A GENETIC ALGORITHM FOR EFFICIENT VIDEO CONTENT REPRESENTATION

A.D. Doulamis, Y.S. Avrithis, N.D. Doulamis and S.D. Kollias
Department of Electrical and Computer Engineering
National Technical University of Athens
Email: adoulam@image.ntua.gr

1. Introduction

The rapid development of video and multimedia applications has enabled users to handle large amounts of visual information. At the same time, new requirements have emerged for more intelligent access to such databases, i.e., content-based indexing, retrieval and video browsing. The traditional text-based approach to accessing image or video databases has the drawback that it is difficult to characterize the rich content of images or video based only on text information [1]. For this reason, a new standardization phase is currently in progress by the Moving Picture Expert Group (MPEG) in order to develop an integrated framework for a multimedia content description interface (MPEG-7).

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 [2], [3]. Some of these approaches are now in the first stage of commercial exploitation, such as the VisualSEEK and QBIC [4] prototypes. Color image retrieval has been examined in [5] based on a hidden Markov model, and extraction of detailed image regions for indexing and retrieval has been proposed in [6]. 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 [8] to perform the queries. A new progressive resolution motion indexing has been presented in [9] using 3-D wavelet decomposition of video sequences as well as rigid polygonal shapes. Finally, an approach for automatic video segmentation and content-based retrieval based on a temporally windowed principal component analysis of a sub-sampled version of a video sequence has been reported in [10].

However, 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. To make retrieval in video databases more efficient, a pre-indexing stage should be introduced which extracts a small but meaningful information of the video content. Then, video queries can be directly applied to this small amount of information. In this chapter, we propose an efficient video content representation using optimal extraction of a limited number of key frames and scenes of video sequences. This approach not only provides a more efficient way for video indexing, but also results in reducing storage requirements and thus permits easy management of multimedia databases. Then, video queries are performed on this small but meaningful collection of frames instead of the entire video stream.

The first stage of the proposed algorithm includes a scene cut detection mechanism. Then, 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, and their location, size and shape. 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 for each video frame. The representation of each frame by a feature vector, apart from reducing storage requirements, transforms the image domain to another domain, more efficient for key frame selection. 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.

Optimal selection of representative scenes is performed by minimizing a distortion criterion. This is accomplished by clustering similar scenes and selecting a limited number of cluster representatives. The generalized Lloyd-Max algorithm has been used for this purpose as described in [11]. 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. This approach gives better results than the one proposed in [12], where frame selection was based on detection of feature vectors that reside in extreme locations of the feature vector trajectory. Since similar frames may be characterized by different segments, the latter approach was rather sensitive and heavily dependent on the adopted segmentation algorithm.

Unfortunately, the complexity of an exhaustive search for the minimum value of a correlation measure is such that a direct implementation would be practically unfeasible. For this reason, a genetic algorithm approach [13] is adopted in this chapter. Possible solutions of the optimization problem, i.e., sets of frames, are represented by chromosomes whose genetic material consists of frame numbers (indices). An initial population of chromosomes is then generated by selecting sets of frames whose feature vectors reside in extreme locations of the feature vector trajectory. The objective function used to estimate the fitness values of all chromosomes, is defined as the sum of squares of cross-correlations between all combinations of feature vectors, for all frame

numbers that belong to the genetic material of the respective chromosome. Following a proportionate scheme for parent selection, a set of new chromosomes (offspring) is produced by mating the parent chromosomes and applying uniform crossover and mutation operations.

This chapter is organized as follows: Section 2 briefly describes the feature extraction module, including the scene cut detection as well as the color / motion segmentation procedure. Section 3 refers to the feature vector formulation, while in sections 4 and 5 the scene and frame selection mechanisms are presented respectively. Experimental results illustrating the performance of the proposed scheme are presented in section 6, while conclusions are given in section 7 of this chapter.

2. Feature Extraction

The feature extraction procedure is performed in a way similar to [14] 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. Since visual content is typically stored in MPEG compressed format, it is preferable to perform the feature extraction directly in the compressed domain. As a result, 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 [14].

