Multimed Tools Appl (2009) 44:161–186
DOI 10.1007/s11042-009-0277-9

Combining graph connectivity & dominant set clustering for video summarization

D. Besiris · A. Makedonas · G. Economou · S. Fotopoulos

Published online: 12 May 2009

Abstract The paper presents an automatic video summarization technique based on graph theory methodology and the dominant sets clustering algorithm. The large size of the video data set is handled by exploiting the connectivity information of prototype frames that are extracted from a down-sampled version of the original video sequence. The connectivity information for the prototypes which is obtained from the whole set of data improves video representation and reveals its structure. Automatic selection of the optimal number of clusters and hereafter keyframes is accomplished at a next step through the dominant set clustering algorithm. The method is free of user-specified modeling parameters and is evaluated in terms of several metrics that quantify its content representational ability. Comparison of the proposed summarization technique to the Open Video storyboard, the Adaptive clustering algorithm and the Delaunay clustering approach, is provided.

Keywords Video summary · Prototype set · Connectivity graph · Dominant set

1 Introduction

Recent advances in multimedia technologies have made a large amount of commercial or home videos available to the general public. For the manipulation of the video information researchers developed techniques that are oriented to the production of video abstracts. Video abstract appears as a compact representation of a video sequence and is useful for various video applications as video browsing, indexing and retrieval. A video abstract or summary [3] can be a preview sequence combining a limited number of video segments (video skimming) or a keyframe set properly chosen from the video sequence. Although a keyframe-based video abstract may lose the spatial-temporal properties and the audio content of the original video sequence, it is clearly the simplest method for video representation. The use of a compact set of keyframes reduces the amount of information required in video indexing and provides the framework for dealing with the video content in retrieval applications.

D. Besiris (✉) · A. Makedonas · G. Economou · S. Fotopoulos
Electronics Laboratory, Department of Physics, University of Patras, Rio 26500, Greece
e-mail: dbes@upatras.gr

Among the different methods for the extraction of a keyframe set, the most straightforward approach is by uniformly sampling the video sequence with a certain frame rate. Although this technique has very low complexity, it is possible to disregard a time limited but content essential part of the video sequence. Very often the number of keyframes can be defined a priori according to the requirements of the video application. This approach, so called, rate constraint keyframe extraction, is applicable mostly in mobile communication networks which face bandwidth limitations. For those networks the number of keyframes is allocated according to the storage capacity or the size of the display. On the other hand, the number of keyframes can be set a posteriori or it can be computed automatically according to the video content. Clearly, more keyframes are necessary for the representation of a video with high motion activity than a rather static one.

Earlier efforts in the field of video analysis and summarization were based on frame clustering, relying on the detection of shot boundaries [29] and providing a fixed or variable number of keyframes per shot. The objective of those methods was twofold, maintain the temporal continuity of the extracted keyframes and capture the pictorial information from each shot at the same time. Although their results seem to be good for some genres of video, for others, as films, interviews, athletic events, introduce redundancies since similar pictorial content appear repeatedly in the resulting summary. A solution to this problem was to obtain the video frames using the original video as a whole [12, 32] and cluster the frames that have similar content. This approach places greater burden to the clustering process as the size of the dataset increases considerably.

In line with the above observations, we propose a new approach for automatic video summarization, which is organized around a hybrid frame-based pairwise clustering algorithm. The technique arises as a combination of previous studies on clustering problems [14], [22] over edge-weighted graphs. According to [14], an efficient structural representation of a large dataset can be derived by utilizing the geometrical constraints among a small down-sampled version of the original dataset. The algorithm uses a number of appropriately selected prototypes (considered as cluster centers) to construct their connectivity graph and reveal the video structure. This edge-weighted graph is built by exploiting the membership values of the entire data set towards the prototypes set. The whole procedure in its original form [14] is based on the fuzzy C-means algorithm operating in an over partitioning mode. In the present method a modification is introduced which produce a most representative set of prototypes.

Next, in order to achieve the automatic partitioning of the resulting graph and extract an optimal number of representative keyframes, the robust '*dominant set clustering methodology*' [22] is utilized. According to this work, any pair of prototypes sharing the same content information or having high degree of connectivity is component of the same dominant set (cluster). The technique proceeds by partitioning the prototypes set into coherent groups, through a self-terminating clustering process, defining at each step the corresponding dominant set. The centroids of the dominant sets are selected as key frames, thus formulating the video summary.

The remainder of the paper is organized as follows: Section 2 is a concise report of related works. In Section 3, the proposed method is presented in detail. Experimental results are presented and commented in Section 4, and conclusions are drawn in Section 5.

2 Related works

Previous methods in frame-based video summarization mainly relied in shot boundary detection [2]. A shot is detected when a certain difference measure between consecutive

frames exceeds a threshold. This measure can be computed by using either features containing global information (colour histograms) or more complex features, such as image edges, motion vectors and probability densities. In a very simple approach the first frame [1] or the first and the last frames [23] in each shot are selected as keyframes. Other methods use a certain frame rate in order to extract a down-sample version of video. All these approaches do not consider the dynamics of the visual content or the motion analysis and the type of the shot boundary and they often extract a fixed number of keyframes per shot.

Recent approaches operate on the entire video sequence using techniques like, maximum frame coverage [4], clustering [12, 32], curve simplification [6], SRE [18], motion analysis [7] and interesting events [17]. The maximum frame coverage technique proposed by the Chang et al. [4] is referred as the fidelity based approach. According to this approach each frame is represented in a high dimensional space and the video is viewed as a proximity graph with vertices the frame set of the video segment. The problem can be interpreted as that of finding the minimum fidelity values so that all frames can be represented by the selected set of keyframes. The minimum set-cover problem is a well known NP-complete vertex cover problem, Chang et al., proposed a greedy approach to get an approximate solution. In [28] the coverage of a frame is the number of fixed-size excerpts which contain at least one frame similar to this frame and a dynamic programming procedure is used to select a pre-specified number of frames as the keyframe set. Cooper in [5] assumes the analogy between keyframe extraction and the popular keyword extraction in text information retrieval via the term frequency-inverse document frequency method (TF-IDF).

In clustering-based methods a segment of the video sequence is represented as a set of points in the feature space. The most representative points of the formed clusters are selected as keyframes for the video sequence. Usually the clustering process is implemented in three steps: pre-processing, clustering procedure and filtering. Each step is used to improve the effectiveness of the clustering process. Yeung in [29] introduced a tolerance-band method to extract video keyframes and used it as a pre-processing step for video shot clustering. The tolerance-band method selects the first frame as keyframe. If a subsequent frame is more dissimilar to the existing keyframe according to a designated threshold, this frame is selected as the next keyframe. The process continues until the number of keyframes reaches the desired level. In [25] a set of potential keyframes are selected via the sufficient content change method and are used as input to the clustering algorithm. The clustering procedure is sequentially applied to the video sequence. During the process a frame is assigned to the existing cluster if their similarity is maximal and exceeds a certain threshold. The keyframe set is formulated by the clusters set at the end of the clustering process. The clustering method of [32] requires users to set the maximum cluster size at the beginning of video process. The video frames are grouped into clusters, and the frame that are closest to the cluster centroids are extracted as keyframes. Hanjalic and Zhang in [12] use a partitioning clustering algorithm with cluster-validity analysis to select the optimal number of clusters for each shot. Each frame of the video segment is represented by a cumulative activity level. The keyframe set is selected as the frames located in the middle of the representative range between the breakpoints and the corresponding set of frames in shot. The technique introduced in [10] is based on a hierarchical complete link approach. From the formulated clusters the keyframes are selected only from those that contain at least one uninterrupted nine-second sequence of frames so as to avoid video artefacts. According to the clustering algorithm (GMM) proposed by [9] each frame is transformed into the eigenspace via principal component analysis (PCA). The purpose of this transformation is to perform dimensionality reduction so that the high-dimensional image

space can be represented by a much lower dimension space, whilst retaining the significant variations of the original dataset. The cluster centroids are the centers of each component and are selected to form the keyframe set. In Gong [11] the video summarization is accomplished with a clustering algorithm based on singular value decomposition method (SVD). The video frames are time sampled and visual features are computed. The refined feature vectors are clustered, and a keyframe is extracted from each cluster. Finally the method presented in [31] uses a hierarchical clustering algorithm based on spectral clustering that merges similar frames into clusters in a tree-structured representation with individual frame at the leaves.

The curve simplification approach is related to the clustering method as each frame is represented in the feature space. However, this technique searches for a set of points, such as, that the removal of the remaining points does not affect the basic shape of the curve connecting all points according to the temporal order in the video sequence. In [6] the keyframe extraction problem is considered as fitting the curvature of point trajectory in the feature space, in which the curvature characteristic frames are iteratively selected as keyframes. The approach introduced in [15] uses a standard curve simplification algorithm, namely, discrete contour evolution. The keyframe set is constructed through a polygon simplification procedure, where its frame is appointed as a vertex.

Clustering techniques used in the above works usually rely on user-defined specifications parameters. Most of the clustering techniques require either a predefined number of clusters or a fixed parameter threshold value. Since, these parameters are mostly found experimentally their adjustment is expensive and inefficient for a large dataset. It is some of these shortcomings that the present work aims to tackle.

