Video Summarisation for Surveillance and News Domain

Uros Damnjanovic, Tomas Piatrik, Divna Djordjevic, and Ebroul Izquierdo

Department of Electronic Engineering, Queen Mary, University of London,
London E1 4NS, U.K.
{uros.damnjanovic, tomas.piatrik, divna.djordjevic,
ebroul.izquierdo}@elec.qmul.ac.uk

Abstract. Video summarization approaches have various fields of application, specifically related to organizing, browsing and accessing large video databases. In this paper we propose and evaluate two novel approaches for video summarization, one based on spectral methods and the other on ant-tree clustering. The overall summary creation process is broke down in two steps: detection of similar scenes and extraction of the most representative ones. While clustering approaches are used for scene segmentation, the post-processing logic merges video scenes into a subset of user relevant scenes. In the case of the spectral approach, representative scenes are extracted following the logic that important parts of the video are related with high motion activity of segments within scenes. In the alternative approach we estimate a subset of relevant video scene using ant-tree optimization approaches and in a supervised scenario certain scenes of no interest to the user are recognized and excluded from the summary. An experimental evaluation validating the feasibility and the robustness of these approaches is presented.

Keywords: Spectral clustering, ant-tree clustering, scene detection, video summarization.

1 Introduction

With new video material being created every day, the users need to spend more and more time viewing the content even though they might not be interested into all its aspects. Therefore efficient browsing and access to relevant video items is a crucial application in modern multimedia systems. Not only that new material is created everyday, but also new applications that utilize the creation and distribution of multimedia content and its metadata are emerging. For example WWW applications make the process of video publishing and sharing available to everyone. Video on demand, news broadcasting and syndication, personal video archiving, surveliance and security camera tracking are just some of the examples of exiting applications which are working with huge video databases. Not only time requirement issues but also storage issues have to be taken into account when working with videos. Storing whole video when just a small part is used waists a lot of disk space resulting in decrease of application efficiency. Conventional multimedia systems are still not capable to overcome above mentioned constraints. For example, most existing video players are offering only fast forward and backward option, leaving the user without the possibility to efficiently search the video. Contrary, modern multimedia applications need to be efficient, providing the user only with pieces of content that are of interest. However the definition of what is interesting may vary from user to user, or from application to application making the problem more difficult. But the essence of the approach is to present to the user as much interesting information as possible in the smallest possible time interval. In this aspect, video summaries can be seen as short but highly informative representation of the content, presented to the user who wants to have fast overview of the available information.

Looking at the video from bottom to top, essential part of a video are frames, which are further organized into shots [1-2]. The next level in the video structure is a scene, defined as a group of shots with similar semantic content. Shots can still be seen as low level of video organization, while scenes represent more semantic level of video organization. After the scenes are found one need to choose which part of the scene actually contains the information important to the user and present it. This can be done either as key frame presentation, or as a video skim.

Research efforts put in the problem of video summarization and scene detection resulted in large amount of available literature on this topic. Most of the approaches focuses on either detecting the scenes within the video or on defining some importance measure which is the used for selecting parts of the video which will be presented to the user. Main problem in both approaches is still to connect the low level representation of the video with high level semantic representation. For example problem of scene detection should sometimes group together shots that are visually toatally different. Using only low level representation of video this is very hard task to do. Also, automatic detection of informative parts of a video can be very hard task having in mind subjective nature of importance definition. In [3] authors presented the tool that utilizes MPEG-7 visual descriptors and generates a video index for summary creation. The resulting index generates a preview of the movie and allows non-linear access to the content. This approach is based on hierarchical clustering and merging of shot segments that have similar features and neighboring them in the time domain. In [4] Rasheed and Shah construct a shot similarity graph, and use graph partitioning normalized cut for clustering shots into scenes. In a similar approach [5] authors proposed a novel way to assess clustering quality using “eigengap” measure. Video summarization based on the optimization of viewing time, frame skipping and bit rate constraint is described in [6]. For a given temporal rate constraint the optimal video summary problem is defined as finding a predefined number of frames that minimize the temporal distortion. Otsuka et. al in [7] presented their video browsing system that uses audio to detect sport highlights by identifying segments with mixture of the commentators excite speech and cheering. Video motion analysis can be used for creating video summaries as in [8]. In this approach Wang et al. showed that by analyzing global/camera motion and object motion is possible to extract useful information about the video structure. Motion vectors are also used in [9] for demonstrating semantic inference by using the MPEG-7 low-level motion activity features. In [10] motion based method for video summarization and scene detection is achieved by analyzing the temporal variations of some coefficients of the 2D affine model.

