ELSEVIER

Signal Processing 80 (2000) 1049–1067

SIGNAL PROCESSING

www.elsevier.nl/locate/sigpro

A fuzzy video content representation for video summarization and content-based retrieval

Anastasios D. Doulamis*, Nikolaos D. Doulamis, Stefanos D. Kollias

Department of Electrical and Computer Engineering, National Technical University of Athens, 9, Heroon Polytechniou str., Zografou 157 73, Greece

Received 30 April 1999; accepted 15 November 1999

Abstract

In this paper, a fuzzy representation of visual content is proposed, which is useful for the new emerging multimedia applications, such as content-based image indexing and retrieval, video browsing and summarization. In particular, a multidimensional fuzzy histogram is constructed for each video frame based on a collection of appropriate features, extracted using video sequence analysis techniques. This approach is then applied both for video summarization, in the context of a content-based sampling algorithm, and for content-based indexing and retrieval. In the first case, video summarization is accomplished by discarding shots or frames of similar visual content so that only a small but meaningful amount of information is retained (key-frames). In the second case, a content-based retrieval scheme is investigated, so that the most similar images to a query are extracted. Experimental results and comparison with other known methods are presented to indicate the good performance of the proposed scheme on real-life video recordings.

Zusammenfassung

In dieser Arbeit wird eine unscharfe (fuzzy) Darstellung des visuellen lnhalts vorgeschlagen, die von Nutzen für die neu aufkommenden Multimedia-Anwendungen wie inhaltsbasierte Bildindizierung und -auffindung, Video-Browsing und Video-Zusammenfassung ist. Im speziellen wird ein mehrdimensionales fuzzy Histogramm für jedes Videobild konstruiert, welches auf einer Menge von geeigneten Merkmalen beruht, die mittels Verfahren zur Video-Sequenzanalyse extrahiert werden. Dieser Ansatz wird sowohl auf die Video-Zusammenfassung (in Zusammenhang mit einem inhaltsbasierten Abtastalgorithmus) als auch auf die inhaltsbasierte Indizierung und Auffindung angewandt. Im ersten Fall wird die Video-Zusammenfassung durch Verwerfen von Bildern mit ähnlichem visuellen Inhalt erzielt, so daß nur eine kleine aber aussagekräftige Menge von Information zurückbehalten wird (Schlüsselbilder). Im zweiten Fall wird eine inhaltsbasierte Auffindungsmethode untersucht, so daß die einer Anfrage am meisten entsprechenden Bilder extrahiert werden. Die gute Leistungsfähigkeit der vorgeschlagenen Methode bei echten Videoaufnahmen wird durch experimentelle Ergebnisse und durch Vergleich mit anderen bekannten Methoden gezeigt.

Résumé

Dans cet article, nous proposons une représentation floue du contenu visuel qui est utile pour les nouvelles applications multimédias émergeantes, telles l'indexage et la récupération d'images par le contenu, ou le parcours et le

* Corresponding author.
E-mail address: adoulam@image.ntua.gr (A.D. Doulamis)

0165-1684/00/$ - see front matter
PII: S0165-1684(00)00019-0

résumé vidéo. En particulier, un histogramme flou multidimensionnel est construit pour chaque trame vidéo, sur base d'une collection de caractéristiques appropriées, extraites en utilisant des techniques d'analyse de séquences vidéo. Cette approche est ensuite appliquée à la fois au résumé vidéo, dans le contexte d'un algorithme d'échantillonnage basé sur le contenu, et à l'indexage et la récupération basés sur le contenu. Dans le premier cas, le résumé vidéo est accompli est éliminant des plans ou des trames de contenu visuel similaire, (trames-clés). Dans le second cas, nous investiguons un schéma de récupération sur base du contenu, de sorte que les images les plus semblables à une requête soient extraites. Des résultats expérimentaux et une comparison avec d'autres méthodes connues sont présentées pour indiquer les bonnes performances du schéma proposé sur des enregistrements vidéo de la vie réelle

Keywords: Video summarization; Content-based retrieval; Fuzzy logic; Genetic algorithms

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 approach of keyword annotation to accessing image or video information has the drawback that, apart from the large amount of effort for developing annotations, it cannot efficiently characterize the rich visual content using only text. For this reason, *content-based retrieval* algorithms have been recently proposed and they have attracted a great research interest in the image processing community. [10,19]. Examples of content-based retrieval systems, either academic or in the first stage of commercial exploitation include the QBIC [8], Virage [11] or VisualSeek [21] prototypes. In this framework, the moving picture expert group (MPEG) is currently defining the new MPEG-7 standard [17], to specify a set of descriptors for an efficient interface of multimedia information.

The aforementioned systems are mainly restricted to still images and cannot easily be applied to video databases [4]. This is due to the fact that the standard representation of video as a sequence of consecutive frames results in significant temporal redundancy of visual content and thus it is very inefficient and time consuming to perform queries on every video frame. Furthermore, most video databases are often located on distributed platforms and impose both large storage and transmission bandwidth requirements, even if they are compressed. Such linear representation of video sequences is also not adequate for the new emerging multimedia applications, such as video browsing, content-based indexing and retrieval. For this reason, a *content-based sampling* algorithm is usually applied to video data for extracting a small but "meaningful" amount of the video information [3,13]. This results in a video summarization scheme similar to that used in document search engines, where a brief text summary corresponds to one or multiple documents.

However, efficient implementation of content-based retrieval algorithms and video summarization schemes requires a more meaningful representation of visual content than the traditional pixel-based one. This is due to the fact that there is a lack of semantic information at the pixel level. For this reason, several works have been presented in the literature towards a more efficient image/video representation. A hidden Markov model has been investigated in [15] for color image retrieval, while in [5] an approach of image retrieval based on user sketches has been reported. A hierarchical color clustering method has been presented in [22]. For video summarization, construction of a compact image map or image mosaics has been described in [13], while a pictorial summary of video sequences based on story units has been presented in [24].

In the context of this paper, a *fuzzy representation* of visual content is proposed for both video summarization and content-based indexing and retrieval. This representation increases the flexibility of content-based retrieval systems since it provides an interpretation closer to the human perception

[14]. It also results in a more robust description of visual content, since possible instabilities of the segmentation, used for describing the visual content, are reduced. In particular, the adopted fuzzy representation is applied for both video summarization and content-based retrieval. In the first case, a small set of key-frames is extracted which provides an efficient description of visual content. This is performed by minimizing a cross correlation criterion among the video frames by means of a genetic algorithm. The correlation is computed using several features extracted using a color/ motion segmentation on a fuzzy feature vector formulation basis. In the second case, the user provides queries in the form of images or sketches which are analyzed in the same way as video frames in video summarization scheme. A metric *distance* or *similarity measure* is then used to find a set of frames that best match the user's query.

This paper is organized as follows: In Section 2, the video sequences are analyzed by applying a color/motion segmentation algorithm. The extracted features for each color or motion segment are fuzzy classified as is presented in Section 3. Application of the proposed fuzzy representation schemes to video summarization is discussed in Section 4, while the application to content-based retrieval is discussed in Section 5. Furthermore, several practical implementation issues, such as selected parameters and numerical values, are also mentioned in these sections. Experimental results on a large image/video databases are presented in Section 6 along with comparisons with other known techniques to show the good performance of the proposed scheme. Finally, Section 7 concludes the paper.

2. Video sequence analysis

Semantic segmentation, i.e., extraction of meaningful entities, is essential in a content-based retrieval environment. However, this remains one of the most difficult problems in the image analysis community, especially if no constraints are imposed on the kind of the examined video sequences [6,7,9]. For this reason, a color/motion segmentation algorithm is applied in this paper for visual content description.

A multiresolution implementation of the *recursive shortest spanning tree* (RSST) algorithm, called M-RSST, is adopted, for both color and motion segmentation. The M-RSST recursively applies the RSST to images of increasing resolution. In particular, a truncated image pyramid is created, each layer of which contains a quarter of the pixels of the layer below. In the following steps, an iteration begins so that images of higher resolution are taken into consideration until the highest resolution level is reached.

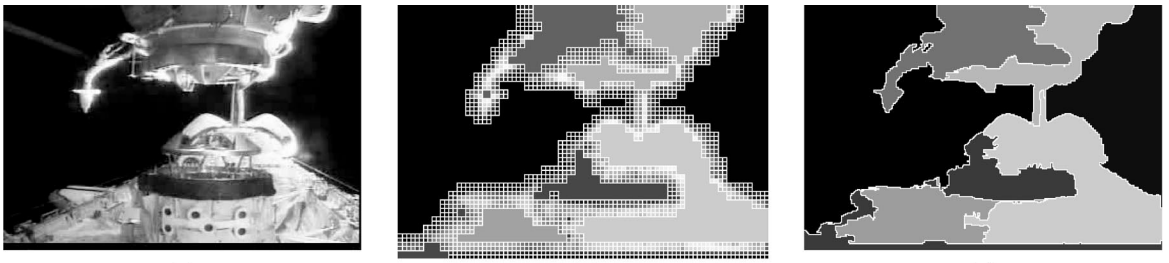
The use of the RSST algorithm is based on the fact that it is considered as one of the most powerful tools for image segmentation compared to other techniques, such as pyramidal region growing, morphological watershed or color clustering [20]. It has been shown by the COST211ter simulation subgroup, using several experiments on a set of generic video test sequences, that the RSST presents the best performance compared to the other methods [1,18]. In particular, it delineates the true content edges while giving as few arbitrary boundaries as possible. As far as the computational cost is concerned, the RSST is also the fastest algorithm among all the examined ones [1].