3 Our method

The proposed video summarization system composes of three stages and is illustrated in Fig. 1. The first stage is a pre-processing one where the original video sequence is down-sampled. Frames are selected by uniformly sampling the initial video sequence at a constant frame-rate. In the second stage, a specific number of *prototype* vectors is selected that are subsequently used to cluster the remainder of the video sequence. The key idea is to reveal the topology of the original video sequence based on the selected prototypes, invoking fuzzy logic procedures. The prototype's set serves as a simplified representation of the whole video sequence and the Fuzzy C-Means clustering methodology is employed to provide the corresponding membership values among all of its elements. These membership values are further processed to compute the connectivity matrix and construct the

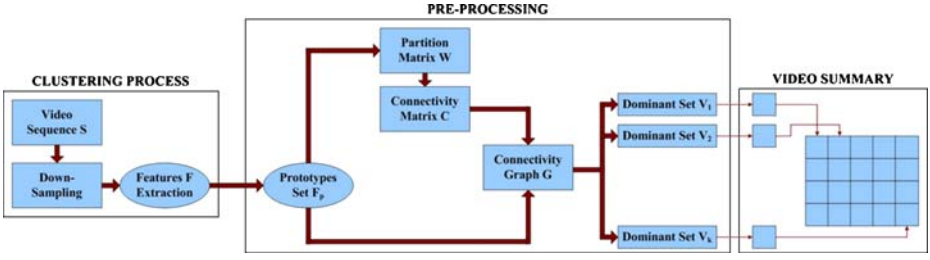

Fig. 1 The structure of the video summarization system

corresponding connectivity graph, which describes the important pairwise relations among prototypes. Afterwards and in order to achieve an optimized grouping of prototypes, the dominant sets among prototypes are extracted through a pairwise clustering graph-based procedure [22]. In the third and final stage, from each dominant set the centroid vector is extracted as keyframe.

3.1 Pre-processing

The video is considered as a discrete sequence of L consequent static images (frames), i.e., $S = \{F(i)|1 \leq i \leq L\}$, where $F(i)$ is the ith frame in the sequence. In order to exploit the inherent temporal redundancy and reduce the required computational time and complexity of the clustering process the video is down-sampled at a frame-rate R. The resulting video sequence$S' \subseteq S$, comprises of N frames. The frame-rate R can be defined a priori or can be set a posteriori according to the duration of the video [11, 21]. Experimental results indicate that, for a value in the range $R \in [1, 30]$ the quality of the video summary is only slightly affected by the pre-sampling process. An indication is given in Fig. 2 where video representation in the 2D (2-dimensional) reduced feature space is given both, in the original form and in its down-sampled version. It is obvious that the topology of the data set is maintained. This is due to the nature of the video, which conveys a great deal of redundant information as most time adjacent frames are quite identical.

The down-sampled version $S'$of video sequence is represented in a d-dimensional feature space by a set of vectors, $F = \{f_i|_{1 \leq i \leq N}\}$, where $f_i = \{f_{ij}|_{1 \leq j \leq d}\}$, is the corresponding feature vector of the ith frame of the video sequence. The feature component f_{ij} represents the jth attribute associated to vector f_i. The feature selected in this study is a 24 bin HSV colour histogram, where the first 16 bins refer to *Hue*, while the remaining 8 bins are equally distributed to *Saturation* and *Value*, respectively.

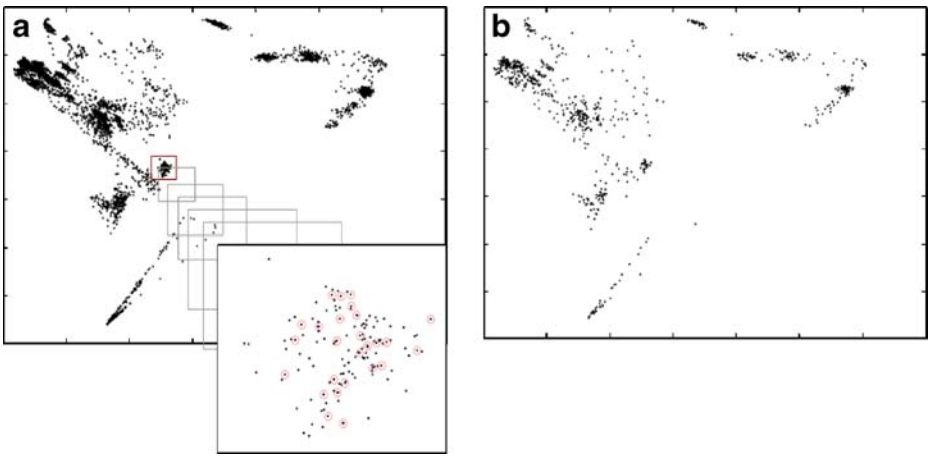

Fig. 2 **a** Representation of the original video sequence S (in the box is indicated a segment of the original dataset and the selected data through the down-sampling procedure, marked with *red circle*). **b** Representation of the down-sampled version of the video sequence S'(the frame-rate is R=5), in the $\Re^2$ feature space. Both biplots were produced in the reduced feature space using PCA analysis. The structural similarity of the two data representations is obvious

3.2 Clustering process

Clustering the frames of a video sequence is an essential part in video summary. Clustering data generally is not an easy task and in our case the situation is more complex due to the high structural organization of the video data. The large number of frames and the high dimensionality adds to the burden of the clustering process. To tackle all these issues and to produce automatically the correct number of clusters the process adopts a graph-based methodology which is accomplished in two steps.

3.2.1 Partition matrix—prototypes set

Using as input the set of vectors F from the pre-processing stage, some *prototype* vectors are extracted. Their number is on purpose selected higher than the true number of classes in the set. Although, these prototypes (due to the over-determined number) do not correspond to the correct classification scheme, however, they can serve as "markers", used to reveal the correct organization of the dataset and guide the final classification stage. The objective of this section is to organize prototypes in a graph structure that reveals their topology and existing connectivity information between them. It should be noticed that this organization is accomplished using the whole data set F so that the new structure retains global information while at the same time is more efficient due to much reduced number of data elements.

The key idea is to compute the connectivity among prototypes utilizing the membership values of all vectors in set F with regard to the prototypes set. This process originates from a recently presented data structure learning methodology [14], which as starting point uses the well known Fuzzy C-Means (FCM) clustering algorithm.

In the present work, the iteration step of the FCM algorithm for the selection of cluster centers is ignored and the prototypes are selected by uniformly sampling the down-sampled version of the video sequence F. This modification of the original algorithm was found necessary due to the strong data-density variation in the feature space of the video sequences. The selected prototypes set in our case, unlike FCM, do not necessarily correspond to high density areas. However there are certain advantages associated with our approach i) time proximity of prototypes is preserved ii) the property of closeness is maintained (the prototypes are subset of the original video sequence) iii) low density areas are well represented in the prototype's set, especially those corresponding to small duration shots.

The number of the selected prototypes n must be sufficiently large for the competent representation of the entire video sequence. A relatively small number of prototypes may not represent the true clusters contained in the dataset, while a large number of prototypes might leave insufficient number of test vectors to run the process and reveal the true connectivity structure. It should be noticed here that, although, in our method the prototypes are not produced from the FCM algorithm the subsequent step of the algorithm that computes membership values is utilized.

Let us denote $F_p = \{f_{pi}|_{1\le i\le n}\}$ the selected set of prototype vectors and $F_d = \{f_{dj}|_{1\le j\le N-n}\}$ the remaining vectors of the initial set F, where $F = F_p \cup F_d$ and $f_{pi} \neq f_{dj}$, $\forall i,j$. Each prototype vector f_{pi} is compared to the vectorsf_{dj}, providing the corresponding dissimilarity matrix $D = [d_{ij}]_{n\times(N-n)}$, i.e.

$$d_{ij} = \sum_{k=1}^{d} \left\| f_{pik} - f_{djk} \right\|^2 \quad , \ \forall\, i \in [1,\ n] \text{ and } \forall\, j \in [1, N-n] \tag{1}$$

where d denotes the dimension of the selected feature space.

The partition matrix is computed from the dissimilarity matrix, as:

$$w_{ij} = \frac{1}{\sum_{k=1}^{n} \left\{ \frac{d_{ij}}{d_{kj}} \right\}^2} \tag{2}$$

The resulting matrix $W = \left[w_{ij}\right]_{n\times(N-n)}$ depicted in Fig. 3, is a $n\times(N\text{-}n)$ matrix comprising the membership value of each vector f_{dj} with regard to the set of prototype vectors F_p, with $0\leq w_{ij}\leq 1$ and $\sum_{i=1}^{n} w_{ij} = 1$, where:

$$W = \left[w_{ij}\right]_{n\times(N-n)} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N-n} \\ w_{21} & w_{22} & \cdots & w_{2N-n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nN-n} \end{bmatrix}$$

3.2.2 Connectivity graph

The partition matrix $W = \left[w_{ij}\right]_{n\times N-n}$ contains information regarding the relative proximity of each vector to the prototypes set F_p. The membership values indicate the strength of the relationship between the data element j and a particular prototype i, which is considered as a cluster centre. The assignment of data elements to one or more clusters based on the partition matrix $W = \left[w_{ij}\right]_{n\times N-n}$ is an essential process in fuzzy clustering. The information included in the partition matrix is very rich and can be used advantageously to detect existing similarities between prototypes and subsequently compute their connectivity. Looking in one column j of the partition matrix illustrated in Fig. 3, it is seen that each element j has stronger relations with a certain number of prototypes. This observation could be also translated as an indication that those prototypes may share some common characteristics and belong to the same class.

