First publ. in: Computers in Human Behavior ; 27 (2011), 5. - S. 1571-1578

Shot retrieval based on fuzzy evolutionary aiNet and hybrid features

Xian-Hui Li [a,b,*], Yong-Zhao Zhan [b], Jia Ke [b], Hong-Wei Zheng [c]

[a] China Realtime Database Co. LTD., Sgepri, Nanjing 210003, China
[b] School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, 212013 Jiangsu, China
[c] Department of Computer and Information Science, University of Konstanz, Germany

ARTICLE INFO

Keywords:
Shot retrieval
Fuzzy evolutionary aiNet
Hybrid features
Probabilistic distance
Similarity measure
Key-frame extraction

ABSTRACT

As the multimedia data increasing exponentially, how to get the video data we need efficiently become so important and urgent. In this paper, a novel method for shot retrieval is proposed, which is based on fuzzy evolutionary aiNet and hybrid features. To begin with, the fuzzy evolutionary aiNet algorithm proposed in this paper is utilized to extract key-frames in a video sequence. Meanwhile, to represent a key-frame, hybrid features of color feature, texture feature and spatial structure feature are extracted. Then, the features of key-frames in the same shot are taken as an ensemble and mapped to high dimension space by non-linear mapping, and the result obeys Gaussian distribution. Finally, shot similarity is measured by the probabilistic distance between distributions of the key-frame feature ensembles for two shots, and similar shots are retrieved effectively by using this method. Experimental results show the validity of this proposed method.

1. Introduction

With the rapid development of multimedia technology, multimedia data are increasing exponentially. Consequently, how to get the video data we need efficiently from abundant video databases becomes very important and urgent. In order to cope with this problem, Content-based Video Retrieval (CBVR) has become a research hotspot (Gao, Li, & Feng, 2009; Shao et al., 2008; Snoek et al., 2007). In the process of Content-based Video Retrieval, the video data are divided into key-frames, shots and scenes by analyzing video structure. In the units of the shots, according to the user-submitted video examples, the similar video clips in the video database can be found and displayed in accordance with their similarity.

Since a shot is captured under the same scene, there is no doubt that the frames in the same shot are highly correlative and have a lot of repetitive information. Meanwhile, one shot might contain hundreds or thousands of frames, which makes it a very time-consuming task to deal with the shot. Therefore, it is necessary to extract the key-frames of a shot in order to improve the retrieval efficiency. One of the most commonly used key-frame extraction methods is the unsupervised clustering algorithm. For instance, Song and Fan (2005) have put forward Sequential Forward Floating Selection method to extract the key-frames. In addition, an unsupervised clustering method based on HSV color features was introduced by Zhuang, Rui, Huang, and Mehrotra (1998), and the frame closest to the cluster center is chosen as the key-frame to represent for a given video shot. Although the above methods are effective to some extent, they need to get the type of cluster categories and the number of clusters with prior experience before cluster analysis, which is very difficult when no knowledge of the video content has been learned previously.

In the area of similarity measure for shots, Kim and Park (2002) extracted the key-frames with Cumulative Directed Divergence method, and utilized the Modified Hansdorff Distance to carry out shot retrieval. In addition, a method of dynamic programming has been proposed to measure the similarity of two shots (Chen & Chua, 2001). In order to extract key-frames, Nearest Feature Line (NFL) has been proposed by Zhao and Wei (2000). Although these methods are effective to a certain extent, the internal correlation among frames in the same shot has been overlooked. Moreover, they have ignored that a shot is a whole of frames.

In view of the shortcomings of the above methods, a new key-frame extraction method based on fuzzy evolutionary aiNet (artificial immune Network) is proposed in this paper. Artificial immune Network can implement unsupervised data clustering effectively, and it doesn't need to determine the number of clusters in advance. Moreover, it can consider the internal correlations among data more reasonably with the evolutionary mechanisms of immunity and cloning, and accordingly it can solve the problem of data clustering effectively. Therefore, based on the characteristics mentioned above, our fuzzy evolutionary aiNet based key-frame extraction method can overcome the disadvantages of the methods mentioned above effectively. The experimental results have also showed the feasibility of this method. In order to improve the

* Corresponding author. Address: School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, 212013 Jiangsu, China. Tel.: +86 013951687315.
E-mail address: lxh0417@163.com (X.-H. Li).

recall and precision of shot retrieval, hybrid features which include color feature, texture feature and spatial structure feature are extracted. Then the same types of extracted key-frames' features in the same shot are taken as an ensemble, and mapped to the high dimension space by kernel functions and non-linear mapping. In this high dimension space, we suppose that the ensemble obeys the Gaussian distribution. Based on this, we can measure the similarity of two shots by calculating the probabilistic distance of the corresponding Gaussian distributions. Finally, we can find the validity of the method from the experimental results.

2. Shot retrieval

2.1. Key-frame extraction based on fuzzy evolutionary aiNet

The artificial immune technique is enlightened by the immunology. It simulates the functions, elements and models of biology immune system to solve the complex problem with exceptional phenomenon. As we know, the matching degree between antibody and antigen is fuzzy and the boundary definition of key-frame is also fuzzy. There, based on evolutionary aiNet (de Castro & Von Zuben, 2000; Li, Gao, & Jiao, 2004), an algorithm of fuzzy evolutionary aiNet for key-frame extraction is proposed, aiming at effectively extracting the key-frames which can express the nature of a shot. This algorithm can overcome the disadvantages of traditional key-frame extraction methods, such as the lower probability of global convergence, sensitivity to the initial value, proneness to premature, essential usage of a prior knowledge to determine the number of cluster categories, and so on (Jiao & Du, 2003).

