Dynamic Video Content Decomposition: An Efficient Scheme for Personalized Video Content Navigation over The Web

Nikolaos Doulamis+, Panagiotis Karagiannis*, Angelo Sifaleras* and Konstantinos Paparrizos*
* Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece
+ National Technical University of Athens, Athens, Greece email: ndoulam@cs.ntua.gr

ABSTRACT

In this paper, an adaptable hierarchical video summarization scheme is proposed which dynamically organizes video files to fit the current user's information needs and preferences. The architecture non-linearly (non-sequentially) decomposes a video sequence into content resolution levels of representative shots/frames based on a tree structure scheme. However, instead of a static tree construction, video decomposition is dynamically performed to fit the current user's information needs and preferences. In particular, each time the user selects a tree node for further navigation, the node is considered as relevant. On the contrary, irrelevant nodes are marked as those nodes in which the user backtracks to previous levels. In the sequel, an on-line learning strategy is adopted for estimating the user's profile according to the set of relevant/irrelevant selected nodes. Then, the decomposition tree is updating to meet user's needs. Experimental results and comparisons with other conventional static schemes indicate the outperformance of the proposed architecture compared to other approaches.

1. INTRODUCTION

Video content delivery in a cost effective and quality guaranteed manner over heterogeneous and distributed platforms, such as the Internet, still remains one of the most changeling problems in the image processing and communication society. This is due to the fact that digital video, even in compressed domain, imposes very large bandwidth requirements. It takes hours to completely download a video file from an Internet connection of average rate.

The traditional organization of a digital video as a sequence of consecutive frames, each of which corresponds to a constant time interval, while being appropriate for viewing a video in a movie-mode, it is not efficient for the new multimedia technologies and knowledge management over the Web. Currently, video information can be delivered using either video file downloading or video streaming, techniques that are both tedious and time consuming.

To overcome the above mentioned problems, algorithms for video summarization and video content decomposition have been proposed in the literature [1]- [15]. The first approaches deal with the construction of a small but meaningful "abstract" of video information, which can be first delivered before transmitting the entire video data. The second approaches aim at re-organizing video in a non-linear (non-sequential) way so that a low video resolution is first delivered and then video quality gradually enhances so that the users are allowed to easily and quickly preview video sequences at various resolutions and zoom in on segments of their interest.

Construction of compact image maps or image mosaics for each shot has been proposed in the literature for video summarization. More specifically, in [1], the dominant object of a shot has been used for frame alignment, while in [2] a panoramic view for all frames of a shot has been depicted. These approaches, however, yield efficient performance only for simple scenes which is not the case of real life video sequences. Video summarization using the motion intensity is presented in [3]. Furthermore, in [4] a method for analyzing video and building a pictorial summary has been presented, while in [5] a fuzzy visual content representation has been proposed with application to video summarization and content based indexing and retrieval. Color and depth information have been appropriately combined in [6] to summarize stereoscopic video sequences. The class separation measure presented in [7] has been used to investigate the effect of the number of classes on visual content. Video content summarization using concepts of the interpolation theory has been presented in [8].

Some approaches for hierarchical video summarization can be considered the works of [9], [10]. In particular, in [9] video frames are decomposed in space and frequency domain, while in [10], extension of [9] has been investigated using spatial-temporal filter banks. However, both approaches decompose (downsample) visual information in a linear way without taking into account any knowledge about video content. Hierarchical video summarization has been also adopted in the framework of the MPEG-7 standard [11], through the HierarchicalSummary Description scheme. The standard provides a syntax for hierarchically decomposing a video file and suggests an algorithm for constructing hierarchical summaries. The technique uses only key-frame organization and clusters video segments according to the visual content and temporal coherency [11]. Other approaches include the work of [12], where a hierarchical video shot classification scheme has been proposed, and the work of [13], where hierarchical brushing of video collection has been reported. In [14] a hierarchical video decomposition scheme is presented for sport scene analysis.