In compact formulation the connectivity strength c_{ij} is computed from the partition matrix W as a point-to-point correlation of the membership values [14], where

$$c_{ij} = \sum_{k=1}^{N-n} w_{ik} w_{jk} \tag{3}$$

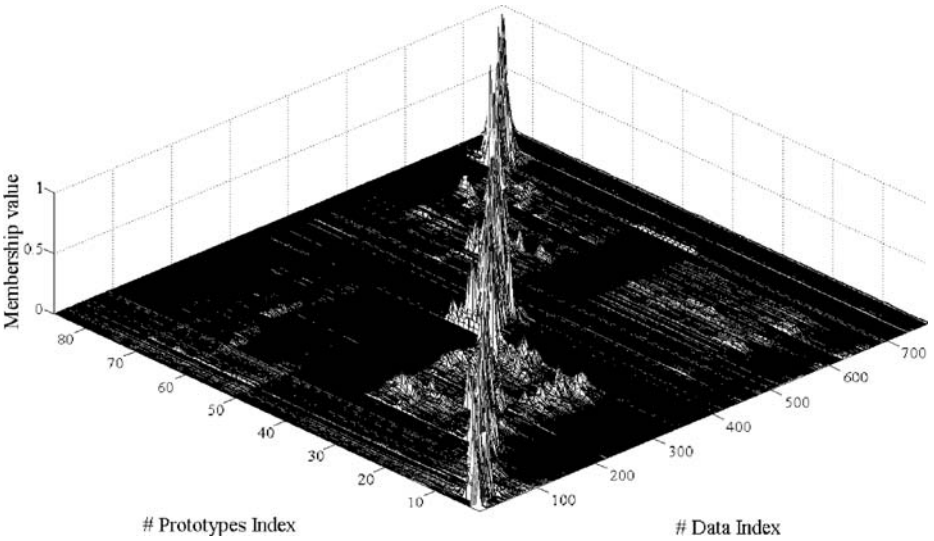

Fig. 3 Visualization of the partition matrix W for a certain video data set

The derived connectivity matrix $C = [c_{ij}]_{n\times n}$ is a symmetric $n\times n$ matrix. An example is given in Fig. 4, for the same data set of Fig. 3.

Having produced the matrix C we construct the corresponding connectivity graph $G=(V, E,a)$, where $V = \{1,\ldots,n\}$ is the vertex set, $E \subseteq V \times V$ the edge set, and $a : E \rightarrow \Re^+$ is the weight function. Vertices of G denote the prototypes, while edge-weights reflect the connectivity strength between pairs of linked vertices. The graph G is represented by the corresponding weighted similarity matrix A, where $A = [a_{ij}]_{n\times n}$ with entry values computed as:

$$a_{ij} = \begin{cases} c_{ij},\ if\ i \neq j \\ 0,\ otherwise \end{cases} \tag{4}$$

i.e., matrix A is symmetric and all elements on the main diagonal (self-loops) are zero. The resulting connectivity graph is a fully connected undirected graph, and is illustrated in Fig. 5(a).

In order to emphasize the strong connectivity relations only, a thresholding scheme has been introduced in the partition matrix. Membership values below a threshold value τ are zeroed and do not contribute in the estimation of the correlations. Using this modification the connectivity values are re-estimated based on the new membership values, as:

$$c_{ij}' = \sum_{k=1}^{N-n} w_{ik}' w_{jk}' \tag{5}$$

where w_{ij}'corresponds to the re-estimated membership value: $w_{ij}' = w_{ij} \cdot \theta(w_{ij} - \tau)$, with $\theta(\cdot)$ the step function. Non zero values of C' indicate now a strong connection for the corresponding pair of prototypes while all zero values are denoted in the produced connectivity graph G by the absence of the corresponding edges, Fig. 5(b).

When the threshold value is increased the number of connecting edges decreases, resulting to the partitioning of the connectivity graph, as it is illustrated in Fig. 5(b–d). Although this partitioning based on threshold value τ could be efficiently utilized to

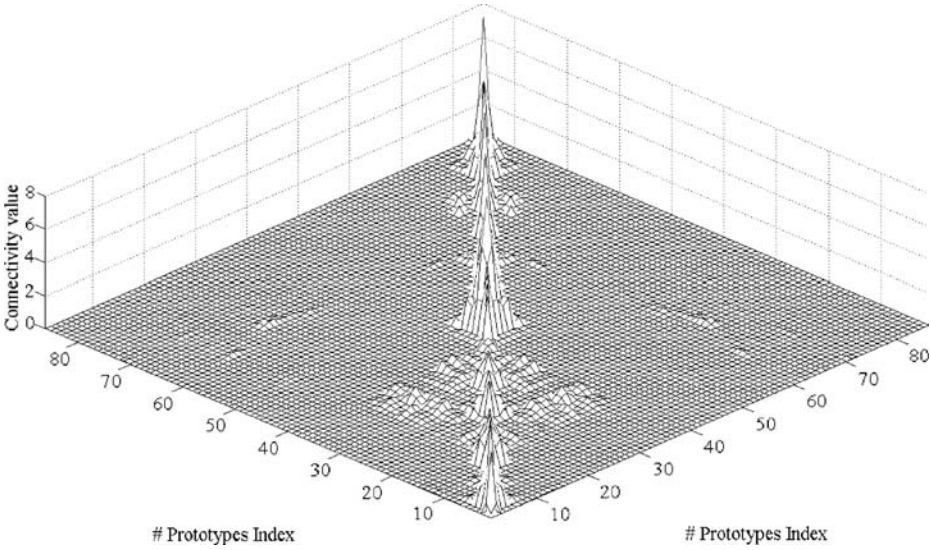

Fig. 4 Visualization of the connectivity matrix of the prototype set

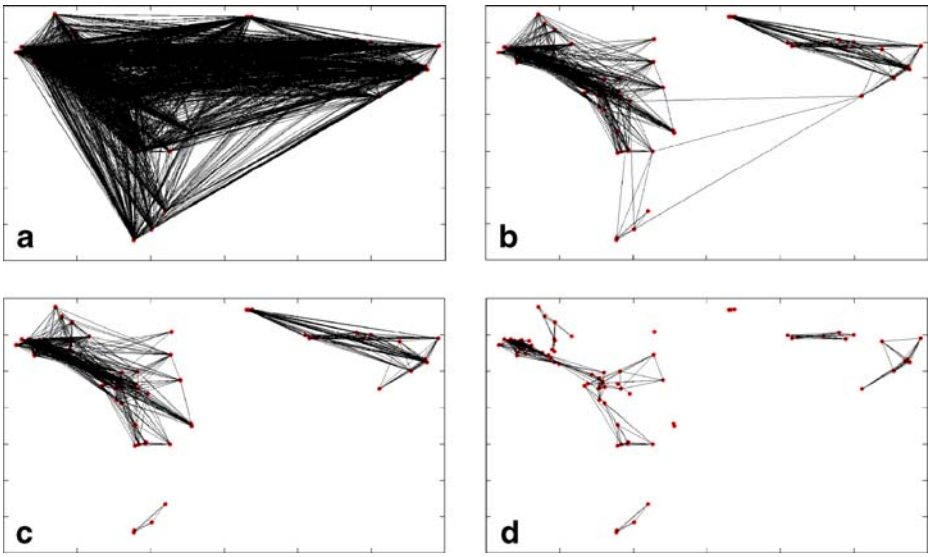

Fig. 5 **a** Visualization of the connectivity graph G, **b** the connectivity graph after the deletion of the weakest edges based on the estimation of the critical threshold value and **c**–**d** the connectivity graph with the selection of a relatively large threshold value (with *red circles* are denoted the prototypes and with solid black lines the corresponding edges)

produce the appropriate number of classes the more robust "dominant set" clustering algorithm [22] has been adopted here. In order to apply this algorithm, described in the next paragraph, the graph should be connected. A self-terminating algorithm to compute the critical threshold value τ, which keeps the graph connected, is presented in Table 1.

3.2.3 Dominant sets extraction

The dominant sets extraction step that follows serves to partition the created connectivity graph and determine the true prominent classes of the video sequence.

Let us consider that the connectivity graph G can be partitioned into m disjoint sub-graphs, each one comprising a set of vertices, $V = V_1 \cup V_2 \cup \cdots V_m$, where $V_i \cap V_j = \emptyset$ $\forall\ (i, j) \in [1, m]$. The objective in the partitioning process is to identify the most cohesive sets of vertices, denoted as *dominant* sets, based on the information provided by the weighted similarity matrix A.

The dominant set is defined as the cluster comprising of a group of vertices with: a) high intra homogeneity and b) high inter in-homogeneity. For the set of vertices in the graph this is equivalent to the selection of the group of vertices with large weight values for edges within the cluster and small weight values for edges connecting different clusters. So, a high quality cluster is the one where elements are strongly associated with large similarity values in matrix A.