Definition: The fuzzy evolutionary aiNet can be defined as an edge-weighted graph, which is not necessarily fully connected, composed of a set of nodes called cells, and sets of node pairs called edges. Each edge has an assigned number, called weight or connection strength.

To compute the affinity of two network cells, we should first compute the distance of corresponding feature vectors. Euclidean distance is a very common method for distance measure, it can be described as:

$$d^2(x_i, y_j) = (x_i - y_j)^T (x_i - y_j),$$

where x_i and y_j are the feature vectors of the frames which will be extracted and described in Section 2.2.

Then the affinity of two network cells can be defined as:

$$D_{ij} = \frac{1}{1 + d^2(x_i, y_j)} \tag{1}$$

The affinity of the network cell with the given antigen (frame in the same shot) f_{ij} can be improved by the following expression:

$$K = K - \alpha(K - S), \tag{2}$$

where K is the feature vector of network cell (key-frame), S is the feature vector of antigen, α is the mutation rate, whose value is set according to the antigen–antibody affinity, the higher the affinity, the smaller the α.

In this paper, the main steps of the algorithm include choosing the initial antibody, cloning, inhibiting and stimulating. It is supposed that a shot contains n frames, and substitutes the frames with the corresponding feature vectors extracted in Section 2.2. The specific steps of the algorithm are described as the following:

Step 1: Initialization. Choose $n/25$ frames randomly to construct the initial key-frames (antibody) of aiNet. Initialize the network compression threshold σ_s and cloning mortality σ_d;

Step 2: Constructing antigen–antibody affinity matrix D. Take every frame f_j in shot S as an antigen, and the node of aiNet as an antibody which is the key-frame we want to extract, and then we use Eq. (1) to compute the affinity. The sub-steps are given in the following:

Step 2.1: Cloning operation. In the aiNet, sort the antibody according to degree of affinity with f_j. Clone h key-frames that are one of the key-frames with h highest affinity, according to the principle of that the higher affinity, the larger scale will be cloned. The total number of the key-frames (antibodies) cloned is N_c.

Step 2.2: Apply Eq. (2) to these N_c cells.

Step 2.3: Determine D for these improved cells.

Step 2.4: Take out the a% antibodies with the highest affinity as the network memory cells, and store them in M_P.

Step 2.5: Inhibit operation. In M_P, kill the cells whose antigen–antibody affinity are less than σ_d.

Step 2.6: Construct the new nodes of aiNet and add the remaining nodes in M_P into aiNet.

Step 3: Construct the antibody–antibody affinity matrix G. In matrix G, assume the number of cells in the same row whose values are less than σ_s is k, if k is less than σ_f, the antibody to which the row corresponds will be deleted from aiNet.

Step 4: Randomly select r% of frames (antigens) from shot S to replace the antibodies that have poor-affinity in aiNet.

Step 5: Determination of the termination for iteration. If the time of iteration is less than the given number denoted as *Num*, and the number of antibodies is less than $n/10$, then go to step 2. Otherwise, the iteration is terminated.

In the steps above, σ_s is the network compression threshold, which can compress the immune network, and can be determined by repeating the experiment. D is the antigen–antibody affinity matrix with elements $D_{i,j}(Ag - Ab)$. G is the antibody–antibody affinity matrix with elements $G_{i,k}(Ab - Ab)$. For one key-frame, assume that the number of key-frames in a shot whose affinities to this key-frame are less than σ_s is denoted as k, and if k is less than σ_f, delete this key-frame from aiNet, which will improve the tolerance of the network. When the time of iteration is larger than the number of *Num*, or the number of key-frames is larger than $n/10$, the iteration will be terminated.

2.2. Shot features extraction

2.2.1. Color histogram extraction

Each frame of a shot is firstly pre-processed in order to reduce the working load of computation. According to the Attention Model Theory (Ma, Lu, Zhang, & Li, 2002), people are more concerned about objects at the center. A key-frame in one shot in this paper is therefore divided into 4×4 areas, in which four of the central areas are endowed with the highest weight of 1/8, four corner areas are endowed with the least weight of 0, and the rest eight areas are endowed with the weight of 1/16, respectively. They are shown in Table 1.

In this paper, the HSV (Hue-Saturation-Value) color space is adopted. According to human's different perceptions to hue, saturation and value, we quantize them into $9 \times 3 \times 3$(9H × 3S × 3 V) levels. Therefore, the HSV color space is divided to 81 sub-color-spaces, and the color information of a frame can be expressed by an 81-dimensional feature vector H_C, which can be calculated by

Table 1
The weight of each sub-area.

