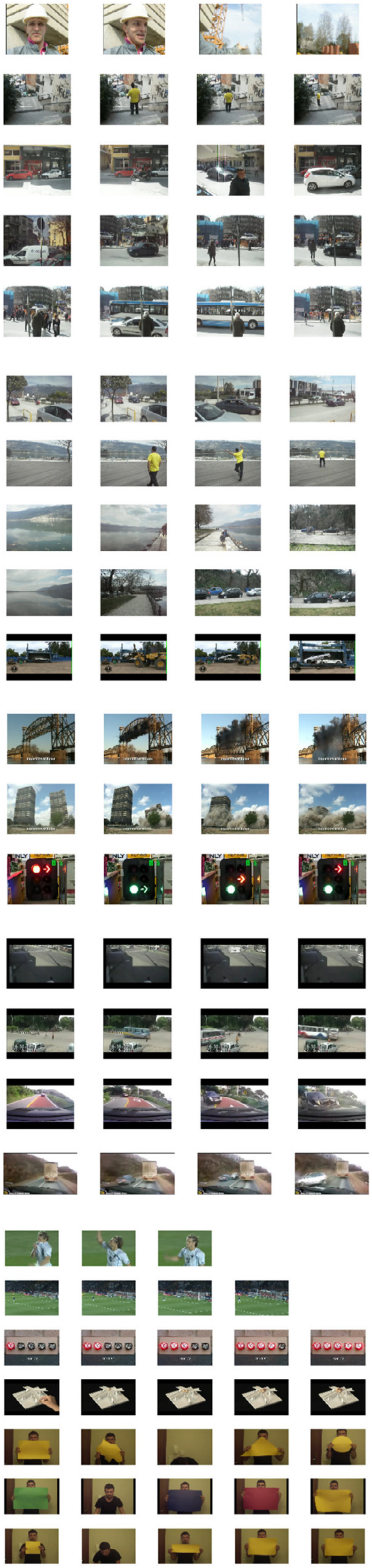


Fig. 3. Selected frames from the video sequences of the first dataset.

news, sports, commercials, tv-shows and home videos) and their duration varies from 1 to 10 min. For the last two datasets, the ground truth is also available in the form of user summaries (sets of key-frames) [8]. These summaries were created from 50 users, each one dealing with five videos, meaning that each video has five video summaries created by five different users.

4.2. Performance measure

As mentioned previously, performance evaluation for the first dataset has been based on the visual comparison (separately by each of the two persons) of the key-frames extracted from each method against the ones in the ground truth set. Suppose we are given M video sequences $VS = \{VS_1, \dots, VS_M\}$ and let G_i $(i = 1, \dots, M)$ the number of ground truth key-frames of sequence VS_i. For each sequence VS_i, the key-frame extraction method is applied setting the number of key-frames (clusters) equal to G_i and the number F_i of successfully found ground truth key-frames is determined through visual assessment. Thus, F_i/G_i is the percentage of successfully found ground truth key-frames per video sequence. Our performance measure m (mean accuracy) is computed as the average accuracy over all video sequences:

$$m = \frac{1}{M}\sum_{i=1}^{M}\frac{F_i}{G_i}. \tag{10}$$

It is apparent that separate performance results (namely “GT1” and “GT2”) have been computed for each of the two persons that made the visual assessment.

For the second and third datasets each summary extracted for a video from any approach (Automatic Summary (AS)) is compared with all the corresponding to this video User Summaries (US). Color histograms and Manhattan distance [8] are employed to measure the distance between two summaries. Two key-frames are similar if the distance between them is less than a predetermined threshold $d = 0.5$. Once two frames are matched, they are removed from the next iteration of the comparing procedure. The quality of the any generated summary is assessed by two metrics, called accuracy rate CUS_A and error rate CUS_E, which are defined as follows:

$$CUS_A = \frac{n_{mAS}}{n_{US}}, \tag{11}$$

$$CUS_E = \frac{n_{\tilde{m}AS}}{n_{US}}, \tag{12}$$

where n_{mAS} is the number of matching key-frames from automatic summary (AS), $n_{\tilde{m}AS}$ is the number of non-matching key-frames from AS and n_{US} is the number of key-frames from user summary (US). Note that we seek for high values of CUS_A and low values of CUS_E.

4.3. Experimental results

In Tables 2 and 3 we present comparative results for the first dataset on ground truth assessments “GT1” and “GT2”, respectively. Each of the six image descriptors (see Section 3) was also tested in a single-view experiment, using the methods proposed in [6] and [20]. Performance results of these experiments are presented as “HSV1D”, “HSV3D”, “CEN”, “WAV”, “SIFT20” and “SIFT50”. Eight pairs of image descriptors were tested in the multi-view experiment. Experiment “SP-UNWEIGHTED” corresponds to the solution using the spectral clustering algorithm, when the similarity matrix is built as an *unweighted sum* of the individual view kernels. Experiment “SP-MULTIVIEW” corresponds to the solution using the spectral clustering algorithm, when the similarity matrix is built as a weighted sum of the individual view kernels, with

Table 2
Comparative results using mean accuracy (m in (%)) based on assessment “GT1”, for both sampling rates ($Sr = 1$ and $Sr = 5$) for the first dataset. Daggers indicate statistical significance.