According to a recently introduced graph-theoretic algorithm [22] the extraction of the most cohesive group is equivalent to the maximization of the following objective function (*cohesiveness* function)

$$J(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i a_{ij} x_j \tag{6}$$

Table 1 The algorithm for the computation of the critical threshold value

Initialize the threshold value
$\tau \leftarrow 0$
The procedure continues until at least one node is disconnected
while $\min_i \left(\sum_{j=1}^{n} \alpha_{ij}' \right) = n-1$
For the current threshold value compute the re-estimated connectivity c_{ij} *and weight values* α_{ij} *according to Eq.(5) and Eq.(4). For each node, search all possible node connections*
for i=1 to n
for j=1 to n
For each pair of node compute the adjacency weight value and entry a new edge connection

$$\alpha_{ij}' = \alpha_{ji}' = \begin{cases} 1 & , if\ \alpha_{ij} \neq 0 \\ 0 & , otherwise \end{cases}$$

end for
end for
Increase the threshold value by a fixed value and repeat the previous procedure
$\tau \leftarrow \tau + 0.001$
end while

α_{ij}' is the corresponding adjacency weight of graph G. It is used only during the critical threshold computation.

according to the *participation vector* **x** [20], subject to

$$x = \{x_i|_{1 \le i \le n}\} \text{ with } x_i \ge 0 \text{ and } \sum_{i=1}^{n} x_i = 1, \forall i \tag{7}$$

Each component of vector **x** express the participation degree of the corresponding vertex to the cluster. If a component has a small value, then the corresponding vertex is weakly associated with the cluster, whereas if it has a large value, the vertex is strongly associated with it. For those vertices that do not participate in the cluster the corresponding values are zero.

A straightforward and efficient way to find the local solution of the quadratic problem in Eq. (6) was given by the so-called *replicator dynamics* [24]. According to this approach the participation vector can be computed through a simple recursive process, according to the following model:

$$x_i^{(t+1)} = x_i^{(t)} \cdot \frac{x_i^{(t)} \sum_{k=1}^{n} a_{ik} x_k^{(t)}}{\sum_{k=1}^{n} \sum_{j=1}^{n} x_k^{(t)} a_{kj} x_j^{(t)}} \tag{8}$$

where $x_i^{(t)}$ denotes the value of the ith component of the participation vector **x** during the tth iteration of the process. After a fixed number of iterations, the support of **x** is computed providing the set of vertices participating (all vertices corresponding to no zero values) in the dominant sub-graph, as described in Table 2.

The full partitioning of the graph is accomplished by repeating the previous procedure. At each step, the algorithm extracts the dominant set of vertices corresponding to the

Table 2 The algorithm for the extraction of the dominant set

Initialize the clustering procedure
t←1
Set the participation vector $\mathbf{x}^{(t)}$ *as the simplex barycenter*
$x_i^{(t)} = 1/n$, for i∈[1, *n*]
Normalize the participation vector $\boldsymbol{x}$ *based on the restriction in Eq.(7) and compute the cohesiveness function* $J\left(\mathbf{x}^{(t)}\right)$ *according to Eq.(6)*
while $J\left(\mathbf{x}^{(t+1)}\right) \geq J\left(\mathbf{x}^{(t)}\right)$
Compute the re-estimated participation vector $\mathbf{x}^{(t+1)}$ *and the corresponding cohesiveness function* $J\left(\mathbf{x}^{(t+1)}\right)$ *(the participation vector must first be normalized)*
The procedure continues until the cohesiveness function stables to a fixed value
t←t+1
end while
Define the dominant set according to the arguments of the estimated cluster vector $\mathbf{x}$

current sub-graph, as resulted through the bipartition process. The current group of vertices is removed from the initial graph and the formed sub-graph is used as input the next iteration of the algorithm. The algorithm with regard to the participation vector **x**, computes the corresponding cohesiveness function. Groups of vertices with cohesiveness value smaller than a certain threshold value are not accepted. The algorithm terminates when the threshold value is reached. This threshold value is defined as:

$$J_T(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} e_i a_{ij} e_j \tag{9}$$

where **e** is a n-dimensional vector consisting of unit entries (normalized in order to fulfill the restriction, introduced in Eq. (7)). The threshold value is a measure of the cohesiveness of the initial set of vertices in the connectivity graph. The whole procedure is described in Table 3.

The algorithm is self-terminating according to the threshold value, in Eq. (9). Any unprocessed vertex is assigned to the nearest dominant set according to its connectivity value.

3.2.4 Keyframes extraction

The video summary is easily derived from the partitioned connectivity graph. For each individual subgraph the dominant vertice is extracted, identified as the centroid of the corresponding cluster, i.e.

$$v_i = V_i(k) = \underset{k}{argmax} \left\{ \sum_{k=1}^{n_i} [C(V_i(k), V_i)] \right\} \tag{10}$$

where v_i is the corresponding dominant node of the dominant set V_i, n_i is the number of vertices comprising in the *i*th cluster, and $C(\cdot)$ denotes the connectivity measure between the *k*th node set and the dominant set V_i.

The resulting dominant node-set $v = \{v_i|_{1 \leq i \leq m}\}$ contains the centroids of the identified clusters. We select to appoint the unitary sets directly to the nearest dominant node-set, as they comprise of only one vertex and it is unlike to represent meaningful keyframes.

Table 3 The algorithm for the extraction of the dominant sets

Initialize the clustering procedure
$i \leftarrow 1$
$\mathbf{x}' = \mathbf{x}$
Compute the threshold value $J_T(\mathbf{x})$ *according to Eq.(9)*
while $J(\mathbf{x}') \geq J_T(\mathbf{x})$
Identify the dominant set of the graph, based on the procedure in Table 2
Extract the corresponding set of vertices V_i *from the initial vertex set V*
$V_i \cap V = \varnothing$
The procedure continues until the cohesiveness function reaches the predefined threshold
Compute the corresponding cohesiveness value $J(\mathbf{x}')$ *and repeat the procedure*
$i \leftarrow i + 1$
end while

$x'$is a subset of the participation vector **x**, provided after the extraction of the dominant set V_i from the original vertex set V.

4 Experimental study

4.1 Evaluation criteria

One of the most difficult tasks in the field of the video abstraction is the evaluation of the produced abstracts over a video sequence. This is because, unlike other vision research areas such as object detection and recognition, the evaluation of a video abstract is not a straightforward task due to the lack of an objective criterion (ground-truth). Much of the problem comes from the absence of standardized metrics, due to the disparities between feature-based low level analysis and higher level semantics of the video content. In most cases, it is difficult for someone to decide if one video abstract is better than another, making the summarization task application-dependent.

Several evaluation techniques have been proposed during the last years. In its simple form, the testing method is applied to a few video sequences and the resulting abstract is described or discussed [12, 30, 32]. Other evaluation procedures rely on the opinion of a panel of users judging the quality of the generated video summaries. In [8] each keyframe is classified as “good”, “fair”, or “poor” according to the original video sequence. In a similar but in a more systematic and subjective study Liu et al. [16] set the evaluation framework based on a large group of testers. Each tester has the ability to assign a score, denoted as “good”, “acceptable”, or “bad”, regarding the extracted keyframe set. These scores are then used to evaluate the proposed technique on a large collection of videos of variable genres, as news, sports and home movies. Although this is probably the most realistic approach of evaluation, especially when keyframes are extracted for user-based tasks, it could not be easily employed to a wide range of video applications, since it is very difficult to determine the parameters of the experiment.

On the other hand, the selection of objective criteria which can be applied automatically to all video sequences without the need of video experts is more attractive. According to [13] a video abstract should maintain three attributes in order to provide an effective video representation: *conciseness*, *comprehensive coverage* and *coherence*. Conciseness is straightforward associated to the length of the produced video summary. Comprehensive coverage ensures that the selected key-frame set can efficiently represent the visual diversity of the video sequence. Coherence is associated to the consecutiveness of the

produced abstract. This attribute is usually addressed to video applications which maintain the temporal continuity and the dynamic of the video sequence. Based on these observations, we select four metrics, in order to evaluate the results of a video summary: *Fidelity, Compression rate*, *Recall* and *Precision.*

Fidelity is used as a measure of the *comprehensive coverage* of a video. It is based on the metric of the semi-Hausdorff distance, first introduced by Chang et al. [4]. The Fidelity measure is computed as the maximum of the minimum distances between the keyframe set and the frames in the original video sequence, i.e.

$$d_f = \max_i \left(\min_j \left(d_{ij} \right) \right), \ \forall i \in [1,\ N], \forall j \in [1,\ m] \tag{11}$$

where $d_{ij} = D\left(f_i, Kf_j\right)$ is a dissimilarity measurement between the *i*th frame of the video sequence and the *j*th keyframe of the video summary. Usually, the Fidelity is computed in its normalized form, as:

$$Fidelity = 1 - \frac{d_f}{\max_i \left(\max_j \left(d_{ij} \right) \right)} \tag{12}$$

High Fidelity values indicate that the extracted keyframe set provides a good global description of the visual content of the video sequence.

Compression rate is used as a measure of video *conciseness*. It is computed as:

$$CR = \frac{m}{N} \tag{13}$$

where m is the number of keyframes and N is the total number of frames in the original video. This metric gives an indication of the size of the summary with respect to the size of the original video.

Recall is used as a measure of the *coherence* of a video. It is defined as:

$$Recall = \frac{N_c}{N_h} \tag{14}$$

where N_c is the number of the correctly detected shots and N_h the number of shots annotated by human subjects and used in the experimentation as ground-truth. Although, it is reported mainly for the evaluation of shot-based algorithms, it is applicable to this work too.

Precision is introduced as a measure of the optical *comprehensiveness* of a video. It is defined as:

$$Precision = \frac{N_c}{N_c + N_m} \tag{15}$$

where N_c is the number of the correctly detected shots and N_m the number of false detected shots, corresponding to gradual transition segments in the video sequence like fade-in/out, wipes and dissolves.

4.2 Algorithms tested

We have compared the results of our method with three other keyframe extraction methods: 1) Adaptive unsupervised clustering algorithm (ADC) [32], 2) Delaunay clustering algorithm (DCA) [21] and 3) Open Video Project (OVP) website results in [26].