In [15], a content-based video decomposition scheme has been presented to organize visual information at different levels of content hierarchy. The method represents video information in a tree structure, the level of which corresponds to a content resolution, while tree nodes to the video segments that are partitioned at this level. Shots and frames are considered as the basic elements for the hierarchical video decomposition. In addition, optimal extraction of key-shot and frame representatives is accomplished along with the optimal selection of the number of nodes of the tree so that the average transmitted information is minimized.

The main drawback, however, of all the above mentioned, is that they use a static *decomposition of the video sequences.* A more effective video organization is accomplished if video is dynamically organized so as to re-organize the structure of video segments according to the user current information needs and preferences.

These difficulties are addressed in this paper by proposing a dynamic hierarchical video organization based on a reconfigurable architecture. Reconfiguration is performed using an on-line learning strategy, which estimates the current user's information needs. More specifically, the structure of [15] is adopted to organize video files at different levels of content hierarchy. However, in the presented architecture the structure of the tree is *dynamically* updating. Upon

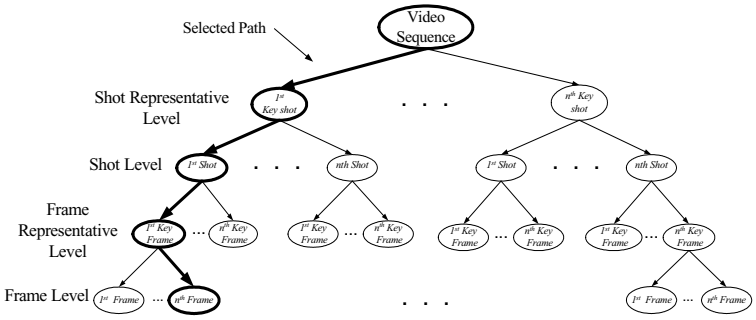

Figure 1: Hierarchical Video Content Decomposition.

user's navigation through the tree structure, a learning algorithm estimates the user's profile and in the sequel video is re-organized to satisfy user's needs. As the user zoom in on video segments of his/her interest, the selected tree nodes are considered as relevant. On the contrary, upon a user's backtrack to previous levels of the tree, the respective nodes are irrelevant. This information is used for updating the user's profile. In the following, video is re-organized (tree structure updating) so that video content that is more relevant to user's needs is clustered together.

2. Hierarchical Video Content Decomposition

2.1 Content Tree Construction

In this section, we present the tree structure used for hierarchically organizing video files into different levels of content resolutions. In particular, four content resolution levels are adopted; the shot representatives, the shots, the frame representatives and the frame themselves. An example of the adopted tree-structure video decomposition scheme is shown in Figure 1.

The first level of content hierarchy is a description about the video file to be decomposed (e.g. video title). This level corresponds to the root of the tree. At the following content resolution, shot representatives are considered as the basic elements for video file organization. This means that the video sequence is represented by a set of characteristic shots. At the second level of content hierarchy, the basic elements are the shots associated with a representative shot. Therefore, upon a user's selection of a representative shot, all the respective shots (related to the selected representative) are transmitted (expansion to the second content resolution level).

Then, for each shot, the representative frames (key-frames) are considered as the elements of the third resolution level. This means that upon a user's selection for a particular shot, video file is decomposed into the set of key-frames of this shot. Finally, the fourth level of content hierarchy includes the video frames themselves associated with a particular key-frame. For example, in Figure 1, to access a frame of video sequence the path illustrated in bold line should be selected by the user.

2.2 Dynamic Representative Class Construction

A) Shot/Frame Representatives Extraction

Let us denote in the following as $\mathbf{g}_i$ a feature vector, which corresponds either to the ith shot of a video sequence (in case that we refer to a shot representative) or to the ith frame of a shot (in case that we refer to a frame representative). Vector $\mathbf{g}_i$ includes several descriptors able to represent the content of the respective shot/frame as in [15].