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 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 has been adopted both for reducing the required computational time and for exploiting information that already exists in the MPEG coding standard. To avoid oversegmentation problems we have proposed a hierarchical block-based segmentation algorithm described in [14]. Moreover, object tracking is supported by taking into account motion compensated segmentation results of previous frames [12]. Apart from information provided by color or motion segmentation other features are included in the feature vector, such as information provided by color and motion histograms or appropriate ac coefficients of the DCT transform.

3. Feature Vector Formulation

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 to different classes, causing erroneous comparisons, a degree of membership is allocated to each class, resulting in a fuzzy classification [15].

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.

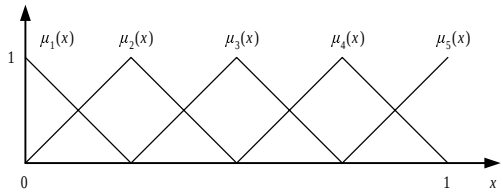

Figure 1: One-dimensional fuzzy classification.

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

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

then fully describes the properties of segment S_i using a total of L segment features. Since the length of vectors $\mathbf{c}$, $\mathbf{v}$ and $\mathbf{l}$ is 3×1, 2×1 and 2×1 respectively, L will be equal to 8 in the above formulation. However, it can be greater if we also include segment shape or texture information. 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 $M=Q^L$ 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. Scene Selection.

The first stage for an efficient video content representation is the extraction of a small but sufficient number of scenes that satisfactorily represent the video content. This is accomplished by clustering similar scene feature vectors (that is, vectors whose distance in the feature space is small) 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. The extraction of the most representative scenes can be used in applications such as automatic generation of low resolution video clip previews.

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 $\mathbf{c}_i$:

$$Z_i = \{\mathbf{s} \in S : d(\mathbf{s}, \mathbf{c}_i) < d(\mathbf{s}, \mathbf{c}_j) \; \forall \; j \neq i\} \tag{4}$$

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

$$D(\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{K_S}) = \sum_{i=1}^{K_S} \sum_{\mathbf{s} \in Z_i} d(\mathbf{s}, \mathbf{c}_i) \tag{5}$$

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:

$$(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, ..., \hat{\mathbf{c}}_{K_S}) = \underset{c_1, c_2, ..., c_{K_S} \in \Re^M}{\arg\min} D(\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{K_S}) \tag{6}$$

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 [16]. 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 $n \geq 0$:

$$Z_i(n) = \{\mathbf{s} \in S: \; d(\mathbf{s}, \mathbf{c}_i(n)) < d(\mathbf{s}, \mathbf{c}_j(n)) \; \forall \; j \neq i\} \tag{7a}$$

$$\mathbf{c}_i(n+1) = cent(Z_i(n)) \tag{7b}$$

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

$$cent(Z_i(n)) = \frac{1}{|Z_i(n)|} \sum_{\mathbf{s}_i \in Z_i(n)} \mathbf{s}_i \tag{8}$$

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 $(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, ..., \hat{\mathbf{c}}_{K_S})$:

$$\hat{\mathbf{s}}_i = \arg\min_{\mathbf{s} \in S} d(\mathbf{s}, \hat{c}_i), \quad i = 1,2,\ldots,K_S \tag{9}$$

5. Genetic Algorithm for Frame Selection.

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, the key frames are selected as the ones with the minimum correlation among them. The selection could also be performed using the previous optimization technique. However, that approach does not exploit the temporal relation of feature vectors, which is significant for the frame selection procedure, as it is described in the sequel.