The ADC algorithm is a dynamic clustering method. It works in a sequential way, clustering the video frames according to a user-defined threshold value. At each step of the algorithm the entry frame is compared to the centroids of the existing clusters and it is appointed to the most similar one, according to a threshold value. If not, it is appointed to a new cluster. Prior to the next step, the algorithm refines the clusters centroids and repeats the procedure for all frames in the video sequence. When the clusters are formed, the frame which is closest to the cluster centroid is selected as the keyframe.

The DCA algorithm uses the Delaunay Triangulation in order to cluster the frames of the video sequence in a fully automatic way obviating any need for parameter definition, as the previously described ADC algorithm does. Initially each frame is represented in a multidimensional feature space (256 HSV colour histogram). By using PCA analysis it reduces the dimensions of the feature vectors, which serve as nodes for the corresponding Delaunay diagram. The clusters are formulated from the partition of the diagram by removing the separating edges. The frame that is nearest to the center of each cluster is selected as keyframe.

The Open Video Project (OVP) is a valuable resource, providing the storyboard results for a very large number of videos. Each video summary is generated using the algorithm from [6] together with some manual intervention that refines results. The overall procedure is described in [19]. It is well suited for video summary applications and it is referenced by many published works.

4.3 Theoretical complexity

The theoretical complexity of the three keyframe extraction algorithms is shown in Table 4. All costs are computed using the specified algorithms' parameters. The OVP algorithm is not included, since the keyframe set is provided directly the corresponding website. The complexity was computed considering logical and mathematical operations, all with unit cost. We have not taken into account memory usage or the cost required to decode a frame as in each algorithm the pre-processing step is ignored. Let N, the number of the processed video sequence (we consider the same size for all algorithms), n the number of prototypes selected in our algorithm (due to sampling n<<N) and m the number of the extracted keyframes. The complexity is relative to the number of operations required during the clustering procedure. Although the construction of the Delaunay diagram of the DCA algorithm is fast, the PCA pre-processing phase penalises the algorithm. More than half of the operations required by our algorithm are used to compute the connectivity threshold. By employing a fixed threshold value the complexity is reduced to $O(n^2)$.

4.4 Video dataset

We tested our method on 20 selected video segments, each of length more than 2 min, pertaining to documentaries, as shown in Table 5. The video segments were MPEG compressed and downloaded from the Open Video Project's shared digital video repository

Table 4 Complexities of the keyframe extraction algorithms tested in this work

Algorithm	Complexity
Our method	$O(nN+n^2)$
ADC	$O(n\text{-}1)$
DCA	$O(N\log N+N^2)$

Table 5 The video set used in the experimental results

VideoName	Video File	Duration (mm:ss)	Resolution (W×H)	TNF	NS	NF
Nasa 25th Anniversary Show, Segment 3	anni003	02:22	320×240	4,267	27	2
Nasa 25th Anniversary Show, Segment 4	anni004	02:09	320×240	3,895	19	5
Exotic Terrane, Segment 4	UGS01_004	02:40	352×240	4,797	23	5
Exotic Terrane, Segment 7	UGS01_007	03:06	352×240	5,601	23	13
Exotic Terrane, Segment 10	UGS01_010	02:13	352×240	3,999	20	7
Exotic Terrane, Segment 11	UGS01_011	02:00	352×240	3,606	18	15
America's New Frontier, Segment 4	UGS02_004	02:03	352×240	3,705	10	0
America's New Frontier, Segment 5	UGS02_005	02:40	352×240	4,896	22	1
America's New Frontier, Segment 6	UGS02_006	04:49	352×240	8,670	20	3
America's New Frontier, Segment 7	UGS02_007	02:00	352×240	3,615	19	5
America's New Frontier, Segment 8	UGS02_008	04:13	352×240	7,608	23	5
America's New Frontier, Segment 9	UGS02_009	03:49	352×240	6,879	20	1
America's New Frontier, Segment 10	UGS02_010	02:41	352×240	4,830	9	1
Ocean Floor Legacy, Segment 3	UGS07_003	02:38	352×240	4,749	18	6
Ocean Floor Legacy, Segment 5	UGS07_005	02:35	352×240	4,665	25	4
The Future of Energy Gases, Segment 4	UGS03_004	04:27	352×240	8,007	28	5
The Future of Energy Gases, Segment 6	UGS03_006	02:02	352×240	3,660	19	7
The Future of Energy Gases, Segment 10	UGS03_010	02:51	352×240	5,142	9	5
Moon, Segment 2	moon002	03:43	320×240	6,709	22	26
New Indians, Segment 11	indi011	04:14	320×240	7,640	51	2

TNF is the total number of frames in the video sequence.

NS is the number of shots according to the ground-truth.

NF is the number of transitional regions (fade-in/out, wipes, dissolves).

[26]. They were first decompressed using the official MPEG codec from MPEG Software Simulation Group [27]. In Table 5 we also present the number of shots that are used as ground-truth during the computation of Recall and Precision along with the number of transitional regions in the video sequence.

Seven out of twenty videos are selected in order to test the ability of our algorithm to exclude duplicate keyframes from the video summary. These videos are presented in Table 6, along with the number of duplicate shots presented in the corresponding video sequence.

Table 6 The seven videos present duplicate shots in the video sequence

Video File	anni004	UGS01 007	UGS02 004	UGS02 006	UGS02 008	UGS02 009	UGS02 010
NR	3	2	2	2	3	2	1

NR is the total number of duplicate shots in video.

4.5 Experimental results

We evaluated the proposed system on two aspects: i) the computational performance of the tested algorithms and ii) applying the four criteria presented in Section 4.1. For both experiments, the parameters set of the algorithms was adjusted based on the reported values in the original papers. For the ADC algorithm, the parameter of the threshold value that controls the density of clusters was adjusted in order to achieve the same number of keyframes as the OVP algorithm. On the other hand since both, our algorithm and the DCA method extracts the keyframes in a totally automatic way, the results depend on both the number of keyframes extracted and the selection of the processing technique.

In our technique frame rate (pre-processing stage) is set to $R=5$, while prototypes are selected by uniformly sampling the down-sampled version of the video sequence. In order to evaluate the number of prototypes n, an experiment is conducted. For each video the set of prototypes is selected by using a fixed sampling-rate in the range [2–20]. The produced key frame set is evaluated by using the Recall measure. The scope of the experiment is to measure the influence of the selected number of prototypes to the performance of the system, i.e., to the coverage ability based on the identified set of different shots in the video. The results, as depicted in Fig. 6, present the average Recall measurement per step value for all the 20 videos of the dataset. As the results indicate, there is a reduction in the Recall value as the sampling-rate increases, especially for values greater than 10 (Recall<0.95). This is due to the reduction on the representative ability of the system. In line with this observation, we selected a sampling-rate of one prototype per ten frames.

4.5.1 Computational time

Table 7 shows the computational time of the algorithms tested. All algorithms were implemented in Matlab 6.5 development environment with the default optimization turned on. The computer used for the comparison was an Intel Pentium (4) 3 GHz with 512 MB of RAM and running under the Windows XP 2002 Professional operating system. The pre-processing step in each algorithm was omitted. A test session was performed by processing all videos sequences under the same variable definition (the length of the processed video

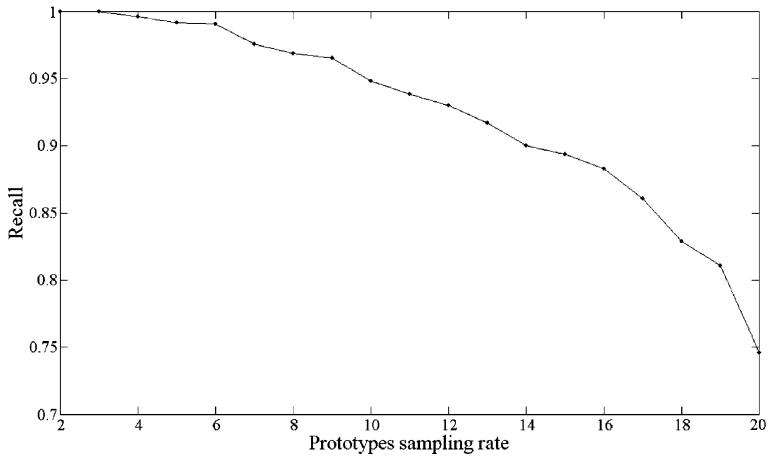

Fig. 6 Recall value versus prototypes sampling rate

Table 7 Computational time of the video set, reported in seconds and thousands of seconds

Video file	Our method		ADC	DCA
	Threshold computation	Fixed threshold		
anni003	3:03	0:38	1:18	4:63
anni004	3:22	0:34	1:43	4:01
UGS01_004	2:72	0:70	0:79	5:17
UGS01_007	5:44	1:03	1:56	6:07
UGS01_010	3:03	0:36	0:69	4:32
UGS01_011	2:56	0:36	0:58	3:81
UGS02_004	5:73	1:83	0:48	4:22
UGS02_005	3:45	0:53	0:75	5:21
UGS02_006	10:18	2:98	1:31	9:67
UGS02_007	2:62	0:34	0:84	3:93
UGS02_008	7:35	1:92	2:73	8:26
UGS02_009	8:51	2:15	1:17	7:50
UGS02_010	3:15	0:75	0:80	5:23
UGS07_003	3:59	0:83	1:22	5:07
UGS07_005	3:53	0:55	0:57	5:02
UGS03_004	8:53	2:03	2:10	8:69
UGS03_006	2:83	0:48	0:72	3:98
UGS03_010	3:44	1:08	1:31	5:65
moon002	6:52	2:01	2:23	7:88
indi011	13:55	4:28	3:47	8:39

sequence) repeated for three times. For each video the computational time reported refers to average time of the three test sessions. The OVP was not included, since its results are provided in the corresponding website page [26], obviating any need for simulation in the current work.