Let us also denote as K either the total number of shots for a video sequence or the total number of frames for a shot and as $U=\{1,2,\ldots,K\}$ a set, which contains the time instances (indices) of all shots or frames of a video sequence. Let us also assume that P indices of U are selected as representatives. The number P (nodes of the tree) is optimally estimated so that the total "entropy" measured as the difficulty for a user to find a video segment of his/her interest is minimized [15].

In the approach of [15], the most characteristic shots or frames are extracted by minimizing a cross-correlation criterion among the P *possible* vectors as,

$$G(\mathbf{x}) = G(x_1,\ldots,x_P) = \frac{2}{P(P-1)} \sum_{i=1}^{P-1} \sum_{j=i+1}^{P} \rho(\mathbf{g}_{x_i}, \mathbf{g}_{x_j})^2 \qquad (1)$$

where as $\mathbf{x} = (x_1,\ldots,x_P)$, we denote an index vector of possible indices of the P representative shots or frames and as $\rho(\mathbf{g}_{x_i}, \mathbf{g}_{x_j})$ the correlation function of the feature vectors $\mathbf{g}_{x_i}$ and $\mathbf{g}_{x_i}$. Based on the above definition, it is clear that searching for a set of P minimally correlated feature vectors is equivalent to searching for an index vector $\mathbf{x}$ that minimizes $G(\mathbf{x})$. Minimization of (1) is performed using an optimization algorithm similar to the one presented in [15].

B) Personalized Shot/Frame Representative Extraction

The above mentioned algorithm yields a *static* estimation of shot/frame representatives. For a given video sequence, the shot / frame representatives cannot change for a given content description.

To allow a dynamic estimation of the representative shots/frames, a degree of importance is assigned to each visual descriptor. Such a policy permits modification of the shot/frame representatives with respect to the user's needs. More specifically, equation (1) is written as follows

$$G(\mathbf{x}) = G(x_1,\ldots,x_P) =$$
$$= \frac{2}{P(P-1)} \sum_{i=1}^{P-1} \sum_{j=i+1}^{P} \rho(\mathbf{w}.*\mathbf{g}_{x_i}, \mathbf{w}.*\mathbf{g}_{x_j})^2 \qquad (2)$$

where operator *'.*'* indicates the element by element multiplication of two vectors. Vector $\mathbf{w}$ corresponds to the weight vector, which includes the degree of importance of the visual descriptors. Initially, equal values are assigned to all elements of the weight vector $\mathbf{w}$, meaning that all descriptors are of the same importance. Instead, during user's navigation, the weight vector is modified to fit the actual user's information needs. The adopted method for the weight adaptation is discussed in section 3.

Using equation (2) and the minimization algorithm presented in [15], we can estimate the optimal shot/frame representatives $\hat{\mathbf{x}}$ as

$$\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_P) = \arg\min_{\text{for all } \mathbf{x}} G(\mathbf{x}) \quad (3)$$

C) Personalized Shot/Frame-Class Construction

Having estimating the shot/frame representatives, the next step is to dynamically construct the shot/frame classes. For this reason, an influence zone is first defined for each, say $\hat{x}_k$, out of the P shot/frame representatives as follows

$$Z(\hat{x}_k) = \{i \in U : \rho(\mathbf{w}.*\mathbf{g}_i, \mathbf{w}.*\mathbf{g}_{\hat{x}_k}) > \rho(\mathbf{w}.*\mathbf{g}_i, \mathbf{w}.*\mathbf{g}_{\hat{x}_m}) \quad \forall\, m \text{ with } m \neq k\} \quad (4)$$

In previous equation, $\mathbf{g}_{\hat{x}_k}$ and $\mathbf{g}_{\hat{x}_m}$ correspond to the feature vector of the kth and mth representative shot of the sequence (or frame of a shot) among all P available, and $\mathbf{w}$ to the weight vector.