Sampling rate	***Sr* = 1**	***Sr* = 5**	***Sr* = 1**	***Sr* = 5**
Descriptors	[6]		[20]	
HSV1D	63.08 ± 16.43	65.30 ± 17.70	61.90 ± 14.54	68.99 ± 15.40
HSV3D	64.52 ± 13.09	65.88 ± 12.04	69.28 ± 12.85	70.08 ± 12.69
SIFT20	61.96 ± 20.04	62.57 ± 15.32	71.28 ± 14.47	68.29 ± 12.05
SIFT50	61.16 ± 15.70	65.78 ± 17.30	69.65 ± 13.61	67.86 ± 14.05
CEN	62.98 ± 12.22	62.55 ± 12.36	68.41 ± 15.39	68.26 ± 11.39
WAV	66.03 ± 18.60	64.30 ± 16.62	66.47 ± 16.61	69.49 ± 15.66
	SP-UNWEIGHTED		**SP-MULTIVIEW**	
HSV1D - SIFT20	73.17 ± 14.88	70.76 ± 14.94	**78.82 ± 17.34** †‡	**80.26 ± 14.89** †‡
HSV1D - SIFT50	72.70 ± 13.35	72.61 ± 13.10	**79.44 ± 17.80** †‡	**79.65 ± 15.10** †‡
HSV3D - SIFT20	74.25 ± 13.15	72.92 ± 16.45	**83.26 ± 11.71** †‡	**79.09 ± 15.93** †‡
HSV3D - SIFT50	73.94 ± 14.76	74.03 ± 16.14	**83.88 ± 12.14** †‡	**82.18 ± 14.75** †‡
HSV1D - CEN	70.98 ± 16.84	68.91 ± 17.64	**83.26 ± 15.23** †‡	**81.50 ± 15.79** †‡
HSV1D - WAV	69.74 ± 15.68	71.25 ± 14.79	**81.72 ± 15.23** †‡	**82.36 ± 15.14** †‡
HSV3D - CEN	69.74 ± 15.27	68.91 ± 17.64	**80.79 ± 12.10** †‡	**80.45 ± 16.06** †‡
HSV3D - WAV	70.42 ± 16.48	72.92 ± 15.10	**82.03 ± 14.12** †‡	**80.76 ± 13.70** †‡

Table 3
Comparative results using mean and standard deviation of accuracy (m in (%)) based on assessment “GT2”, for both sampling rates ($Sr = 1$ and $Sr = 5$) for the first dataset. Daggers indicate statistical significance.

Sampling rate	***Sr* = 1**	***Sr* = 5**	***Sr* = 1**	***Sr* = 5**
Descriptors	[6]		[20]	
HSV1D	60.43 ± 14.84	57.41 ± 16.42	68.66 ± 15.12	66.48 ± 14.99
HSV3D	61.08 ± 14.87	62.66 ± 14.18	69.40 ± 15.48	69.04 ± 14.44
SIFT20	57.24 ± 21.32	61.71 ± 16.51	64.47 ± 16.11	64.75 ± 14.53
SIFT50	56.51 ± 17.33	61.18 ± 16.50	64.27 ± 15.37	65.52 ± 14.55
CEN	57.94 ± 15.52	56.92 ± 14.18	68.86 ± 15.22	68.46 ± 15.57
WAV	59.26 ± 16.74	58.13 ± 15.90	65.99 ± 14.97	64.97 ± 14.52
	SP-UNWEIGHTED		**SP-MULTIVIEW**	
HSV1D - SIFT20	67.40 ± 13.49	68.46 ± 16.01	**76.72 ± 14.56** †‡	**76.02 ± 13.32** †‡
HSV1D - SIFT50	71.17 ± 14.93	68.52 ± 11.49	**77.83 ± 13.70** †‡	**76.17 ± 11.73** †‡
HSV3D - SIFT20	70.21 ± 15.14	67.10 ± 14.36	**78.94 ± 13.69** †‡	**79.38 ± 13.60** †‡
HSV3D - SIFT50	69.59 ± 16.46	71.23 ± 14.25	**81.41 ± 14.27** †‡	**80.06 ± 14.75** †‡
HSV1D - CEN	68.91 ± 18.96	69.44 ± 16.08	**78.91 ± 13.09** †‡	**79.10 ± 12.20** †‡
HSV1D - WAV	69.32 ± 14.31	68.73 ± 17.65	**80.34 ± 12.26** †‡	**77.44 ± 11.48** †‡
HSV3D - CEN	67.43 ± 16.50	69.51 ± 16.49	**78.76 ± 14.33** †‡	**80.49 ± 15.66** †‡
HSV3D - WAV	64.32 ± 17.72	70.43 ± 14.39	**79.37 ± 14.15** †‡	**79.07 ± 11.92** †‡

