Optimal Multiscale Organization of Multimedia Content for Fast Browsing and Cost-Effective Transmission

KLIMIS S. NTALIANIS and STEFANOS D. KOLLIAS
Electrical and Computer Engineering Department
National Technical University of Athens
9, Iroon Polytechniou str., Zografou 15773, Athens
GREECE
http://www.image.ntua.gr

Abstract: - In this paper, an interactive framework for efficient browsing and transmission of video sequences is presented, based on an optimal content-based video decomposition scheme. In particular each video file is analyzed to provide a multiscale structure of different "content resolution levels". This structure can be seen as a tree structure, each level of which corresponds to a particular content resolution, while the tree-nodes contain viewing elements, representing the visual content of a segment of the sequence. The multiscale optimal video organization is performed by minimizing a cross correlation criterion so that the most representative shots (key-shots) from a video sequence or frames (key-frames) from a video shot are extracted. Experimental results on real-life video sequences show that the proposed multiscale video organization technique enables users to detect content of interest much faster, compared to the conventional sequential video scanning method, and thus it leads to significant reduction of the viewed/transmitted information.

Key-Words: - video sequence, non-linear analysis, tree structure, content organization

1 Introduction

The increasing number of video sequences induces an increasing need for fast browsing and efficient transmission of video content. Currently the only way to get an overview of a video sequence is the functions forward/backward or to jump to a random time-code. The first approach is usually time-consuming and tedious while the second does not guarantee the number of steps until the content of interest is detected. If for example a user would like to browse a video sequence of 2 hours duration and he/she fasts it forward by 4 times compared to the normal speed, then he/she would need about 30 minutes to preview the visual content of the sequence. Furthermore if the user looks for a particular content, he/she would spend 15 minutes on average to detect it, assuming that each video frame presents the same probability to be accessed.

To make things worse, suppose that the aforementioned video sequence is located on a distributed platform and is remotely accessed through the Internet. Then, the needed time would greatly increase since the visual content should first be transmitted and afterwards viewed. For example a video file of 2 hours duration, compressed in an MPEG-1 SIF format at 1.5 Mbits/sec, occupies about 1.3 Gbytes and thus it requires about 40 hours to be completely downloaded assuming an average throughput over the Internet of 80 Kbits/sec.

To overcome the aforementioned difficulties, a non-linear video representation scheme is introduced to enable fast browsing and cost effective video transmission. More specifically, video content is organized into different resolution levels, providing a multiscale tree structure. In particular the lowest resolution contains a coarse representation of the visual content, while the highest resolution gives the finest possible detail.

Several works have been proposed in literature for representing a video sequence in a compact way. In particular in [1] a video news magazine is presented that contains a text-based table of contents for browsing each video sequence. However, the scheme is not suitable for content-based browsing and restricts searching to the titles of the table of contents. Other approaches are oriented to video abstraction and summarization. In particular in [2] images of a video sequence are chosen at regular time intervals leading to a storyboard presentation. In [3] three-dimensional iconic cubes are constructed, namely micons, which contain the representative frame of a shot, together with camera breaks and relative duration (depth of the cube). Arman et. al. [4] extract a single frame for a shot, surrounded by irregular lines, indicating camera motion. However, single frame extraction cannot sufficiently describe the content of shots with long duration and rapid motion. Other methods deal with the construction of compact image maps or image mosaics for each shot. More specifically, in [5], the dominant object of a shot has been used for frame alignment, while in [6] a panoramic view for all frames of a shot has been depicted. However these interesting methods cannot provide satisfactory results in many real world shots, where complicated camera effects are usually encountered. In [7], a method for analyzing video and building a pictorial summary has been presented, while in [8] a fuzzy visual content representation scheme has been proposed with application to video summarization and content based indexing and retrieval. Color and depth information has appropriately been combined in [9] to summarize stereoscopic video sequences. A fast technique, which exploits the temporal variation of the feature vector trajectory has been proposed in [10]. Finally, in [11], the class separation measure presented in [12] has been used to investigate the effect of the number of classes on the visual content.

Although the aforementioned video summarization and abstraction schemes provide a more compact representation of a video sequence, they discard a significant amount of visual information. On the contrary, in an effective video browsing and navigation approach, no information should be lost. Furthermore, video summarization algorithms are also not appropriate for efficient video transmission. For instance, a video file of 2 hours duration consisting of 1000 shots is represented by 4000 key-frames, assuming that 4 key-frames on average are extracted per shot. However, this representation, apart from discarding a significant amount of visual information, it also incurs a heavy load to the network since 4000 frames require approximately 40 Mbytes.