Let us denote by $\mathbf{f}_i \in \Re^M$, $i \in V = \{1,\ldots,N_F\}$ the feature vector of the *i*-th frame, where N_F is the total number of frames of a scene, and suppose that the K_F most characteristic ones should be selected. The correlation coefficient of the feature vectors $\mathbf{f}_i, \mathbf{f}_j$ is defined as $\rho_{ij} = C_{ij} / (\sigma_i \sigma_j)$ where $C_{ij} = (\mathbf{f}_i - \mathbf{m})^T (\mathbf{f}_j - \mathbf{m})$ is the covariance of the two vectors, $\mathbf{m} = \sum_{i=1}^{N_F} \mathbf{f}_i / N_F$ is the average feature vector of the scene and $\sigma_i^2 = C_{ii}$ is the variance of $\mathbf{f}_i$. In order to define a measure of correlation between K_F feature vectors, we first define the *index* vector $\mathbf{x} = (x_1, \ldots, x_{K_F}) \in W \subset V^{K_F}$ where

$$W = \{(x_1, \ldots, x_{K_F}) \in V^{K_F} : x_1 < \cdots < x_{K_F}\} \tag{10}$$

is the subset of V^{K_F} which contains all sorted index vectors $\mathbf{x}$. Thus, each index vector $\mathbf{x} = (x_1, \ldots, x_{K_F})$ corresponds to a set of frame numbers. The *correlation measure* of the feature vectors $\mathbf{f}_i$, $i = x_1, \ldots, x_{K_F}$ is then defined as

$$R(\mathbf{x}) = R(x_1, \ldots, x_{K_F}) = \left(\sum_{i=1}^{K_F - 1} \sum_{j=i+1}^{K_F} (\rho_{x_i, x_j})^2 \right)^{1/2} \tag{11}$$

Based on the above definitions, it is clear that searching for a set of K_F minimally correlated feature vectors is equivalent to searching for an index vector $\mathbf{x}$ that minimizes $R(\mathbf{x})$. Searching is limited in the subset *W*, since index vectors are used in order to construct sets of feature vectors, therefore any permutations of the elements of

$\mathbf{x}$ will result in the same sets. The set of the K_F least correlated feature vectors, corresponding to the K_F most characteristic frames, is thus represented by

$$\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_{N_F}) = \arg\min_{\mathbf{x} \in W} R(\mathbf{x}) \tag{12}$$

Unfortunately, the complexity of an exhaustive search for the minimum value of $R(\mathbf{x})$ is such that a direct implementation would be practically unfeasible, since the multidimensional space W includes all possible sets (combinations) of frames. A dramatic reduction in complexity is achieved, however, through *logarithmic search*, which has been introduced in [11] and is performed in a way similar to the search for block motion estimation in video sequences. The algorithm is restricted to the special case $N_F = 2^G$ and its implementation includes the definition of an *initial step size* $S_0 = 2^{G-2} = N_F / 4$ and an *initial index* $\mathbf{x}_0 \in W$ as the element of W which is closest to the *middle point* $\tilde{\mathbf{x}}_0 = (\mu, \ldots, \mu)$, where $\mu = 2^{G-1} - 1$. Successive index vector estimates are then obtained by the recursive equations

$$\mathbf{x}_n = \arg\min_{\mathbf{x} \in N(\mathbf{x}_{n-1}, S_{n-1})} R(\mathbf{x}), \; S_n = S_{n-1} / 2 \tag{13}$$

where the *neighborhood* $N(\mathbf{x}, S)$ of $\mathbf{x}$ is a small set of index vectors whose distance from $\mathbf{x}$ is S. The final result $\hat{\mathbf{x}} = \mathbf{x}_{G-2}$ is obtained by applying the above recursion for $n = 1, \ldots, G-2$ (until $S_n = 1$). The algorithm, whose implementation details are fully described in [11], provides a very fast convergence to a sub-optimal solution. However, since the search procedure is by definition confined to a very small, pre-defined subset of the search space W, there is always a significant possibility of converging to a local minimum of $R(\mathbf{x})$, resulting in poor performance.

For this reason, a genetic algorithm (GA) [13] approach is adopted in this chapter. This approach seems to be very efficient for the particular optimization problem, given the size and dimensionality of the search space and the multimodal nature of the objective function. Possible solutions of the optimization problem, i.e., sets of frames, are represented by chromosomes whose genetic material consists of frame numbers (indices). Chromosomes are thus represented by index vectors $\mathbf{x} = (x_1, \ldots, x_{K_F}) \in V^{K_F}$ following an integer number encoding scheme, that is, using integer numbers for the representation of genes $x_i \in V, i = 1, \ldots, K_F$.