weights determined by the weighted multi-view CMM algorithm. In another experiment we carried out, instead of employing all available frames of a video sequence, we sampled every five frames in order to reduce execution time. Dagger (†) and double dagger (‡) superscripts denote that the proposed method has a statistically significant difference from all single experiments and the corresponding experiment with equal weights, respectively, according to t-test (the significance level is taken as 0.05).

At first, it must be noted that it has been empirically confirmed that there is no single-view descriptor surpassing the other single-view descriptors. From the results in Tables 2 and 3 it is clear that the proposed method achieves the best performance compared to all single-view methods and the method with equal weights regardless of the pair of descriptors employed. Moreover, even for different ground truth assessments, the proposed method provides the best results, indicating that the number of clusters (key-frames) does not affect the performance. It must be noted that the weights assigned to the views by the Weighted Multi-View CMM are the same, regardless the number of clusters (key-frames). Thus, the proposed algorithm guarantees that the quality of each view, associated with its weight, is independent of the ground truth assessment. In Table 6, we present the average of the weights assigned to the descriptors by the weighted multi-view CMM algorithm for the two sampling rates. Furthermore, it should be emphasized that the application of the method on the 20% of each video sequence (sampling every five frames) still provides very good results indicating the robustness of the proposed approach.

In Table 4 we present performance results on the first dataset when more than two descriptors are combined at the same time, for both ground truth assessments and for both sampling rates. In what concerns the first ground truth assessment (“GT1”), performance is very good, similar to the performance when pairs of descriptors are used, whereas for the second ground truth assessment (“GT2”), there is a considerable performance improvement for both sampling rates compared to the cases where pairs of descriptors are used. However, the use of many descriptors may increase the computational cost of the method, thus making it inefficient. In other words, a tradeoff between speed and accuracy must be set in such an approach. The cost of using a lot of descriptors could be compensated by using only the 20% of the video sequence (sampling every five frames). Despite frame subsampling, high performance is retained as indicated in Table 4 for $S_r = 5$. In Table 7, we present the average of the weights assigned to the descriptors by the weighted multi-view CMM algorithm for the two sampling rates when more than two descriptors are combined at the same time. Note that, due to the addition of a third relevant descriptor, the HSV weight has been reduced.

Table 4
Performance results using mean and standard deviation of accuracy (m in (%)) using more than two descriptors and two sampling rates ($Sr = 1$ and $Sr = 5$) for the first dataset. Daggers indicate statistical significance.

Ground truth	**GT1**		**GT2**	
Sampling rate	***Sr* = 1**	***Sr* = 5**	***Sr* = 1**	***Sr* = 5**
	SP-UNWEIGHTED			
HSV1D - CEN - WAV	70.02 ± 12.72	71.53 ± 15.20	71.54 ± 17.05	70.15 ± 18.17
HSV1D - CEN - WAV - SIFT20	72.80 ± 07.49	73.32 ± 11.71	72.53 ± 17.33	73.02 ± 17.46
	SP-MULTIVIEW			
HSV1D - CEN - WAV	**82.83 ± 13.06 †‡**	**81.50 ± 12.21 †‡**	**84.25 ± 11.57 †‡**	**80.80 ± 12.07 †‡**
HSV1D - CEN - WAV - SIFT20	**82.83 ± 13.06 †‡**	**82.43 ± 12.64 †‡**	**85.17 ± 11.85 †‡**	**81.73 ± 12.56 †‡**

Table 5
Performance results using mean and standard deviation of CUS_A and CUS_E measures (in %) for the second and third datasets. Daggers indicate statistical significance.

Descriptors	**Second dataset**		**Third dataset**	
	CUS_A	CUS_E	CUS_A	CUS_E
	SP-SINGLE			
HSV1D	73.28 ± 11.87	49.78 ± 26.67	57.58 ± 16.23	51.86 ± 29.53
CEN	73.20 ± 11.81	49.82 ± 28.92	56.74 ± 16.76	52.70 ± 32.67
WAV	69.98 ± 12.70	53.06 ± 29.67	57.22 ± 17.50	52.18 ± 31.09
SIFT50	66.70 ± 14.37	56.44 ± 31.75	54.44 ± 17.34	54.68 ± 31.67
	SP-UNWEIGHTED			
HSV1D - CEN - WAV	73.94 ± 13.27	49.16 ± 29.19	57.44 ± 16.57	52.14 ± 31.56
HSV1D - CEN - WAV - SIFT50	74.36 ± 13.70	48.78 ± 29.80	57.22 ± 14.89	52.28 ± 30.17
	SP-MULTIVIEW			
HSV1D - CEN - WAV	**76.02 ± 13.21 †‡**	**47.12 ± 27.43**	**59.34 ± 15.78** $\bar{\dagger}\,\bar{\ddagger}$	**50.14 ± 30.24**
HSV1D - CEN - WAV - SIFT50	**76.38 ± 12.23 †‡**	**46.68 ± 27.55**	**58.66 ± 15.37** $\bar{\dagger}\,\bar{\ddagger}$	**50.78 ± 29.56**