Algorithms dealing with progressive retrieval of images or video sequences were presented in [13] and [14]. These techniques initially transmit a low resolution of an image, followed by additional residual information. In particular, in [14], the algorithm is applied to still images, which are decomposed in the space and frequency domain. In [13], extension to video sequences has been investigated using spatial-temporal filter banks. However, both approaches linearly downsample visual information on the spatial-temporal domain at fixed units of pixels or frames. As a result, such a video decomposition does not consider variations of the visual content of the sequence.

In this paper, a multiscale content-based video sequence organization is proposed for fast browsing and cost effective transmission of video files especially over IP-based networks. More specifically, each video sequence is hierarchically analyzed in the "content domain", meaning that visual information is partitioned into regions of relevant content and thus different content resolution levels are created. The scheme results in an interconnected tree structure, which allows users to easily and quickly preview video sequences at various content resolutions and zoom in on segments of their interest. Each level of the tree corresponds to a different content resolution layer and the nodes of the tree represent the regions that the sequence is partitioned at this level. Starting from the tree-root and interactively selecting the nodes of interest, a user can quickly move to the leaves of the tree, where the full video resolution is available.

The fundamental structural elements of the proposed multi-content video representation scheme are the shot representatives, the key-shots and the key-frames. A shot representative describes the visual content of a shot, a key-shot the visual content of a group of shots while a key-frame the visual content of a group of frames within a shot. For each shot, one shot representative is selected based on the distance between the feature vectors of all frames of the shot and the shot feature vector. Key-shots (key-frames) are optimally extracted for each sequence (shot) by minimizing a cross-correlation criterion. Then the extracted structural elements are properly linked to each other and to the rest of the frames, to construct the proposed multi-content video organization tree.

This paper is organized as follows. Section 2 describes the concepts of the progressive content-based video decomposition scheme. In section 3 selection of the fundamental structural elements is investigated. Section 4 describes the way that the extracted structural elements are organized, to produce the multiscale tree structure. In section 5 experimental results are presented and finally section 6 concludes the paper.

2 Overview of the Multiscale Video Content Decomposition Tree Structure

A simple example of the proposed multi-content video decomposition scheme is presented in Figure 1. Each node of the tree corresponds to a particular resolution level, while its children indicate how this content is analyzed at a higher resolution level. More specifically, the root-node contains a text description of the video sequence, for example the title. Then, a shot cut detection algorithm is applied and for each shot of the sequence, one representative frame is extracted, namely the shot representative. Gathering together all shot representatives, a set is

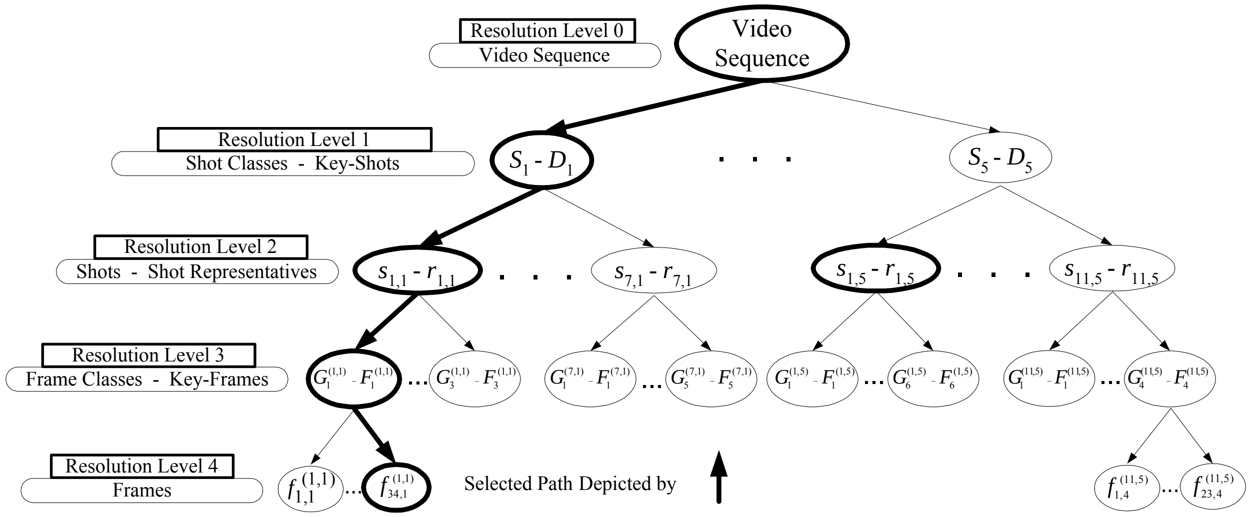


Figure 1: An example of a decomposition tree. The bold nodes correspond to nodes selected by a user (i.e., the frame $f_{34,1}^{(1,1)}$ is of particular interest).