Let us first concentrate on the case of shot class construction. Let us also assume that an index vector $\hat{\mathbf{x}}(V)$ has been estimated [using equation (3)], which contains the shot representatives of sequence V. As can be seen, we have added, in this case, the dependence of $\hat{\mathbf{x}}$ on V to indicate that this vector contains the time instances of representative shots, instead of frames. Then, using equation (4), the shot classes S_k are constructed by gathering together all shots s_i of a video file, the indices of which belong to the same influence zone.

$$S_k = \{s_i : i \in Z(\hat{x}_k(V))\} \text{ with } k=1,2,\ldots,P \quad (5)$$

Similarly, the frame classes are constructed.

3. USER'S PROFILE ESTIMATION

In this section, we present the algorithm used for automatically estimating the current user's information needs. In other words, the weight vector $\mathbf{w}$, which includes the degree of importance of visual descriptors, is computed.

To implement this, initially, we need to construct a set of relevant / irrelevant data. In particular, in our case, upon a user's selection for a shot or frame for further decomposition, the respective tree node is considered as *relevant.* On the contrary, each time the user backtracks the respective selection is marked as *irrelevant*.

3.1 Learning Strategy

Let us assume that m tree nodes have been selected by the user for further decomposition. These elements are considered as relevant. Let us denote as $\mathbf{y}_i$ i=1,…,m, the feature vectors of these elements. Vectors $\mathbf{y}_i$ coincide with the feature vectors $\mathbf{g}_i$ of shot/frames of the selected nodes. In case that a node is selected as irrelevant, the respective feature vector is multiplied by -1.

The concept of the adopted learning strategy is that if a particular feature element, say for example the $y_{i,k}$, captures the user's information needs and preferences, then the values of the elements $y_{i,k,}$ for all i=1,2,…,m will be consistent and thus the standard deviation of the respective feature element over all selected m relevant samples,

$$\sigma_k = \frac{1}{m-1}\sqrt{\sum_{i=1}^{m}(y_{i,k} - \mu_k)^2} \quad (6)$$

should be small [16]. In equation (6), μ_k is the average value of the kth feature element over the m selected samples, i.e., $\mu_k = (1/m)\cdot\sum_{i=1}^{m} y_{i,k}$. Instead, large values of σ_k, indicate that the respective feature element is not of user's interest. As a result, the weight vector elements w_k of $\mathbf{w}$ are estimated as follows

$$w_k = \frac{1}{\sigma_k} \quad (7)$$

This degree of relevance w_k re-configures the tree structure with respect to the actual users' information needs. In particular, taking into account the relevance weight as in equations (2,4), a new tree construction is activated and a new video decomposition is accomplished resulting in a dynamic tree structure.

4. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed dynamic hierarchical video decomposition scheme and compare it with conventional static approaches using an objective criterion, which expresses the "difficulty" for a user to locate relevant video segments.

In a hierarchical video organization, the efficiency is measured as the number of tree nodes (information) that should be selected in order to locate a frame of interest.

$$E_D = N(\textit{selected tree nodes}) \quad (8)$$

where function $N(\cdot)$ returns the number of the argument.

On the contrary, in a sequential video organization scheme, the average number of information transmitted equals the *half of the total number of frames of the video sequence.* The former statement is true only under the assumption that all frames present the same probability to be selected as interest by a user. Therefore,

$$E_S = N_V / 2 \quad (9)$$

where E_S refers to the efficiency in the linear case and N_V to the total number of frames of a video file V.

Then, the improvement ratio, say IR, of using the hierarchical video decomposition approach and a linear organization scheme is computed as

$$IR = E_D / E_S \quad (10)$$

To estimate the improvement ratio IR, three MPEG coded video sequences each of duration of 45 minutes (total duration of 2 hours and 15 minutes) have been used. The sequences have been selected to represent different thematic areas. A randomly selected frame of the sequences is considered each time as of user's interest and the transmitted information required for being accessed is calculated. The experiment is conducted by submitting 3,000 randomly selected frames of interest and then estimating the average transmitted information.