However, the complexity of the RSST still remains very high especially for images of large size. Instead, the proposed M-RSST approach yields much faster execution time, while simultaneously keeping the same performance. Comparison of the computational load of the RSST and M-RSST at different image sizes is depicted in Table 1, using a C implementation on a Sun Ultra 10 (333 MHz) workstation. Another benefit of the M-RSST is that it eliminates regions of small segments, which, in general, are not preferable in the context of content-based indexing and retrieval. This is due to the fact that the algorithm begins with a low image resolution, while no segments are created or destroyed at higher resolution levels.

The initial image resolution level is selected to be 3 (downsampling by 8×8 pixels) so that information directly available in the MPEG stream is exploited [6]. This selection results in a reduction of computational complexity since no decoding of the images at the initial resolution level is required. At higher-resolution levels, only the "boundary" blocks are decoded. Since these blocks compose

Table 1
Execution times of color segmentation, fuzzy representation and color histogram

	Execution times			
	RSST (s)	M-RSST (ms)	Fuzzy scheme ($Q = 3$, Triangular) (ms)	Color histogram (ms)
176×144 (QCIF)	5.65	132.01	1.32	8.67
352×288 (CIF)	44.21	382.34	1.32	28.35
720×576 (PAL)	534.22	1360.90	1.32	103.45



(a) (b) (c)

Fig. 1. Color segmentation results, using the M-RSST algorithm, (a) original frame; (b) segment splitting at the initial resolution level; (c) final segmentation result.

(a) (b) (c)

Fig. 2. Motion segmentation results for frame #29 of the Monitor sequence; (a) motion vectors derived from the MPEG sequence; (b) motion vectors after post processing; (c) final segmentation.

a small percentage of image blocks, as shown in Fig. 1(b), the overall complexity for segmentation is decreased.

Another parameter, which affects the segmentation efficiency, is the selection of threshold used for terminating the algorithm. In our case, an adaptive procedure is used for threshold estimation. Particularly, at each iteration, the Euclidean distance of the color or motion intensities between two neighboring segments, weighted by the harmonic mean of their areas, is first calculated and then the distance histogram is created. The half of maximum histogram value is considered as the appropriate threshold for the given iteration. Finally, the segmentation is terminated if no segments are merged from one step to another. Fig. 1(c) depicts the segmentation results for the image of Fig. 1(a). Fig. 1(b) presents the boundary blocks, which are split into four new segments, at the initial resolution of the algorithm. Fig. 2(c) shows the motion

segmentation results using the image of Fig. 2(a), while Fig. 2(b) presents the motion vectors after being post-processed with a median filter so that possible noise effect is removed.

3. Fuzzy visual content representation

The size, location and average color components of all color segments are used as color properties. In a similar way, motion properties include the size, location and average motion vectors of all motion segments. Since the segment number is not constant for each video frame, the aforementioned properties cannot be directly included in a feature vector, because the size of this vector is not constant. Thus, direct comparison between vectors of different frames is practically impossible. For example, a frame consisting of 10 segments results in a 10×1 feature vector, while a frame of five segments results in a 5×1 feature vector. To overcome this problem, we classify color/motion properties into pre-determined classes. In this framework, each element of the feature vector corresponds to a specific class.

To avoid 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 [14]. As a result, each sample, i.e., segment in our case, can belong to several or all classes, but with different degrees of membership. Then, *a fuzzy multidimensional histogram* is created. In a conventional classification, each sample (segment property) belongs only to one class, so that it is possible for two samples (segments) of similar properties to be assigned to different classes if they are located near the class boundary. Instead, in a fuzzy classification approach, such samples would slightly differ in their degree of membership to adjacent classes.

To clearly explain the proposed fuzzy classification approach, we first consider the simple case that only one property, say s, is used for each segment. We further assume, without loss of generality, that s takes values in [0,1] and it is classified into Q classes (partitions) using Q *membership functions* $\mu_n(s)$, $n = 1, 2, \ldots, Q$. Functions $\mu_n(s)$ denote the degree of membership of s in the nth class and take values in the range [0,1]. Values of $\mu_n(s)$ near unity (zero) indicate high (low) degree of membership of feature s in the nth class.

The exact type and shape of the membership functions $\mu_n(s)$ can be greatly varied [14] and in general depends on the specific problem. Several functions have been proposed in the literature as membership functions. One of the simplest are the triangular ones, which are illustrated in Fig. 3 for $Q = 3$ partitions. Higher-order functions have been also proposed and depicted in Fig. 4(a) along with the triangular ones for comparison purposes. As the order increases the membership functions provide "harder" classification closer to the binary logic. Functions of trapezoid shape are illustrated in Fig. 4(b) for different line slopes β. As expected, the higher the slope is the "harder" the classification becomes. In all the above cases, "symmetric" functions have been used since there is no reason to give more importance to a specific class.

Let us also assume that the examined video frame consists of K segments or samples. First, for each feature s_i, $i = 1, \ldots, K$, of the ith segment, the degree of membership of feature s_i in all Q classes is evaluated. Then, the degree of membership of all K segments of the respective frame to the nth class is calculated through the *fuzzy histogram*, say $H(n)$, which is defined as follows:

$$H(n) = \frac{1}{K} \sum_{i=1}^{K} \mu_n(s_i), \quad n = 1, 2, \ldots, Q. \tag{1}$$

An arithmetic example is presented in the following for clarification purposes. In particular, we assume that $K = 2$ segments have been extracted while their average luminance values, 120 and 238, respectively, are considered as the only segment property. If $Q = 2$ and triangular functions are used, then the degree of membership of each segment to the two available classes is: $\mu_1(120/255) = 0.53$, $\mu_2(120/255) = 0.47$ for the first segment and $\mu_1(238/255) = 0.07$, $\mu_2(238/255) = 0.93$ for the second segment. (The division by 255 is used for normalization in the interval [0 1].) Then, from (1), the 2×1 fuzzy histogram vector is $[H(1)\ H(2)]^{\mathrm{T}} = [(0.53 + 0.07)/2\ (0.47 + 0.93)/2]^{\mathrm{T}} = [0.3\ 0.7]^{\mathrm{T}}$. Instead, if binary classification was

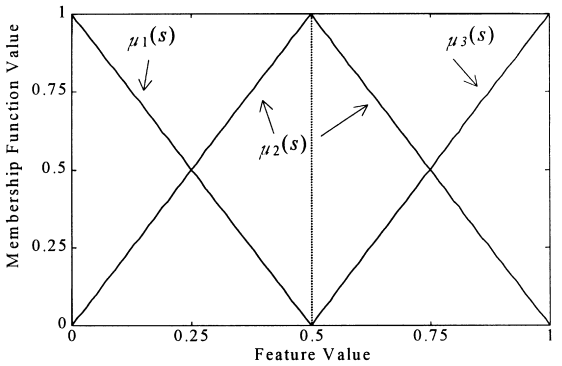


Fig. 3. Triangular membership functions in the case of $Q = 3$ partitions.

used, the histogram vector would be $[1\ 1]^{\mathrm{T}}$ and thus the two classes would be of equal importance.

The more general case of more than one segment properties is considered next. In particular, let us assume that K^{c} color and K^{m} motion segments have been extracted. Then, for each color segment S_i^{c}, $i = 1, \ldots, K^{\mathrm{c}}$, an $L^{\mathrm{c}} \times 1$ vector $\boldsymbol{s}_i^{\mathrm{c}}$ is formed, while for each motion segment S_i^{m} $i = 1, 2, \ldots, K^{\mathrm{m}}$ an $L^{\mathrm{m}} \times 1$ vector $\boldsymbol{s}_i^{\mathrm{m}}$ is formed as follows:

$$\boldsymbol{s}_i^{\mathrm{c}} = [\boldsymbol{c}^{\mathrm{T}}(S_i^{\mathrm{c}})\ \boldsymbol{l}^{\mathrm{T}}(S_i^{\mathrm{c}})\ a(S_i^{\mathrm{c}})]^{\mathrm{T}}, \tag{2a}$$

$$\boldsymbol{s}_i^{\mathrm{m}} = [v^{\mathrm{T}}(S_i^{\mathrm{m}})\ \boldsymbol{l}^{\mathrm{T}}(S_i^{\mathrm{m}})\ a(S_i^{\mathrm{m}})]^{\mathrm{T}}, \tag{2b}$$

where a denotes the size of the color or motion segment, and $\boldsymbol{l}$ is a 2×1 vector indicating the horizontal and vertical locations of the segment center; the 3×1 vector $\boldsymbol{c}$ includes the average values of the three color components of the respective color segment, while the 2×1 vector $\boldsymbol{v}$ includes the average motion vector of the motion segment. Thus, $L^{\mathrm{c}} = 6$ for color segments and $L^{\mathrm{m}} = 5$ for motion segments. For the sake of notational simplicity, the superscripts c and m will be omitted in the sequel; each color or motion segment will be denoted as S_i and will be described by the $L \times 1$ vector $\boldsymbol{s}_i$ where $L = 5$ or $L = 6$ depending on the segment type.

