AN ADAPTIVE APPROACH TO VIDEO INDEXING AND RETRIEVAL USING FUZZY CLASSIFICATION

Yannis S. Avrithis, Anastasios D. Doulamis, Nikolaos D. Doulamis and Stefanos D. Kollias

National Technical University of Athens, Department of Electrical and Computer Engineering
Heroon Polytechniou 9, 157 73 Zografou, Greece
E-mail: {iavr, adoulam, ndoulam}@image.ntua.gr, stefanos@cs.ntua.gr

ABSTRACT

An integrated framework for content-based indexing and retrieval in video databases is presented in this paper, which has the capability of adapting its performance according to user requirements. Video sequences are represented by extracting a small number of key frames or scenes and constructing multidimensional feature vectors using fuzzy classification of color, motion or texture segment properties. Queries are then performed by employing a parametric distance between feature vectors, and adaptation is achieved by estimating distance parameters according to user requirements, resulting in a content based retrieval system of increased performance and flexibility.

1. INTRODUCTION

The increasing amount of digital image and video data has stimulated new technologies for efficiently searching, indexing, content-based retrieving and managing multimedia databases. The traditional text-based approach to accessing image or video information has the drawback that it is difficult to characterize the rich content of an image or video using text [13]. Thus, with the growing use of digital acquisition and storage, a number of applications will require content-based access to their data. For this reason, the MPEG group has recently begun a new standardization phase (MPEG-7) for multimedia content description interface [11]. This standard will specify a set of content descriptors that can be used to describe any multimedia information.

Many techniques have been developed in this research area and many image/video retrieval systems have been built. The active research effort has been reflected in many special issues of leading journals dedicated to this topic [14, 15]. Examples of products, which are now in the first stage of commercial exploitation, include the Virage, VisualSEEK, Photobook and QBIC [6] prototypes. However, these commercial prototypes are mainly restricted to still images and use simple features, such as color histogram, to perform their queries. Moreover, these systems cannot be easily extended to video databases since it is practically impossible to perform queries on every video frame. Furthermore, due to the strong temporal correlation of video frames, examination of each frame is very inefficient.

Color image retrieval has been examined in [10] based on a hidden Markov model. Extraction of detailed image regions for indexing and retrieval has been proposed in [1]. Object modeling and segmentation for indexing in video databases has been reported in [7] while single frame extraction based on the frame properties has been proposed in [2] to perform the queries. A new progressive resolution motion indexing has been presented in [12] using 3-D wavelet decomposition of video sequences as well as rigid polygonal shapes. An approach for automatic video segmentation and content-based retrieval based on a temporally windowed principal component analysis of a subsampled version of a video sequence has been reported in [8].

In the context of this paper, we propose an integrated framework for content-based indexing and retrieval in video databases, which has the capability of adapting its performance according to user requirements. In particular, in the first stage of our method a collection of a small number of key frames or scenes is extracted, that provides sufficient information about the video sequence under examination, in order to reduce the strong temporal correlation of video frames. Such video representation is necessary since redundant information is rejected and video queries can be performed faster and more efficiently [3]. Then, queries are performed based on this small but meaningful collection of frames.

Video processing and image analysis techniques are applied to each video frame for extracting color, motion and texture information. Color information is extracted by applying a hierarchical color segmentation algorithm to each video frame. Consequently, apart from the color histogram of each frame additional features are collected concerning the number of color segments, their location and their respective shape and size. Motion information is also extracted in a similar way by using a motion estimation and segmentation algorithm.

All the above features are gathered in order to form a multidimensional feature vector used for performing video queries. Since similar frames can be characterized by different color or motion segments due to imperfections of the segmentation algorithms, a fuzzy representation of feature vectors is adopted in order to provide more robust searching capabilities. In particular, we classify color as well as motion and texture segments into pre-determined classes forming a multidimensional histogram and a degree of membership is allocated to each category so that the possibility of erroneous comparisons is eliminated.

Traditional searching algorithms use a metric distance to find to the best *M* frames for a given user query. However different applications or different users may require different types of indexing such as color, motion or texture indexing. This means that some elements of the multidimensional feature vectors (e.g., the motion elements for motion indexing) should be taken into account to a higher or lower degree. Therefore a parametric distance has been adopted in this paper to increase the flexibility of the system in these cases. Moreover, a recursive algorithm is proposed for properly estimating the distance parameters according to the application or user requirements. This approach increases the performance and the flexibility of the proposed content-based retrieval system.