formed (the shot-representatives' set). On the next step, key-shots are extracted from the shot representatives' set and an influence zone is estimated for each key-shot to classify the remaining shots with respect to these representatives. As a result, shot-classes are created and each class is visualized by the respective key-shot. The created shot classes (level 1 of the tree structure) are the children of the root node as is illustrated in Figure 1.

In particular let us denote as D_i , i = 1, 2…, K_s, the K_s key-shots of a sequence V and as S_i, i = 1, 2, …, K_s, the respective shot classes. Since each shot is assigned only to one class, it is held that

$$S_i \cap S_j = \varnothing \text{ with } i \neq j \text{ and } V = \bigcup_{i=1}^{K_s} S_i \ , \qquad (1)$$

Additionally, the kth shot class S_k is expressed as the union of all shots that this class contains

$$S_k = \bigcup_{i=1}^{P(S_k)} s_{i,k} \qquad (2)$$

where $s_{i,k}$ is the ith shot of class S_k and $P(S_k)$ corresponds to the number of shots that the class S_k. contains. It should be mentioned that index i of $s_{i,k}$ refers to the ith shot (element) of class S_k and not to the ith shot of video sequence V, since class S_k does not contain temporally consecutive shots.

In the presented example of Figure 1, K_s =5, meaning that five classes are constructed (S_1,…,S_5), each represented by one key-shot (D_1,…, D_5).

Each node of level 1 is further decomposed into level 2 to describe the node content in more detail. In particular, a node of level 1, which corresponds to a specific shot-class, is partitioned into the shots that this class contains. Shot $s_{i,k}$ is represented by one frame, namely the shot-representative, which is denoted as $r_{i,k}$.

Afterwards the content of each shot is further decomposed by extracting key-frames, so that the shot visual content is analyzed in more detail. The remaining frames of the shot are classified with respect to key-frames generating frame classes. In particular, let us denote as $F_i^{(l,k)}$, i=1,2,…,$K_f^{(l,k)}$ the $K_f^{(l,k)}$ key-frames of the shot $s_{l,k}$ and as $G_i^{(l,k)}$, i=1,2,..,$K_f^{(l,k)}$ the corresponding frame classes of this shot, which satisfy the following equation,

$$G_i^{(l,k)} \cap G_j^{(l,k)} = \varnothing$$

$$\text{with } i \neq j \text{ and } s_{l,k} = \bigcup_{i=1}^{K_f^{(l,k)}} G_i^{(l,k)} \qquad (3)$$

Equation (3) indicates that each frame of shot $s_{l,k}$ is assigned only to one frame-class. In a similar way, we denote as $f_{i,m}^{(l,k)}$ the ith frame of class $G_m^{(l,k)}$. Thus,

$$G_m^{(l,k)} = \bigcup_{i=1}^{P(G_m^{(l,k)})} f_{i,m}^{(l,k)} \qquad (4)$$

where $P(G_m^{(l,k)})$ returns the number of frames belonging to class $G_m^{(l,k)}$.

In the presented example, three key-frames are extracted for shot $s_{1,1}$ ($F_1^{(1,1)}, F_2^{(1,1)}, F_3^{(1,1)}$) and 34 frames are associated with key-frame $F_1^{(1,1)}$ ($f_{1,1}^{(1,1)}$,…, $f_{34,1}^{(1,1)}$).

In the proposed scheme, to indicate the non-linear way that visual information can be accessed by a user, we assume in the following that frame $f_{34,1}^{(1,1)}$ is of particular interest. In this case, initially the five key-shots (D_1,…,D_5) will be presented to the user and D_1 will be selected. Afterwards, seven new shot representatives will appear ($r_{1,1}$,…,$r_{7,1}$) corresponding to shots $s_{1,1}$,…,$s_{7,1}$ of the first shot-class S_1 respectively. The next user's selection will be the $r_{1,1}$ shot representative, leading to the three key-frames of shot $s_{1,1}$ ($F_1^{(1,1)}, F_2^{(1,1)}, F_3^{(1,1)}$). Finally the key-frame $F_1^{(1,1)}$ is chosen as the most appropriate and then the 34 frames $f_{1,1}^{(1,1)}$,…, $f_{34,1}^{(1,1)}$ are presented, one of which contains the desired visual information (frame $f_{34,1}^{(1,1)}$). In this scenario, the total number of browsed frames is 49, assuming that frame $f_{34,1}^{(1,1)}$ is reached in the first attempt. The described path from root (video sequence) to leaf (frame $f_{34,1}^{(1,1)}$) is indicated in Figure 1 with bold line.