Table 6
Average weight values (π^1 and π^2) assigned to views V_1 and V_2 for two sampling rates ($Sr = 1$ and $Sr = 5$) for the first dataset.

Sampling rate		***Sr* = 1**		***Sr* = 5**	
V_1	V_2	π^1	π^2	π^1	π^2
HSV1D	**SIFT20**	0.90	0.10	0.79	0.21
HSV1D	**SIFT50**	0.93	0.07	0.85	0.15
HSV3D	**SIFT20**	0.88	0.12	0.80	0.20
HSV3D	**SIFT50**	0.93	0.07	0.84	0.16
HSV1D	**CEN**	0.58	0.42	0.52	0.48
HSV1D	**WAV**	0.72	0.28	0.69	0.31
HSV3D	**CEN**	0.53	0.47	0.49	0.51
HSV3D	**WAV**	0.72	0.28	0.65	0.35

Table 7
Average weight values assigned to views for two sampling rates ($Sr = 1$ and $Sr = 5$) when more than two descriptors (views) are combined at the same time for the first dataset.

Sampling rate		***Sr* = 1**	***Sr* = 5**	***Sr* = 1**	***Sr* = 5**
Descriptors	**Weights**	**3 views**		**4 views**	
HSV1D	π^1	0.44	0.38	0.43	0.37
CEN	π^2	0.38	0.40	0.36	0.34
WAV	π^3	0.18	0.22	0.18	0.21
SIFT20	π^4	–	–	0.03	0.08

Table 8
Average weight values assigned to views for the second and third datasets when more than two descriptors (views) are combined at the same time.

		Second dataset		**Third dataset**	
Descriptors	**Weights**	**3 views**	**4 views**	**3 views**	**4 views**
HSV1D	π^1	0.20	0.19	0.25	0.23
CEN	π^2	0.73	0.73	0.71	0.68
WAV	π^3	0.07	0.06	0.04	0.04
SIFT50	π^4	–	0.02	–	0.05

In Table 5 we present performance results using mean and standard deviation of CUS_A and CUS_E measures on four single descriptors and their combination for the second and third datasets. In must be noted that the proposed method does not estimate the number of extracted key-frames but takes it as granted. For this reason, we set the number of key-frames per video equal to the number of key-frames extracted from "VSUMM1" approach presented in [8]. Moreover, we set the sampling rate to one frame per second (we sample every 30 frames) [8]. Furthermore, in contrast with [8] we do not attempt to eliminate meaningless frames, usually caused by fade in/out effects. This is done in order not to affect performance, since the main research objective in this work is to explore whether the combination of image descriptors with different weights performs better that single descriptors and their combination with equal weights. It can be observed that in both datasets the proposed methodology provides better results compared to all single view experiments and the method with equal weights. Dagger (†) and double dagger (‡) superscripts denote that the proposed method has a statistically significant difference from all single experiments and the corresponding experiment with equal weights, respectively, according to t-test (the significance level is taken as 0.05). A line above these symbols denotes statistically significant difference when significance level is taken as 0.1. In Table 8, we present the average of the weights assigned to the descriptors by the weighted multi-view CMM algorithm for the second and third datasets.

4.4. Visual examples

In Fig. 4, Fig. 5 and Fig. 6 we present three examples of the extracted key-frames for two of the videos of the first dataset and one video of the second dataset, respectively. In the first example (Fig. 4), there are five lights that initially are turned on, then