In this paper we present two frameworks for efficient video summarization. One based on spectral clustering methods for temporal segmentation of the video followed by representative key frame extraction based on the motion analysis of detected scenes. In this approach we assume that scenes that are rich in motion are more interesting to the user that static ones. It is a logical choice for news and surveillance video domain, having in mind that usually events and highlights are related to some dynamic process within videos. For example, in news video domain, showing the actual event is much more interesting to the user than presenting anchor person speaking about the event. Motion analysis of scenes results in key-frames that are rich in the local motion opposed to the global motion. Finally, we propose the approach for analyzing the structure or resulting clusters, in order to define importance measure for every part of the scene based on the statistical analysis of eigenvectors used in clustering process. We show that is possible to have information about the motion activity in the scene, without extracting motion vectors from MPEG stream, and to use this information to create video skims that will be presented to the user. In the alternative approach, ant-tree clustering with automatic partitioning of video scenes is used for video summarization. It is combined with semi-supervised approach for recognizing non-relevant scenes (e.g. anchor person for the news domain and static indoor or outdoor scenes for the surveillance video). The remaining clusters and their relevance in regards to the timeline is used for obtaining video summaries. We compare the obtained video skims with the manually generated ground truth in both domains.

Structure of the paper is as follows. In section 2 the spectral clustering approach for building video summaries is presented. This also incorporates the eingen vector analyses and definition of importance measure for every part of the scene. Section 3 presents the ant-tree approach for building video summaries together with scene recognition. In section 4 the experimental results are depicted by comparing the generated summaries with ground truth data. Section 5 presents some conclusion.

2 Spectral Clustering for Video Summarization

We present a video summarization approach based on spectral normalized cut algorithm. Since spectral clustering methods have proved to be efficient in detecting block structure of a similarity matrix for some dataset, in this section we show how this block structure can be formed over a set of video frames, and then used to detect scenes in a video. As mentioned scenes are neighboring video segments with similar visual content. The task of detecting scenes can be seen as one of clustering frames based on their visual properties and temporal position. Once the scenes are detected one needs to present each scene to the user. The goal is to find a small set of frames that provide the core information about the whole scene. Each frame is then ranked based on the motion activity, and finally video summary is created. The ranking is based on the fact that the structure of the similarity matrix of eigenvectors corresponds to the structure of the changes in the video.

2.1 Spectral Clustering

Spectral clustering has its origin in a spectral graph partitioning,. a popular algorithm in high performance computing. Now days, spectral clustering has many applications in machine learning, exploratory data analysis, computer vision and speech processing. The success of such algorithms depends heavily on the choice of the metric, but this choice is generally not treated as part of the learning problem. Spectral algorithms use information contained in the eigenvectors of data affinity matrix to detect structures.

Given a set of data points, the pair wise similarity matrix is defined as matrix W with elements w_{ij} representing a measure of similarity between points *i* and *j*. Spectral clustering techniques make use of the spectrum of the data similarity matrix to cluster the points. The matrix mechanics is closely related to the more general singular value decomposition. Square matrices have an eigenvalue/eigenvector equation with solutions that are the eigenvectors $x_{\lambda.}$ and the associated eigenvalues λ . The main idea behind all spectral clustering algorithms is similar. Initially a pair-wise similarity has to be defined and a similarity matrix built. Eigen decomposition of the similarity matrix results in eigenvalues and eigenvectors which are finally used to cluster the dataset.

The set of data points are denoted by *V,* with cardinality $|V| = N$. For each pair of points $i, j \in V$, a similarity $w_{ij} = w_{ji} \geq 0$ can be viewed as weight of the undirected edge of a graph *G* over *V*. The matrix $W = \lfloor w_{ij} \rfloor$ plays the role of *"real valued"* similarity matrix for *G*. Let d_i represent a degree of node *I:*

$$d_i = \sum_{j \in I} w_{ij} \tag{1}$$

And let the volume of a set $A \subset V$ be defined as:

$$VolA = \sum_{i \in A} d_i \tag{2}$$

Let D be a $N \times N$ diagonal matrix with values d_i, $i \in [1, N]$ on its diagonal. Then Laplacian matrix of the graph *G* is defined as:

$$L = D - W \tag{3}$$

The set of edges between two disjoint sets $A, B \subseteq V$ is called *edge cut* or in short *cut* between *A* and *B*:

$$cut(A, B) = \sum_{i \in A, j \in B} w_{i,j} \tag{4}$$

Once the specific eigenvalues and its corresponding eigenvectors are found, membership of each point from the dataset is determined by investigating specific entries of eigenvectors. Every entry of eigenvector corresponds to exactly one point from the dataset. By comparing the entry to some value that is chosen to be the splitting point, membership of a point to one or the other partition is determined. A set of clusters $C = \{C_1, C_2, ...C_k\}$ defines partitioning of *V* into nonempty mutually disjoint subsets.

