Interactive Content-Based Retrieval in Video Databases Using Fuzzy Classification and Relevance Feedback

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

Department of Electrical and Computer Engineering
National Technical University of Athens
{adoulam,iavr,ndoulam}@image.ntua.gr

Abstract

This paper presents an integrated framework for interactive content-based retrieval in video databases by means of visual queries. The proposed system incorporates algorithms for video shot detection, key-frame and shot selection, automated video object segmentation and tracking, and construction of multidimensional feature vectors using fuzzy classification of color, motion or texture segment properties. Retrieval is then performed in an interactive way by employing a parametric distance between feature vectors and updating distance parameters according to user requirements using relevance feedback. Experimental results demonstrate increased performance and flexibility according to user information needs.

1. Introduction

The increasing amount of digital image and video data has stimulated new technologies for efficient searching, indexing, content-based retrieving and managing multimedia databases. The traditional keyword annotation approach to accessing image or video information has the drawback that, apart from the large amount of effort for developing annotations, it is not efficient to characterize the rich content of an image or video using only text [8]. For this reason, the MPEG group has recently begun a new standardization phase (MPEG-7) for a multimedia content description interface. This standard will specify a set of content descriptors that can be used to describe any multimedia information.

Several tools and algorithms have been proposed in the recent literature for image and video analysis, segmentation or representation, which can be used for the purposes of *content-based image retrieval* (CBIR). For example, object modeling and segmentation for indexing in video databases has been reported in [9] while a progressive resolution motion indexing has been presented in [12] using 3-D wavelet decomposition of video sequences as well as rigid polygonal shapes. A visual image retrieval approach by means of user sketches has been reported in [5], while user interaction for still image retrieval using relevance feedback has been proposed in [4] and [13]. Many CBIR systems have been built, either academic or in the first stage of commercial exploitation, including the QBIC [8], Virage [10], and VisualSeek [14] prototypes. However, most of these prototypes are mainly restricted to still images and cannot be easily extended to video databases since, due to the strong temporal correlation of video frames, performing queries on each video frame is very inefficient.

In the context of this paper, we propose an interactive framework for content-based indexing and retrieval in video databases. This work has been motivated by previous results on video analysis and content description using *key-frames* [6]. This description has later been extended in [7] by introducing *key-shots* and applying optimization techniques for determining the "best" combination of key-frames/shots. An efficient video content representation has thus been derived in [3] using feature vectors based on *fuzzy classification* of frame segment properties for all key-frames/shots. This representation has been utilized for content-based retrieval in [1]. This paper combines the above video content representation with the *relevance feedback* approach presented in [13] for still image databases in order to provide interactive content-based retrieval for video databases.

In the first stage of the proposed method, video processing and image analysis techniques are applied to each video frame for extracting color, motion, shape and texture information. Color and motion segment information such as their location, shape and size is gathered in order to form a multidimensional feature vector using fuzzy classification, and a small set of key-frames/shots is extracted. At this point the problem of content-based retrieval from a video database has actually reduced to still image retrieval [1]. In a *query by example* environment a user can thus provide queries in the form of still images and a *parametric similarity measure* is employed to find a set of frames that best match a given user query. A relevance feedback approach [13] is then adopted, where the retrieval process is *interactive* between human and computer. This way the user is relieved from the task of expressing his query in terms of low level features, and the computer is provided with a means of

mapping low level features to high level queries and coping with the subjectivity of user requirements.

2. Feature-Based Content Representation

A block diagram of the proposed architecture for video content representation is illustrated in Figure 1, consisting of four modules; shot cut detection, video sequence analysis, fuzzy classification and key frame extraction.

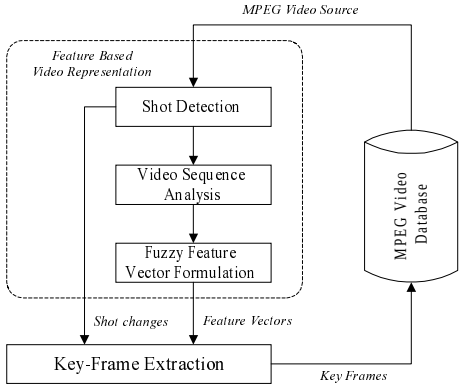


Figure 1. Block diagram of the proposed architecture for video content representation.

2.1. Shot Detection

Since a video sequence is a collection of different shots, each of which corresponds to a continuous action of a single camera operation, a shot detection algorithm is applied first. The algorithm proposed in [32] has been adopted for shot detection since it is based on information directly available in the case of intracoded frames of MPEG video sequences, while for the intercoded ones, it requires a minimal decoding effort, resulting in significant reduction of the required computations.