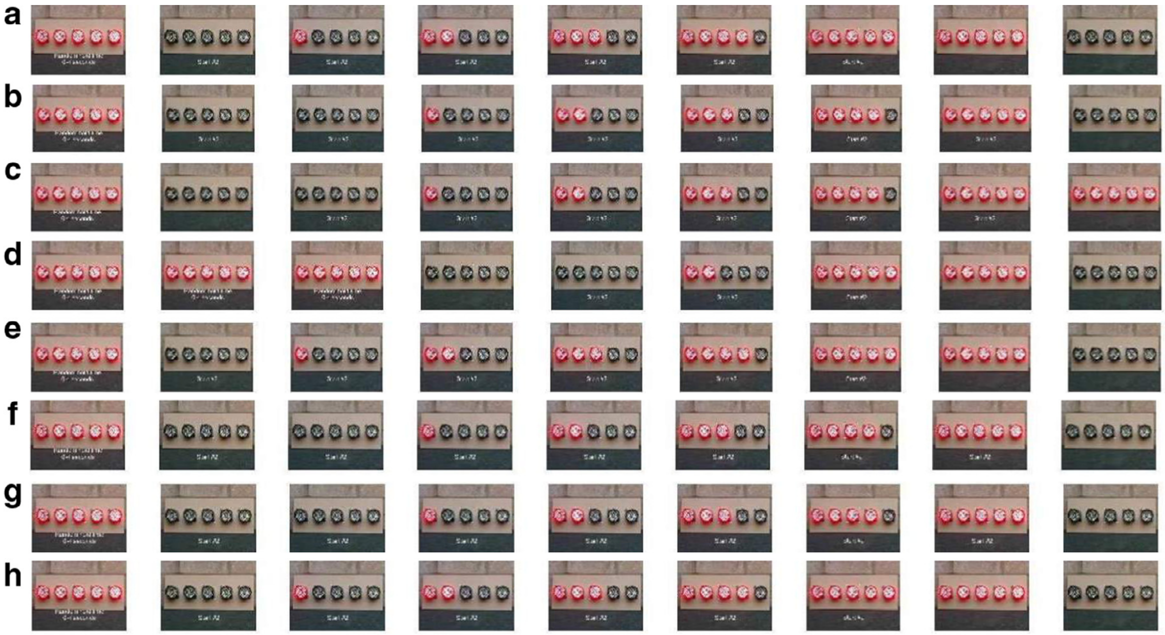


Fig. 4. First visual example (better seen in color). (a) Ground truth. Solutions using: (b) HSV3D, (c) SIFT50 (d) Centrist (CEN), (e) Wavelets (WAV), (f) HSV3D - SIFT50, (g) HSV3D - CEN, (h) HSV3D - WAV. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