0	1/16	1/16	0
1/16	1/8	1/8	1/16
1/16	1/8	1/8	1/16
0	1/16	1/16	0

counting each sub-color-space in the same frame associated with the weight mentioned above. At the same time, we normalize H_C, and the color histogram of each frame is finally obtained, which could be described as below:

$$H_C = [c_1, c_2, \ldots, c_{81}] \tag{3}$$

2.2.2. Conformation of spatial structure histogram

The entire or partial spatial structure information of an image is a comparatively key feature to the image itself. In this paper, the idea introduced by Lin, Zhang, Feng, and Shi (2002) is adopted, and the spatial structure histogram is formed to represent the spatial information of an image. Spatial structure histogram can very well complement the color histogram, for color histogram lacks the spatial distribution of color information.

First of all, the *K*-means algorithm is used to quantize color in the HSV space to obtain the color block graphs of each key-frame. These color block graphs can be described as $R_j (j = 0, \ldots, N_b - 1)$, where N_b is the number of the color block graphs. Then, we calculate the area histogram H_{area} of the color block graphs, the location histogram H_{pos} and the image region variance histograms in both *X* and *Y* directions (H_{vx} and H_{vy}), as well as the region length histograms in the *X* and *Y* directions (H_{sx} and H_{sy}), respectively. They are defined as follows:

$$H_{area}(i) = \sum_{R_j \in \Omega_i} \text{Area}(R_j), \; i = 0, 1, \ldots, 7,$$

where $\Omega_i = \{R_j | \text{Area}(R_j) \in [A_k, A_{k+1}), j = 0, 1, \ldots, N_b - 1\}, A_0 = 0, A_k = 1/2^{8-k}, (k = 0, 1, \ldots, 7)$, Area($R_j$) represents the area percentage of the *j*th color block graph. When an image is eventually divided into 16 small pieces, we can get $H_{pos}(i)$ as follow:

$$H_{pos}(i) = \sum_{R_j \in \Omega_i} \text{Area}(R_j), \; i = 0, 1, \ldots, 15,$$

where $\Omega_i = \{R_j | \text{ Center } (R_j) \in \text{Black } (i), j = 0, 1, \ldots, N_b - 1\}$, Black ($i$) is the *i*th piece and Center (R_j) represents the center of the *j*th color block graph. To get the value of H_{vx}, we should firstly compute $\sigma_x(R_j)$, which is the standard deviation in the *x* direction. The value of H_{vx} can be described as follow:

$$H_{vx}(i) = \sum_{R_j \in \Omega_i} \text{Area}(R_j), \; i = 0, 1, \ldots, 7,$$

where $\Omega_i = \{R_j | \sigma_x(R_j) \in (B_k, B_{k+1}], j = 0, 1, \ldots, N_b - 1\}, B_0 = 0, B_k = 1/2^{8-k}, (k = 0, 1, \ldots, 7)$. Finally, the region length histogram in the *X* direction can be calculated as:

$$H_{sx} = \sum_{R_j \in \Omega_i} \text{Area}(R_j), \; i = 0, 1, 2 \ldots, 7,$$

where $\Omega_i = \{R_j | \text{Width}(R_j) \in (B_k, B_{k+1}], j = 0, 1, \ldots, N_b - 1\}, B_0 = 0, B_k = 1/2^{8-k}, (k = 0, 1, \ldots, 7)$, Width($R_j$) is the width of the minimum bounding rectangle which contains R_j dividing the width of the image. H_{vy} and H_{sy} share the similar calculation with H_{vx} and H_{sx}.

As the amount of blocks denoted as *k* in one frame is variable. In order to obtain the spatial information of a frame simply and effectively, spatial information histogram is exploited. In this way, we quantize every spatial parameter extracted above into eight levels. For every parameter, the histograms of eight levels in a frame can be accounted, which will form 48 spatial information features. Finally, the spatial information of one frame can be described as a 48-dimensional spatial structure feature vector as follow:

$$H_S = [s_1, s_2, \ldots, s_{48}] \tag{4}$$

2.2.3. Texture feature extraction

We make use of a co-occurrence matrix to extract texture features. A co-occurrence matrix (Wikipedia) is defined over an image to be the distribution of co-occurring grayscale values at a given offset. Mathematically, a co-occurrence matrix *P* is defined over an $N \times M$ image *I*, parameterized by an offset $\delta = \sqrt{\Delta x^2 + \Delta y^2}$, as:

$$P_{\Delta x, \Delta y}(i,j) = \sum_{p=1}^{N} \sum_{q=1}^{M} \begin{cases} 1, I(p,q) = i, I(p + \Delta x, q + \Delta y) = j \\ 0, \text{otherwise} \end{cases}$$

It makes the statistics of the probability of the image with intensity *j* which has a distance δ from the image with intensity *i*.

Based on a co-occurrence matrix, we can calculate the texture feature parameters, including the angular second moment, contrast, correlation, variance, variance and the sum of average value, entropy, difference entropy, difference variance and trade deficit variance, etc. Although more than ten characteristics parameters are able to express some specific information of the texture, there are still some problems like miscellaneous information, duplicated statements. Therefore, the action of screening and classifying should be taken to get the most representative and independent characteristics parameters. Here, four representative parameters are chosen, they are entropy, contrast, energy and relativity,

Entropy:

$$F = \sum_{i=1}^{N} \sum_{j=1}^{M} p(i,j) \log p(i,j)$$

Contrast:

$$I = H = \sum_{i=1}^{N} \sum_{j=1}^{M} (i-j)^2 \log p(i,j)$$

Energy:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{M} [p(i,j)]^2$$

Relativity:

$$R = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{ijp(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

Here $p(i, j)$ is an element of the co-occurrence matrix, *N* and *M* are the number of pixels in columns and rows.