In the graph theoretical paradigm a clustering represents a *multiway cut* in the graph *G*. After creation of a specific graph structure, the objective function can be easily defined. Usually this is some measure that describes the relations between two or more separate clusters. The clustering problem can be seen as a problem of finding partitions of a dataset in a way that similarities inside clusters are large, and similarities between different clusters are small. Most of the algorithms can be thought of as consisting of three stages:

- **Pre-processing:** This is a form of normalization of the similarity matrix *W* in order to avoid ill conditioned adjacency matrix.
- **Spectral Mapping:** Top k eigenvectors of the pre-processed similarity matrix are computed. Each data point is represented by the value of a specific component in the aforementioned eigenvectors.
- **Post-processing/Grouping:** A grouping algorithm clusters the data based on the respective values of eigenvectors.

Normalized Cuts algorithm, *Ncut*, was first introduces by Shi and Malik in [11] as a heuristic algorithm aiming to minimize the *Normalized Cut* criterion. Originally this approach was created to solve the problem of perceptual grouping in the image data. The normalized cut between two sets $A, B \subseteq I$ is defined as:

$$NCut(A,B) = Cut(A,B)\left(\frac{1}{VolA} + \frac{1}{VolB}\right) \tag{5}$$

The algorithm consists of minimizing (5) by solving the generalized eigensystem:

$$L\,x = \lambda D\,x \tag{6}$$

Ncut algorithm focuses on the second smallest eigenvalue of (6) and its corresponding eigenvector, λ^L and x^L respectively. In [12] it is shown that when there is a partitioning *A, B* of *V* such that

$$x^L = \begin{cases} \alpha, i \in A \\ \beta, i \in B \end{cases} \tag{7}$$

then *A*, *B* is the optimal cut and the value of the cut itself is λ^L. This result represents the basis of spectral segmentation by normalized cuts. After the dataset is divided in two groups, the binary clustering algorithm can be run recursively.

In [12] it is shown that multway cut approach gives better results then recursively applying binary clustering, until k clusters are reached. In a mulitwaycut approach instead of using only one eigenvector of (6), top k eigenvectors are used, where k is predefined number of clusters. Let matrix X be $N \times k$ matrix, created by stacking the top k eigenvectors in columns. Each row of X corresponds to a point in the dataset and is represented in a k-dimensional Euclidian space. Finally, k clusters are obtained by applying the K-means algorithm over the rows of X.

2.2 Video Summarization with Spectral clustering

The first step in our algorithm is to create the similarity matrix. Instead of analyzing the video on a key frame level, we are using a predefined ratio of frames in order to stimulate the similarity matrix block structure. For surveillance videos similarity between all frames is high resulting in the similarity matrix which has almost all entries close to 1. Any change that happens in the video will result in a local change of the block structure of the similarity matrix.

In most cases events occurring in the surveillance video are short making the extraction of important events from the whole set of frames a difficult task. The emphasis in the summarization of surveillance videos is to detect small and short changes in the video, and to present them to the user based on importance levels. On the other side news videos are formed of clusters that are significantly different. The main task in the news summarization is to properly cluster scenes, and then to analyze clusters in order to find most informative representatives.

The similarity matrix W is made of pair wise similarities $w_{i,j}$ between two frames i and j defined as:

$$w_{i,j} = \exp\left(-\frac{d(i,j)^2}{2\sigma^2} \right) \tag{8}$$

Where $d(i,j)$ is a distance is over the set of low- level feature vectors and σ is scaling parameter. Biger the σ is, higher the sensitivity of the similarity measure. After creating the similarity matrix and solving the generalized eigensystem from (6), the next step is to determine the number of clusters in the video. This number is then passed to a classical k-means algorithm. Automatic determination of the number of clusters is not a trivial task. In [13] is shown that every similarity matrix have a set of appropriate number of clusters depending on the choice of the parameter σ used in equation (8). For automatic detection of number of clusters for fixed σ we use the results of matrix perturbation theory [14]. The matrix perturbation theory states that the number of clusters in a dataset is highly dependent on the stability of eigenvalues/eigenvectors determined by the eigengap δ :

$$\delta_i = \left| \lambda_i - \lambda_{i+1} \right| \tag{9}$$

With λ_i and λ_{i+1}, being two consecutive eigenvalues of (6). The number of clusters k is then found by searching for the maximal eigengap over a set of eigenvalues:

$$k = \left\{ i \middle| \delta_i = \max_{j \in [1,N]} (\delta_j) \right\} \tag{10}$$