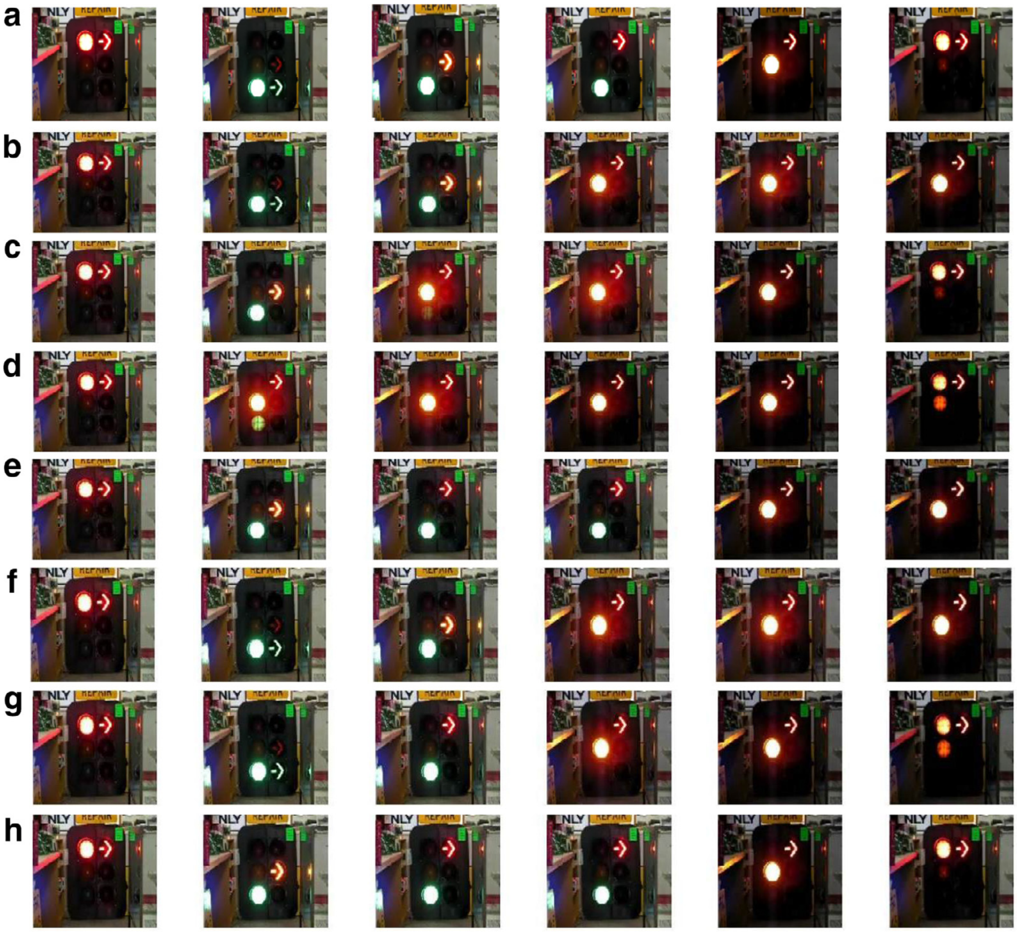
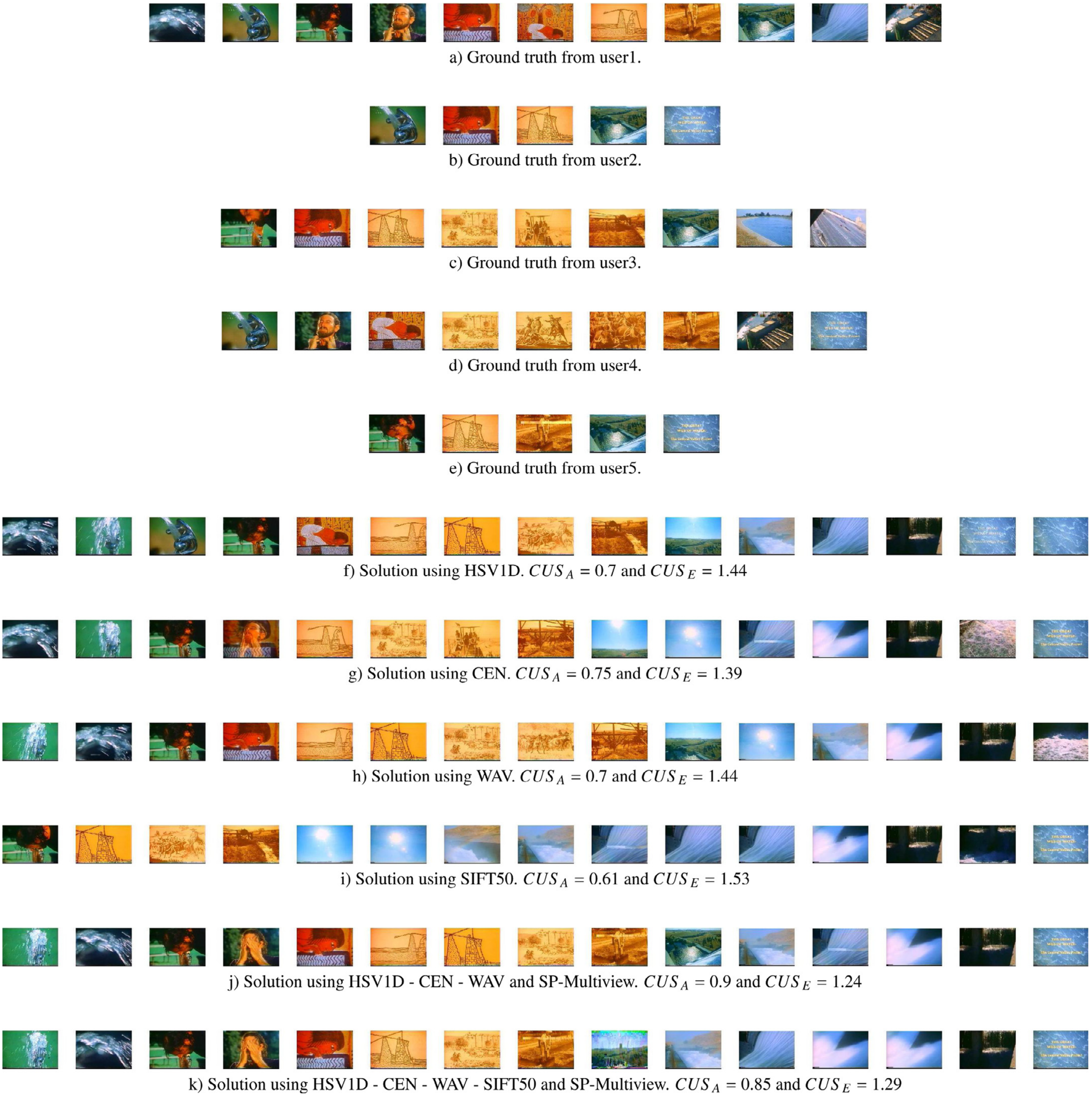


Fig. 5. Second visual example (better seen in color). (a) Ground truth. Solutions using: (b) HSV3D, (c) SIFT50 (d) Centrist (CEN), (e) Wavelets (WAV), (f) HSV3D - SIFT50, (g) HSV3D - CEN, (h) HSV3D - WAV. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



a) Ground truth from user1.

b) Ground truth from user2.

c) Ground truth from user3.

d) Ground truth from user4.

e) Ground truth from user5.