Based on the above, $\boldsymbol{s}_i = [s_{i,1}\ s_{i,2}\ \ldots\ s_{i,L}]^{\mathrm{T}}$ is a vector containing all properties extracted from the ith segment, S_i. For example $s_{i,\,1}$ corresponds to the average value of the first color component of segment S_i. Each element $s_{i,j}, j = 1, 2, \ldots, L$ of vector $\boldsymbol{s}_i$ is then partitioned into Q regions by means of Q membership functions $\mu_{n_j}(s_{i,j})$ $n_j = 1, 2, \ldots, Q$. As in the previous case, $\mu_{n_j}(s_{i,j})$ denotes the degree of membership of $s_{i,j}$ to the n_jth class. Then, the product of $\mu_{n_j}(s_{i,j})$ over all $s_{i,j}$ of $\boldsymbol{s}_i$ defines the degree of membership of vector $\boldsymbol{s}_i$ to the L-dimensional class $\boldsymbol{n} = [n_1 n_2\ \ldots\ n_L]^{\mathrm{T}}$ the elements of which express the class to which the elements of $\boldsymbol{s}_i$ belong.

$$\mu_{\boldsymbol{n}}(\boldsymbol{s}_i) = \prod_{j=1}^{L} \mu_{n_j}(s_{i,\,j}). \tag{3}$$

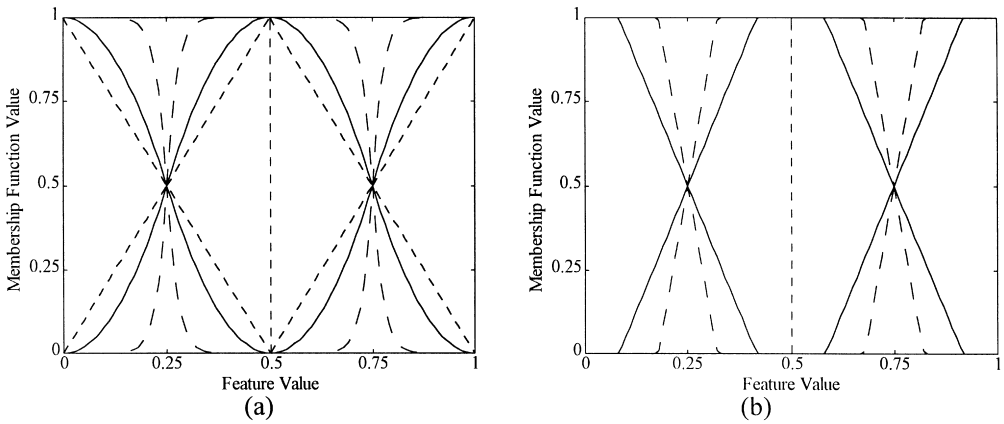


Fig. 4. Different membership functions: (a) quadratic (solid line), triangular (dotted line) and functions of order 10 (dashed line); (b) trapezoid membership functions, slope $\beta = 3$ (solid line) and $\beta = 7$ (dashed line).

The product operator in the previous equation is one of the most used T- norms in fuzzy logic [14].

Gathering all segments of a frame, a *multidimensional fuzzy histogram* is created exactly as in the previous case [Eq. (1)].

$$H(\boldsymbol{n}) = \frac{1}{K}\sum_{i=1}^{K} \mu_{\boldsymbol{n}}(\boldsymbol{s}_i) = \frac{1}{K}\sum_{i=1}^{K}\prod_{j=1}^{L} \mu_{n_j}(s_{i,j}). \tag{4}$$

$H(\boldsymbol{n})$ thus can be viewed as a degree of membership of a whole frame to class $\boldsymbol{n}$. A frame feature vector $\boldsymbol{f}$ is then formed by gathering all values of $H(\boldsymbol{n})$ for all classes $\boldsymbol{n}$, i.e., for all Q^{L} combinations of indices, resulting in a vector of Q^{L} elements: $\boldsymbol{f} = [f_1\, f_2\, \ldots\, f_{Q^{\mathrm{L}}}]^{\mathrm{T}}$. In particular, the vector $\boldsymbol{f}$ is constructed from $H(\boldsymbol{n})$ using an index function $z(\boldsymbol{n})$ which maps the class $\boldsymbol{n}$ to an integer between 1 and Q^{L},

$$z(\boldsymbol{n}) = 1 + \sum_{j=1}^{L} (n_j - 1)Q^{L-j}. \tag{5}$$

Then, the elements f_i, $i = 1, \ldots, Q^{\mathrm{L}}$, of $\boldsymbol{f}$ are calculated as $f_{z(n)} = H(\boldsymbol{n})$ for all classes $\boldsymbol{n}$. In fact, since the above analysis was based on features $\boldsymbol{s}_i^{\mathrm{c}}$ and $\boldsymbol{s}_i^{\mathrm{m}}$ of color S_i^{c} and motion S_i^{m} segments, respectively, two feature vectors will be calculated; a color feature vector $\boldsymbol{f}^{\mathrm{c}}$ for color segments and a motion feature vector $\boldsymbol{f}^{\mathrm{m}}$ for motion segments. Thus, the total feature vector of an image is formed as follows:

$$\boldsymbol{f} = [(\boldsymbol{f}^{\mathrm{c}})^{\mathrm{T}}\,(\boldsymbol{f}^{\mathrm{m}})^{\mathrm{T}}]^{\mathrm{T}}. \tag{6}$$

To further explain the proposed fuzzy video content representation a simple example is presented in the following. We first assume that $K = 2$ segments have been extracted and only the three average color components (RGB) are used as segment properties. In particular, we consider that the values of vectors $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, which describe the segment properties [Eq. (2a) and (2b)] are $\boldsymbol{s}_1 = [53\ 145\ 21]^{\mathrm{T}}$ and $\boldsymbol{s}_1 = [200\ 2\ 144]^{\mathrm{T}}$. If $Q = 2$ and triangular membership functions are used then the three-dimensional class $\boldsymbol{n} = [n_1 n_2 n_3]^{\mathrm{T}}$ has $2^3 = 8$ different combinations since n_1, n_2, n_3 take values 1 or 2. The value $n_1 = 1$ indicates low R color component, whereas $n_1 = 2$ corresponds to high R color component. Similarly, low G (B) is represented by $n_2 = 1$ ($n_3 = 1$), while high G (B) by $n_2 = 2$ ($n_3 = 2$). As a result, the class $\boldsymbol{n} = [1\ 1\ 1]^{\mathrm{T}}$ indicates that all color components (R,G,B) take "low" values, while $\boldsymbol{n} = [1\ 1\ 2]^{\mathrm{T}}$ means that the R and G components are low while the B is high. Eq. (3) is used to estimate the degree of membership of vector $\boldsymbol{s}_1$ or $\boldsymbol{s}_2$ to a given class $\boldsymbol{n}$. For instance, for $\boldsymbol{n} = [1\ 1\ 1]^{\mathrm{T}}$ the $\mu_{[1\ 1\ 1]^{\mathrm{T}}}(\boldsymbol{s}_1) = 0.314$ while for $\boldsymbol{s}_2$ $\mu_{[1\ 1\ 1]^{\mathrm{T}}}(\boldsymbol{s}_2) = 0.09$. Hence, the $H([1\ 1\ 1]^{\mathrm{T}}) = 0.202$ [Eq. (4)].

Let us denote as γ the computational complexity for estimating $\mu_{\boldsymbol{n}}(\boldsymbol{s}_i)$ [Eq. (3)] for a given combination of $\boldsymbol{n}$. Then, the total complexity for all K extracted segments and for all combinations (classes) of $\boldsymbol{n}$ is $K \cdot Q^L \cdot \gamma$. As can be seen the computational load increases *exponentially* with respect to the partition number Q, and *proportionally* to the number of segments K and membership function complexity γ. Thus, the complexity load is mainly affected by the partition number Q. Furthermore, large partition numbers Q also increase storage requirements since a $Q^L \times 1$ feature vector is assigned to each video frame.

However, the partition numbers Q as well as the shape of the membership functions, apart from the computational complexity, affects the performance of the proposed fuzzy representation method to the problem of content-based retrieval. Intuitively, small number of partitions is not able to describe with high efficiency the visual content. On the contrary, large number of classes leads to "noisy" classification. The appropriate selection of membership functions and number of partitions is described in Section 5.1 of this paper. In this section, we also present the effect of different types of membership functions and partition numbers on computational complexity. The application of the proposed fuzzy visual content representation to the problem of video summarization is described in Section 4, while Section 5 presents the application to the problem of content-based retrieval.

4. Video summarization

Fig. 5 depicts the block diagram of the proposed video summarization scheme. 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 cut detection*

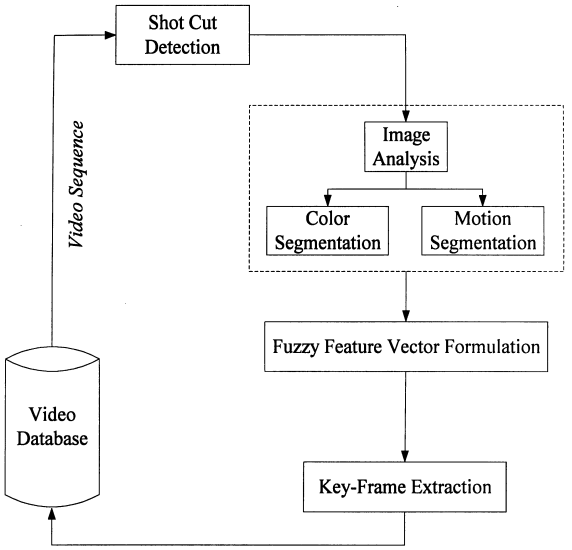


Fig. 5. A block diagram of the proposed scheme for video summarization.

algorithm is first applied, to identify video frames of similar visual content. The algorithm proposed in [22] has been adopted for this purpose, since it presents high accuracy and low computational complexity compared to other techniques [3]. Then, analyzing the video sequence as presented in Section 2, color and motion properties are extracted for each frame and then classified using fuzzy formulation. Finally, a content-based sampling algorithm is applied so as to discard frames, which present similar visual content.