Merging the parameters above, then the texture feature vector of a key-frame can be described as:

$$I_T = [F, I, E, R]. \tag{5}$$

2.3. Shot similarity measure based on hybrid features

2.3.1. Probabilistic distance

In probabilistic statistics, the difference between two distributions can be expressed by the probabilistic distance (Zhou & Chellappa, 2006). The probabilistic distance is an effective method for measuring the similarity of two samples with uncertain feature values. When the two samples are in compliance with the Gaussian distribution model, calculating their probabilistic distance will become easier. In order to simplify calculation, we should first make features obey Gaussian distribution. The kernel-based method can map the feature space to the high-dimensional space without having to know the specific form of mapping. That's to say with the help of the kernel technology (Bach & Jordan, 2003), the feature vectors of frames can be mapped to a high-dimensional space obeying Gaussian distribution. So, it is possible to measure the similarity of two shots easier by using the probabilistic distance.

First of all, we adopt the Mercer Kernel method to map the key-frame feature vectors ensembles of the original sampled space to the Reproducing Kernel Hilbert Space (RKHS) with a non-linear

mapping technique. It is assumed that the data obey the Gaussian distribution in the RKHS. The mapping functions are described as below:

$$\Phi : \Omega \to R^f,$$

$$K(\alpha, \beta) = \Phi(\alpha)^T \Phi(\beta),$$

where α and β are feature vectors in the original vector space, $\Phi(\beta)$ and $\Phi(\alpha)$ are their corresponding values in the RKHS.

As the mapped vectors should obey the Gaussian distribution, in this paper, we select the Radial Basis Function (RBF) as a kernel function:

$$K(x, y) = \exp(-||x - y||^2 / 2\sigma^2),$$

where x and y are feature vectors of a frame, which are one of the feature vectors H_C, H_S, I_T that are extracted in Section 2.2.

It is supposed that there are two shots, there are $S_1 = \{f_{1,1}, f_{1,2}, \ldots, f_{1,n}\}$ and $S_2 = \{f_{2,1}, f_{2,2}, \ldots, f_{2,m}\}$. After being mapped to the high-dimensional space using the non-linear mapping methods, they will be represented by $\Phi(S_1) = \{\Phi(f_{1,1}), \Phi(f_{1,2}), \ldots, \Phi(f_{1,n})\}$ and $\Phi(S_2) = \{\Phi(f_{2,1}), \Phi(f_{2,2}), \ldots, \Phi(f_{1,m})\}$, respectively. Here, the dot product matrix of two feature vectors in the high-dimensional space is defined as follow:

$$\begin{pmatrix} \Phi_1^T \\ \Phi_2^T \end{pmatrix} (\Phi_1 \quad \Phi_2) = \begin{pmatrix} \Phi_1^T\Phi_1 & \Phi_1^T\Phi_2 \\ \Phi_2^T\Phi_1 & \Phi_2^T\Phi_2 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix},$$

where $K_{ij} = \Phi_i^T \Phi_j$ and $K_{ij} = K_{ji}^T$.

The general methods of distance measurement in probabilities include the Cherrnoff distance, the Bhattacharyya distance, the Mahalanobis distance, the KL divergence, the Kolmogorv distance, the Patrick–Fisher distance, and so on (Devijver & Kittler, 1982). The followings are four kinds of distance measurement for probabilistic distributions that we will take into account for the comparison in this paper. Here, we present the detailed computational formula.

(1) Chernoff distance measurement:

$$J_c(p_1, p_2) = \frac{1}{2}\alpha_1\alpha_2(\mu_1 - \mu_2)^T[\alpha_1\Sigma_1 + \alpha_2\Sigma_2]^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\log\frac{|\alpha_1\Sigma_1 + \alpha_2\Sigma_2|}{|\Sigma_1|^{\alpha_1}|\Sigma_2|^{\alpha_2}}$$

(2) KL divergence calculating method:

$$J_D(P_1, P_2) = \frac{1}{2}(\mu_1 + \mu_2)^T\left(\Sigma_1^{-1} + \Sigma_2^{-1}\right)(\mu_1 + \mu_2) + \frac{1}{2}tr\left[\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 - 2I_d\right]$$

(3) Patrick–Fisher distance measurement:

$$J_P(P_1, P_2) = \left[(2\pi)^d|2\Sigma_1|\right]^{1/2} + \left[(2\pi)^d|2\Sigma_2|\right]^{1/2} - 2\left[(2\pi)^d|\Sigma_1 + \Sigma_2|\right]^{1/2} \times \exp\left\{-\frac{1}{2}(\mu_1 + \mu_2)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 + \mu_2)\right\}$$

(4) Mahalanobis distance measurement:

$$J_M(P_1, P_2) = (\mu_1 + \mu_2)^T\Sigma_1^{-1}(\mu_1 + \mu_2)$$

Here p_1, p_2 are two Gaussian distributions, μ_1, μ_2 and Σ_1, Σ_2 are the means and variances, $0 < \alpha_1, \alpha_2 < 1$, $\alpha_1 + \alpha_2 = 1$, and d is the dimension.

