Efficient Content Representation in MPEG Video Databases

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

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

Abstract

In this paper, an efficient video content representation system is presented which permits automatic extraction of a limited number of characteristic frames or scenes that provide sufficient information about the content of an MPEG video sequence. 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, resulting in faster and more efficient video queries. Moreover, the proposed system can be used for automatic generation of low resolution video clip previews (trailers), giving the ability to browse databases on web pages. Finally, direct content-based retrieval with image queries is possible using the feature vector representation incorporated in our system.

1. Introduction

Multimedia systems have recently led to the development of a series of applications which make use of different kinds of information, such as text, voice, sounds, graphics, animation, images and video. The resulting multimedia databases require new technologies and tools for their organization and management, especially for digital video databases, mainly due to the size of the information involved. For this reason, a new standardization phase is currently in progress by the MPEG group in order to develop algorithms for audiovisual coding (MPEG-4 [9]) and content-based video storage, retrieval and indexing, based on object extraction (MPEG-7 [10]).

Several prototype systems, such as Virage, WebSEEK and QBIC [3] have already been developed and are now in the first stage of commercial exploitation, providing indexing capabilities. However, most of them are restricted to still images and use simple features, e.g. color histogram, to perform the queries. Moreover, these systems cannot be easily extended to video databases since it is practically impossible to perform queries on every video frame.

Several approaches have been proposed in the recent literature for content-based video indexing which mainly deal with scene cut detection [1], video object tracking [4] as well as with single frame extraction [3,5] or image retrieval based on hidden Markov models [7].

In the context of this paper, another approach has been adopted for representing video content. In particular, we present a system that, apart from providing visual search capabilities, also permits automatic extraction of a limited number of characteristic frames or scenes which provide sufficient information about the content of a video sequence. The scene/frame selection mechanism is based on a transformation of the image to a feature domain, which is more suitable for image comparisons, queries and content based retrieval.

A similar approach has been proposed in one of our earlier works [2]. However, better performance is now achieved by integrating object-tracking functionality in the color segmentation algorithm, thus resulting in smoother trajectories of the feature vectors and in a selection mechanism, which is less susceptible to noise. Moreover, the most representative scenes are selected in an optimal way and all the calculations involved are directly applied to MPEG coded video sequences, resulting in a robust and fast implementation which is not very far from real-time.

The proposed system can be very useful for multimedia database management because it provides a means for reduction of the necessary amount of information, which needs to be stored in order to provide search capabilities. 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.

Multimedia interactive services are another interesting application of our system. It is possible to automatically generate low resolution video clip previews (trailers) or still image mosaics which play exactly the same role for video sequences as “thumbnails” for still images. These previews can be used to improve the user interface of digital video databases or browse databases on web pages.

2. System Overview

The proposed system consists of several modules using color and motion information. The partitioning of the input video stream into scenes is performed in the first stage of the proposed system, using a scene cut detection technique. This is achieved by computing the sum of the block motion estimation error over each frame and

detecting frames for which this sum exceeds a certain threshold. Since we use MPEG coded video streams, the block motion estimation error is available and the resulting implementation is very fast.

A multidimensional feature vector is then generated for each frame, containing global frame characteristics, such as color histogram and texture complexity, as well as object characteristics. Unsupervised color and motion segmentation is performed and information such the number of segments, their location, size, mean color and motion vectors are used in the construction of the feature vector. Object tracking is supported by taking into account motion compensated segmentation results of previous frames.

The feature vector is formed as a multidimensional "histogram" using fuzzy classification of object properties into predefined categories. Based on feature vectors of all frames within each scene, a scene feature vector is computed which characterizes the respective scene. The scene vector is then fed as input to a clustering mechanism, which optimally extracts the most representative scenes by minimizing a distortion criterion. Finally, the most characteristic frames of the selected scenes are extracted, based on the temporal fluctuation of frame vectors.

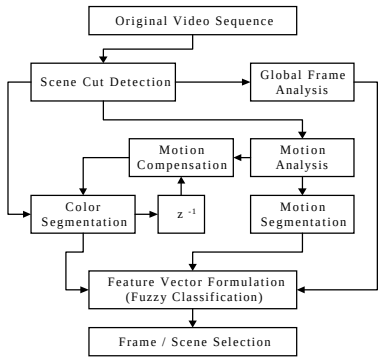


Figure 1. System architecture.