3 Selection of Structural Elements

As mentioned, the fundamental structural elements of the proposed scheme are key-shots (level 1), shot representatives (level 2) and key-frames (level 3). In this section, we describe the way that these elements are extracted, while in Section 4 the way that these elements are linked together to form the proposed multiscale tree structure is analyzed.

3.1 Correlation Measure Formulation

A cross correlation criterion is used in our case as a similarity measure between the feature vectors of two frames or shots. More particularly let us denote as $\mathbf{g}_i$ and $\mathbf{g}_j$ two feature vectors corresponding either to a pair of frames or shots. Then the correlation coefficient of these vectors is estimated by:

$$\rho(\mathbf{g}_i,\mathbf{g}_j) = \frac{C(\mathbf{g}_i,\mathbf{g}_j)}{\sqrt{C(\mathbf{g}_i,\mathbf{g}_i)}\cdot\sqrt{C(\mathbf{g}_j,\mathbf{g}_j)}} \tag{5a}$$

with $C(\mathbf{g}_i,\mathbf{g}_j) = (\mathbf{g}_i - \mathbf{m})^T(\mathbf{g}_j - \mathbf{m})$ and

$$\mathbf{m} = \frac{1}{L}\sum_{i=1}^{L}\mathbf{g}_i \tag{5b}$$

In equation (5), $C(\mathbf{g}_i,\mathbf{g}_j)$ is the covariance of vectors $\mathbf{g}_i$ and $\mathbf{g}_j$, while $\mathbf{m}$ indicates the average feature vector over all frames within a shot or all shots within a sequence, the number of which is expressed by the variable L in equation (5b).

3.2 Selection of Shot Representatives

Initially, shot representatives are extracted and afterwards they are used for estimating the key-shots. The role of a shot representative is to give a clue of what the visual content of a shot is. For this reason let us consider that a feature vector is formulated for each frame of a shot [9]. Then, the mean shot feature vector $\mathbf{s}$ of a shot *s,* is estimated by averaging all frame feature vectors within this shot,

$$\mathbf{s} = \frac{1}{P(s)}\sum_{i=1}^{P(s)}\mathbf{f}_i \tag{6}$$

where $\mathbf{f}_i$ is the feature vector of the ith frame of shot *s* and *P*(*s*) the function which returns the number of frames belonging to this shot. In equation (6), we have omitted the subscripts in the variable *s* since we refer to any shot of the sequence.

In the proposed scheme, the correlation coefficient of equation (5a) is estimated for all pairs of gi, gj, where in this case vector (shot feature vector), while vector (feature vector of a frame within the shot s). Then, the shot representative, say r, of shot s is selected as the frame of that shot whose feature vector results in maximum correlation with respect to the shot feature vector

$$r = \{ f_i : \rho(\mathbf{f}_i,\mathbf{s}) > \rho(\mathbf{f}_j,\mathbf{s})\ \forall f_i, f_j \in s\} \tag{7}$$

Gathering the shot representatives for all shots of the sequence together, the set *SR* is constructed

$$SR = \{r_1, r_2, \cdots, r_{P(V)}\} \tag{8}$$

where r_i refers to the shot representative of the ith shot of the sequence, while $P(V)$ expresses the number of shots in the sequence.

3.3 Selection of Key- Frames/Shots

In the following, let us consider that the feature vector $\mathbf{g}_i$ corresponds either to the ith shot of a video sequence (when addressing extraction of key-shots) or to the ith frame of a shot (when addressing extraction of key-frames). Then the correlation coefficient between two shot representatives (or frames) with feature vectors $\mathbf{g}_i$ and $\mathbf{g}_j$, $i \neq j$ is given by equation (5). Thus $\mathbf{m}$ expresses the mean vector of all shot representatives or the average vector of a particular shot.

Let now U={0,1,…, L-1} be a set, containing the time instances (indices) of all shots of the sequence or frames of the shot. Let us also assume that *K* indices of *U* are selected as key-shots or key-frames.