4.1. Extraction of key-frames

Key-frames are extracted by minimizing a cross-correlation criterion, so that the selected frames are not similar to each other.

Let us denote by $\boldsymbol{f}_k$ the feature vector of the kth frame of a shot, with $k \in V = \{1, 2, \ldots, N_F\}$, where N_F is the total number of frames of the given shot. Let us also denote by K_F the number of key-frames that should be selected. In order to define a measure of correlation among K_F feature vectors, an index vector is first defined $x = (x_1, \ldots, x_{K_F}) \in W \subset V^{K_F}$, where $W = \{(x_1, \ldots, x_{K_F}) \in V^{K_F} : x_1 < \cdots < x_{K_F}\}$ is the subset of V^{K_F} containing all sorted index vectors $\boldsymbol{x}$ which contains the frame numbers or time indices of candidate key-frames. Then, the correlation measure among the K_F feature vectors is given by following equation:

$$R(\boldsymbol{x}) = R(x_1, \ldots, x_{K_F}) = \frac{2}{K_F(K_F-1)} \sum_{i=1}^{K_F-1} \sum_{j=i+1}^{K_F} \rho^2_{x_i, x_j}, \tag{7}$$

where ρ_{x_i, x_j} denotes the correlation coefficient between feature vectors $\boldsymbol{f}_{x_i}, \boldsymbol{f}_{x_j}$, which corresponds to frames with numbers x_i and x_j. Function $R(\boldsymbol{x})$ takes values in the interval [0,1]. Values close to zero mean that the K_F feature vectors are uncorrelated while values close to one indicate that K_F feature vectors are strongly correlated.

Based on the above definition, it is clear that searching for a set of K_F minimally correlated feature vectors is equivalent to searching for an index vector $\boldsymbol{x}$ that minimizes $R(\boldsymbol{x})$. Searching is limited in the subset W, since the correlation measure of the K_F features is independent of the feature arrangement. Consequently, the set of the K_F least-correlated feature vectors is found by

$$\hat{\boldsymbol{x}} = (\hat{x}_1, \ldots, \hat{x}_{K_F}) = R^{-1}(\boldsymbol{x}). \tag{8}$$

Unfortunately, the complexity of an exhaustive search for obtaining the minimum value of $R(\boldsymbol{x})$ is such that a direct implementation of the method is practically unfeasible. For example, about 264 million combinations of frames should be considered (each of which requires several computations for estimation of $R(\boldsymbol{x})$) if we wish to select 5 representative frames out of a shot consisting of 128 frames. For this purpose, a logarithmic search algorithm has been proposed in [3] for efficient implementation of the optimization procedure. Although this approach provides very fast convergence, it is highly probable for the solution to be trapped to a local minimum resulting in a sub-optimal solution. This drawback is alleviated by the use of a guided random search procedure implemented by a *genetic algorithm* (GA) [16].

4.1.1. The genetic approach

In the GA approach, the index vector $\boldsymbol{x} = (x_1, \ldots, x_{K_F})$ is considered as a chromosome, while

the elements $x_1, \ldots, x_{K_F}$ of $\boldsymbol{x}$ as the genetic material of the respective chromosome. For instance, in case that we are interested in extracting four ($K_F = 4$) key-frames from a shot composing $N_F = 100$ frames, a chromosome of the form (5,10,52,90) indicates that the frames with numbers 5, 10, 52 and 90 are possible key-frames. Initially, a population of m chromosomes is created, say $P(0)$, consisting of m randomly selected index vectors $\boldsymbol{x}$. That is, $P(0) = \{\boldsymbol{x}_1(0), \ldots, \boldsymbol{x}_m(0)\}$, where $\boldsymbol{x}_i(0)$, $i = 1, \ldots, m$ corresponds to the ith chromosome, i.e., $K_F \times 1$ index vectors of population $P(0)$.

As can be seen, the codification of the examined problem presents similarities to the traveling salesman problem (TSP). In particular, in the TSP chromosomes of a population are represented by vectors which express a salesman's tour, while the vector elements (genetic material) indicate the cities that the salesman visits in this tour (see page 218 of [16]). The object of TSP is to find that chromosome which yields the minimum travelling cost. Similarly, in our approach, chromosomes, which are represented by index vectors, express candidate key-frames (like tour of the TSP), while their elements express the frame indices (like cities of the TSP) of the candidate key-frames. Different combinations of frame indices result in different correlation measures, while in the TSP different permutations of cities gives different travelling costs (see p. 211 of [16]).

The correlation measure of (7) is used to evaluate the performance of all chromosomes in population $P(n)$ of the nth iteration. However, the lower the correlation measure of a chromosome the higher is its performance. For this reason, the fitness function, which is responsible for measuring the chromosome quality (proportional to the chromosome performance), is completed related to the correlation function [16]. Particularly, for the ith chromosome of the nth population $\boldsymbol{x}_i(n)$ the fitness function is given by

$$F(\boldsymbol{x}_i(n)) = D - R(\boldsymbol{x}_i), \tag{9}$$

where the constant D is selected such that negative values of the fitness function are avoided. In our case $D = 1$ since the correlation measure is normalized in the interval [0 1].

Based on the fitness values, $F(\boldsymbol{x}_i(n))$, $i = 1, \ldots, m$, for all chromosomes of the current population, appropriate "*parents*" are selected so that a fitter chromosome gives a higher number of offspring and thus has a higher chance of survival in the next generation. In particular, in our case, a probability is assigned to each chromosome, equal to $F(\boldsymbol{x}_i(n))/\sum_{i=1}^{p} F(\boldsymbol{x}_i(n))$ and then $p_o < m$ chromosomes are randomly selected based on their assigned probabilities as candidate parents (roulette *wheel selection* procedure [16]).

A set of new chromosomes (offspring) is then produced by mating the genetic material of the parents using a *crossover* operator. A simple crossover mechanism is illustrated in Fig. 6(a) and produces a chromosome whose genetic material, until a random position (*crossover point*), comes from the

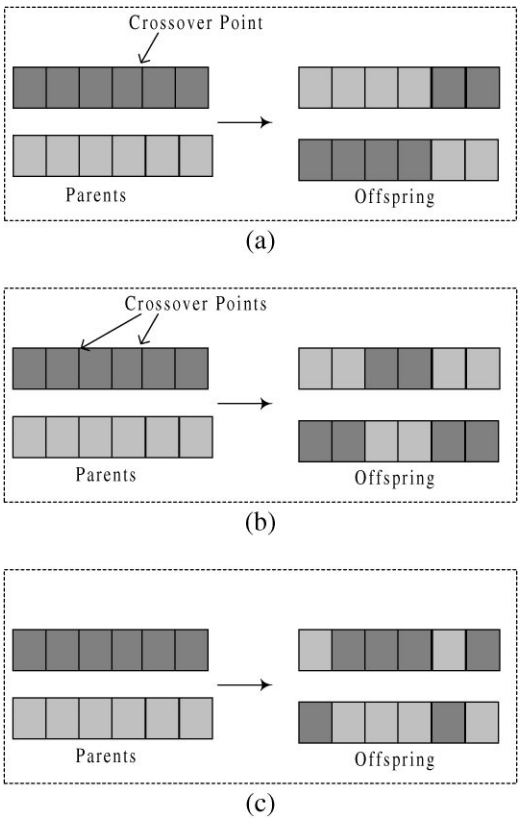


Fig. 6. A graphical representation of the three examined crossover mechanisms: (a) 1-random crossover operator; (b) 2-random crossover operator; (c) uniform crossover operator.

first parent while all the others come from the second parent. This is called 1-*random* crossover mechanism in the following. Another method is to use more than one, usually two, crossover points (2-*random* crossover) as explained in Fig. 6(b). In a more general case, the selected material of two parents is randomly mated resulting in a *uniform*-crossover operator as in Fig. 6(c).

The next step of the algorithm is to apply *mutation* 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 our case, the genetic material of a chromosome is mutated to a randomly selected frame number (index) of the examined shot with a probability p_m. Otherwise, the gene remains intact. After that, the next-generation population $P(n+1)$ is created by inserting the new chromosomes and deleting the older ones. Several GA cycles take place by repeating the procedures of fitness evaluation, parent selection, crossover and mutation, until the population converges to an optimal solution. The GA terminates when the best chromosome fitness remains constant for a large number of generations, indicating that further optimization is unlikely.

4.2. Implementation issues

As can be seen from the previous subsection, several parameters are involved in the GA implementation. In this section, we discuss the impact of these parameters on the performance of the GA to the problem of video summarization.