2. VIDEO CONTENT REPRESENTATION

Video content representation is performed using several modules, which exploit color, motion and texture information. The procedure is performed in a way similar to [4] and is briefly discussed in the sequel.

Scene Cut Detection. The first stage of the feature extraction procedure includes a scene cut detection technique, in order to locate the main shots of a video stream. In our approach scene cut detection is achieved by computing the sum of the block motion estimation error over each frame and detect frames for which this sum exceeds a certain threshold [4].

Color/Motion Segmentation. Color and motion segmentation provides a powerful representation of each video frame, more oriented to the human perception. In general, the number, size, shape and location of objects as well as their color, motion, or texture characteristics give more meaningful information for an image than raw pixels. Thus, a color and motion segmentation technique is applied to each video frame. Block resolution is adopted both for reducing computational complexity and for exploiting information, which already exists in the MPEG coding standard. To avoid oversegmentation problems we have proposed a hierarchical block-based segmentation algorithm described in [4]. Apart from information provided by color or motion segmentation other features are also included in the feature vector, such as information provided by color and motion histograms or appropriate ac coefficients of the DCT transform.

Key Frame/Scene Extraction. Using color and motion segment information, a limited number of key frames or scenes that provide sufficient information about the content of a video sequence is extracted. This can be used for reduction of the amount of stored information that is necessary in order to provide search capabilities in a multimedia database. Moreover, instead of performing a query on all available video frames, one can only consider the selected ones, because they include most information about the content of the database. The proposed method is similar to that reported in [5]. In particular, an unsupervised classification is first adopted for scene selection. Our approach is based on the generalized Lloyd or *k*-means algorithm. After extracting the most representative scenes, the next step is to select the key frames within each one of the selected scenes. This is achieved by minimizing a correlation criterion, so that the selected frames are not similar to each other. In particular, a set of minimally correlated frames is extracted from each scene. The query mechanism is then applied on the above key frames or scenes, which contain a very small amount of the data but retain video content as much as possible.

3. FEATURE VECTOR FORMULATION UNIT GENERATION

All of the above frame features are gathered in order to form a multidimensional feature vector which is used for collection of information content for each frame. Properties of color or motion segments cannot be used directly as elements of feature vectors, since its size will differ between frames. To overcome this problem, we classify color as well as motion segments into pre-determined classes, forming a multidimensional histogram. To eliminate the possibility of classifying two similar segments at different classes, causing erroneous comparisons, a degree of membership is allocated to each class, resulting in a fuzzy classification [9].

This kind of classification is illustrated in Figure 1 for the simple case of a single one-dimensional feature *x*, normalized between 0 and 1 (e.g., normalized segment size). A fuzzy partition of the feature space [0,1] into *Q*=5 classes is defined by using *Q* membership functions $\mu_n(x) \in [0,1]$, $n = 1, \ldots, Q$. Triangular membership functions with 50% overlap between successive partitions are used in Figure 1, but their exact shape and overlap percentage can be greatly varied. Using this partition scheme for feature *x*, a fuzzy histogram can be constructed from a large number of feature samples, corresponding to different image segments. Moreover, this histogram can be meaningful even when the total number of segments is small, since similar features always produce similar classification results.

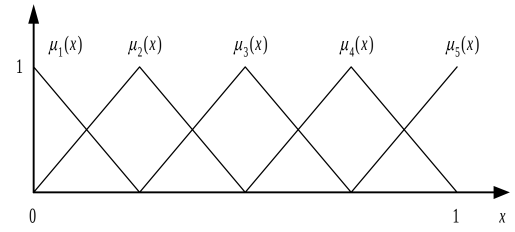

Figure 1. One-dimensional fuzzy classification.