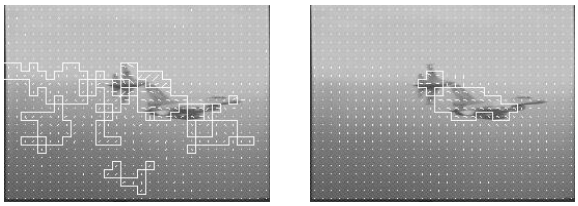
2.2. Video Sequence Analysis

The next step is segmentation of each shot into semantically meaningful objects and extraction of essential information describing those objects. Color and motion segmentation is applied for this purpose, while color and motion information is kept distinct in order to provide a flexible video content representation where each piece of information can be handled separately.

The *Recursive Shortest Spanning Tree* (RSST) [11] algorithm is our basis for color segmentation. Despite its relative computational complexity, it is considered as one of the most powerful tools for image segmentation. In order to yield faster execution, a new approach is proposed in [2], which recursively applies the RSST algorithm on images of increasing resolution. The results are depicted in Figure 2 for a target number of segments equal to 5. It is shown that it the exact segment contours can be obtained at the highest resolution level even without knowledge of the image at that level, making it possible to segment frames in MPEG video streams with minimal decoding of a very small percentage of blocks.

(a) (b)

Figure 2. Color segmentation: (a) initial image, (b) final result.



(a) (b)

Figure 3. Motion segmentation results (a) without, and (b) with smoothing.

Motion segmentation is performed by applying the recursive RSST algorithm at the MPEG block resolution, while motion vector differences are used instead of color differences. We have chosen to exploit the motion vector information that is directly available in MPEG streams, thus eliminating the need for motion analysis altogether. Although an extremely fast implementation is achieved in this way, a post-processing median filtering step for motion field smoothing is indispensable [6]. It is clear from Figure 3 that without motion vector smoothing, wrong segmentation results are produced, even in a uniform and almost stationary background. On the contrary, only the actually moving objects are extracted in the case of smoothed motion vectors.

2.3. Fuzzy Feature Vector Formulation

All features extracted by the video sequence analysis module (i.e., size, location, color or motion of each segment) can be used to describe the visual content of each video frame. However, since there can be absolutely no correspondence between such features of two different frames, making comparisons unfeasible, we classify color as well as motion segments into pre-determined classes, forming a multidimensional histogram. In this framework, each feature vector element corresponds to a specific feature class (equivalent to a histogram bin) and contains the number of segments that belong to this class. In order

to eliminate the possibility of classifying two similar segments to different classes, causing erroneous comparisons, a degree of membership is allocated to each class, resulting in a fuzzy classification formulation [2].

For each color segment S_i, i=1,…,K, in a frame consisting of K segments, an $L\times1$ vector $\mathbf{s}_i$ is formed:

$$\mathbf{s}_i = [a(S_i)\,\mathbf{c}^T(S_i)\,\mathbf{l}^T(S_i)]^T \quad (1)$$

where a denotes the size of the color or motion segment, the 3×1 vector $\mathbf{c}$ includes the average values of the color components of the color segment, and $\mathbf{l}$ is a 2×1 vector indicating the horizontal and vertical location of the segment center (so that L=6). A similar 5×1 vector is formed for each motion segment.

The domain of each element $s_j^{(i)}$, j=1,2,…,L of vectors $\mathbf{s}_i$, i=1,2,…,K is then partitioned into Q regions by means of Q membership functions $\mu_{n_j}(s_j^{(i)})$, n_j=0,1,…,Q-1. For a given real value of $s_j^{(i)}$, $\mu_{n_j}(s_j^{(i)})$ denotes the degree of membership of the element $s_j^{(i)}$ to the class with index n_j. Gathering class indices n_j for all elements j=1,2,…,L, an L-dimensional class $\mathbf{n} = [n_1,\cdots,n_L]^T$ is defined. Then, the degree of membership of each vector $\mathbf{s}_i$ to class $\mathbf{n}$ can be performed through a product of the membership functions $\mu_{n_j}(s_j^{(i)})$ of all individual elements $s_j^{(i)}$ of $\mathbf{s}_i$ to the respective elements n_j of $\mathbf{n}$:

$$\mu_{\mathbf{n}}(\mathbf{s}_i) = \prod_{j=1}^{L} \mu_{n_j}(s_j^{(i)}) \quad (2)$$