In order to tune the most appropriate parameters of the GA, an experiment was carried out by examining about 170 shots of a video database. The number of key-frames has been selected to be $K_F = 6$. The average correlation measure over all shots is used as objective function in this case. Furthermore, triangular membership functions and $Q = 3$ partitions are used to classify the color and motion features. This is due to the fact that these parameters provide better representation of the visual content as explained in Section 5.1. Fig. 7 illustrates the convergence of the GA for the three different crossover mechanisms depicted in Fig. 6. In particular, Fig. 7(a) presents the results obtained when no mutation is used, while Fig. 7(b and c) show the results for mutation probability $p_m = 0.04$ and 0.1 respectively. In all cases, the population consists of 80 chromosomes, while, at each iteration, 30 parents are selected. It is observed that, for low mutation rates, the uniform crossover operator outperforms the other two examined crossover methods since it converges faster to the optimal solution. However, as p_m increases, the effect of the three crossover operators to the GA performance becomes minimal, though the uniform one still yields slightly better results on average. The same conclusions are drawn from Fig. 8(a), which presents the average correlation measure versus p_m for the three aforementioned crossover mechanisms. In this case, the number p_o of chromosomes used as parents has been selected to be 10 to show the effect of crossover operators to a different number of p_o, while 100 GA cycles have been used to terminate the iteration process. Consequently, the uniform crossover operator seems to provide better results and this is selected in the following.

The effect of mutation probability, p_m, on the GA performance is also depicted in Fig. 8(a). It seems that the minimum correlation is given in the range of [0.04–0.06] for all crossover mechanisms. Small mutation probability may trap the solution to a local minimum. Instead, large probability leads to random search, which deteriorates the GA performance. Fig. 8(b) presents the mutation impact on the GA performance for different values of p_o in case of uniform crossover operator and 100 GA cycles. Mutation probabilities around [0.04 0.06] provide better results in this case too. The effect of the mutation probability on the average computational load is shown in Table 2. In this case the results have been obtained on a Sun Ultra 10, using uniform crossover operator and $p_o = 10$. It should be mentioned that the execution times refer to the total GA procedure but with different mutation probabilities. Particularly, Table 2 presents the average load per 100 GA cycles, the average number of iterations required to achieved a correlation measure less than 0.278 over all the examined 170 shots and the total load. In conclusion, mutation probability around 0.06 provides the best results.

Then, the effect of the number of parents selected p_o is evaluated. Fig. 9 shows the average correlation

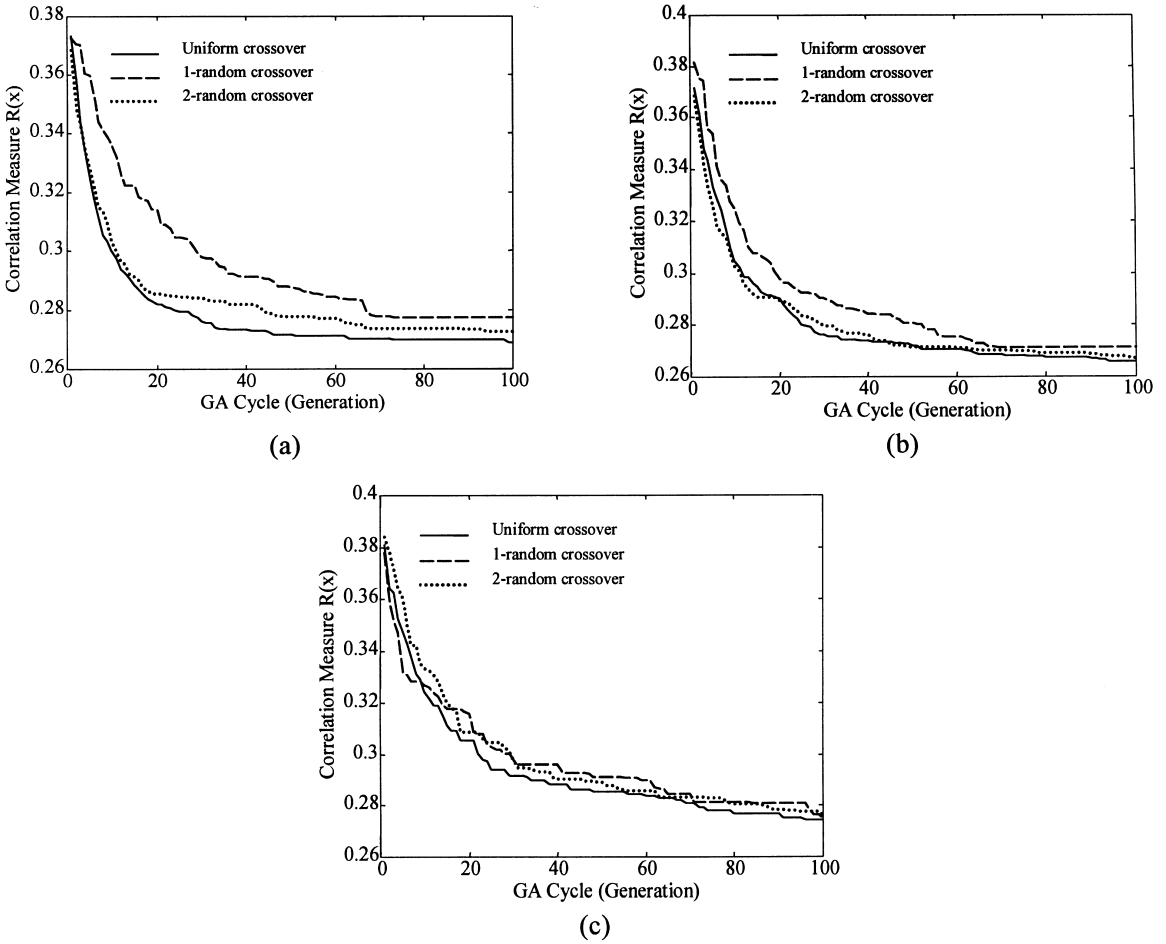


Fig. 7. The genetic algorithm convergence for different crossover mechanisms in the case of $p_o = 30$: (a) no mutation; (b) mutation probability 0.04; (c) mutation probability 0.1.

measure versus GA cycles for four different values of p_o (10 30 50 and 60, respectively) and $p_m = 0.06$ and uniform crossover operator as they have been selected by the previous analysis. It can be seen that as the value of p_o increases, the algorithm reaches the optimal solution in fewer GA cycles. However, the number p_o significantly affects the computational complexity of the GA. Table 3 presents the average computational load per 100 GA cycles for different values of p_o over all the 170 shots along with the average iterations required to achieve a correlation less than 0.26. It can be seen that $p_o = 30$ provides a faster convergence although it requires greater number of genetic cycles than other p_o values. As a result, the most appropriate GA parameters are; uniform crossover operator, $p_m = 0.06$ and $p_o = 30$.

5. Content-based retrieval

The problem of content-based retrieval from image and video databases is discussed in this section. Particularly, for content-based *video* retrieval the aforementioned video summarization scheme is applied so that all the redundant temporal video information is discarded. At this point, the problem of content-based retrieval from a video database

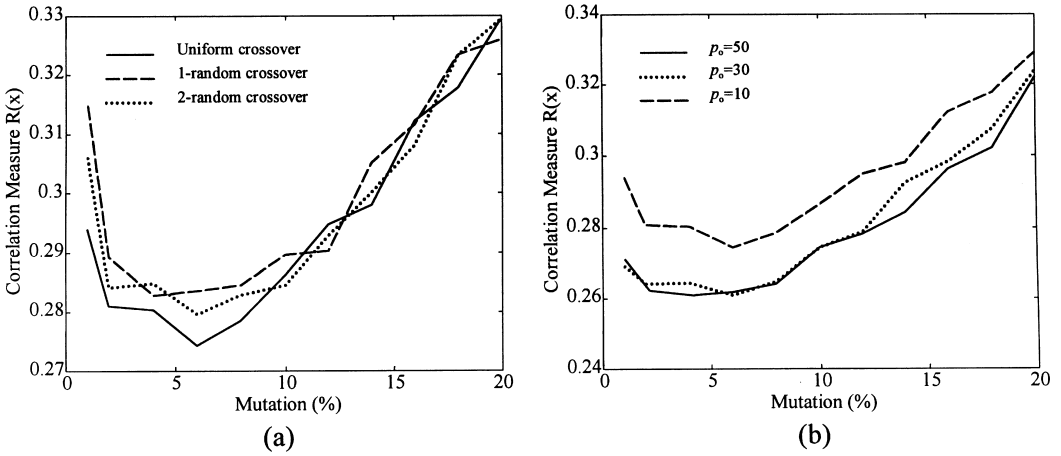


Fig. 8. The correlation measure versus mutation probability for 100 GA cycles: (a) different crossover operators and $p_o = 10$; (b) different values of p_o and uniform crossover.

Table 2
Average execution of the genetic algorithm for different mutation probabilities in case of uniform crossover mechanism and $p_0 = 10$

Mutation rate (%)	Cost/(100 GA cycles) (ms)	Average iterations	Total cost (ms)
1	79.4	334.4	265.51
2	82.82	295.6	244.82
4	89.1	130.2	116.01
6	93.92	98.2	92.22
8	97.22	210.2	204.36
10	101.12	395.6	400.03

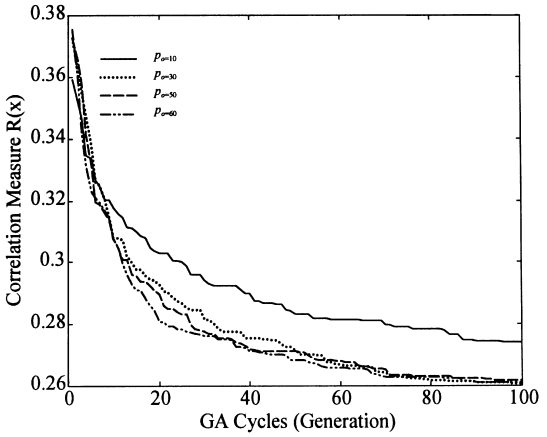


Fig. 9. The genetic algorithm convergence for different values of p_o in case of mutation probability 0.06 and uniform crossover operator.

has actually reduced to still image retrieval since *video queries* are applied on the selected key-frames. The proposed fuzzy representation scheme is then employed by assigning a fuzzy feature vector to each still image or key-frame of the database.