The overall system architecture is depicted in Figure 1. Note that the Motion Analysis step in this figure is actually not implemented in our system, as the motion vectors of the MPEG sequence are directly used for motion segmentation and compensation.

3. Segmentation

The most important task involved in video representation is the extraction of features such as motion, luminosity, color, shape and texture. For this purpose, a feature vector for each frame is calculated, containing global frame characteristics, such as color histogram and texture complexity, obtained through global frame analysis, as well as object characteristics, obtained through color and motion segmentation. Segmentation is studied next.

3.1 Hierarchical block-based color segmentation

A hierarchical block-based color segmentation scheme is adopted for providing color information. Images consisting of uniform areas will be characterized by large segments, while images containing many small objects or noise, by small and distributed segments.

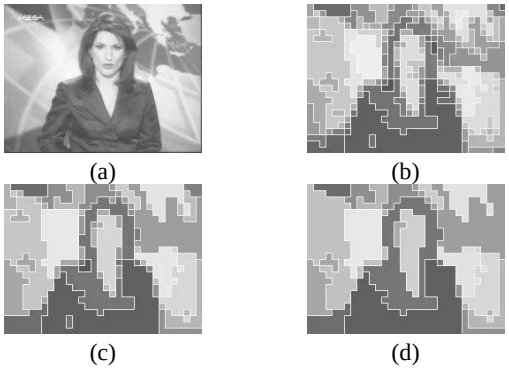


Figure 2. Color segmentation. (a) original frame, (b) 1st iteration of segmentation, (c) 3rd iteration, and (d) final result (6th step).

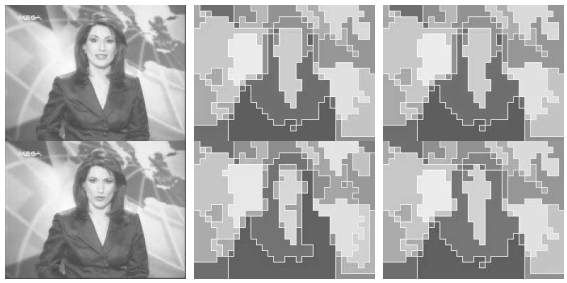


Figure 3. Tracking capabilities. (a) two successive frames, (b) segmentation results without tracking, and (c) with tracking.

Block resolution is used in order to reduce computational time and to exploit information available in MPEG sequences (dc coefficient of the block DCT transform). Since oversegmentation usually results in "noisy" feature vectors, hierarchical merging of similar segments is proposed, which takes into account segment sizes apart from spatial homogeneity, in order to eliminate small segments while preserving large ones. The aforementioned algorithm, which is fully described in one of our earlier works [2], is enhanced by introducing video object tracking capabilities. This is achieved by taking into account motion compensated segmentation results of previous frames.

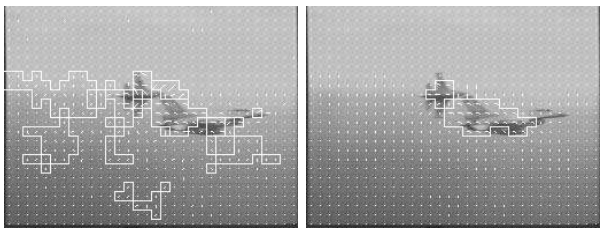
In particular, the decision for merging two adjacent regions is modified by adding to the threshold function a positive or a negative constant, depending on whether the two regions belong to the same segment in previous frame or not. Thus, connected regions are encouraged to remain connected in successive frames. However, in order to take

account of motion in the picture, the color segmentation results are first motion compensated.

Figure 2 depicts the color segmentation results of a frame extracted from a TV news program. It can be clearly seen that in each iteration, more and more segments are merged together, and that the smallest segments are only merged in the last iteration. Our segmentation algorithm has similar performance with the Recursive Shortest Spanning Tree (RSST) technique [8] but is much faster. Moreover, the "noise" effects which appear due to the random order of the merging procedure [2] are now eliminated with the use of object tracking. This is illustrated in Figure 3, where segmentation results are shown for two successive frames, with and without tracking. A stationary background gives different segmentation results without tracking, whereas with tracking only the moving parts of the image result in different segments.