After the right number of cluster is determined, k-means algorithm is initialized with k equidistant rows of the matrix X (defined in section 2.1). Results of the k-means algorithm are clusters that give importance information for various applications Fig. 1. clusters which came from the k-means are used in the decision process, which will actually perform the summarization of the video, Fig. 2. Decision process analyzes the structure of the clusters and decides how to present them to the user. Let

$\tau_{i,j}$ be the length of the j-th continuous sequence of cluster i as in Fig. 2, and let $\sum_{j\in[1,n_i]} \tau_{i,j} = T_i$ be the total length of the cluster i, where n_i is the total number of disconnected continuous segments of the cluster i. T_v is the length of the whole video, T_s is the length of the resulting summary and $\delta_{m,n}^{(p)}$ is the Euclidean distances between centroids of m-th and n-th cluster in p-dimensional feature space.

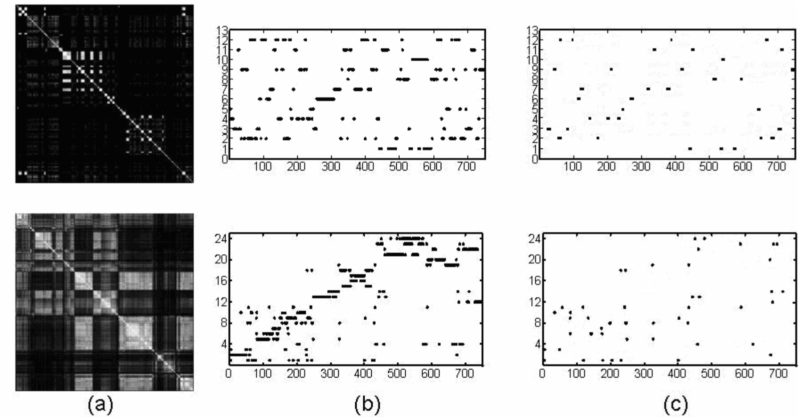

Fig. 1. The video summarization process, the top row represents news domain example, the bottom row surveillance video example. a) Similarity matrix. (b) Result of the k-means clustering. Cluster indicators are plotted over the time axis. Every point on the horizontal axis corresponds to the fixed time interval. (c) Final summary structure. For the news videos, long continuous segments are used to build the summaries. In the surveillance domain, long segments are removed and only short sequences are used in building the summary.

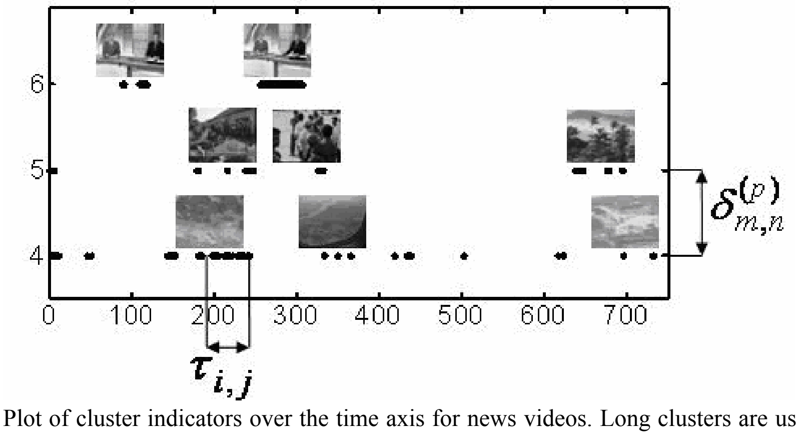

Fig. 2. Plot of cluster indicators over the time axis for news videos. Long clusters are used to give short segments that will be used for building the summary. Parameters $\tau_{i,j}$ and $\delta_{m,n}^{(p)}$ are used in the analysis of clusters. $\tau_{i,j}$ is the length if the continuous segment, while $\delta_{m,n}^{(p)}$ is the distance between centroids of two clusters in p-dimensional feature space.