K is the same as K_s and $K_f^{(l,k)}$, but in this case, we have omitted the subscripts and superscripts since we refer either to a shot or to a frame representative. In our approach, the K most characteristic shot representatives (key-shots) or frames (key-frames) are extracted by minimizing the cross-correlation criterion, formed using equation (5). In particular, initially an index vector $\mathbf{z}$ is formed, which contains possible indices of the K key-shots or key-frames. That is,

$$\mathbf{z} = [z_1, \ldots, z_K]^T \in Z \subset U^K \tag{9a}$$

where

$$Z = \{[z_1, \ldots, z_K]^T \in U^K : z_1 < \cdots < z_K\} \tag{9b}$$

is the subset of UK, which contains only sorted indices , in contrast to , since any re-arrangement of the elements of z results in the same representatives. Then, correlation among K shots or frames is given as:

$$J(\mathbf{z}) = J(z_1, \ldots, z_K) = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \rho(\mathbf{g}_{z_i}, \mathbf{g}_{z_j})^2 \tag{10}$$

Based on the above definition, it is clear that searching for a set of K minimally correlated feature vectors is equivalent to searching for an index vector $\mathbf{z}$ that minimizes $\underline{J}(\mathbf{z})$. As a result, the index vector, say $\hat{\mathbf{z}}$, which contains the elements of the K most representative feature vectors is given as

$$\hat{\mathbf{z}} = [\hat{z}_1, \ldots, \hat{z}_K]^T = \underset{\text{for all } \mathbf{z} \in Z}{\arg\min}\; J(\mathbf{z}) \tag{11}$$

Minimization of (11) is performed using a stochastic logarithmic search algorithm [15].

4 Viewing Elements Organization - Tree Construction

In the previous sections the problem of extracting structural elements, i.e. shot-representatives, key-shots and key-frames, has been investigated. However the extracted components should be properly organized and linked with the rest of the frames, so that the resulting structure of the video sequence is suitable for fast browsing and efficient transmission. Towards this direction shot/frame classes are constructed, each class of which uses a structural element as viewing element, corresponding to a node of the tree.

More specifically, let us recall that the optimal vector $\hat{\mathbf{z}} = [\hat{z}_1, \ldots, \hat{z}_K]^T$ contains the indices of key-shots or key-frames. Then, the remaining shots (or frames of a shot) are classified with respect to key-shots (or key-frames). This is performed by constructing an influence zone for each element $\hat{z}_j$, j=1,2,…,K of $\hat{\mathbf{z}}$ [16]:

$$\begin{aligned} IZ(\hat{z}_j) = \{ & i \in U : \rho(\mathbf{g}_i, \mathbf{g}_{\hat{z}_j}) > \rho(\mathbf{g}_i, \mathbf{g}_{\hat{z}_m}) \\ & \forall\, m \in \{1, 2, \cdots, K\} \text{ and } m \neq j\} \end{aligned} \tag{12}$$

In the previous equation, $\mathbf{g}_{\hat{z}_i}$ corresponds to the feature vector of the ith key-shot of the sequence or key-frame of a shot among K available. Thus, $IZ(\hat{z}_j)$ contains those time indices of shot representatives or frames, for which the correlation coefficient of the respective feature vectors is closer to $\mathbf{g}_{\hat{z}_i}$ than to $\mathbf{g}_{\hat{z}_m}$, $m \neq j$. Under this consideration, for each key-shot (or key-frame) $\hat{z}_j$, j=1,2,…,K, a class of shot representatives (or frames) is constructed according to its influence zone $IZ(\hat{z}_j)$. The shot and frame classes are constructed as described below.

Let us first concentrate on the case of shot class construction. Let us also assume that an index vector $\hat{\mathbf{z}}(V)$ has been estimated using equation (11), which contains the key-shots of the sequence V. Then, using the notation of section 2, shot classes S_k are constructed as:

$$S_k = \{s_i : i \in IZ(\hat{z}_k(V))\} \quad k = 1, 2, \cdots, K_S \tag{13}$$

where we denote as s_i the ith shot of the sequence V, and as $\hat{z}_k(V)$ the kth element of index vector $\hat{\mathbf{z}}(V)$. It should be mentioned that in this case, index i does not refer to the ith element of class S_k but to the ith shot of sequence V. Equation (13) indicates that the shot classes S_k are constructed by gathering together all shots of V, the indices of which are located in the influence zone of the kth key-shot. The lth shot of class S_k is denoted as $s_{l,k}$ and is obtained by renumbering all shots within this class, so that they always start from one.