It is now possible to construct a multi-dimensional fuzzy histogram from the segment feature samples $\mathbf{s}_i$, i=1,…,K. The value of the fuzzy histogram, $H(\mathbf{n})$, is defined as the sum, over all segments, of the corresponding degrees of membership $\mu_{\mathbf{n}}(\mathbf{s}_i)$:

$$H(\mathbf{n}) = \frac{1}{K}\sum_{i=1}^{K} \mu_{\mathbf{n}}(\mathbf{s}_i) = \frac{1}{K}\sum_{i=1}^{K}\prod_{j=1}^{L} \mu_{n_j}(s_j^{(i)}) \quad (3)$$

$H(\mathbf{n})$ thus can be viewed as a degree of membership of a whole frame to class $\mathbf{n}$. A frame feature vector $\mathbf{f}$ is then formed by gathering values of $H(\mathbf{n})$ for all classes $\mathbf{n}$, i.e., for all combinations of indices, resulting in a total of Q^L feature elements. Since the above analysis is repeated for both color and motion segments, the final feature vector of length $N=Q^6+Q^5$ is constructed by gathering the color and motion feature vectors. All object information is retained and can be separated for retrieval purposes by applying appropriate weight combinations on the feature vectors. In other words, each derived feature element has a specific semantic meaning, so that one can look, for example, for "small black" objects located "near the top".

2.4. Extraction of Key Frames / Shots

Once a feature-based representation of each frame is available, a *shot feature vector* can be constructed, characterizing a whole shot. This information can be exploited for extracting a set of representative shots (*key shots*) using a shot clustering algorithm. The generalized Lloyd or *K-means* algorithm is employed for clustering similar shot feature vectors and selecting a limited number of cluster representatives. The optimization algorithm used was proposed in [7]. *Key-frames* can then be selected from the key shots, so that the final video content representation consists of the set of feature vectors of the selected key-frames and shots. The key-frame selection algorithm is based on an optimization method for locating a set of minimally correlated feature vectors. A *genetic algorithm* approach, proposed in [2], is employed here since it is more efficient for the particular optimization problem, given the size and dimensionality of the search space and the multimodal nature of the objective function.

3. Content-Based Retrieval

At this point, the problem of content-based retrieval from a video database has actually reduced to still image retrieval [1]. Once the video content representation has been generated (off-line) for all sequences in a video database, content-based retrieval is possible by means of queries in the form of frames (images) or shots.

3.1. Video Queries

Each still frame or video shot that is given as an input query by a user is analyzed in the same way as video sequences in the database, and its feature vector $\mathbf{x}$ is calculated. A comparison is then performed between $\mathbf{x}$ and feature vectors $\mathbf{y}$ of the key video frames/shots of the database, and the best M frames/shots of the database are selected and provided to the user. A *parametric (weighted) distance* or *similarity measure* is employed for feature vector comparisons, permitting some elements of the feature vectors to be taken into account to a higher or lower degree according to the user information needs. The distance between the input feature vector $\mathbf{x}$ and feature vectors $\mathbf{y}$ of the key video frames/shots of the database is defined as follows:

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

where $\mathbf{w}$ is an $N\times1$ weight vector, N is the feature vector length, $\mathbf{e}=\mathbf{x}-\mathbf{y}$ 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 (shots) corresponding to the M feature vectors $\mathbf{y}_i$, i=1,…,M with the M minimum distances $d_{\mathbf{w}}(\mathbf{x},\mathbf{y}_i)$ is returned as a query result to the user.

3.2. Relevance Feedback

Since the end user is not always able to express his query in terms of low-level features, a *relevance feedback* approach [13] is adopted, so that the retrieval process is *interactive* between human and computer. The user is actually able to select a subset of the retrieved objects, that is, the m frames/shots out of M, which he considers that best match his original query. Those frames are marked as "relevant", while the remaining M-m frames/shots are marked as "irrelevant".

The relevance information is fed back to the system and used to automatically update or refine similarity measure weights $\mathbf{w}$ so that the updated distance of the input feature vector $\mathbf{x}$ from the "relevant" vectors decreases, while its distance from the "irrelevant" vectors increases. This way the next retrieval is a better approximation to the original information needs [4]. The user is thus relieved from the task of selecting weights, while the computer is provided with a means of mapping low level features to high level queries and coping with the subjectivity of user requirements.

3.3. Weight Update Mechanism

Let $\mathbf{y}_i$, i=1,..,m (m<M) be the feature vectors of the frames (shots) selected by the user as relevant to the original video query. 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. A cost function defined as

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

should thus be minimized with respect to w, subject to the constraint that the magnitude of $\mathbf{w}$ is constant. 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,…,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 $y_k^{(i)}$, $k = 1, \ldots, N$ are the elements of vector $\mathbf{y}_i$.