An *initial population* of P chromosomes, $\mathbf{X}(0) = (\mathbf{x}_1, \ldots, \mathbf{x}_P)$ is then generated by selecting sets of frames whose feature vectors reside in extreme locations of the feature vector trajectory. This selection is accomplished by locating points where the magnitude of the second-order derivative of feature vector trajectory is locally maximized. Traditionally, initial populations are randomly generated, but the above approach exploits the temporal relation of feature vectors and increases the possibility of locating sets of feature vectors with small correlation within the first few GA cycles. Note that this approach has been used directly for key frame selection in [12], [14]. The initial population $\mathbf{X}(0)$ is used for the creation of new generation populations $\mathbf{X}(n)$, $n>0$. The creation of $\mathbf{X}(n)$ at generation (or GA cycle) n is performed of by applying a set of

operations on population $\mathbf{X}(n\text{-}1)$, described below. This procedure is repeated in an iterative way, until $\mathbf{X}(n)$ converges to an optimal solution of the problem.

The correlation measure $R(\mathbf{x})$ is used as an objective function to estimate the performance of all chromosomes $\mathbf{x}_i, i = 1,\ldots,P$ in a given population. However, a *fitness function* is used to map objective values to fitness values, following a *linear normalization scheme*. In particular, chromosomes $\mathbf{x}_i$ are ranked in ascending order of $R(\mathbf{x}_i)$, since the objective function is to be minimized. Let $r(\mathbf{x}_i) \in \{1,\ldots,P\}$ be the rank of chromosome $\mathbf{x}_i, i = 1,\ldots,P$. Defining an arbitrary fitness value F_B for the best chromosome, the fitness of the i-th chromosome is given by the linear function

$$F(\mathbf{x}_i) = F_B - [r(\mathbf{x}_i) - 1]D, \quad i = 1,\ldots,P \tag{14}$$

where D is a decrement rate. Thus, the average objective value of the population is mapped into the average fitness [17]. After fitness values, $F(\mathbf{x}_i), i = 1,\ldots,P$, have been calculated for all members of the current population, *parent selection* is then applied so that a fitter chromosome gives a higher number of offspring and thus has a higher chance of survival in the next generation. A *proportionate scheme*, implemented by the *roulette wheel selection* procedure [18], is used for parent selection, ensuring that each chromosome has a growth rate proportional to its fitness value.

A set of new chromosomes (offspring) is then produced by mating the selected parent chromosomes and applying a *crossover operator*. The genetic material of the parents is combined in a random way in order to produce the genetic material of the offspring. Figure 2 depicts an example of the crossover operator with four crossover points used for exchanging genes. A generalized *uniform crossover* scheme is employed in the context of this chapter, by considering each parent gene to be a potential crossover point. *Mutation* is then applied to the newly created chromosomes, introducing random gene variations that are useful for restoring lost genetic material, or for producing new material that corresponds to new search areas. In particular, each offspring gene x_i is replaced by randomly generated one $x_i' \in V = \{1,\ldots,N_F\}$, if a probability test is passed. A small mutation probability p_m ensures that only a small gene proportion is altered in each generation.

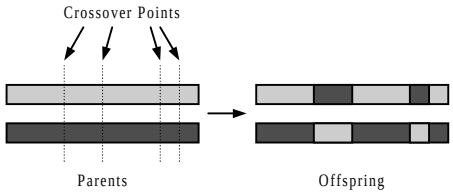

Figure 2: Example of the crossover operator.