Frame classes are constructed in a similar way. In this case, we refer to a shot $s_{l,k}$ and we denote as $\hat{\mathbf{z}}(s_{l,k})$ the vector, which contains the time instances of all key-frames of this shot $s_{l,k}$. Then, the mth frame class $G_m^{(l,k)}$ includes those frames of the shot $s_{l,k}$, whose indices are located in the influence zone of the mth key-frame,

$$G_m^{(l,k)} = \{ f_i^{\{l,k)} \in s_{l,k} : i \in IZ(\hat{z}_m(s_{l,k}))\} \tag{14}$$

where $f_i^{(l,k)}$ indicates the ith frame of the shot $s_{l.k}$ and $\hat{z}_m(s_{l,k})$ the mth element of index vector $\hat{\mathbf{z}}(s_{l,k})$, which contains the indices of key-frames of the shot $s_{l,k}$. The ith frame of class $G_m^{(l,k)}$ is denoted as $f_{i,m}^{(l,k)}$ and is obtained by renumbering all frames within this class so that they always start from one.

At the final step the tree structure is constructed. In particular, at each node of the tree a viewing element is presented, the type of which depends on the level of content resolution (tree-level) that the node belongs to.

Let us denote as $v(\cdot)$ an operator, which returns the viewing elements of a node. Description of the tree is made, starting from the top level and moving to the bottom. In particular, the viewing element of the root is a text description (e.g., sequence title) associated with an advertising photo (e.g. containing the actors). Thus:

$v(V)$=text-description and/or advertising photo (15)

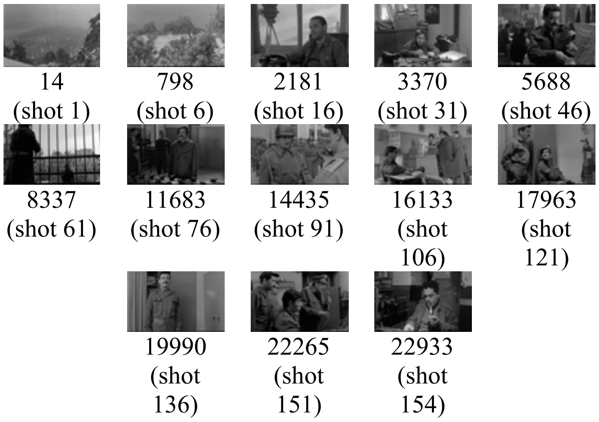

14 (shot 1) 798 (shot 6) 2181 (shot 16) 3370 (shot 31) 5688 (shot 46)
8337 (shot 61) 11683 (shot 76) 14435 (shot 91) 16133 (shot 106) 17963 (shot 121)
19990 (shot 136) 22265 (shot 151) 22933 (shot 154)

Figure 2: Shot representatives of 1 every 15 shots, for the examined film

The nodes of level 1 correspond to the shot classes and thus, the viewing elements are the respective key-shots. In particular, for a shot class S_k, we have that

$v(S_k) = D_k$ (16)

In a similar way, at level 2, each node corresponds to a shot representative and therefore the viewing element associated to shot s_k is given by

$v(s_k) = r_k$ (17)

where we recall that r_k is the shot representative of shot s_k.

The third level is comprised of frame classes G_i, each of which is represented by the respective key-frame F_i:

$v(G_i) = F_i$ (18)

Finally, the viewing elements of level 4 are the frames themselves,

$v(f_i) = f_i$ (19)

Each time a node of the tree is selected, the viewing elements of all children of this node are presented/transmitted. In this way, the content of this node is expressed in more detail, moving to a higher content resolution level.

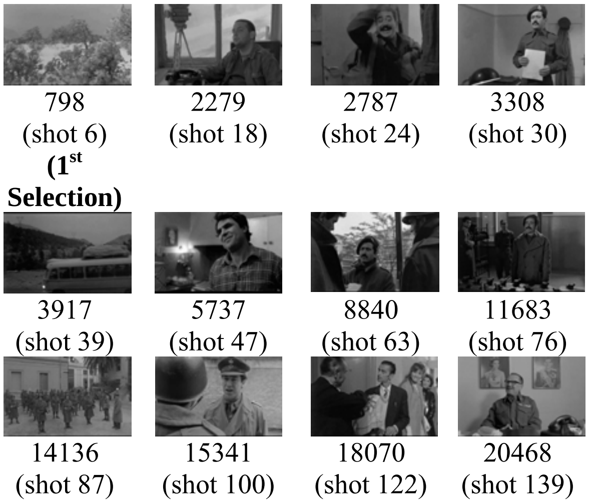

798 (shot 6) **(1st Selection)** 2279 (shot 18) 2787 (shot 24) 3308 (shot 30)
3917 (shot 39) 5737 (shot 47) 8840 (shot 63) 11683 (shot 76)
14136 (shot 87) 15341 (shot 100) 18070 (shot 122) 20468 (shot 139)

Figure 3: The shot representatives' level for the film sequence: The twelve (12) representative shots.