2.3.2. Shot similarity with hybrid features

In the above steps, through the key-frame extraction and the feature extraction of every key-frame, the features of a shot are expressed as a color histogram ensemble, a co-occurrence matrix vector ensemble and a space histogram vector ensemble. Meanwhile, with the help of the non-linear mapping technology, these feature vectors are non-linearly mapped to RKHS. In this high-dimensional space, it is assumed that samples obey the Gaussian distribution. Thus, the similarity of two shots can be measured using the probabilistic distribution formulae:

$$simH_C(H_C(S_1), H_C(S_2)) = F(\Phi(H_C(S_1)), \Phi(H_C(S_2))),$$

$$simH_S(H_S(S_1), H_S(S_2)) = F(\Phi(H_S(S_1)), \Phi(H_S(S_2))),$$

$$simI_T(I_T(S_1), I_T(S_2)) = F(\Phi(I_T(S_1)), \Phi(I_T(S_2))),$$

where the function F is one of the probabilistic distances among the Cherrnoff distance, the Mahalanobis distance, the KL divergence and the Patrick–Fisher distance which are introduced in Section 2.3.1. In the following experimental section, we compare the effects of the four probabilistic distance measurements for the shot retrieval. $H_C(S_i) = \{H_C(f_{i1}), H_C(f_{i2}) \ldots, H_C(f_{in})\}$ is the color feature vectors ensemble of shot S_i. $H_S(S_i) = \{H_S(f_{i1}), H_S(f_{i2}) \ldots, H_S(f_{in})\}$ is the spatial structure feature vectors ensemble. $I_T(S_i) = \{I_T(f_{i1}), I_T(f_{i2}) \ldots, I_T(f_{in})\}$ is the texture feature vectors ensemble. $\Phi(H_C(S_i)) = \{\Phi(H_C(f_{i1})), \Phi(H_C(f_{i2})) \ldots, \Phi(H_C(f_{in}))\}$, $\Phi(I_T(S_i)) = \{\Phi(I_T(f_{i1})), \Phi(I_T(f_{i2})) \ldots, \Phi(I_T(f_{in}))\}$ and $\Phi(H_S(S_i)) = \{\Phi(H_S(f_{i1})), \Phi(H_S(f_{i2})) \ldots, \Phi(H_S(f_{in}))\}$ are the corresponding feature vectors in the high dimension space.

Finally, the similarity of two shots is measured by fusing these similarities with weights. The formula is described as below:

$$sim(S_1, S_2) = \omega_1 simH_C(H_C(S_1), H_C(S_2)) + \omega_2 simH_S(H_S(S_1), H_S(S2)) + \omega_3 simI_T(I_T(S_1), I_T(S_2)),$$

where $\omega_1 + \omega_2 + \omega_3 = 1$. $\omega_1, \omega_2, \omega_3$ can be different depending on the type of videos. They can also be determined by the user feedback mechanisms to the system. For example, if color feature is more important than the other two features, then we assigns ω_1 a larger value. As experiments, we think that the color feature, the spatial structure feature and the texture feature are all most take the same weight. Therefore, we take $\omega_1 = 0.4$, $\omega_2 = 0.3$ and $\omega_3 = 0.3$.

3. Experimental results and analysis

Our experiments are based on TREC Video Retrieval Evaluation database (TRECVID 2007). In order to verify the effectiveness of our key-frame extraction method, we randomly choose some of those videos as our experimental test videos, which include 267 sports video shots, 106 move shots, 302 cartoon shots and 238 news program shots, and conducted a large number of experiments. We choose four types of videos so as to validate whether our method is insensitivity to video type. The correctness of the shot retrieval result is evaluated by integrating several persons' subjective judgments.

Figs. 2 and 3 are one result of the experiments. In that experiment, we get a LBC news program from TRECVID 2007, with 225 frames and 19 key-frames shown in Fig. 1, which are extracted subjectively. Fig. 2 shows the key-frames extracted by the method of unsupervised cluster introduced by Hanjalic and Zhang (1999). Fig. 3 shows the key-frames extracted by the method of fuzzy evolutionary aiNet proposed in this paper. As shown in Fig. 2, some of the frames are quite similar, such as the 1st frame and the 6th frame, the 17th frame and the 19th frame, the 58th frame and the 65th frame, etc. Therefore, we only need to extract one frame of each similar group. Just as Fig. 3 shows, our proposed method has less redundant information. Accurately, 6 key-frames are not