Multiple relevance feedback is also possible, by means of multiple, consecutive queries. In this more general case, 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. Past weight adaptations are also taken into account by means of a "memory" factor λ (0<λ<1) by which previous optimization results are multiplied. 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

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

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

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

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

This recursive implementation of the adaptation scheme results in a great reduction of the required time consumption for the parameter update.

4. Experimental Results

An MPEG video database consisting of real life video sequences has been used to test the performance of the proposed algorithm. The database consists of video sequences of total duration about 3.5 hours, and includes several shots of news programs, films, commercials, sports and cartoons. The shot detection, feature extraction and key-frame/shot selection algorithms have been applied off-line to all sequences, so that the feature-based video content representation is stored in the database and is readily available.

Figure 4. User input query.

An example user input query, containing a person in foreground, is shown in Figure 4, while the resulting M=8 video frames corresponding to the above query are depicted in Figure 5. Although the objective of the original query was to locate one person in the foreground, only three of the retrieved frames (shown in black border) are "relevant" to this objective. The rest are still similar to the input frame, but mainly regarding the image background, so they are considered "irrelevant". The relevance information is fed back to the retrieval mechanism and

similarity measure weights are updated so as to reflect the high-level query of "one person in the foreground" in terms of parameters corresponding to low-level features. The query results after weight adaptation are depicted in Figure 6.

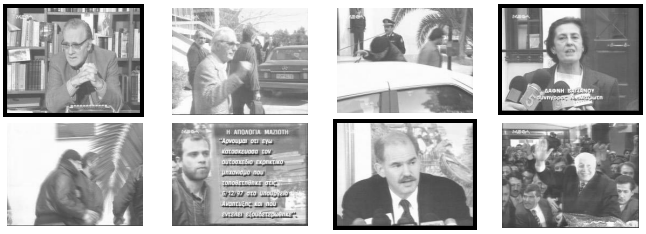


Figure 5. Initial retrieval results for M=8. "Relevant" frames are shown in black border.

Figure 6. Retrieval results after weight adaptation.

5. Conclusions – Further Work

The proposed video representation provides a sufficient framework for many multimedia applications. Examples include video content visualization and summarization, efficient management of large video databases, interactive content-based indexing and retrieval, fast video browsing and access to video archives. Further improvement of the proposed techniques can be achieved by applying more robust object segmentation algorithms. Another objective is the implementation of semantic object segmentation and tracking so that meaningful entities of a video frame can be extracted. Finally, an object graph can be incorporated into the fuzzy classification so that the location and the relationship among different video objects are exploited.

6. References

[1] Y. Avrithis, A. Doulamis, N. D. Doulamis and S. Kollias, "An Adaptive Approach to Video Indexing and Retrieval Using Fuzzy Classification," *Proc. of VLBV*, Urbana, IL, Oct. 1998.

[2] Y. Avrithis, A. Doulamis, N. Doulamis and S. Kollias, "A Stochastic Framework for Optimal Key Frame Extraction from MPEG Video Databases," *Computer Vision and Image Understanding*, May 1999 (accepted for publication).

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

[4] B. Bhanu, J. Peng and S. Qing, "Learning Feature Relevance and Similarity Metrics in Image Databases," *Proc. of CBAIVL*, Santa Barbara CA, June 1998.

[5] A. Del Bimbo and P. Pala, "Visual Image Retrieval by Elastic Matching of User Sketches," *IEEE Trans. Pat. Anal. and Mach. Intell. (PAMI)*, Vol. 19, No. 2, pp. 121-132, Feb. 1997.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] A. Hamrapur, A. Gupta, B. Horowitz, C. F. Shu, C. Fuller, J. Bach, M. Gorkani and R. Jain, "Virage Video Engine," *SPIE Proc. Storage and Retrieval for Video and Image Databases V*, pp. 188-197, San Jose, CA, Feb. 1997.

[11] O. J. Morris, M. J. Lee and A. G. Constantinides, "Graph Theory for Image Analysis: an Approach based on the Shortest Spanning Tree," *IEE Proceedings*, Vol. 133, pp.146-152, April 1986.

[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. S. Huang, M. Ortega and S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," *IEEE Trans. Circ. Syst. for Video Techn.*, Vol. 8, No. 5, Sept. 1998.

[14] J. R. Smith and S. F. Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System," *ACM Multimedia Conf.*, pp. 87-98, Boston, MA, Nov. 1996.

[15] B. L. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Videos," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 5, pp. 533- 544, Dec. 1995.