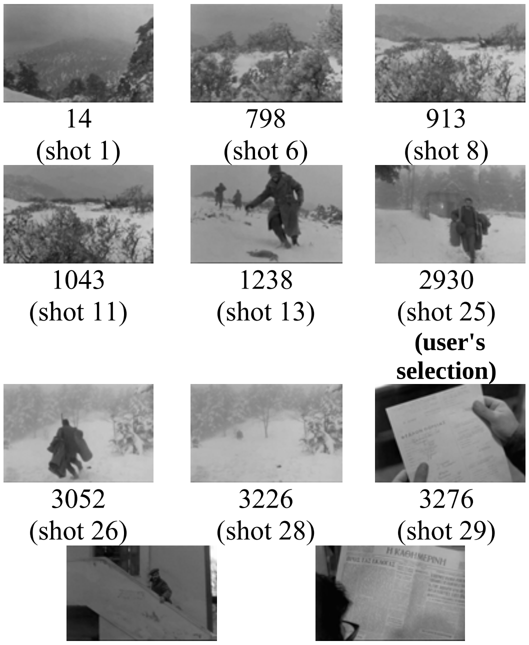
5 Experimental Results

In the following, the proposed scheme is applied to a part of a film consisting of 23,045 frames. For the film sequence 154 shots have been detected. For presentation purposes only 1 every 15 shot representatives is presented in Figure 2. Thus, set *SR* consists of 154 elements. Then, key-shots are extracted from *SR*. Figure 3 shows the 12 key-shots extracted from the film.

Let us now suppose that a user is interested in the first key-shot, i.e., the D_1. By selecting this key-shot, the viewing elements of the shot class, associated to the selected key-shot are presented. Figure 4 illustrates the shot representatives of the shots belonging to shot-class S_1.

Let us now assume that the sixth shot of class S_1, denoted as $s_{6,1}$, (which corresponds to frame 2930 of the sequence and is the shot representative of shot 25), is selected by the user for a more detailed analysis of its visual content. Then, the key frames of the selected shot are extracted and transmitted. In our case, 4 key frames have been estimated, the visual content of which is depicted in Figure 5.

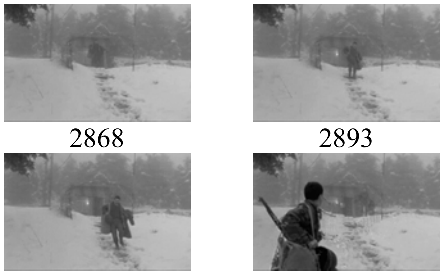
As is observed, the 4 extracted key-frames provide a good visualization of the shot content. Finally, by selecting the third key-frame of this shot (denoted as $F_3^{(6,1)}$), we reach the leaves of the tree, in which the full resolution of the sequence is available (level 4).



14 (shot 1)
798 (shot 6)
913 (shot 8)
1043 (shot 11)
1238 (shot 13)
2930 (shot 25) **(user's selection)**
3052 (shot 26)
3226 (shot 28)
3276 (shot 29)
10001 (shot 67)
16024 (shot 105)

Figure 4: Content decomposition of shot class S_1 (shot level): The shots belonging to shot class S_1.

The respective class of key frame $F_3^{(6,1)}$ contains 32 frames, the content of which is shown in Figure 6, where, for presentation purposes, only 1 every 5 frames is depicted.