In the more general case of multiple segment properties (such as size, color and motion), multidimensional classification is applied. Let $P(S_i)$, $\mathbf{c}(S_i)$ and $\mathbf{l}(S_i)$ denote the size, color and location of the *i*-th color segment S_i. The $L\times1$ vector

$$\mathbf{x}^{(i)} = [P(S_i)\ \mathbf{c}(S_i)\ \mathbf{l}(S_i)]^T \tag{1}$$

then fully describes the properties of segment S_i using a total of *L* segment features. A motion segment is also described by a similar vector as that in (1) with the difference that the color properties are substituted for the motion ones. Each feature space is then partitioned into *Q* regions and a partition index $n_j \in \{1,2,\ldots,Q\}$ is assigned to the *j*-th feature element, $x_j^{(i)}$, of $\mathbf{x}^{(i)}$. The degree of membership of segment S_i into the *L*-dimensional class $\mathbf{n} = [n_1 \ldots n_L]^T$ is defined as

$$M_i(\mathbf{n}) = \prod_{j=1}^{L} \mu_{n_j}(x_j^{(i)}) \quad \in [0,1] \tag{2}$$

where $\mu_{n_j}(x_j^{(i)})$ is the degree of membership of feature $x_j^{(i)}$ in partition n_j. The sum, over all segments, of the corresponding degrees of membership results in a fuzzy classification of a whole frame in class $\mathbf{n} = [n_1 \ldots n_L]^T$:

$$F(\mathbf{n}) = \sum_{i=1}^{K} M_i(\mathbf{n}) = \sum_{i=1}^{K} \left\{ \prod_{j=1}^{L} \mu_{n_j}(x_j^{(i)}) \right\} \tag{3}$$

where K is the total number of segments of the frame. The above summation actually corresponds to a multidimensional histogram, using segments S_i (or equivalently features $\mathbf{x}^{(i)}$) as samples. Finally, the frame feature vector is formed by gathering values $F(\mathbf{n})$ for all categories $\mathbf{n}$, i.e., for all combinations of indices $n_1, \ldots, n_L \in \{1,2,\ldots,Q\}$, resulting in a total of $N=Q^L$ feature elements.

Global frame characteristics, obtained through global frame analysis, are also included in the feature vector. In particular, the color histogram of each frame is calculated using YUV coordinates for color description and the average texture complexity is estimated using the high frequency DCT coefficients of each block derived from the MPEG stream. Finally, a scene feature vector, which characterizes a whole scene, is constructed by calculating the mean value of feature vectors over the whole duration of a scene.

4. VIDEO QUERIES

Once the feature vector is formed as a function of time, a video database can be searched in order to retrieve frames which possess particular properties, such as dark frames, frames with a lot of motion and so on. The feature vector space is ideal for such comparisons, as it contains all essential frame properties, while its dimension is much less than that of the image space. Moreover, a dramatic reduction is achieved in the number of frames that are required for retrieval, browsing or indexing. Instead of examining every frame of a video sequence, queries are performed on a very small set of key frames, which provide a meaningful representation of the sequence.

For each still frame or video scene (small sequence of frames) that is given as input by a user, color and motion information is extracted using color and motion segmentation. Then, the segment features are gathered into a fuzzy representation scheme and the search engine is activated. In particular, the whole set of key frames/scenes of the video database is searched for frames or scenes that possess similar properties with the input frame/scene, and the best M frames/scenes of the database are selected using a parametric distance metric and then provided to the user. The search is implemented by calculating the feature vector $\mathbf{x}$ of the input frame (scene) and performing a comparison between $\mathbf{x}$ and feature vectors $\mathbf{y}$ of the key video frames (scenes). The parametric (weighted) distance function between $\mathbf{x}$ and $\mathbf{y}$ is defined as follows:

$$d_{\mathbf{w}}(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{N} w_j (x_j - y_j)^2 = \sum_{j=1}^{N} w_j e_j^2 \tag{4}$$

where $\mathbf{w}$ is an N×1 weight vector, $\mathbf{e}=\mathbf{x}-\mathbf{y}$ is an error vector and x_j, y_j, w_j and e_j, are the elements of vectors $\mathbf{x}$, $\mathbf{y}$, $\mathbf{w}$ and $\mathbf{e}$ respectively. The set of M key frames (scenes) corresponding to the M feature vectors $\mathbf{y}_i$, $i=1,\ldots,M$ with the M minimum distances $d_{\mathbf{w}}(\mathbf{x},\mathbf{y}_i)$ is returned as a query result to the user.