The cluster analysis for the surveillance videos search for clusters with short continuous segments using the following rule:

if $\tau_{i,j} < T_{tr}$ *the sequence i of the cluster j contains important information.*

Or in the case of the news domain the rule is as follows:

if $\tau_{i,j} > T_{tr}$ *the sequence i of the cluster j contains important information.*

Where T_{tr} is experimentally determined threshold. Motivation for the decision rules is that in the surveillance videos long scenes correspond to static scenes where nothing important is actually happening in the video. While for the news videos long scenes usually contain information that is important to the user.

After important clusters are extracted, next step is to decide how to present them to the user. The main task of the video summarization is to save time for browsing the video. Time saving is achieved by choosing short segments that represent clusters. Let T_{si} be the duration of each cluster i representative segment. With representative segment we mean part of the each cluster that will be used to build the summary. Duration T_{si} in the surveillance video is set to a fixed value, while in the case of news video it is proportional to the length of the cluster. To calculate the duration T_{si} for each cluster, distances between cluster centroids are found. If $\delta_{m,n}^{(p)} < \delta_{tr}$, then clusters m and n will contribute together to the overall duration of the summary T_s with: $T_{smn} = T_{sm} = T_{sn}$. Now, T_{si} is calculated using the following equation:

$$T_{si} = T_s \cdot \frac{T_i}{T_v} \qquad \textbf{(11)}$$

The last step in our algorithm is to find representative segments of the specified duration for each cluster. We propose simple method for motion activity estimation that is based on the statistical theory of spectral clustering. The authors in [15] showed a connection between spectral clustering and Markov random walks where pair wise similarities are seen as flows in Markov random walks within a probabilistic framework for properties of eigenvectors and eigenvalues. In the case of the ideal dataset with high similarities within the same clusters and with sum of inter-cluster similarities close to zero, resulting eigenvectors will be piecewise constant. In other words, eigenvector values on the same side of a cut will have almost same values. In our approach we studied properties of datasets which resulted in non-constant eigenvectors. By applying statistical analysis of eigenvector entries, it is possible to get the insight of the motion activity level. We use statistical analysis of eigenvectors on scene level to extract representatives for each interesting segment. The algorithm searches for the segments with largest values of accumulated absolute change between consecutive eigenvector entries. Areas with high change in the eigenvector entries correspond to the regions with high motion activity levels. If t_r is the start time of a representative scene and $\Delta E_i = |E_{i+1} - E_i|$ is the absolute difference between consecutive entries of eigenvectors, then t_r can be found by using the following formula:

$$t_r = t_i \left| \sum_{i \in (i, i+T_{si})} \Delta E_i = \max_{i \in [1,N]} \sum_{j \in (j, j+T_{si})} \Delta E_j \right. \quad \textbf{(12)}$$

Segments extracted using equations (12) are finally used to build video summaries, which are then presented to the user.

3 Ant-Tree clustering for Video Summarization

Many researchers in computer science have been inspired by real ants [16] and have defined artificial ants paradigms for dealing with optimization or machine learning problems [18-20]. Previous models involve the ability of ants to sort objects [20-21] or to build a colonial odor [22] and mechanical structures by a self-assembling behavior [23]. In this section, we present a video summarization approach based on ant-tree clustering algorithm. We model the ability of ants to build live structures with their bodies [24] in order to discover, in a distributed and unsupervised way, a tree-structured organization and summarization of the video data set.

3.1 Ant-tree Clustering

The ant-tree clustering method was inspired by self-assembling behavior of African ants and their ability to build chains by their bodies in order to link leaves together [23]. The main principles of the algorithm are depicted in Fig. 3: each ant represents a node of the tree to be assembled, i.e. a data to be clustered. On the basis of a root (initial support node) on which the tree will be built, ants will gradually fix themselves until all ants are attached to the structure.

The movement and fixing of ants in a position depends on the similarity value of the data and on the local neighborhood of the moving ants. Similarity measure between two data points is denote by $sim(i, j)$ which gives, for a couple of data points (x_i, x_j), $i, j \in [1, N]$, a value in the range [0, 1] where N is the number of data points. Thus, for each ant a_i the following concepts are defined:

- the outgoing link of a_i is the link that a_i can maintain toward another ant;
- the incoming links of a_i are the links that the other ants maintain towards a_i, these bonds can be the legs of the ant;
- the data x_i is resented by a_i;
- There are two thresholds one for higher- similarity $T_{sim}^{H}(a_i)$ and one for lower-similarity (dissimilarity) $T_{sim}^{L}(a_i)$, which are locally updated by a_i.

At each step, an ant a_i will connect itself or move according to the similarity with its neighbors. While there is still a moving ant a_i, we simulate its action according to the movement. The first ant is directly connected to the support. Next, for each ant a_i, the two following cases need to be considered.