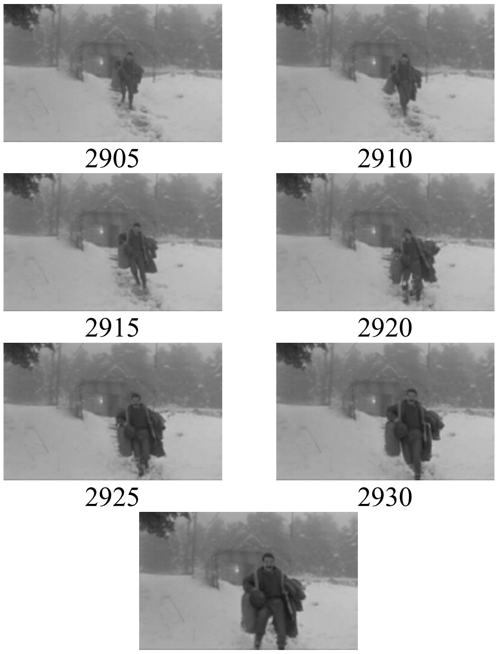


2868
2893
2913 **(user's selection)**
2964

Figure 5: The four key-frames of shot $s_{6,1}$

6 Conclusions

The conventional way of getting an overview of the visual information in a video file or detecting content of interest, is to sequentially scan each frame of a sequence. However, this approach is time consuming and usually leads to network congestion when video content should first be transmitted, e.g., over the Internet.



2905
2910
2915
2920
2925
2930
2935

Figure 6: One every 5 frames of frame class $G_3^{(6,1)}$ selected by a user in Figure 8.

In this paper, the aforementioned problem is addressed by proposing an automatic non-sequential video organization scheme. In particular, the proposed scheme is based on a progressive content-based video decomposition algorithm, which analyzes a video and organizes its visual information at different content resolution levels. Initially the low resolution of the visual information is presented, which becomes more and more specific at the following layers, creating a hierarchy from the lowest to the highest resolution level. The proposed content-based video decomposition scheme results in a tree structure organization of video files which is very suitable for interactive navigation of video sequences over communication networks, fast browsing of visual content and cost-effective / directive transmission of video segments of particular interest.

References:

[1] L. Compton and P. D. Bosco, "Internet CNN NEWSROOM: A digital video news magazine and library," in *Int. Conf. on Multimedia Computing and Systems*, 1996, pp. 296-301.

[2] M. Mills, J. Cohen, and Y. Y. Wong, "A magnifier tool for video data," in Proc. *ACM Computer Human Interface* (CHI), May 1992, pp. 93--98.

[3] S. W. Smoliar and H. J. Zhang, “Content-based video indexing and retrieval,” *IEEE Multimedia*, pp.62-72, Summer 1994.

[4] F. Arman, R. Depommier, A. Hsu and M.Y Chiu, “Content-based browsing of video sequences”, *ACM Multimedia*, pp. 77-103, Aug. 1994.

[5] M. Irani and P.Anandan,” Video indexing based on mosaic representation,” *Proceedings of the IEEE*, Vol. 86, No. 5., pp. 805-921, May 1998.

[6] N. Vasconcelos and A. Lippman, “A spatiotemporal motion model for video summarization,” Proc. of *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR), pp. 361- 366, Santa Barbara, CA, June 1998.

[7] M. M. Yeung and B.-L. Yeo, “Video visualization for compact presentation and fast browsing of pictorial content,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 7, No. 5, pp. 771- 785, October 1997.

[8] A. Doulamis, N. Doulamis, and S. Kollias, "A fuzzy video content representation for video summarization and content-based retrieval," *Signal Processing*, Vol. 80, pp. 1049-1067, June 2000.

[9] N. Doulamis, A. Doulamis, Y. Avrithis, K. Ntalianis and S. Kollias, "Efficient summarization of stereoscopic video sequences," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 10, No. 4, pp. 501-517, June 2000.

[10] A. Doulamis, N. Doulamis, and S. Kollias, “Non-sequential video content representation using temporal variation of feature vectors,” *IEEE Trans. on Consumer Electronics*, Vol. 46, No. 3, pp. 758-768, August 2000.

[11] A. Hanjalic and H. Zhang, "An integrated scheme for automated abstraction based on unsupervised cluster-validity analysis," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 8, pp. 1280-1289, December 1999.

[12] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. on Patt. Anal. Machine Intell.*, vol. PAMI-1, pp. 224-227, Aprl. 1979.

[13] J. R. Smith, "VideoZoom: Spatio-temporal video brwoser," *IEEE Trans. on Multimedia*, vol. 1, No. 2, pp. 157-171, June 1999.

[14] J. R. Smith, V. Castelli and C.-S. Li, "Adaptive storage and retrieval of large compressed images," in *Storage & Retrieval for Image and Video Databases*, VII, M.M Yeung, B.L. Yeo and C. A. Bouman Eds. *Proc. SPIE*, vol. 3656, pp. 467-487, Jan. 1999.

[15] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw Hill, 1984.

[16] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1993.