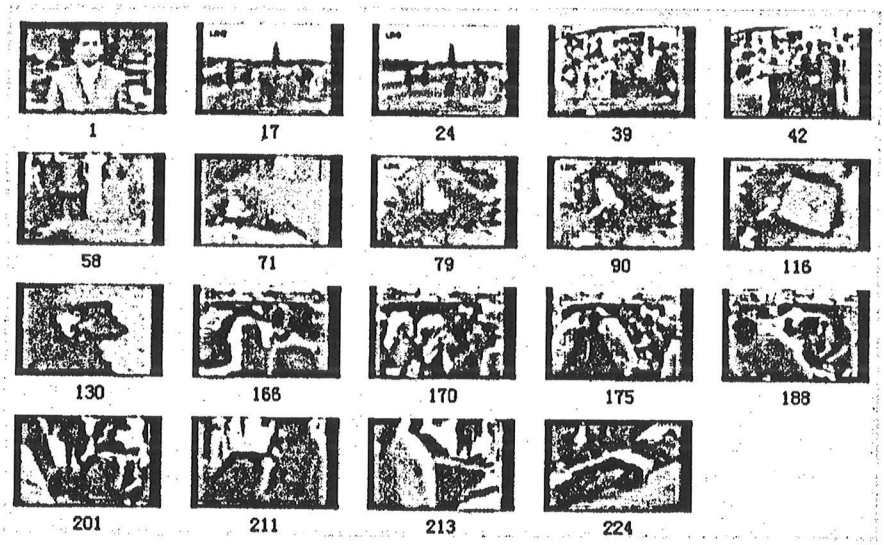

Fig. 1. Key-frames extracted subjectively.

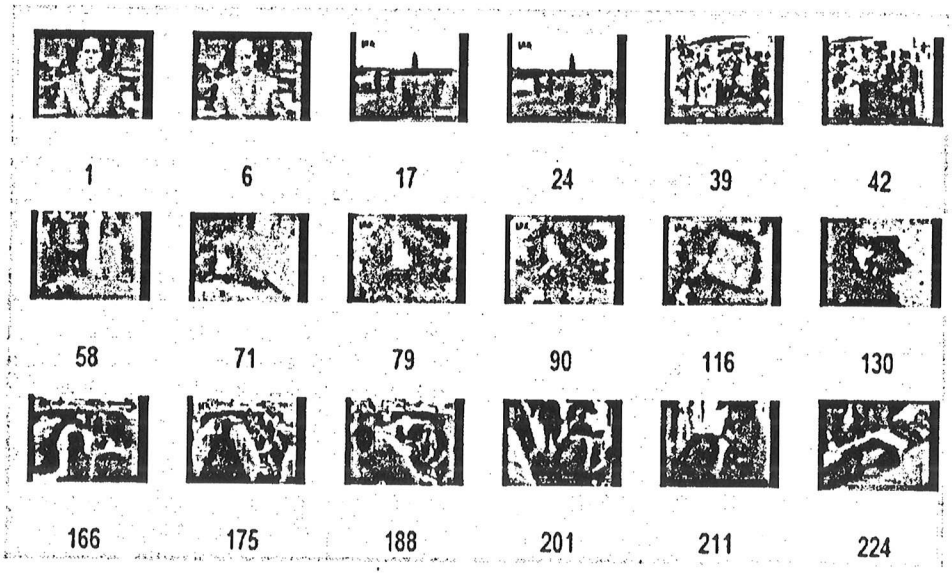

Fig. 2. Key-frame extracted by unsupervised cluster.

extracted in Fig. 2, which are 24th, 39th, 79th, 201st, 211st and 213rd , while only 2 key-frames are ignored in Fig. 3, which are 170th and 213rd. This shows that our method has better accuracy.

There are different effects if different distance measurements of probability are selected. The shot retrieval experiments have been done with different measurements which are the Cherrnoff distance, the Mahalanobis distance, the KL divergence and the Patrick–Fisher distance, respectively. The corresponding performances are shown in Fig. 4.

In the Fig. 4, we can see that the KL distance and the Mahalanobis distance are better than the Chernoff distance and the Patrick–Fisher distance apparently.

In order to explain the effectiveness of the method in a better way, we select the similar shots subjectively, and then take the KL divergence as the probabilistic distance measurement function. We also use the performance parameters of recall (Rec.) and precision (Prec.) to measure the retrieval results. The experimental results are shown in Table 2 (ST: shot type, TS: total shots of the corresponding shot type, ASS: actually similarity shots, SBR: shots are retrieved by our method, RSBR: right shots are retrieved by our method, RSBNR: right shots are not retrieved by our method, MS: Miscarriage Shots). It shows that in the condition of ensuring the recall is greater than or equal to 80%, the precision is more than 70%. Actually, most of the shots that should be extracted have been correctly selected. Besides, we also do some experiments using different features in shot retrieval process. In Table 3, we compare the performances of methods by using different features, such as the color histogram (CH), the co-occurrence matrix vector (COMV), the space histogram vector (SH) and the weighted hybrid features (HF). What we can observe from the table is that the method of hybrid features is not sensitive to the video type, and also its performance is better than the others for the most types of videos (precision: prec.)

As it is known to all, video processing is time-consuming. In the process of shot retrieval, a good method should not only retrieve accurately, but also do it fast. Consequently, retrieval speed should