The first case is when ant a_i is placed on the support; if a_i is similar enough to a^+ so that $sim(a_i, a^+) > T_{sim}^H(a_i)$, where a^+ is the ant which is the most similar to a_i among the ants already connected to the support; then a_i is moved towards a^+ in order for both ants to be clustered in the same subtree, i.e. the same cluster.

Otherwise if a_i is dissimilar enough to a^+ that is $sim(a_i, a^+) < T_{sim}^L(a_i)$, then it is connected to the support and it becomes a representative ant of a new subtree. This means that the new sub-tree is created, ant its ants will be as dissimilar as possible to representative ants in other sub-trees connected to the support. Finally, if a_i is not similar or dissimilar enough to a^+, the thresholds is updated in order to allow a^+ to be more tolerant and to increase its probability of being connected the next time it will be considered (in the meantime, other ants will have changed the tree). The updating procedure of the proposed threshold for improving summarization is present in section 3.2.

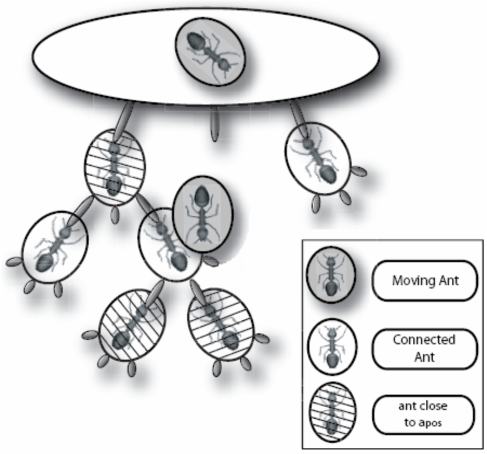

Fig. 3. Building of a tree with artificial ants.

The second case is when a_i is on top of another ant denoted by a_{pos}; if there is a free incoming link for a_{pos} and if a_i is similar enough to a_{pos}, that is if the following holds true $sim(a_i, a^+) > T_{sim}^H(a_i)$ and at the same time dissimilar enough to ants connected to a_{pos}, then a_i is connected to a_{pos}. In this case, a_i represents the root of a new sub-tree or sub-cluster below a_{pos}. Its dissimilarity with other ants directly connected to a_{pos} is such that sub-clusters of a_{pos} will be well "separated" from each other (while being similar to a_{pos}).

Otherwise, a_i is randomly moved toward a neighbor node of a_{pos} and its thresholds are updated in the same way as described in the previous case. So, a_i will move around within the graph, in order to find the optimal location. The algorithm ends when all ants are connected.

3.2 Video Summarization by Ants

To obtain a video summary, an ant tree is built so that nodes correspond to video frames and the edges are what needs to be discovered in order to summarize the content. One should notice that this tree will not be strictly equivalent to a dendogram as used in standard hierarchical clustering techniques. Here each node corresponds to one data point, while in the case of dendograms data points correspond to leaves.

A fundamental step in our video summarization approach is to create the similarity matrix and organize video frames into the tree-structure using ant-tree clustering method as depicted in Fig. 4. Results of the ant-tree algorithm are clusters which are used in the decision process for creating video summaries.

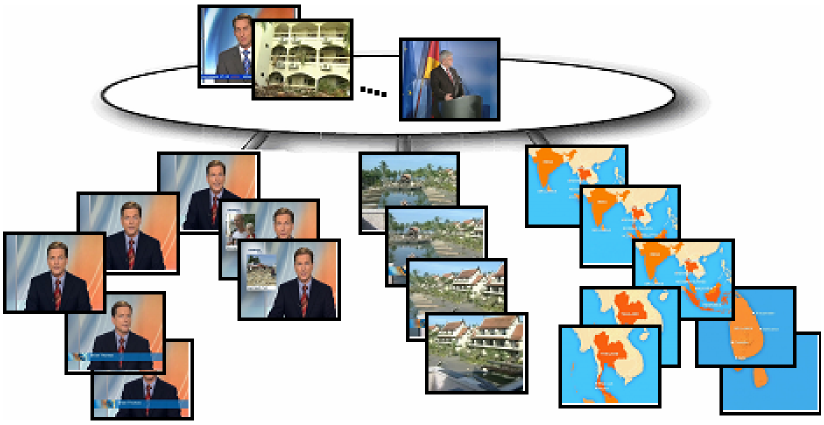

Fig. 4. Tree structuring of video frames by ant-tree clustering.