The user's queries can be submitted to the system in the form of images (*query by example*) or sketches (*query by sketch*). The goal in all cases is to retrieve the best M images from the database, whose visual content is close to the user's query. The submitted image or sketch is first analyzed similarly to the images of the database, by applying the M-RSST segmentation algorithm, and then the extracted segment properties are classified using the proposed fuzzy representation scheme. A distance or similarity measure is used to find the set of images that best match the user's query. Let us denote as $\boldsymbol{f}_q$ the feature vector of the user's query and as $\boldsymbol{f}_i$, the respective vector of the ith image in the database. Then, a weighted Euclidean distance is adopted as similarity measure between the vector $\boldsymbol{f}_q$ and all the vectors $\boldsymbol{f}_i$ in the database,

$$d(f_q, \boldsymbol{f}_i) = \sum_{j=1}^{Q^L} w_i (f_{q,\,j} - f_{i,\,j})^2, \tag{10}$$

Table 3
Average execution of the genetic algorithm for different values of p_o in case of uniform crossover mechanism and mutation probability 0.06

p_0	Cost/(100 GA cycles) (ms)	Average iterations	Total cost (ms)
10	89.10	699.8	623.52
20	181.24	325.2	589.39
30	275.40	146.0	402.08
40	366.24	122.1	447.18
50	461.04	108.0	497.92
60	579.12	102.4	593.01

where $f_{q,j}$ and $f_{i,j}$ are the jth element of vectors $\boldsymbol{f}_q$ and $\boldsymbol{f}_i$, respectively. The parameter w_i regulates the importance of each feature element to the similarity distance. The set of weights can be assigned either by the user, according to his information needs or can be adjusted automatically through a *relevance feedback mechanism* [2]. In this paper, all the weights are selected to be equal to one, $w_i = 1$ for all i, meaning that we impose the same importance on all features elements. The set of M images in the database with minimum distances $d(\boldsymbol{f}_q, \boldsymbol{f}_i)$ are returned as the most appropriately retrieved images to the user's query.

5.1. Implementation issues

The performance of the content-based retrieval mechanism depends on the selected values of fuzzy representation; the shape of membership functions and the number of partitions Q. In this section, we present the practical details of the proposed fuzzy video content representation scheme, i.e., we present the values of fuzzy parameters used and the reason why these values give appropriate representation of visual content.

The following experiment is carried out to estimate the parameters of the fuzzy representation. Particularly, 15 image queries are submitted to a database consisting of 1250 images and keyframes about the space. Each time, the 10 most appropriate images are returned. To evaluate the performance of the content-based retrieval system the following procedure is used. First, the similarity measure [Eq. (10)] is normalized in the interval [0 1] as follows:

$$d_{\mathrm{nrm}}(\boldsymbol{f}_q, \boldsymbol{f}_i) = d_w(\boldsymbol{f}_q, \boldsymbol{f}_i)/(d_{\max} - d_{\min}), \tag{11}$$

where $d_{\mathrm{nrm}}(\boldsymbol{f}_q, \boldsymbol{f}_i)$ denotes the normalized distance and $d_{\min} = \min_i\{d(\boldsymbol{f}_q, \boldsymbol{f}_i)\}$, $d_{\max} = \max_i\{d(\boldsymbol{f}_q, \boldsymbol{f}_i)\}$ the minimum and maximum distances over all images of the database for a given query.

Normalization is performed to allow comparisons between different user's queries, which in general give different distance values. Then, for an image query, a *similarity degree*, say t_i, is assigned to all images of the database, which indicates how similar is the content of the ith image to the query. In our case, three similarity degrees are used; zero degree meaning that the image is quite similar to the user's query, 1° for irrelevant images and 0.5° for somehow relevant images. The absolute difference E_A of the normalized distance and similarity degree over all the best M retrieved images is used to evaluate the system performance to the user's query,

$$E_A = \frac{1}{|S_M|} \sum_{i \in S_M} \|d_{\mathrm{nrm}}(\boldsymbol{f}_q, \boldsymbol{f}_i) - t_i\|, \tag{12a}$$

where S_M is the set containing the best M retrieved images for a given user's query and $|S_M|$ its cardinality. The difference E_A expresses how close are the best M retrieved images to the user's query. Another approach is to examine the performance of the system over all relevant images to the user's query, i.e., images of similarity degree $t_i = 0$.

$$E_B = \frac{1}{|S_t|} \sum_{S_t = \{i:\ t_i = 0\}} d_{\mathrm{nrm}}(\boldsymbol{f}_q, \boldsymbol{f}_i), \tag{12b}$$

where $|S_t|$ is the cardinality of set S_t.

Fig. 10(a) illustrates the average error E_A for all the 15 examined image queries versus the number of partitions Q for triangular, trapezoid with line slope $\beta = 3$ and quadratic membership functions. In the same figure, the results obtained using binary classification are also depicted. It is observed that a partition number $Q = 3$ yields the best performance for membership functions. We also observe that the triangular functions give better results compared to the other examined functions for most

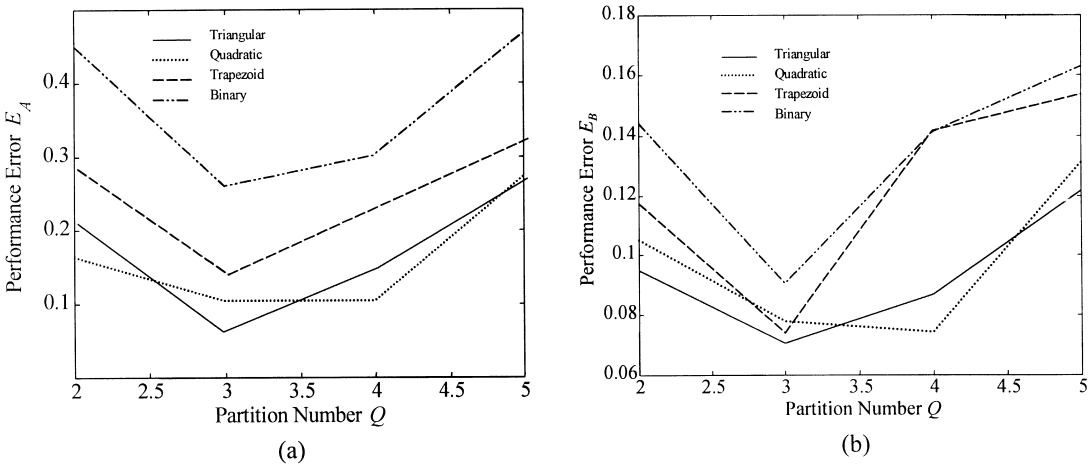


Fig. 10. Average performance error over 15 queries for different partition numbers and membership functions.

of the partition numbers. Similar results are presented in Fig. 10(b) where the average error E_B over all 15 queries is presented. Consequently, triangular membership functions and partition number $Q = 3$ provides the best performance.

Table 4 shows the execution times for different partitions Q and membership functions to indicate the complexity of the proposed fuzzy representation scheme. The results have been obtained on a Sun Ultra 10 system using $L^c = 6$ color and $L^m = 5$ motion properties. As expected, the computational load increases exponentially with respect to partition numbers. Instead, the times remain almost the same for different membership functions with functions of high order presenting the highest load. This is due to the fact that in the implementation additional costs are involved which are the same for all function types such as comparisons, procedure callings and so on. However, the execution times remain very small, just few ms, even for large partition numbers ($Q = 5$). A comparison of the execution times for fuzzy representation (in the case of $Q = 3$ and triangular membership functions) with the required times for color segmentation is presented in Table 1 at different image sizes. It seems that the required time for segmentation is much greater compared to the time of fuzzy representation especially in cases of images of large size. Moreover, large Q increases exponentially the storage requirements.

Table 4
Execution times of the fuzzy representation for different membership functions and partition numbers

	Membership function type			
Partion number Q	Triangular (ms)	Trapezoid (ms)	Quadratic (ms)	Binary (ms)
2	0.11	0.10	0.13	0.10
3	1.32	1.23	1.98	1.02
4	5.45	5.21	7.23	4.98
5	20.67	20.02	26.65	19.86

6. Experimental results

The proposed fuzzy representation of visual content has been evaluated both for video summarization and content-based indexing and retrieval, using a large database consisting of MPEG coded video sequences and several images compressed in JPEG format. The Optibase Fusion MPEG encoder at a bit-rate of 2 Mbits/s has been used to encode the video sequences.