As it is shown from testing results, the computational time of the DCA algorithm depends mostly on the length of the processed video. On the other hand, our algorithm and the ADC algorithm depend on the number of the resulting clusters along with the length of the video sequence, since both algorithms include an iterative stage which refines the value of the cohesiveness function and the centroid point of the clusters. The most time consuming part in our algorithm proved to be the computation of the critical threshold value related to the connectivity matrix. Replacing that component with a fixed threshold value the computational time increases its speed up to 313%. Comparing the simulation results with the other two algorithms we observe an improvement up to 4% and 368% with respect to the ADC and DCA algorithm, respectively.

4.5.2 Evaluation metrics

Figure 7 summarizes the results for the Fidelity measure. In order to make a fair comparison, all four algorithms were tested under the same feature, the 256 HSV colour histogram. During these experiments two different measures were computed: overall Fig. 7(a) and local Fidelity Fig. 7(b).

The overall Fidelity computes the overall content coverage of the video sequence by the keyframe set. As experimental results indicate, our algorithm presents better results in 16 out of 20 tested videos (shown in bold). For the rest four videos, OVP gains higher Fidelity

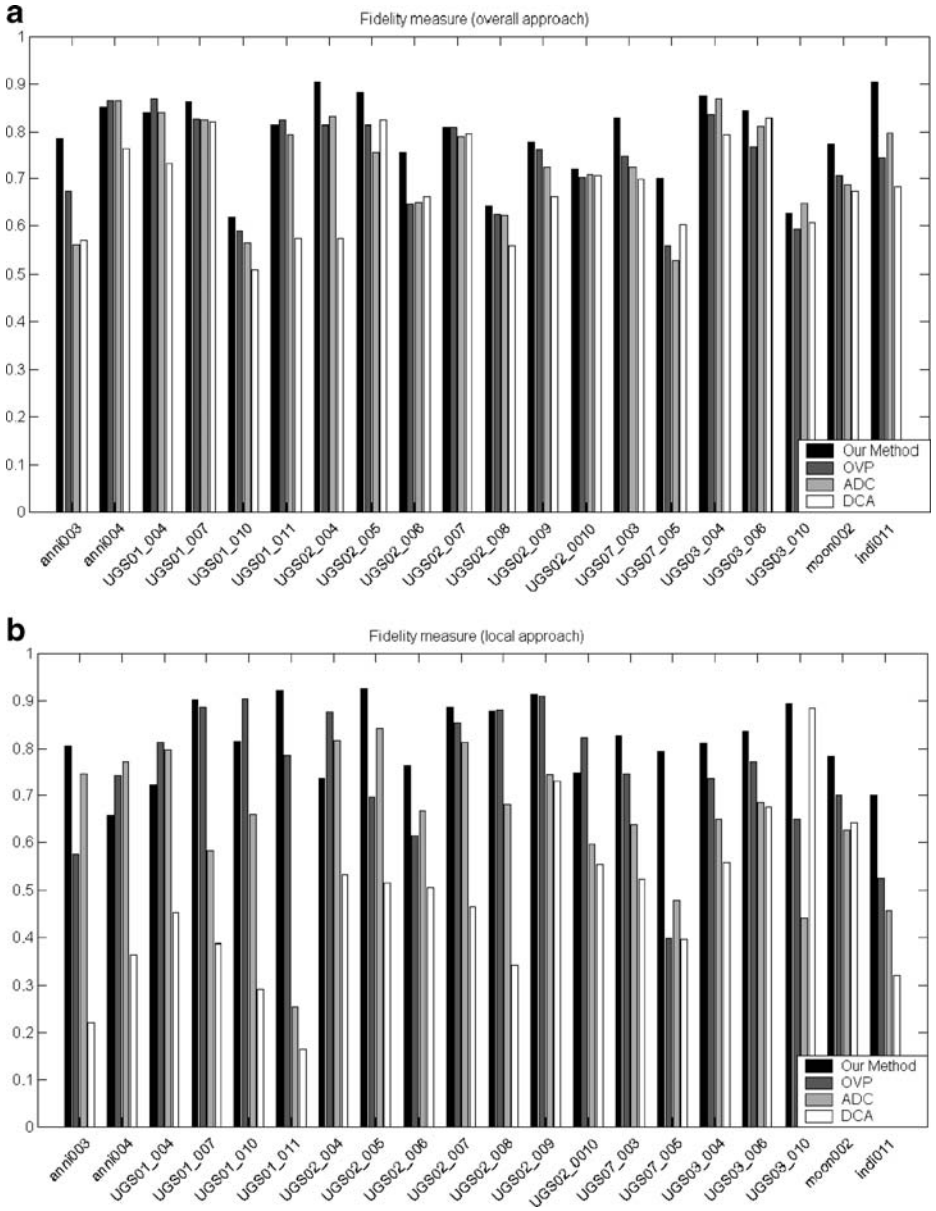

Fig. 7 Comparison of the video summarization algorithms via the overall and local Fidelity measure

value in three of them (anni004, UGS01_004 and UGS01_011), while the ADC algorithm is better only in one video (UGS03_010). We noticed that both OVP and ADC algorithm produce almost the same Fidelity quality. This is not strange, as both algorithms share the same number of keyframes. On the other hand, DCA algorithm produces the smallest Fidelity values in almost all tested videos, compared with the other three methods. On average, our algorithm presents an improvement of up to 6.35%, 7.65% and 15.15% with regard to the OVP, ADC and DCA algorithms. Especially, in videos: anni003, UGS02_004,

UGS02_006, UGS07_003, UGS_007_005 and indi011 this improvement is increased up to 20%. In detail we summarize the average, the minimum and the maximum detected Fidelity values of four algorithms in Table 8, along with the corresponding videos.

Since, keyframes extracted from different shots may have similar feature representation but different pictorial content (semantic information), we must take this into account by computing the corresponding local quality. For each shot the Fidelity measure is computed for the corresponding keyframes. If an algorithm does not extract keyframes from a shot, the corresponding Fidelity measurement is set to the worst case value (zero). The final local Fidelity measure is computed as the average of all Fidelity measures for the shots and is presented in Fig. 7.

As experimental results indicate, the efficiency decreases presenting better results in 14 out of 20 tested videos, with regard to the overall Fidelity measurement. The quality of the OVP is increased by two videos (notice that the maximum local Fidelity value is not presented for the same videos compared with the overall Fidelity value), while the ADC is better only in one video. Although the average improvement is still up to ~10% compared to the other summarization techniques the ability of detecting representative frames in each shot is much lower, compared to the ability of the global representation of the video sequence. This is partly because our algorithm has the tendency to ignore duplicate frames that appear in different but pictorial identical shots. Although this improves the comprehensive coverage of the video sequence it cannot be clearly evident by the use of the specific metric. In Table 9 we summarize the average, the minimum and the maximum detected local Fidelity values of four algorithms, along with the corresponding videos. All measurements indicate that the performance of our algorithm is better.

Table 10 summarizes the compression rate results. The first column presents the corresponding number of shots (and the corresponding refined values after the detection of duplicate shots); the next three columns indicate the number of the extracted keyframes per algorithm while the final three columns portray the computed compression rate. The number of shots is a basic indication of the correct number of keyframes. The selection of one keyframe per shot is considered as the most efficient technique since it maintains the conciseness of the video summary while at the same time it captures the comprehensive information of the video segment.

As experimental results indicate, our algorithm produces on average seven additional keyframes, OVP and ADC produce two additional keyframes, while DCA produces five less keyframes, all compared to the ground-truth (NS). On average, our algorithm presents

Table 8 Comparison of the video summarization algorithms via the overall Fidelity measure

Algorithms	AVF	MAXF	MINF	Video
Our Method	0.7852	0.9036		UGS02_004
			0.6180	UGS01_010
OVP	0.7383	0.8671		UGS01_004
			0.5598	UGS07_005
ADC	0.7294	0.8675		UGS03_003
			0.5282	UGS07_005
DCA	0.6819	0.8269		UGS03_006
			0.5083	UGS01_010

AVF is the average Fidelity value.

MAXF is the maximum Fidelity value.

MINF is the minimum Fidelity value.

Table 9 Comparison of the video summarization algorithms via the local Fidelity measure

Algorithms	AVF	MAXF	MINF	Video
Our Method	0.8156	0.9251		UGS02_005
			0.6570	anni004
OVP	0.7442	0.9080		UGS02_009
			0.3978	UGS07_005
ADC	0.6474	0.8431		UGS02_005
			0.2542	UGS01_011
DCA	0.4759	0.8832		UGS03_010
			0.1643	UGS01_011

AVF is the average Fidelity value.

MAXF is the maximum Fidelity value.