In the proposed content-based retrieval application, the user can interact to the query process. Therefore, he is able to select a subset of the query results, that is, the m frames out of M, which he considers that are the most satisfactory according to his query. In this case, the remaining M-m frames are assumed to have been erroneously retrieved. The weights $\mathbf{w}$ of the distance are adapted in this case according to the correctly selected frames in order to weigh some feature components to a higher degree and some others to a lower. As a result, in the following phase of the query mechanism, frames that are closer to the user requirements are selected since the parameters of the metric distance have been properly adapted.

Without loss of generality, let $\mathbf{y}_i$, $i=1,..,m$ ($m<M$) be the feature vectors of the frames (scenes) selected by the user. Then the distances between $\mathbf{x}$ and $\mathbf{y}_i$, $i=1,..,m$ should be minimized while the distances between $\mathbf{x}$ and $\mathbf{y}_i$, $i=m+1,..,M$ should be maximized for future queries. The cost function is thus defined as

$$J(\mathbf{w}) = \sum_{i=1}^{m} d_{\mathbf{w}}(\mathbf{x},\mathbf{y}_i) - \sum_{i=m+1}^{M} d_{\mathbf{w}}(\mathbf{x},\mathbf{y}_i) \tag{5}$$

and is minimized with respect to w, subject to the constraint that the magnitude of $\mathbf{w}$ is constant, since distance metrics are only used for comparisons, therefore weights $\mathbf{w}$ of different magnitudes will produce the same results. Without loss of generality, let $\|\mathbf{w}\| = 1$:

$$\hat{\mathbf{w}} = \arg\min_{\|\mathbf{w}\|=1} J(\mathbf{w}) \tag{6}$$

This minimization is performed by setting $\partial J(\mathbf{w}) / \partial w_k = 0$ for $k=1,\ldots,N$, and the result is

$$\hat{w}_k = A_k \left(\sum_{l=1}^{N} A_l^2 \right)^{-1/2}, \quad k = 1,\ldots,N \tag{7}$$

where

$$A_k = \sum_{i=1}^{m} (x_k - y_k^{(i)})^2 - \sum_{i=m+1}^{M} (x_k - y_k^{(i)})^2 \tag{8}$$

and $y_k^{(i)}, \quad k=1,\ldots,N$ are the elements of vector $\mathbf{y}_i$. The above minimization procedure actually calculates the optimal weights $\mathbf{w}$ for a specific input frame (scene) with feature vector $\mathbf{x}$. In the more general case of multiple, consecutive queries, the input and output vectors $\mathbf{x}$ and $\mathbf{y}_i$ can be considered as discrete time sequences $\mathbf{x}(n)$ and $\mathbf{y}_i(n)$ respectively. In this case we introduce a "memory" factor λ ($0<\lambda<1$) by which we multiply previous optimization results. Past query results and their respective weight adaptations are thus taken into account to a small degree, when adapting the current weight parameters. The above equations are modified as follows:

$$\hat{w}_k(n) = B_k(n) \left(\sum_{l=1}^{N} B_l^2(n) \right)^{-1/2}, \quad k = 1,\ldots,N \tag{9}$$

where

$$B_k(n) = \sum_{j=0}^{\infty} \lambda^j A_k(n-j) \qquad (10)$$

$$A_k(n) = \sum_{i=1}^{m} (x_k(n) - y_k^{(i)}(n))^2 - \sum_{i=m+1}^{M} (x_k(n) - y_k^{(i)}(n))^2 \qquad (11)$$

Furthermore, calculation of factors $B_k(n)$ reduces to the recursive equation

$$B_k(n) = A_k(n) + \frac{1}{\lambda} B_k(n-1), \quad k = 1, \dots, N \qquad (12)$$

This recursive implementation of the adaptation scheme results in a great reduction of the required time consumption for the parameter update. Moreover, this recursive implementation also handles previous knowledge of the system and as a result, during a real-life operation, the distance parameters are not updated from scratch each time.

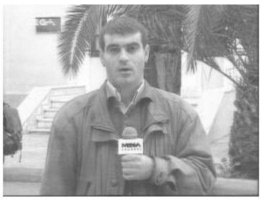

Figure 2. User input query.