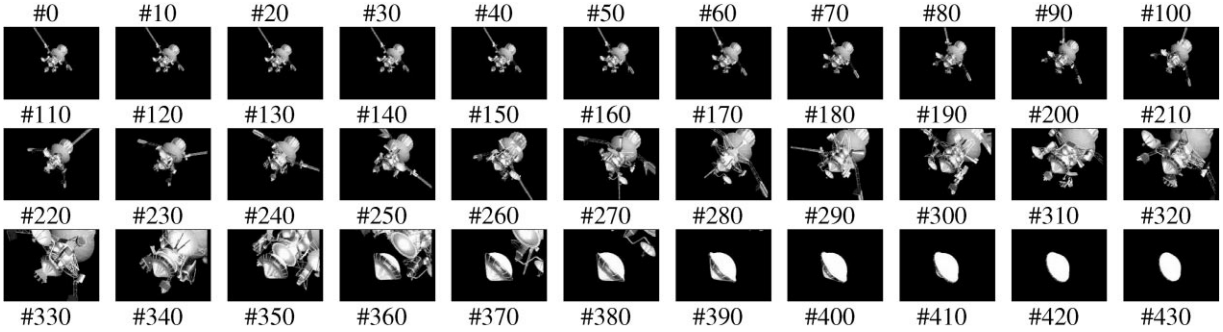


Fig. 11. A video shot about the space consisting of 334 frames, shown with one frame out of every 10.

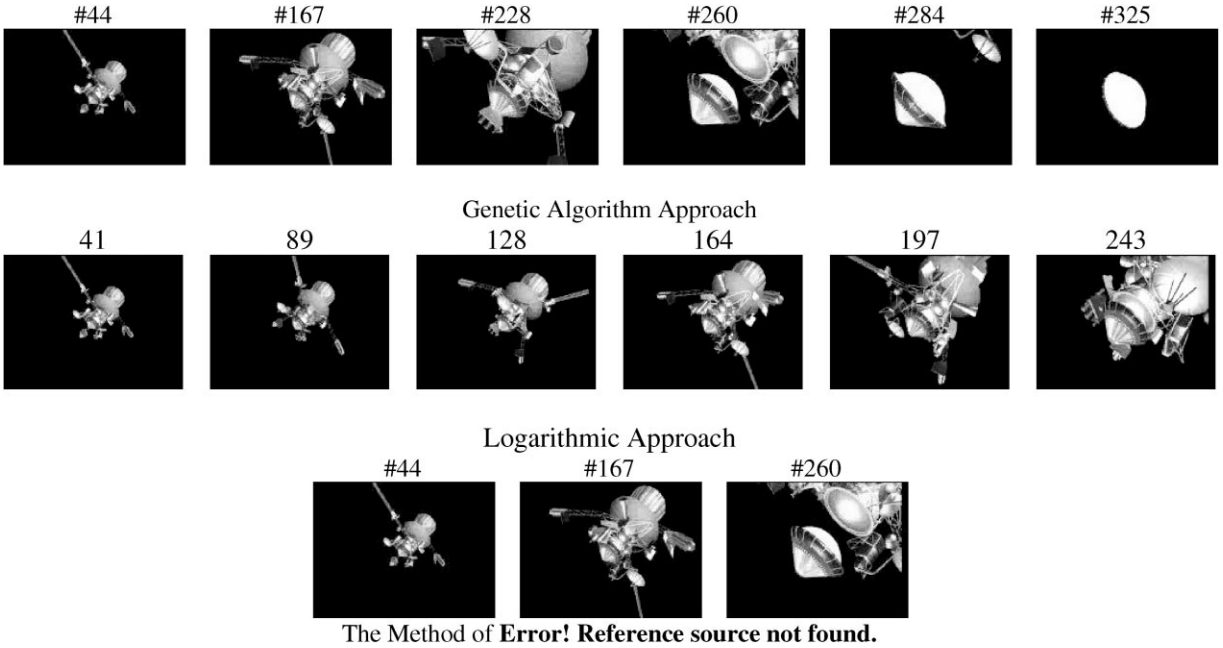


Fig. 12. The six key-frames selected using the genetic algorithm, the logarithmic approach and the method of [23].

Fig. 11 illustrates a shot used to demonstrate the performance of the key-frame extraction algorithm. The shot comes from an educational series about the space and consists of $N_F = 324$ frames. One out of every 10 frames is depicted, resulting in 32 frame thumbnails. Fig. 12 presents the six extracted key-frames of this shot obtained from the minimization of the cross correlation criterion of (7) using the GA approach. The parameters selected in Section 4.2 are used in this case, while $K_F = 6$. The number of key-frames extracted is estimated using the MDL criterion described in [12]. Key-frame selection is also compared with the logarithmic method presented in [3]. The respective results are presented in Fig. 12 for the same shot. It is observed that the GA approach provides a much better description of the visual content compared to the logarithmic method. In particular, the first three and the last

two frames present similar visual content, while there is no key-frame representing the last action of the shot. Finally, the method of [23] is used for comparing the proposed video summarization scheme. In this case, key-frames are extracted at time instances when the accumulated differences of the DC images exceed a pre-determined threshold [23]. This method highly depends on the selection of a threshold value, which is, in general an ad hoc process. In our case, the threshold has been tuned so that the average number of key-frames extracted for the whole sequence is the same as that of the proposed method. However, using this threshold, only three key-frames are extracted from the previous shot, which are not adequate for describing the visual content. Particularly, two similar frames have been extracted, while there is no key-frame for the last action of the sequence.

In order to estimate the efficiency of the algorithm in terms of the obtained correlation measure $R(\boldsymbol{x})$, a test of 100,000 random index vectors is performed, and a histogram of $R(\boldsymbol{x})$ is constructed, as depicted in Fig. 13. The optimal value of $R(\boldsymbol{x})$ obtained through the genetic algorithm (vertical solid line) and the logarithmic approach (vertical dotted line) is presented in this figure. As observed, the minimum value obtained through the genetic algorithm is lower than those of the random test and the logarithmic approach. It should be mentioned that the random test requires about 100 times more computational time, while the logarithmic approach is 3.55 times slower [3].

Content-based retrieval is next examined. In Fig. 14(a) an image of a space shuttle is submitted as user's query. The retrieval results are displayed in Fig. 14(b), using the fuzzy parameters selected in Section 5.1. Fig. 15 presents a comparison of the proposed method with two other methods; binary classification and the traditional method of color histogram [21] (Fig. 15(b and c)). The comparison is also performed quantitatively using both the

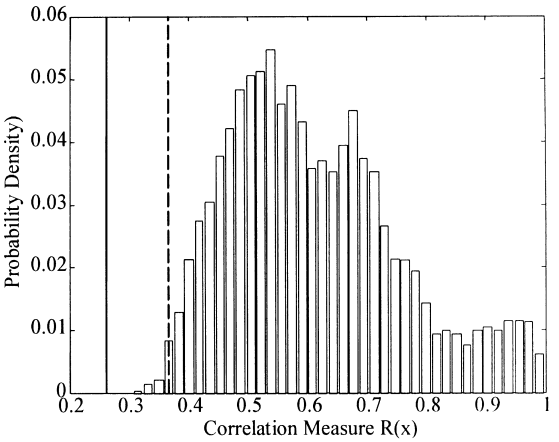


Fig. 13. Histogram of correlation measure $R(\boldsymbol{x})$ together with the minimum value obtained for the genetic algorithm (solid line) and the logarithmic method (dotted line).

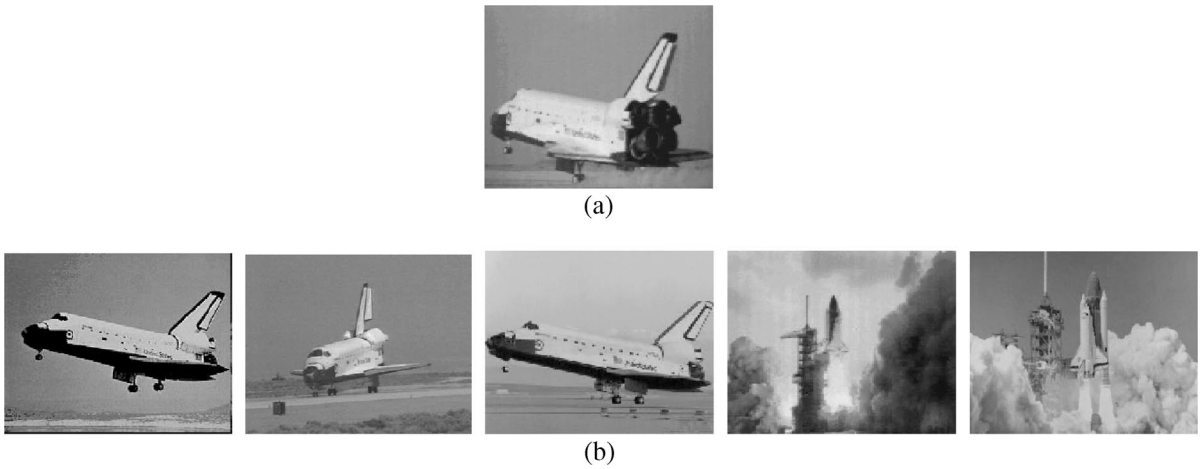


Fig. 14. An image query; (a) the submitted image; (b) the best five retrieved images.