MINF is the minimum Fidelity value.

an increase up to 25% and 88% of the compression rate, with regard to the OVP/ADC and the DCA algorithm, respectively. Although this penalizes the conciseness of the produced video summary, it describes analytically the details included in high motion segments of a video sequence. As it can be seen in Fig. 8, our storyboard result (the keyframe set is re-organized according to the index of each frame in the video sequence) contains more semantically rich information since: a) describes a larger number of shots (without

Table 10 Compression-rate comparison of the video summarization algorithms

Video file	NS	Number of keyframes			CR (%)		
		Our method	OVP ADC	DCA	Our method	OVP ADC	DCA
anni003	27	32	26	7	0.75	0.61	0.16
anni004	19 (16)	27	29	13	0.69	0.74	0.33
UGS01_004	23	28	28	16	0.58	0.58	0.33
UGS01_007	23 (21)	31	24	26	0.55	0.43	0.46
UGS01_010	20	27	21	11	0.68	0.53	0.28
UGS01_011	18	21	18	9	0.58	0.50	0.25
UGS02_004	10 (8)	20	13	10	0.54	0.35	0.27
UGS02_005	22	35	23	15	0.71	0.47	0.31
UGS02_006	20 (18)	42	37	14	0.48	0.43	0.16
UGS02_007	19	22	22	17	0.61	0.61	0.47
UGS02_008	23 (20)	32	28	18	0.42	0.37	0.24
UGS02_009	20 (18)	26	24	19	0.38	0.35	0.28
UGS02_010	9 (8)	22	15	8	0.46	0.31	0.17
UGS07_003	18	20	19	14	0.42	0.40	0.30
UGS07_005	25	31	12	16	0.66	0.26	0.34
UGS03_004	28	40	30	24	0.50	0.37	0.30
UGS03_006	19	29	20	15	0.80	0.55	0.41
UGS03_010	9	13	9	13	0.25	0.18	0.25
moon002	22	26	24	18	0.39	0.36	0.27
indi011	51	42	29	20	0.55	0.38	0.26

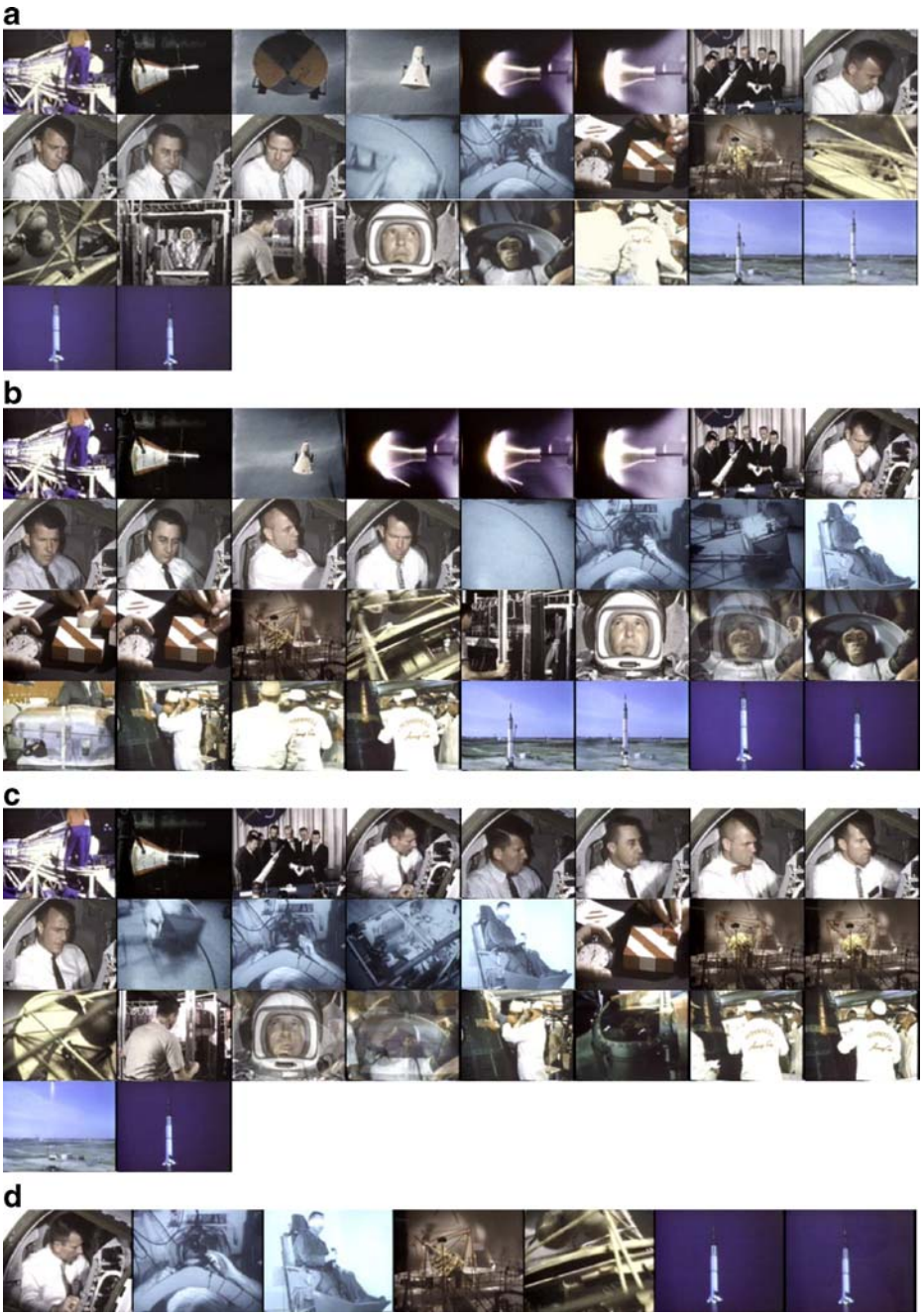

Fig. 8 The keyframe sets extracted by the tested algorithm, for the video file anni003 **a** OVP, **b** our method, **c** ADC and **d** DCA

duplicates) and b) provides an additional keyframe in segments of the video sequence with high motion activity. It should be noticed that ADC is adjusted in order to produce the same number of keyframes as that provided by the OVP storyboard.

Figure 9 summarizes the results of the studied algorithms for Recall and Precision. For both metrics our algorithm gives better results an indication that it is able to detect all the correct keyframes (high Recall value), while at the same time avoids transitional frames (high Precision).

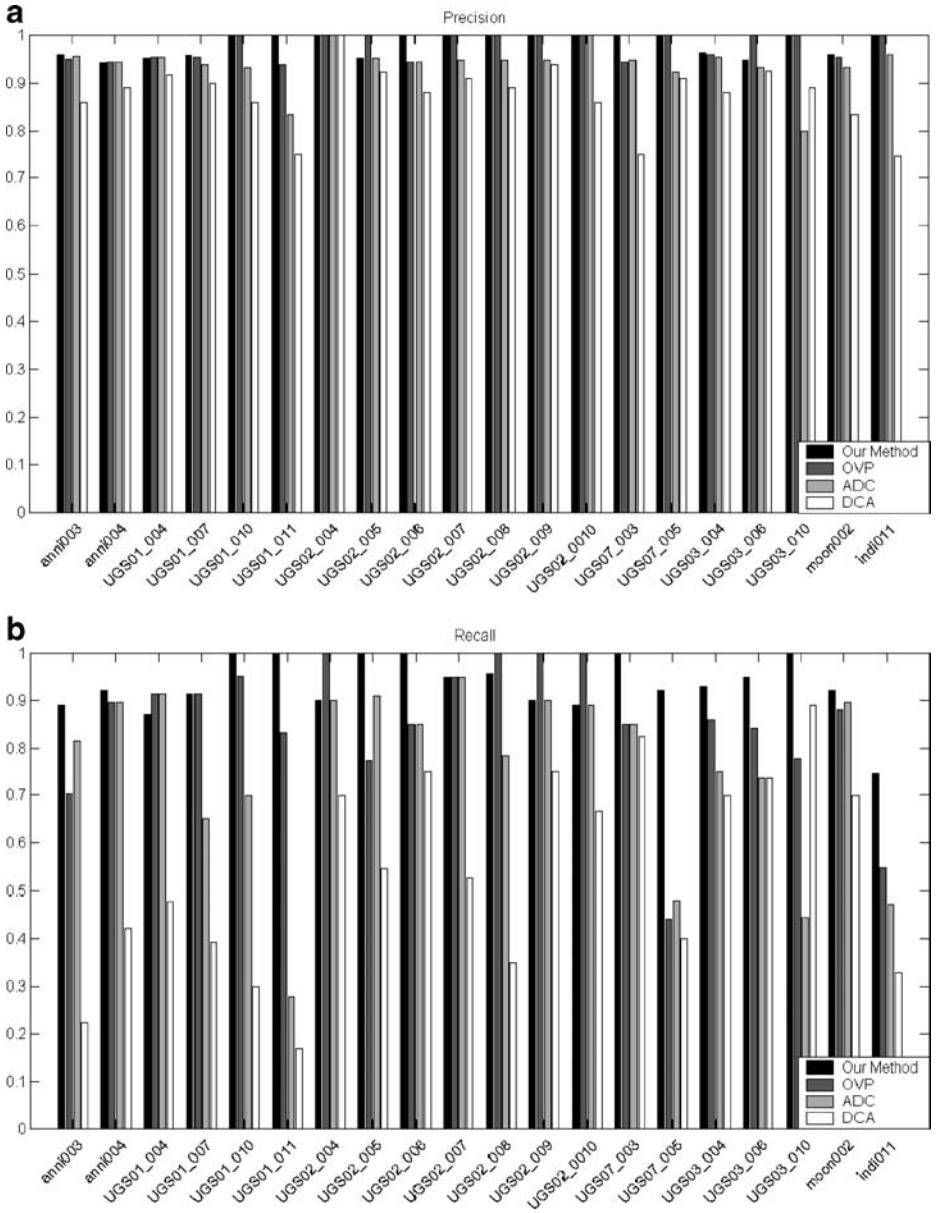

Fig. 9 Comparison of the video summarization algorithms via Precision and Recall

In detail we summarize the average, the minimum and the maximum computed Precision and Recall values of the four algorithms in Table 11. In all cases our algorithm produces better results.