3.2 Block-based motion segmentation

A similar approach is carried out for the case of motion segmentation. The proposed procedure is still at block resolution for exploiting properties of the MPEG bit stream. However, the motion vectors, which can either be computed with a motion analysis algorithm, as shown in Figure 1, or taken directly form the MPEG stream (as is done in our experiments), usually appear "noisy" due to luminosity fluctuations. To achieve smoothness of motion vectors within a moving area, a median filter is used for eliminating "noise" while preserving "edges" between regions of different motion.



(a) (b)

Figure 4. Motion segmentation (a) without, and (b) with filtering.

After the appropriate motion vectors are extracted, we use a technique similar to the previous one to derive the motion segments, except for the fact that no tracking of objects takes place here (since tracking is based on the motion vectors). Figure 4 illustrates the motion segmentation results of a frame extracted from a TV news program. It is clear that without filtering the motion vectors, wrong segmentation results are produced, even in a uniform and almost stationary background (Figure 4(b)). On the contrary, only the actually moving objects are extracted in the case of filtered motion vectors (Figure 4(c)).

4. Fuzzy Classification

All of the above features are gathered in order to form a multidimensional feature vector for each frame. Properties of color or motion segments cannot be used directly as elements of the feature vector, since its size will differ between frames. We therefore classify color as well as motion segments into pre-determined categories, forming a multidimensional "histogram". In order to eliminate the possibility of classifying two similar segments at different categories, a degree of membership is allocated to each category, resulting in a fuzzy classification. The feature vector is then constructed by calculating the sum, over all segments, of the corresponding degrees of membership, and gathering these sums into the appropriate categories:

$$F(\mathbf{n}) = \sum_{i=1}^{K} \left\{ \prod_{j=1}^{L} \mu_{n_j}(f_j^{(i)}) \right\} \quad (1)$$

where K is the total number of color or motion segments, L is number of features (e.g., size, color or location) of segments that are taken into account for classification, $\mathbf{n} = [n_1 \ldots n_L]^T$ specifies the "category" into which a segment is classified, and $n_j \in \{1,2,\ldots,Q\}$ is an index for each feature, where Q is the number of regions into which each feature space is partitioned. The j-th feature, $f_j^{(i)}$, of the i-th segment, S_i, is the j-th element of the vector $[P(S_i)\ \mathbf{c}(S_i)^T\ \mathbf{l}(S_i)^T]^T$ for color segmentation, or $[P(S_i)\ \mathbf{v}(S_i)^T\ \mathbf{l}(S_i)^T]^T$ for motion segmentation, where P, $\mathbf{c}$, $\mathbf{v}$, and $\mathbf{l}$ denote the size, color, motion vector and location of each segment. Finally, $\mu_n(f)$ is the degree of membership of feature f in partition n. Triangular membership functions μ_n are used with 50% overlap between partitions.

The feature vector is formed by gathering values $F(\mathbf{n})$ for all combinations of $n_1,\ldots,n_L$, for both color and motion segments. Global frame characteristics 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.

5. Scene and Frame Selection

The trajectory of the feature vector for all frames within a scene indicates the way in which the frame properties fluctuate during a scene period. Consequently, a vector that characterizes a whole scene is constructed by calculating the mean value of feature vectors over the whole duration of the scene. The scene feature vectors are used for the selection of the most representative scenes, as described below.

5.1 Scene selection.

The extraction of a small but sufficient number of scenes that satisfactorily represent the video content is accomplished by clustering similar scene feature vectors and selecting only a limited number of cluster representatives. For example, in TV news recordings, consecutive scenes of the same person would reduce to just one. Let $\mathbf{s}_i \in \Re^M$, $i = 1,2,...,N_S$ be the scene feature vector for the *i*-th scene, where N_S is the total number of scenes. Then $S = \{\mathbf{s}_i, i = 1,2,\ldots,N_S\}$ is the set of all scene feature vectors. Let also K_S be the number of scenes to be selected and $\mathbf{c}_i$, $i = 1,2,...,K_S$ the feature vectors, which best represent those scenes. For each $\mathbf{c}_i$, an influence set is formed which contains all scene feature vectors $\mathbf{s} \in S$, which are closer to $\mathbf{c}_i$:

$$Z_i = \{\mathbf{s} \in S:\ d(\mathbf{s},\mathbf{c}_i) < d(\mathbf{s},\mathbf{c}_j)\ \forall\, j \neq i\} \tag{2}$$

where $d(\cdot)$ denotes the distance between two vectors. A common choice for $d(\cdot)$ is the Euclidean norm. In effect, the set of all Z_i defines a partition of S into clusters of similar scenes which are represented by the feature vectors $\mathbf{c}_i$. Then the average distortion, defined as

$$D(\mathbf{c}_1,\mathbf{c}_2,...,\mathbf{c}_{K_S}) = \sum_{i=1}^{K_S} \sum_{\mathbf{s}\in Z_i} d(\mathbf{s},\mathbf{c}_i) \tag{3}$$

is a performance measure of the representation of scene feature vectors by the cluster centers $\mathbf{c}_i$. The optimal vectors $\hat{\mathbf{c}}_i$ are thus calculated by minimizing D:

$$(\hat{\mathbf{c}}_1,\hat{\mathbf{c}}_2,...,\hat{\mathbf{c}}_{K_S}) = \underset{c_1,c_2,...,c_{K_S}\in\Re^M}{\arg\min}\ D(\mathbf{c}_1,\mathbf{c}_2,...,\mathbf{c}_{K_S}) \tag{4}$$

Direct minimization of the previous equation is a tedious task since the unknown parameters are involved both in distances $d(\cdot)$ and influence zones. For this reason, minimization is performed in an iterative way using the generalized Lloyd or K-means algorithm. Starting from arbitrary initial values $\mathbf{c}_i(0)$, $i = 1,2,\ldots,K_S$, the new centers are calculated through the following equations for $n \geq 0$:

$$Z_i(n) = \{\mathbf{s} \in S:\ d(\mathbf{s},\mathbf{c}_i(n)) < d(\mathbf{s},\mathbf{c}_j(n))\ \forall\, j \neq i\} \tag{5}$$

$$\mathbf{c}_i(n+1) = cent(Z_i(n)) \tag{6}$$

where $\mathbf{c}_i(n)$ denotes the *i*-th center at the *n*-th iteration, and $Z_i(n)$ its influence set. The center of $Z_i(n)$ is estimated by the function

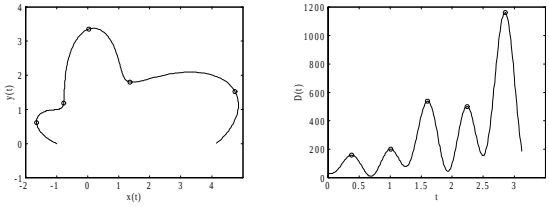
$$cent(Z_i(n)) = \frac{1}{|Z_i(n)|} \sum_{\mathbf{s}_i \in Z_i(n)} \mathbf{s}_i \tag{7}$$

where $|Z_i(n)|$ is the cardinality of $Z_i(n)$. The algorithm converges to the solution $(\hat{\mathbf{c}}_1,\hat{\mathbf{c}}_2,...,\hat{\mathbf{c}}_{K_S})$ after a small number of iterations. Finally, the K_S most representative scenes are extracted, as the ones whose feature vectors are closest to $(\hat{\mathbf{c}}_1,\hat{\mathbf{c}}_2,...,\hat{\mathbf{c}}_{K_S})$:

$$\hat{\mathbf{s}}_i = \underset{\mathbf{s}\in S}{\arg\min}\ d(\mathbf{s},\hat{c}_i),\quad i = 1,2,\ldots,K_S \tag{8}$$

5.2 Frame selection.

After extracting the most representative scenes, the next step is to select the most characteristic frames within each one of the selected scenes. The decision mechanism is based on the detection of those frames whose feature vector resides in extreme locations of the feature vector trajectory. For this purpose, the magnitude of the second derivative of the feature vector with respect to time is used as a curvature measure.