In order to increase quality of clustering and for automatic detection of a number of clusters, the following definition of threshold updating is proposed:

$$T^{H}_{sim}(a_i) = T^{H}_{sim}(a_i).\frac{\alpha-1}{\alpha}, \quad T^{L}_{sim}(a_i) = T^{L}_{sim}(a_i)\cdot\frac{\beta}{\gamma} \qquad \textbf{(13)}$$

$$\beta_{(t)} = \frac{\max\delta^{(p)}_{m,n} - \min\delta^{(p)}_{m,n}}{\max\delta^{(p)}_{m,n}},\ t \in [0,R] \qquad \textbf{(14)}$$

where α represents sensitivity, $\delta^{(p)}_{m,n}$ is distance between support frames of two clusters m,n in p-dimensional feature space, R is iteration number of the clustering approach and γ is used for normalization. The higher α is, the more sensitive the clustering algorithm to changes among frames. The main idea of our approach is to iterate the clustering process until optimal number of clusters is found. The parameter $\beta_{(t)}$ is initialized for first run of ant-tree algorithm and a new $\beta_{(t+1)}$ is obtained after all frames are connected. The optimal partitioning is achieved when $\beta_{opt} = \left|\beta_{(t)} - \beta_{(t-1)}\right| \le \beta_{tr}$. Finding the β_{opt} is also important in the stage of calculating the duration T_{si} (defined in 2. 2) of each cluster that will be used to build the summary. If some clusters are too similar to each other ($\delta_{m,n}^{(p)}$ is very small), then frames will be put together and will contribute to the overall duration of summary as one scene. Duration of each representative segment of video is calculated using equation (11). In order to present most important information of video content to the user in short time, the following procedure is applied. We defined certain scenes of no interest to the user according to the type of video. In the news domain it is the scene an "anchor person" and in the surveillance domain statical "indoor" and "outdoor" scene. After all frames are clustered, and the above mentioned scenes are recognized we excluded clusters contained these scenes from the summary using constraints.

4 Experimental Evaluation

For experimental evaluation in the news domain we used two 20 minutes long news videos, and in the surveillance domain two 20 minute surveillance videos (one depicting an inside room with people entering and leaving and the other depicting an outdoor parking). Low-level features are computed on the frame level using one frame per second sample ratio. We used MPEG-7 color layout features for representing each selected frame.

Spectral parameter σ from (8) is experimentally chosen in such a way that the number of clusters corresponds to the specific application. When choosing the σ values, we create the similarity matrix and examine the resulting eigenvalues. Specifically we examine the number of clusters in created similarity matrix. For example setting σ high in the news domain will result in few clusters, while in the surveillance domain it result in only one cluster. Different values of σ come from different structure of the specific domains. In news video domain, high level of visual changes of frames is significantly bigger then in the surveillance domain. In the surveillance doman, small visuall changes lead to small changes of the simialrtiy measure that are hard to detect. For the news domain σ is set to 20, while for the surveillance domain σ is set to5. Modification of values for parameter σ influence changes in sensitivity of the similarity measure and the ability to detect small changes in the video. The spectral clustering approach initializes k-means clustering and the final clustering is obtained after 10 to 15 iterations.

In the case of ant-tree based summarization after empirical evaluations $\beta_{(0)} = 0.1$ and parameter α from (13) has different values in respect of the type of changes present in the particular domain. For news videos we set $\alpha = 100$ since the events are depicted with dynamic scenes with large variations in content. For the surveillance videos the sensitivity is defined with higher granularity (one person or a car moving) and therefore we set $\alpha = 1000$.

The spectral method gives better performances in case of the surveillance domain (87% of events are detected) since it can detect regions with high motion activity, while the ant-tree clustering approach detects 74 % of events. In the news domain where anchor person precedes most of the stories (events) the scene recognition properties of the ant-tree approach lead to higher performances 86.2% of correctly detected event comparing to 82% for the spectral approach. In this case most of the relevant scenes have high-motion activity; hence this feature is not discriminative enough of its own.

Table 1. Results of the video summarisation.

Method	Video type	Number of detected events	Number of missed events	Total length of videos	Total length of summaries
Spectral	News	24	5	40 min	3 min
	Surveil-lance	27	4	40 min	2 min
Ant-tree	News	25	4	40 min	3 min
	Surveil-lance	23	8	40 min	1,5 min

Overall the experimental results show that our proposed summarization approaches abridged the original video with a compression factor up to 16:1 while capturing most of user defined relevant scenes.