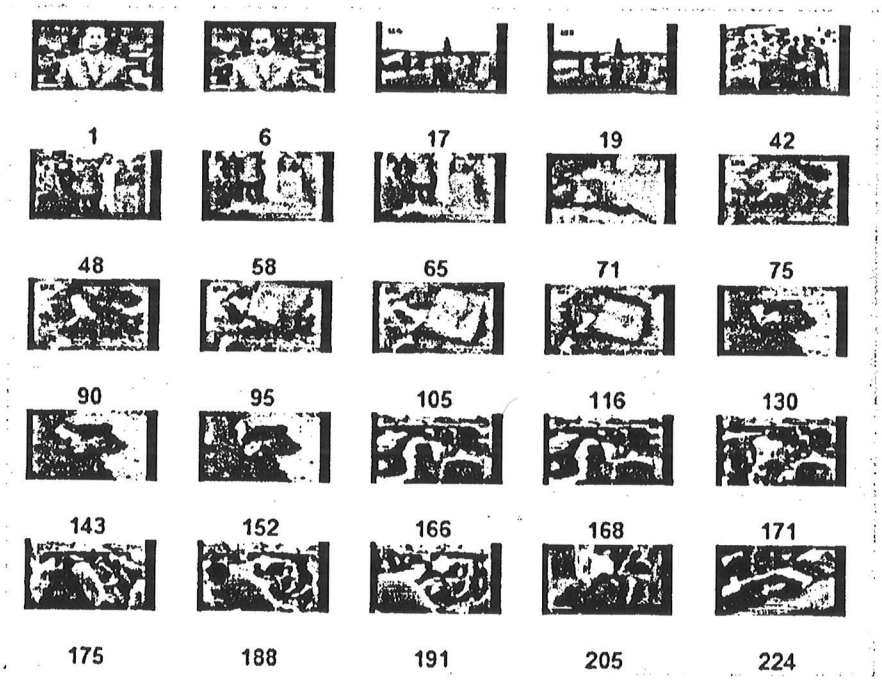

Fig. 3. Key-frames extracted by fuzzy evolutionary aiNet.

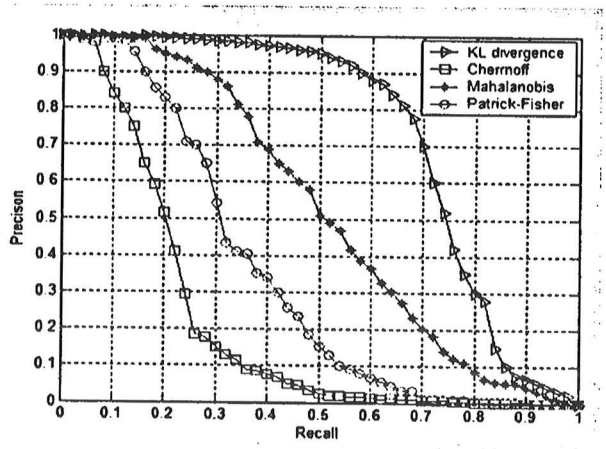

Fig. 4. Performances of four probabilistic distance measurements.

Table 2
The retrieval experimental results of our shot retrieval method.

ST	TS	ASS	SBR	RSBR	RSBNR	MS	Rec. (%)	Prec. (%)
Sports	267	23	25	19	4	6	82	76
Move	106	15	16	12	3	4	80	75
Cartoon	302	30	35	26	4	9	87	74
News	238	20	21	16	4	5	80	76

be taken into account during the measurement of retrieval efficiency. We compared the average recall (Rec.), precision (Prec.) and consumed time (Speeds(s): Spd.) among these four methods with using our key-frame extraction method (*Y* for short) and not using it (*N* for short). As shown in Table 4, the using of our key-frame extraction method might not have substantive contribution to recall and precision, but it makes the retrieval process more than twice as fast as that not using our key-frame extraction method.

4. Comparison with related methods

An easy way of key-frame extraction is to use the frames at specific locations as the shot's key-frames, regardless of the shot's visual complexity. This approach to key-frame extraction is relatively fast. However, it doesn't effectively capture the visual content of the video shot, since a frame at specific location is not necessarily a key-frame. Wlof (1996) has proposed a motion-based approach to extract key-frame. He first computes the optical flow for each frame, and then calculates a simple motion metric based on the optical flow. Finally he analyzes the metric as a function of time to select key-frames at the local minima of motion. This method has reasonably considered the motion feature, though it is computationally expensive and its underlying assumption of local minima is not necessarily correct. Hanjalic and Zhang (1999) have put forward a method for automated video abstraction based on unsupervised cluster-validity analysis. And also, Joshi et al. have applied the fuzzy clustering method to the key-frame extraction for gradual change video sequences (Joshi, AuePhanwiriyaku1, & KrishnaPurm, 1998). These approaches reasonably take the frames with great difference as the shot's key-frames, but they have the following two main problems. On the one hand, the number of clusters needs to be pre-specified, while for different shots, it is difficult to be known in advance. On the other hand, if there are more abrupt shots in a long video sequence, there may be more error clusters. Calic and Izquierdo (2002) have introduced a real-time algorithm for key-frame extraction that generates the frame difference metrics by analyzing statistics of the macro-block features extracted from the MPEG compressed stream. Wang et al. has also proposed a key-frame extraction method based on rough set (Wang, Wu, & Chen, 2007), which extracts motion information from compressed MPEG streams. Both the two approaches have less computational time, but they depend on the corresponding video compression standards.

For key-frame extraction, we propose a method based on the fuzzy evolutionary aiNet. This method can effectively overcome the disadvantages of traditional key-frame extraction methods, such as lower global convergence probability, sensitivity to the initial

Table 3
Recall and precision rates among four methods with four types of video.

	CH		COMV		SH		HF	
	Recall (%)	Prec. (%)	Recall (%)	Prec. (%)	Recall (%)	Prec. (%)	Recall (%)	Prec. (%)
Sports	85	71	87	78	65	76	82	76
Move	53	41	68	66	70	78	80	75
Cartoon	68	52	56	53	46	49	87	74
News	49	60	71	80	49	57	80	76

Table 4
Recall, precision and speed among four methods with using key-frame extraction or not.