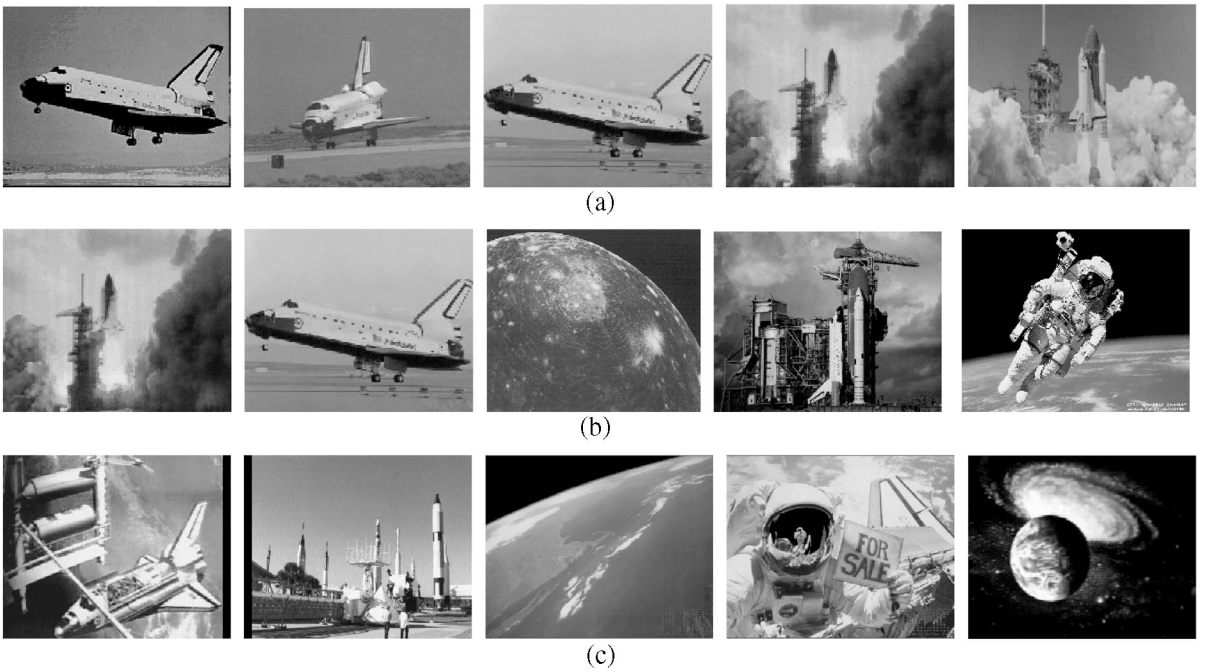


Fig. 15. Comparison of the proposed method with other techniques for the best five retrieved images of the query of Fig. 14: (a) the proposed fuzzy classification; (b) binary classification; (c) color histogram.

errors E_A and E_B over all the 15 images queries of the experiment carried out in Section 5.1. The results for binary classification are illustrated in Fig. 10, where it can be seen that the average performance error is higher in all cases compared to the fuzzy approach. For the color histogram method we have measured an average error $E_A = 0.52$ and $E_B = 0.24$. As can be seen by comparing these values with that presented in Fig. 10, the color histogram performance is worse for any partition number and membership function since only the global image characteristics are taken into consideration. The computational cost for binary classification is very small and the total cost is mainly affected by the segmentation load, resulting in a similar cost to the fuzzy approach (Tables 1 and 4). Instead, the color histogram method demands smaller computational load compared to segmentation as Table 1 shows. However, the additional cost of the proposed method is justified by its better performance.

7. Conclusions

A new approach for efficient visual content representation has been presented in this paper. In particular, in the proposed framework, the traditional pixel-based representation of visual content is transformed to a fuzzy feature-based one, which is more suitable for the new emerging multimedia applications, such as video browsing, content-based image indexing and retrieval and video summarization. First, an analysis of video sequences is performed by applying a color/motion segmentation algorithm. Then, the extracted segment properties are classified using fuzzy formulation. This representation is applied for both video summarization and content-based retrieval. In the first application, a genetic algorithm has been proposed to efficiently extract a limited but meaningful set of key-frames. Experimental results indicate that this approach outperforms the other methods for both accuracy and computational efficiency. In the

second case, the problem of content-based retrieval is investigated. The results indicate that the fuzzy representation provides a more compact description of the visual content resulting in better retrieving performance compared to traditional techniques. Furthermore, the proposed method provides a more natural interpretation of the visual content giving to the users new capabilities of performing their queries. For example, we can seek for large objects or for rapid moving segments in contrast to traditional methods where only the global color or motion information is taken into consideration.

Another issue for further investigation is the development of the fuzzy adaptive mechanism for estimating the distance weights. This approach increases the system accuracy and simultaneously leads to solutions closer to the user's needs. This can be accomplished by interpreting as fuzzy densities the subjective quality of an image given by a human observer and then constructing a fuzzy measure for weight estimation.

Acknowledgements

The authors would like to thank Georgios Akrivas, for providing them with an efficient implementation of the key-frame selection technique presented in [23].

References

[1] A. Alatan, L. Onural, M. Wollborn, R. Mech, E.Tuncel, T. Sikora, Image sequence analysis for emerging interactive multimedia services-the European cost 211 framework, IEEE Trans. Circuits Systems Video Technol. 8 (7) (November 1998) 802–813.

[2] Y. Avrithis, A. Doulamis, N.D. Doulamis, S. Kollias, An adaptive approach to video indexing and retrieval using fuzzy classification, Proceedings of Work. Very Low Bit Rate Video Coding (VLBV), Urbana-Champaign, IL, October 1998.

[3] Y. Avrithis, A. Doulamis, N. Doulamis, S. Kollias, A stochastic framework for optimal key frame extraction from MPEG video databases, Comput. Vision Image Understanding, 75 (1) (July 1999) 3–24.

[4] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, D. Zhong, A fully automated content-based video search engine supporting spatiotemporal queries, IEEE Trans. Circuits Systems Video Technol. 8 (5) (September 1998) 602–615.

[5] A. Del Bimbo, P. Pala, Visual image retrieval by elastic matching of user sketches, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 19 (2) (February 1997) 121–132.

[6] N. Doulamis, A. Doulamis, D. Kalogeras, S. Kollias, Very low bit-rate coding of image sequences using adaptive regions of interest, IEEE Trans. Circuits Systems Video Technol. 8 (8) (December 1998) 928–934.

[7] A. Doulamis, N. Doulamis, S. Kollias, On line retrainable neural networks: improving the performance of neural networks in image analysis problems, IEEE Trans. Neural Networks 11 (1) (January 2000) 137–155.

[8] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, Query by image and video content: the QBIC system, IEEE Comput. Mag. (September 1995) 23–32.

[9] L. Garrido, F. Marques, M. Pardas, P. Salembier, V. Vilaplana, A hierarchical technique for image sequence analysis, in Proceedings of Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Louvain-la-Neuve, Belgium, June 1997, pp. 13–20.

[10] V.N. Gudivada, J.V. Raghavan (Eds.), Special Issue on Content-Based Image Retrieval Systems, IEEE Comput. Mag. 28 (9) (1995).

[11] A. Hamrapur, A. Gupta, B. Horowitz, C.F. Shu, C. Fuller, J. Bach, M. Gorkani, R. Jain, Virage video engine, SPIE Proceedings of Storage and Retrieval for Video and Image Databases V, San Jose, CA, February 1997, pp. 188–197.

[12] S. Haykin, Adaptive Filter Theory, Prentice-Hall, Englewood Cliffs, NJ, 1996.

[13] M. Irani, P. Anandan, Video indexing based on mosaic representation, Proc. IEEE 86 (5) (May 1998) 805–921.

[14] B. Kosko, Neural networks and fuzzy systems, A Dynamical Systems Approach to Machine Intelligence, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[15] H.-C. Lin, L.-L. Wang, S.-N. Yang, Color image retrieval based on hidden markov models, IEEE Trans. Image Proces. 6 (2) (February 1997) 332–339.

[16] Z. Michalewicz, Genetic Algorithms + Data Structure = Evolution Programs, Springer, Berlin, 1992.

[17] MPEG-7: Context and Objectives (v.5). ISO/IEC /JTC1/SC29/WG11 N1920 MPEG-7, October 1997.

[18] P.J. Mulroy, Video content extraction: review of current automatic segmentation algorithms, Proceedings of Workshop on Image Analysis and Multimedia Interactive Systems (WIAMIS), Louvain-la-Neuve, Belgium, June 1997.

[19] K.N. Ngan, S. Panchanathan, T. Sikora, M.-T. Sun (Guest Eds.), Special Issue on Segmentation, Description and Retrieval of Video Content, IEEE Trans. Circuits Systems Video Technnol. 8 (5) (September 1998).

[20] P. Salembier, M. Pardas, Hierarchical morphological segmentation for image sequence coding, IEEE Trans. Image Process. 3 (5) (September 1994) 639–651.

[21] J.R. Smith, S.F. Chang, VisualSEEk: a fully automated content-based image query system, Proceedings of ACM Multimedia Conferance, Boston, MA, November 1996, pp. 87–98.

[22] X. Wan, C.-C. J. Kuo, A new approach to image retrieval with hierarchical color clustering, IEEE Trans. Circuits Systems. Video Technol. 8 (5) (September 1998) 628–643.

[23] B.L. Yeo, B. Liu, Rapid scene analysis on compressed videos, IEEE Trans. Circuits Systems Video Technol. 5 (December 1995) 533–544.

[24] M.M. Yeung, B.-L. Yeo, Video visualization for compact presentation and fast browsing of pictorial content, IEEE Trans. Circuits Systems Video Technol. 7 (5) (October 1997) 771–785.