5 Conclusion

We have shown that spectral and ant-tree clustering approaches can efficiently be used for summarizing videos. In the first case the main reason is that high similarity between consecutive frames in video sequence strengthens the block structure of the similarity matrix in spectral clustering. Furthermore, without the need for extra processing time the structure of the video can be efficiently mapped to eigenvectors and eigenvalues, and used for discovering important segments of the video in both domains. In the latter approach for video summarization, ant-tree clustering, the emphasis is on scene recognition. The approach generates a set of representative shots and extracts the tree structure of a video sequence. In case of surveillance video, where scenes are very similar to each other, ant-tree based summarization shows lack of consideration of motion activity within scenes. Our future work will encompass the advantages of approaches, joining together motion activity and scene recognition properties underlying the individual approaches.

References

1. Calic, J., Izquierdo, E.: Towards real time shot detection in the MPEG compressed domain. In Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services (2001)
2. Yeo, L. B., Liu, B.: Rapid scene analysis on compressed video. IEEE Transactions on Circuits & Systems for Video Technology, Vol. 5 (1995) 533-544

3. Lee, J., Lee, G. G., Kim, W. Y.: Automatic video summarizing tool using MPEG-7 descriptors for personal video recorder. IEEE Transaction on Consumer Electronics, Vol. 49 (2003). 742-749

4.. Rasheed, Z., Shan, M.: Detection and Representation of scenes in videos. IEEE Transactions on Multimedia, Vol. 7 (2005) 1097-1105

5. Odobez, J., Gatica-Perez, D., Guillemot, M.: Video shot clustering using spectral methods. In 3rd Workshop on Content Based Multimedia Indexing (CBMI) (2003)

6. Li, Z., Schuster, G.M., Katsaggelos, A. K.: MINMAX optimal video summarization. IEEE Transactions on Circuits and Systems for Video Technology. Vol. 15, (2005) 1245-1256

7. Osuka, I., Radharkishnan, R., Siracusa, M., Divakaran, A., Mishima, H.: An enhanced video summarization system using audio features for personal video recorder. IEEE Transactions on Consumer Electronics. Vol. 52 (2006) 168-172

8. Wang, Y., Zhang, T., Tretter, D.: Real time motion analysis towards semantic understanding of video content. Conference on Visual Communications and Image Processing (2005)

9. Peker, K. A., Alatan, A. A., Akansu, A. N.: Low level motion activity features for semantic characterization of video. IEEE International Conference on Multimedia and Expo Vol2. (2000) 801-804

10. Peyrard, N., Bouthemy, P.: Motion-based selection of relevant video segments for video summarization. Multimedia Tools and Applications (2005) 259-276

11. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on PAMI, Vol. 22 (2000) 888-905

12. Alpert, C., Khang, A., Yao, S.: Spectral partitioning: The more eigenvectors, the better. Discrete Applied Mathematics (1999)

13. Manjunath,, B. S., Salembier, P., Sikora, T.: Introduction to MPEG-7. John Willey& Sons (2002)

14. Zheng, X., Lin, X.: Automatic determination of intrinsic cluster number family in spectral clustering using random walk on graph. International Conference on Image Processing, ICIP 04, vol. 5 3471-3474

15. Meila, M., Shi, J.: A random walks view of spectral segmentation. AI and Statistic (2001)

16. Holldobler, B., Wilson, E. O.: The Ants. Springer Verlag (1990)

17. Dorigo, M., Di Caro, G.: Ant Algorithms for Discrete Optimization. Technical Report (1999) 98-10

18. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press (2004)

19. Bonabeau, E, Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artifical Systems, Oxford University Press (1999)

20. Lumer, E., Faieta, B.: Diversity and Adaptation in Populations of Clustering Ants. 3tl1 Conference on simulation and adaptive behavior-: from animals to animats (1994) p. 501-508

21. Kuntz, P., Snyers, D., Layzell, P.: A stochastic heuristic for visualizing graph clusters in a hi-dimensional space prior to partitioning, Journal of Heuristic, (1999) vol 5

22. Labroche, N., Monmarche, N., Venturini G.: A new clustering algorithm based on the chemical recognition system of ants. Proceedings ot the 15th European Conference on Artifical Inteligence (2002)

23. Azzag, N., Monmarch, H., Slimane, M., Venturini, G., Guinot, C.: Antree: a new model for clustering with artificial ants. In IEEE Congress on Evolutionary Computation, (2003) 08-12

24. Lioni, A., Sauwens, C., Theraulaz, G., Deneubourg, J., L.: The dynamics of chain formation in Oecophylia longinoda. Journal of Insect Behavior, (2001) vol. 14, 679-696