Once new chromosomes have been generated for a given population $\mathbf{X}(n)$, $n \geq 0$, the next generation population, $\mathbf{X}(n+1)$, is formed by inserting those new chromosomes into $\mathbf{X}(n)$ and deleting an appropriate number of older chromosomes, so that each population consists of P members. The exact number, C, of old chromosomes to be replaced by new ones defines the *replacement strategy* of the GA and greatly

affects its convergence rate. All of the above description refers to simple GA cycle. Several cycles need to take place, that is, several generations $\mathbf{X}(n), n > 0$ need to be produced until the population converges to an optimal solution. For this reason, the procedures of fitness evaluation, parent selection, crossover and mutation are repeated until a termination criterion is reached. Usually the GA terminates when the best chromosome fitness remains constant for a large number of generations, indicating that further optimization is unlikely.

The above algorithm, as well as the logarithmic search algorithm, is based on the assumption that frames which are close to each other (in time) should have similar properties, and therefore indices which are close to each other (in W) should have similar correlation measures. However, the technique performs equally well even in the case of random feature vectors, as shown by experiments.

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 10 minutes (about 15000 frames) are illustrated in the following Figures. The sequence was first partitioned in 52 scenes, using the scene cut detection procedure described in Section 2. Then, the frame and scene feature vectors were extracted using the aforementioned methodology. In particular, an average feature vector was formulated, for each scene, based on the multidimensional feature vectors of the frames composed the scene. The generalized Lloyd algorithm was used for the selection of the most representative scenes.

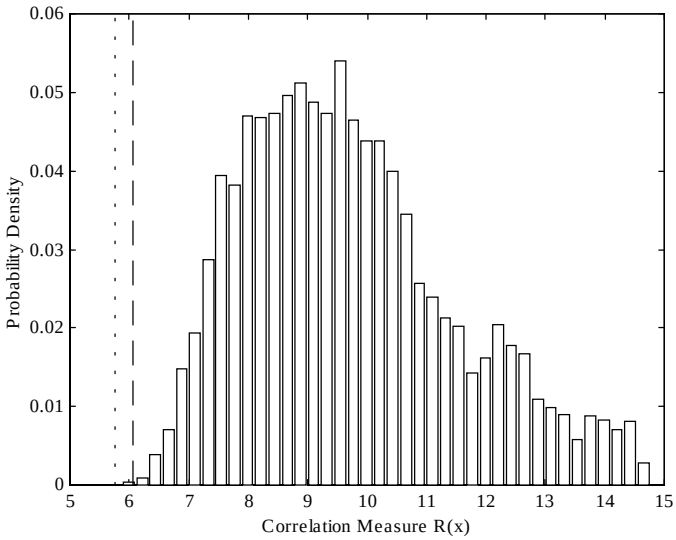

Figure 3. Probability density function of the correlation measure $R(\mathbf{x})$. The vertical dashed line shows the minimum value located by the logarithmic search algorithm, while the dotted line the one located by the genetic algorithm.

The frame selection procedure was then applied to the most representative scenes for extraction of the key video frames. Two methods have been used for selection of K_F=6 key frames out of a total of 293 frames (about 12 sec) of a specific scene: the logarithmic search procedure described in [11], and the genetic algorithm proposed in this chapter. Figure 3 indicates the probability density function of the correlation measure $R(\mathbf{x})$. This function is actually estimated by a histogram obtained through Monte-Carlo simulation, using a large number of random sample vectors $\mathbf{x} = (x_1, \ldots, x_{K_F}) \in V^{K_F}$. In the above Figure, the minimum values obtained by the logarithmic and the genetic search algorithms are also depicted (dashed and dotted line respectively). It is firstly observed that both algorithms return minimum values very close to the actual global minimum of $R(\mathbf{x})$. This, of course, is an approximate result, since the actual minimum value cannot be calculated. Secondly, it is clearly shown that the genetic algorithm provides more accurate results since the logarithmic search procedure can be "trapped" in a local minimum.