(a) (b)

Figure 5. (a) A continuous curve r(*t*)=(*x*(*t*),*y*(*t*)), and (b) the magnitude of the second derivative *D*(*t*) versus *t*.

For example, as shown in Figure 5, and supposing that we have a 2-dimensional feature vector as a function of time *t*, which corresponds to the continuous curve $\mathbf{r}(t)=(x(t), y(t))$ of Figure 5(a), the local maxima of the magnitude of the second derivative

$$D(t) = \left| d^2\mathbf{r}(t)/dt^2 \right| \tag{9}$$

(shown as small circles in Figure 5(a) & (b)) correspond to the extreme locations of the curve $\mathbf{s}(t)$ and provide sufficient information about the curve, since $\mathbf{s}(t)$ could be almost reproduced using some kind of interpolation. Based on the above observation, local maxima of this curvature measure were extracted as the locations (frame numbers) of the characteristic frames in our experiments. Note that although this technique is extremely fast and very easy to implement in hardware, it may not work in all cases. For this reason, other techniques are currently under investigation, such as logarithmic search and genetic algorithms. Optimal selection can be achieved in this way, at the expense of increased computational complexity.

5.3 Video queries.

Once the feature vector is formed as a function of time, a video database can also 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 frames, which provide a meaningful representation of the sequence.

6. Experimental Results

The proposed algorithms were integrated into a system that was tested using several video sequences from video databases. The results obtained from a TV news reporting sequence of total duration 35 seconds (875 frames) are illustrated in Figures 6, 7 and 8. Using scene cut detection, the test sequence was partitioned in 8 scenes and the respective frame and scene feature vectors were calculated. For each scene, the frame whose feature vector is closest to the respective scene feature vector is depicted in Figure 6. Using the scene selection mechanism described above, three scenes were extracted as most representative and are shown in Figure 7. In effect, three scene clusters were generated, each containing scenes with similar properties, such as number and complexity of objects. Moreover, it is clear that the selected scenes give a meaningful representation of the content of the 35 sec video sequence.

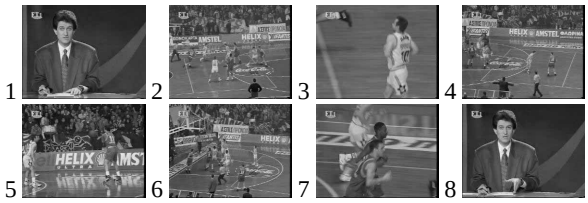


Figure 6. The 8 scenes of a test video sequence.

Figure 7. The 3 scenes that were selected as most representative.

Figure 8. The most representative frames of scene 3.

Scene 3 of Figure 6 was used in order to test the frame selection procedure. Out of a total of 137 frames, only 6 were selected as most representative and are shown in Figure 8. Due to object tracking, the trajectory of the feature vector versus time (frame number) is smoother than the one described in [2] and thus the selection procedure is more reliable. Although a very small percentage of frames is retained, it is obvious that one can perceive the content of the scene by just examining the 6 selected frames.

7. Conclusions - Further Work

An efficient content-based representation has been proposed in this paper. In particular, a small but meaningful amount of information is extracted from a video sequence, which is capable of providing a representation suitable for visualization, browsing and content-based retrieval in video databases.

Several improvements are possible for the proposed system, such as integration of color and motion segmentation results, more robust object tracking algorithm, more intelligent object extraction (e.g., extraction of human faces [6]), enhancement of the frame selection mechanism (based on correlation properties between feature vectors), and interweaving of audio and video information. These topics are currently under investigation.

8. References

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

[2] 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.

[3] 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.

[4] 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.

[5] G. Iyerngar and A.B. Lippman, "Videobook: An Experiment in Characterization of Video," *Proceedings of ICIP*, Sept. 1996, Switzerland.

[6] D. Kalogeras, N. Doulamis, A. Doulamis and S. Kollias, "Low Bit Rate Coding of Image Sequences using Adaptive Regions of Interest," Accepted for pub., *IEEE Trans. Circuits and Systems for Video Technology.*

[7] 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.

[8] 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.

[9] MPEG Video Group, "MPEG-4 Requirements," *ISO/IEC GTC1/SC29/WG11 N1679,* Bristol MPEG Meeting, April 1997.

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