f) Solution using HSV1D. $CUS_A = 0.7$ and $CUS_E = 1.44$

g) Solution using CEN. $CUS_A = 0.75$ and $CUS_E = 1.39$

h) Solution using WAV. $CUS_A = 0.7$ and $CUS_E = 1.44$

i) Solution using SIFT50. $CUS_A = 0.61$ and $CUS_E = 1.53$

j) Solution using HSV1D - CEN - WAV and SP-Multiview. $CUS_A = 0.9$ and $CUS_E = 1.24$

k) Solution using HSV1D - CEN - WAV - SIFT50 and SP-Multiview. $CUS_A = 0.85$ and $CUS_E = 1.29$

Fig. 6. Third visual example (better seen in color). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

they all turn off and start turning on one by one until all traffic lights are turned on again and the turn off again (the ground-truth key-frames are presented in the first row). The next four rows present the key-frames extracted when only one descriptor is employed: (b) HSV3D (8/9 correct key-frames), (c) SIFT20 (7/9 correct key-frames), (d) Centrist (CEN) (6/9 correct key-frames), (e) Wavelets (WAV) (9/9 correct key-frames). Next three rows present the key-frames extracted when pairs of descriptors are employed: (f) HSV3D - SIFT50 (8/9 correct key-frames, π^1=0.95 and π^2=0.05), (g) HSV3D - CEN (8/9 correct key-frames, π^1=0.98 and π^2=0.02) and (h) HSV3D - WAV (9/9 correct key-frames, π^1=0.45 and π^2=0.55). It can be observed that when HSV3D is used with SIFT50 or Centrist it acquires very large weight, thus key-frames are similar to those obtained when only HSV3D is employed. On the other hand, when HSV3D is combined with WAV, their weights have comparable values indicating these are the most relevant descriptors for the specific video sequence.

In the second example (Fig. 5), there are three circle lights (red, orange, green) and three arrow lights (red, orange, green). Ground truth key-frames for this video sequence are presented in the first row. The next four rows provide the extracted key-frames when only one descriptor is employed: (b) HSV3D (4/6 correct key-frames), (c) SIFT50 (4/6 correct key-frames), (d) Centrist (CEN) (3/6 correct key-frames), (e) Wavelets (WAV) (4/6 correct key-frames). Next three rows present the key-frames extracted when pairs of descriptors are employed: (f) HSV3D - SIFT50 (4/6 correct key-frames, π^1=1.00 and π^2=0.00), (g) HSV3D - CEN (5/6 correct key-frames, π^1=0.27 and π^2=0.73) and (h) HSV3D - WAV (5/6 correct key-frames, π^1=0.11 and π^2=0.89). It is obvious that the combination of HSV3D with Centrist or Wavelets yields a better solution.

In the third example (Fig. 6), v21 (The Great Web of Water, segment 01) from the second dataset (Open video project) is used. The first five rows provide ground truth key-frames as selected form five different users [8]. The next four rows provide the extracted key-frames when only one descriptor is employed: (f) HSV1D ($CUS_A = 0.7$ and $CUS_E = 1.44$), (g) Centrist (CEN) ($CUS_A = 0.75$ and $CUS_E = 1.39$), (h) Wavelets (WAV) ($CUS_A = 0.7$ and $CUS_E = 1.44$), (i) SIFT50 ($CUS_A = 0.61$ and $CUS_E = 1.53$). Next two rows present the key-frames extracted when proposed method is employed: (j) HSV1D - CEN - WAV ($CUS_A = 0.9$ and $CUS_E = 1.24$, π^1=0.11, π^2=0.64 and π^3=0.25), (k) HSV1D - CEN - WAV - SIFT50 ($CUS_A = 0.85$ and $CUS_E = 1.29$, π^1=0.18, π^2=0.49, π^3=0.23 and π^4=0.1). It can be observed that the proposed method performs better than single descriptors.

5. Conclusions

A key-frame extraction method has been proposed capable of combining different image descriptors (views). In this way a significant problem is tackled, since the large variations observed in the visual content of videos make inefficient the use of a specific single descriptor. Our method builds upon a weighted multi-view clustering algorithm based on Convex Mixture Models and demonstrates the important property that the weight of each descriptor is automatically estimated to reflect its importance for a specific video sequence. After the weights have been computed, a similarity matrix is built as a weighted sum of the individual kernels corresponding to each descriptor. Finally a spectral clustering algorithm is applied using this similarity matrix to cluster the frames of a shot into groups, from which the representative key-frames (cluster medoids) are extracted. Performance results on several video sequences (also supported by visual examples) indicate that our method efficiently summarizes video shots regardless of the visual content and the combination of image descriptors employed. This is mainly due to its ability to assign high weight to relevant descriptors and almost zero weight to irrelevant ones.

In future work, we aim to perform additional experiments using other types of kernels, creating synthetic kernels for each descriptor as a weighted combination of base kernels and using visual vocabularies of different size. It is also interesting to replace the spectral clustering part of our approach with a segmentation-based technique that would take as input the weighted similarity matrix. Finally, another future research direction would be to enhance the proposed methodology using clustering criteria so as to automatically estimate the number of clusters (key-frames).