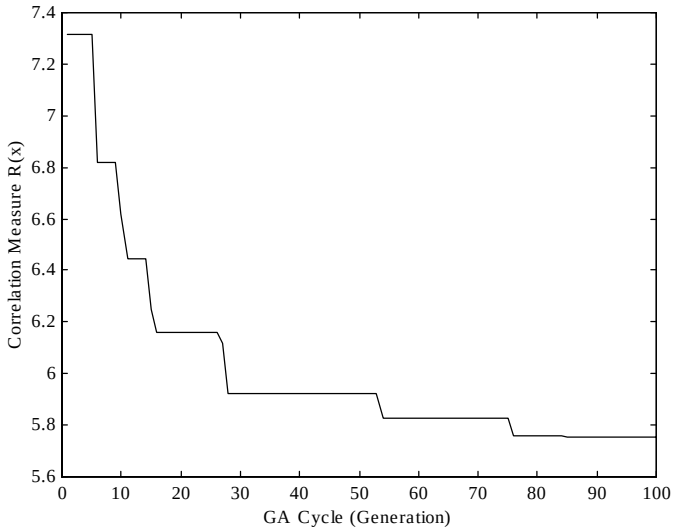

Figure 4: Genetic algorithm convergence: correlation measure $R(\mathbf{x})$ (objective function) versus the GA cycle (generation).

Figure 4 shows the minimum value, over the whole population, of the objective function (or correlation measure $R(\mathbf{x})$) versus the cycle (or generation) of the genetic algorithm. As expected, $R(\mathbf{x})$ decreases as the GA cycle increases, until it reaches a minimum at generation 85. Since in the specific experiment half chromosomes are replaced by new ones at each generation (P=80 and $C=P/2$=40 have been used), there are cases where all generated offspring have lower fitness than their parents. In these cases the value of the $R(\mathbf{x})$ remains at the same level, hence the "stepwise" appearance of the curve in the above Figure. Note that the step "width" increases with the GA cycle, since it is directly related to the probability of further optimization.

Figure 5. Six selected key frames of a scene after applying the genetic algorithm.

The six selected video frames of the given scene are shown in the Figure 5. 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. Consequently, it is clear that the selected frames give a meaningful representation of the content of the 12-sec video sequence.

7. Conclusions

In this chapter, an efficient video content representation system is presented which permits automatic extraction of a limited number of key frames or scenes that provide sufficient information about the content of a video sequence. 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. Queries are then performed in a more efficient way since only a small number of representative frames involved in the process. In our approach a genetic search algorithm have been adopted for the key frame selection.

The GA approach seems to be very efficient for the particular optimization problem, given the size and dimensionality of the search space and the multimodal nature of the objective function. This estimation is supported by experimental results, demonstrating fast convergence to optimal solutions. The performance of the technique could be further improved by considering parallelization methods such as global, migration or diffusion [19]. Several other improvements are also 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]), and interweaving of audio and video information. These topics are currently under investigation.

References

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

[2] 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.
[3] Special issue on visual information management, *Communications of ACM*, Dec. 1997. Guest Editor:Ramesh Jain.
[4] 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.
[5] 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.
[6] D. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos, "Extraction of Detailed Image Regions for Content-Based Image Retrieval," *Proc. of ICASSP*, Seattle, USA, May 1998.
[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] Y. Ariki and Y. Saito, "Extraction of TV News Articles Based on Scene Cut Detection using DCT Clustering," *Proc. of ICIP*, Sept. 1996, Switzerland.
[9] J. Nam and A. Tewfik, "Progressive Resolution Motion Indexing of Video Object," *Proc. of ICASSP*, Seattle, USA, May 1998.
[10] 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.
[11] N. Doulamis, A. Doulamis, Y. Avrithis and S. Kollias, "Video Content Representation Using Optimal Extraction of Frames and Scenes," *Proc*. of *ICIP*, Chicago, USA, Oct. 1998.
[12] Y. Avrithis, N. Doulamis, A. Doulamis and S. Kollias, "Efficient Content Representation in MPEG Video Databases" *Proc. of CVPR*, Santa Barbara, USA, June 1998.
[13] D.E. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, Addison Wesley, 1989.
[14] 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.
[15] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, 1992.
[16] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1993.
[17] K. S. Tang, K. F. Man, S. Kwong and Q. He, "Genetic Algorithms and Their Applications," *IEEE Signal Processing Magazine*, pp. 22-37, Nov. 1996.
[18] H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan Press, 1975.
[19] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.