Video Representation Algorithms	IR
The proposed Method	93.1
The Method of [15]	87.20
The Method of [11] (MPEG-7)	67.42
The Method of [3]	20.18
The Method of [9]	42.30

Table I. The improvement ratio of the proposed method compared with other static video decomposition schemes.

Table I presents the results obtained. In this table, we have also presented the results for other static video decomposition schemes. In the proposed dynamic video decomposition scheme each time a user finds a frame of interest, the tree –structure is updated (according to the set of relevant/irrelevant tree nodes) for the next searching.

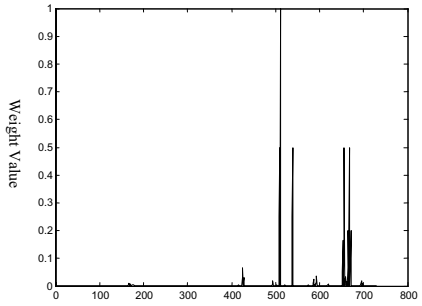

Figure 2: An example of the weight modification.

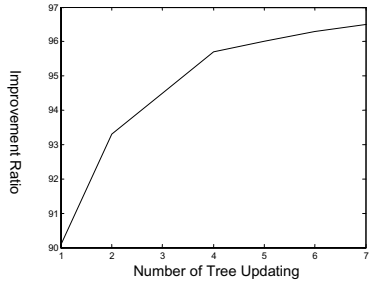

Figure 3: The IR versus the number of the tree updating.

As is observed, the proposed dynamic approach provides higher improvement ratio (smaller number of information required for accessing a frame of interest). This is due to the fact that, the proposed video decomposition is more flexible than the static approaches (The method re-organizes a video so that the user's preferences are fulfilled).

Figure 2 illustrates the weights as they have been estimated by equation (7). As is observed, only some regions are of high importance to the user's needs. These descriptors have a major impact to video organization than the remaining ones.

In Figure 3, we present the improvement ratio (IR) versus the number of the tree-updating. In this scenario, we assume that all the 3000 queries are applied for each updating of the tree. Instead in Table I, the IR has been computed by modifying the tree structure each time the user finds the frame/shot of interest. As is observed, as the number the tree-updating increases, the IR increases too. However, the increase rate decreases.

5. REFERENCES

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

[2] 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.

[3] J. Nam and A. H. Tewfik, "Video Abstract of Video," Proc. of *the IEEE Inter. Workshop on Multimedia Signal Processing*, pp. 117-122, Copenhagen, Denmark, Sept. 2000.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] N. Doulamis, A. Doulamis and K. Ntalianis, "An Optimal Interpolation-based Scheme for Video Summarization," *IEEE Inter. Conf. on Multimedia and Expo* (ICME), Lausanne, Switzerland, August 2002.

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

[10] 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.

[11] ISO/IEC JTC 1/SC 29/WG 11/N3964,N3966**,** "Multimedia Description Schemes (MDS) Group", March 2001, Singapore.

[12] J. Fan, X. Zhu, Ahmed K. Elmagarmid and W. G. Aref, "ClusterView: A Hierarchical Video Shot Classification System", *IEEE Trans. on Multimedia*, Vol.4, Dec. 2002.

[13] D. Ponceleon, A. Dieberger "Hierarchical Brushing in a Collection of Video Data", *Proceedings of the 34th Hawaii International Conference on System Sciences, 2001.*

[14] A. Ekin and A. Tekalp, "Generic Play-Break Event Detection for Summarization and Hierarchical Sports Video Analysis," *Proc of the IEEE International Conference on Multimedia and Expo (ICME)*, Vol. 1, pp. 169-172, July 2003.

[15] A. Doulamis and N. Doulamis, "Optimal Content-based Video Decomposition for Interactive Video Navigation over IP-based Networks," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 14, No. 6, pp. 757-775, June 2004.

[16] Y. Rui, T. S. Huang, M. Ortega and S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," *IEEE Trans. Circuits. Systems for Video Technology*, Vol. 8, No. 5, pp. 644-655, Sept. 1998.