Acknowledgments

The work described in this paper is co-financed by the European Regional Development Fund (ERDF) (2007–2013) of the European Union and National Funds (Operational Programme Competitiveness and Entrepreneurship (OPCE II), ROP ATTICA), under the Action “SYNERGASIA (COOPERATION) 2009”.

References

[1] W. Wolf, Key frame selection by motion analysis, in: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP '96, 1996, pp. 1228–1231.

[2] G. Ciocca, R. Schettini, An innovative algorithm for key frame extraction in video summarization., J. Real-Time Image Process. 1 (1) (2006) 69–88.

[3] C. Panagiotakis, A.D. Doulamis, G. Tziritas, Equivalent key frames selection based on iso-content principles., IEEE Trans. Circuits Syst. Video Technol. 19 (3) (2009) 447–451.

[4] G. Guan, Z. Wang, S. Lu, J.D. Deng, D.D. Feng, Keypoint-based keyframe selection, IEEE Trans. Circuits Syst. Video Technol. 23 (4) (2013) 729–734.

[5] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[6] Y. Zhuang, Y. Rui, T. Huang, S. Mehrotra, Adaptive key frame extraction using unsupervised clustering, in: Proceedings of the International Conference on Image Processing, 1, 1998, pp. 866–870.

[7] Z. Rasheed, M. Shah, Detection and representation of scenes in videos., IEEE Trans. Multimed. 7 (6) (2005) 1097–1105.

[8] S.E.F. de Avila, A.P.B.a. Lopes, A. da Luz Jr., A. de Albuquerque Araújo, VSUMM: a mechanism designed to produce static video summaries and a novel evaluation method, Pattern Recognit. Lett. 32 (1) (2011) 56–68.

[9] R.V. Martn, A. Bandera, Spatio-temporal feature-based keyframe detection from video shots using spectral clustering., Pattern Recognit. Lett. 34 (7) (2013) 770–779.

[10] Z. Cernekova, I. Pitas, C. Nikou, Information theory-based shot cut/fade detection and video summarization, IEEE Trans. Circuits Syst. Video Technol. 16 (1) (2006) 82–91.

[11] A. Girgensohn, J.S. Boreczky, Time-constrained keyframe selection technique., Multimedia Tools Appl. 11 (3) (2000) 347–358.

[12] R. Panda, S. Kuanar, A. Chowdhury, Scalable video summarization using skeleton graph and random walk, in: Proceedings of the 22nd International Conference on Pattern Recognition (ICPR), 2014, pp. 3481–3486.

[13] Y. Cong, J. Yuan, J. Luo, Towards scalable summarization of consumer videos via sparse dictionary selection, IEEE Trans. Multimed. 14 (1) (2012) 66–75.

[14] S. Mei, G. Guan, Z. Wang, S. Wan, M. He, D.D. Feng, Video summarization via minimum sparse reconstruction, Pattern Recognit. 48 (2) (2015) 522–533.

[15] G. Tzortzis, C.L. Likas, Multiple view clustering using a weighted combination of exemplar-based mixture models., IEEE Trans. Neural Netw. 21 (12) (2010) 1925–1938.

[16] A. Ioannidis, V. Chasanis, A. Likas, Key-frame extraction using weighted multi-view convex mixture models and spectral clustering, in: Proceedings of the 22nd International Conference on Pattern Recognition (ICPR), 2014, pp. 3463–3468.

[17] D. Lashkari, P. Golland, Convex clustering with exemplar-based models, in: Proceedings of the Advances in Neural Information Processing Systems, 2008, pp. 825–832.

[18] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[19] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Proceedings of the Advances in Neural Information Processing Systems, 2001, pp. 849–856.

[20] V. Chasanis, A. Likas, N. Galatsanos, Scene detection in videos using shot clustering and sequence alignment, IEEE Trans. Multimed. 11 (1) (2009) 89–100.

[21] A. Likas, N. Vlassis, J.J. Verbeek, The global k-means clustering algorithm, Pattern Recognit. 36 (2) (2003) 451–461.

[22] E.J. Stollnitz, T.D. DeRose, D.H. Salesin, Wavelets for computer graphics: A primer, part 1, IEEE Comput. Graph. Appl. 15 (3) (1995) 76–84.

[23] J. Wu, J. Rehg, Centrist: A visual descriptor for scene categorization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1489–1501.

[24] J. Yang, Y.-G. Jiang, A.G. Hauptmann, C.-W. Ngo, Evaluating bag-of-visual-words representations in scene classification, in: Proceedings of the International Workshop on Multimedia Information retrieval, MIR '07, ACM, New York, NY, USA, 2007, pp. 197–206.