5 Conclusion and future work

In this paper, we proposed an automatic graph-based video summarization technique. This technique generates video summaries by exploiting the connectivity matrix of selected number of prototypes derived by a down-sampled version of the original video sequence. Fuzzy logic formalism is used for the connectivity matrix computations. This process projects to the prototypes the associated connectivity information of the whole data set and reveals its structure. The essential step of cluster extraction is accomplished by the robust action of the dominant set clustering algorithm. The method is free of user-specified modeling parameters and all used thresholds are extracted automatically during the clustering process according to the inherent characteristics of the video data set.

It is well suited for long time videos, obviating any need for shot boundaries detection. Extensive comparisons of the proposed algorithm to OVP storyboard, the adaptive clustering algorithm (ADC) and the Delaunay clustering algorithm (DCA) have been carried out by employing metrics such as, fidelity measure (overall and local), compression rate, recall and precision. All these metrics evaluate the representational ability of the proposed summarization technique. The advantage of our method for batch processing of large videos regarding the processing time is also demonstrated.

In future work, we plan to apply the dominant clustering approach using several additional features besides the HSV color histogram, such as, text, audio and motion features. As it can be seen by our results the clustering performance the algorithm is particularly good in selecting frames with high motion activity and revealing video segments with high semantic information. Furthermore, using the proposed clustering technique as the core layer in an automatic video processing architecture, many other content analysis-based applications can be designed such as, video indexing, searching and retrieval.

Table 11 Comparison of the video summarization algorithms via Precision and Recall

Algorithms	AVP	MAXP	MINP	AVR	MAXR	MINR
Our Method	0.9817	1	0.9412	0.9329	1	0.7451
OVP	0.9772	1	0.9375	0.8487	1	0.4400
ADC	0.9375	1	0.8000	0.7529	0.9474	0.2778
DCA	0.8747	1	0.7454	0.5421	0.8889	0.1667

AVP is the average Precision value.

MAXP is the maximum Precision value.

MINP is the minimum Precision value.

AVR is the average Recall value.

MAXR is the maximum Recall value.

MINR is the minimum Recall value.

Acknowledgments This work was financed by the European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II), and particularly the Program "New graduate programs of University of Patras".

References

1. Behzard S, Gibbon DS (1995) Automatic generation of pictorial transcripts of video programs. Proc SPIE Multimedia Computer Networking 2417:512–518
2. Boreczky JS, Rowe LA (1996) Comparison of video shot boundary detection techniques. Proc Int Conf Storage Retr Still Image Video Databases 5(2):170–179
3. Bovic AC (2000) Handbook of image and video processing. Bovic Academic Press 2000 9(2):705–715
4. Chanq HS, Sull S, Lee SU (1999) Efficient video indexing scheme for content-based retrieval. IEEE Trans Circ Syst Video Tech 9(8):1269–1279. doi:10.1109/76.809161
5. Cooper M, Foote J (2005) Discriminative techniques for keyframe selection. IEEE Int. Conf Multimedia and Expo (ICME) 502–505
6. DeMenthon D, Doermann DS, Kobla V (1998) Video summarization by curve simplification. Proc. ACM Multimedia 211–218
7. Divakaran A, Radhakrishnanp R, Peker KA (2002) Motion activity-based extraction of key-frames from video shots. Int Conf Image Process 1:932–935
8. Dufaux F (2000) Key frame selection to represent a video. Proc ICIP Conf 2:275–278
9. Gibson DNC, Thomas B (2002) Visual abstraction of wildlife footage using Gaussian mixture models. Proc. 15th Int. Conf Vision Interface
10. Girgensohn A, Boreczky J (1999) Time-constrained keyframe selection technique. IEEE Int Conf Multimedia Comput Syst 1:756–761
11. Gong Y, Liu X (2003) Video summarization and retrieval using singular video decomposition. ACM Multimedia Syst 9(2):157–168. doi:10.1007/s00530-003-0086-3
12. Hanjalic A, Zhanq HonqJianq (1999) An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. IEEE Trans Circ Syst Video Tech 9(8):1280–1289. doi:10.1109/76.809162
13. He L, Sanocki E, Gupta A, Grudin J (1999) Auto-Summarization of audio-video presentations. Proc. ACM Multimedia Conf. (ACMMM) 489–498
14. Laskaris NA, Zafeiriou SP (2008) Beyond FCM: graph-theoretic post-processing algorithms for learning and representing the data structure. Pattern Recognit 41(8):2630–2644. doi:10.1016/j.patcog.2008.02.005
15. Latecki LJ, Widldt DD, Hu J (2001) Extraction of key frames from videos by optimal color composition matching and polygon simplification. Proc. Multimedia Signal Process Conf. (France)
16. Liu T, Kender JR (2002) An efficient error-minimizing algorithm for variable-rate temporal video sampling. Proc. Int. Conf. Multimedia Expo (ICME)
17. Liu T, Zhanq H-J, Qi F (2003) A novel video key-frame extraction algorithm based on perceived motion energy model. IEEE Trans Circ Syst Video Tech 13(10):1006–1013. doi:10.1109/TCSVT.2003.816521
18. Liu T, Zhang X, Feng J, Lo K-T (2004) Shot reconstruction degree: a novel criterion for keyframe selection. Pattern Recognit Lett 25(12):1451–1457. doi:10.1016/j.patrec.2004.05.020
19. Marchionini G, Geisler G (2002) The open video digital library. D-Lib 8(12). doi:10.1045/december2002-marchionini
20. Motzkin TS, Straus EG (1965) Maxima for graphs and a new proof of a theorem of Turan. Can J Math 17:533–540
21. Mundur P, Rao Y, Yesha Y (2006) Keyframe-based video summarization using Delaunay clustering. Int J Digit Libr 6(2):219–232. doi:10.1007/s00799-005-0129-9
22. Pavan M, Pelillo M (2007) Dominant sets and pairwise clustering. IEEE Trans Pattern Anal Mach Intell 29(1):167–172. doi:10.1109/TPAMI.2007.250608
23. Ueda H, Miyatake T, Yoshizawa S (1991) Impact: an interactive natural picture dedicated multimedia authoring systems. Proc. SIGCHI Conf Human factors Computer Systems 343–350
24. Weibull JW (1995) Evolutionary game theory. MIT Press
25. Xiong W, Lee JCM, Ma RH (1997) Automatic video data structuring through shot partitioning and key frame computing. Mach Vis Appl 10(2):51–65. doi:10.1007/s001380050059
26. The Open Video Project http://www.open-video.org/
27. The MPEG Software Simulation Group http://www.mpeg.org/MPEG/MSSG/.
28. Yahiaoui I, Merialdo B, Huet B (2001) Automatic video summarization. Proc. CBMIR Conf

29. Yeung MM, Liu B (1995) Efficient matching and clustering of video shots. Proc Int Conf Image Process 1:338–341. doi:10.1109/ICIP.1995.529715
30. Yu X D, Wang L, Tian Q, Xue P (2004) Multi-level video representation with application to keyframe extraction. Proc. Int. Conf. Multimedia Modelling (MMM) 117–121
31. Zhang DQ, Lin CY, Chang SF, Smith JR (2004) Semantic video clustering across sources using bipartite spectral clustering. Proc IEEE Conf Multimedia Expo (ICME) 1:117–120
32. Zhuang Y, Rui Y, Huang TS, Mehrotra S (1998) Adaptive key frame extraction using unsupervised clustering. Proc Int Conf Image Process 1:866–870

Dimitrios Besiris He was born in Agrinio in 1978. He received the B.Sc. degree in Physics in 2002 and the M.Sc. degree in Electronics in 2005. He is currently a Ph.D candidate in Image and Video Processing from the Electronics Laboratory, Dept. of Physics, University of Patras, Greece. His main research interests include image and video processing (browsing, retrieval, summarization and video object tracking), and graph theory.

Andrew Makedonas He was born in Patras in 1977. He received both the B.Sc. degree in Physics in 2001 and the M.Sc. degree in Electronics in 2005 from University of Patras, Greece. He is currently a PhD candidate in image processing at the Electronics Laboratory, Dept. of Physics, University of Patras, Greece. His main research interests include image processing, pattern recognition, data mining and graph theory.

George Economou received the B.S. degree in physics from the University of Patras (UoP), Greece in 1976, the M.S. degree in microwaves and modern optics from University College London in 1978, and the Ph.D. degree in fiber-optic-sensor-systems from the University of Patras in 1989. He is currently an Associate Professor at Electronics Laboratory (ELLAB), Department of Physics, UoP, where he teaches at both undergraduate and postgraduate levels. He has published papers on nonlinear signal and image processing, fuzzy image processing, multimedia databases, data mining, and fiber-optic sensors. He has also served as a referee for many journals, conferences, and workshops. His main research interests include signal and image processing, computer vision, pattern recognition, and optical signal processing.

Spiros Fotopoulos is Professor at the Department of Physics of the University of Patras. He is Director of the M.S. course on electronics and information processing. He is working in the digital signal and image processing area. His research activities include nonlinear digital filters, fuzzy image processing, multimedia databases, computer vision, graph-theoretic approaches, and applications to satellite images and biomedical signals.