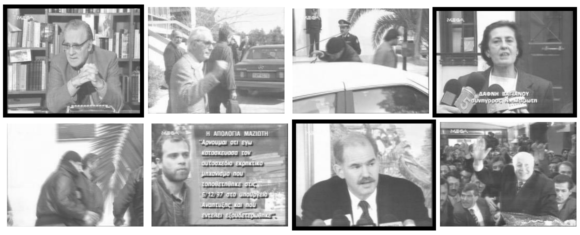

Figure 3. Query results for M=8. A subset of m=3 frames is selected by the user (shown with black borders).

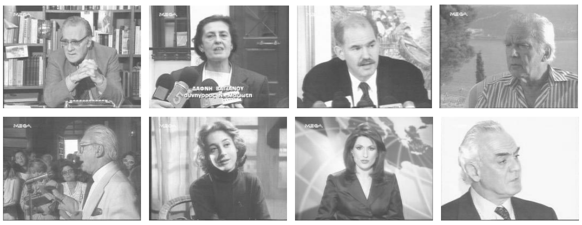

Figure 4. Query results after adaptation of weight parameters.

5. Experimental Results

The proposed algorithms were integrated into a system that was tested using several video sequences from a TV news reporting video database of total duration 145 minutes seconds (approximately 217500 frames). Triangular membership functions with 50% overlap between successive partitions were used in the experiments, with Q=3 partitions for each feature, resulting in a total of N=3^8=6561 elements in each feature vector. The results are presented in the following Figures.

6. REFERENCES

[1] D. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos, "Extraction of Detailed Image Regions for Content-Based Image Retrieval," *Proc. of ICASSP*, Seattle WA, USA, May 1998.

[2] Y. Ariki and Y. Saito, "Extraction of TV News Articles Based on Scene Cut Detection using DCT Clustering," *Proc. of ICIP*, Lausanne, Switzerland, Sept 1996.

[3] Y. Avrithis, N. Doulamis, A. Doulamis and S. Kollias, "Efficient Content Representation in MPEG Video Databases" *Proc. of CVPR,* Santa Barbara CA, USA, June 1998.

[4] A. Doulamis, Y. Avrithis, N. Doulamis and S. Kollias, "Indexing and Retrieval of the Most Characteristic Frames/Scenes in Video Databases," *Proc. of WIAMIS*, June 1997, Belgium.

[5] N. Doulamis, A. Doulamis, Y. Avrithis and S. Kollias, "Video Content Representation Using Optimal Extraction of Frames and Scenes" *Proc. of ICIP,* Chicago IL, USA, Oct. 1998 (to appear).

[6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker, "Query by Image and Video content: the QBIC System," *IEEE Computer Magazine,* pp. 23-32, Sept. 1995.

[7] M. Gelgon and P. Bouthemy, "A Hierarchical Motion-Based Segmentation and Tracking Technique for Video Storyboard-Like Representation and Content-Based Indexing," *Proc.of WIAMIS*, June 1997, Belgium.

[8] K. J. Han and A. H. Tewfik, "Eigen-Image Video Segmentation and Indexing," *Proc. of IEEE ICIP,* pp. 538-541, Santa Barbara, USA, Oct. 1997.

[9] B. Kosko, Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, 1992.

[10] H.-C. Lin, L.-L. Wang and S.-N. Yang, "Color Image Retrieval Based on Hidden Markov Models," *IEEE Trans. Image Processing*, Vol. 6, No. 2, pp. 332-339, Feb. 1997.

[11] MPEG Video Group, "MPEG-7: Context and Objectives (v.3)," ISO/IEC GTC1/SC29/WG11 N1678, Bristol MPEG Meeting, April 1997.

[12] J. Nam and A. Tewfik, "Progressive Resolution Motion Indexing of Video Object," *Proc. of ICASSP,* Seattle WA, USA, May 1998.

[13] Y. Rui, T. Huang and S.-F. Chang, "Digital Image / Video Library and MPEG-7: Standardization and Research Issues," *Proc. of ICASSP*, Seattle WA, USA, May 1998.

[14] Special issue on content-based image retrieval systems, IEEE Computer Magazine, Vol. 28, No. 9, 1995. Guest Editors: Venkat N. Gudivada and Jijay V. Raghavan.

[15] Special issue on visual information management, Communications of ACM, Dec. 1997. Guest Editor:Ramesh Jain.