	CH			COMV			SH			HF		
	Rec. (%)	Prec. (%)	Spd.	Rec. (%)	Prec. (%)	Spd.	Rec. (%)	Prec. (%)	Spd.	Rec. (%)	Prec. (%)	Spd.
Y	64	56	105	70	69	120	57	65	117	82	75	142
N	59	63	251	71	65	302	54	62	297	79	73	396

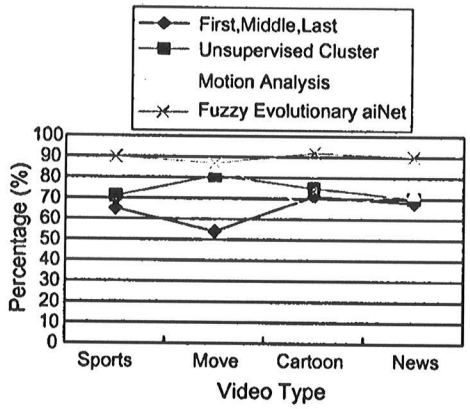

Fig. 5. The results of four different key-frame extraction algorithms compared with subjectively key-frames regarding four types of videos.

values, proneness to premature, essential usage of prior knowledge to determine the number of cluster categories, and so on.

For the measure of shot similarity, a method based on probabilistic distance is proposed. This method much reasonably considers the internal correlation among the various features of the shot.

Fig. 5 is the results of four different key-frame extraction algorithms compared with subjectively key-frames regarding four types of videos. The compared algorithms are: (1). take the first, middle and last frames in the shot as key-frames (FML). (2). Unsupervised cluster key-frame extraction (UC). (3). Motion analysis key-frame extraction. (4). Our fuzzy evolutionary aiNet key-frame extraction method. We can find that our key-frame method is better than the other three.

5. Conclusion

In this paper, a new shot retrieval method is presented, which is based on fuzzy evolutionary aiNet and hybrid features. With the introduction of artificial immune network into key-frame extraction, the key-frames of representative shot are effectively extracted. Simultaneously, with the application of non-linear mapping, the features of key-frames are mapped from the input space to the high dimension RKHS. The probabilistic distance between distributions of the key-frame feature ensembles for two shots is taken as shot similarity measurement. Finally, the shot retrieval is implemented by weighting multi-feature similarities. The experimental results show that our method is effective. In the further work, we may take motion information and high-level semantic information of the shot into account for shot feature extraction in order to further improve the proposed method.

Acknowledgments

This research is partly supported by National Natural Science Foundation of China under Grant No. 60673190, Natural Science Foundation of Jiangsu Province under Grant No. BK2009199, and College Graduate Research and Innovation Plan of Jiangsu Province under Grant No. 1221170010. We would also like to express our thanks to the group of TRECVID, they offered us the video database for testing. The second author appreciates all colleagues of Professor Dr. Dietmar Saupe's group for creating a helpful and friendly working environment during his visit.

References

Bach, F., & Jordan, M. I. (2003). Learning graphical models with Mercer Kernels. In *Advance in neural information proceedings systems*. Cambridge, MA: MIT Press.

Calic, J., & Izquierdo, E. (2002). Efficient key-frame extraction and video analysis. In *Proceedings of the international conference on information technology* (pp. 28–33).

Chen, L., & Chua, T. S. (2001). A match and tiling approach to content-based image retrieval. In *Proceedings of IEEE international conference on multimedia and expo* (pp. 301–304).

de Castro, L. N., & von Zuben, F. J. (2000). An evolutionary immune network for data clustering. In *Proceedings of the IEEE SBRN 2000 Proc.* (Vol. 22(25), pp. 84–89).

Devijver, P., & Kittler, J. (1982). *Pattern recognition: A statistical approach*. Prentic Hall.

Gao, X. B., Li, X. L., & Feng, J. (2009). Shot-based video retrieval with optical flow tensor and HMMs source. *Pattern Recognition Letters, 30*(2), 140–147.

Hanjalic, A., & Zhang, H. J. (1999). An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Transactions on Circuits and Systems for Video Technology, 9*(8), 1280–1289.

Jiao, L. C., & Du, H. F. (2003). Development and prospect of the artificial immune system. *Acta Electronica Sinica, 31*(10), 1540–1548.

Joshi, A., AuePhanwiriyaku1, S., & KrishnaPurm, R. (1998). On fuzzy clustering and content based access to networked video databases. In *Proceedings of the IEEE workshop on research issues in database engineering* (pp. 42–49). Washington, DC: IEEE Computer Society.

Kim, S. H., & Park, R. H. (2002). An efficient algorithm for video sequence matching using the modified Hausdorff distance and the directed. *IEEE Transactions on Circuits and Systems for Video Technology, 12*(7), 592–596.

Li, J., Gao, X. B., & Jiao, L. C. (2004). A novel clustering method with network structure based on clonal algorithm. *Acta Electronica Sinica, 32*(7), 1195–1199.

Lin, T., Zhang, H. J., Feng, J. F., & Shi, Q. Y. (2002). Shot content analysis for video retrieval applications. *Journal of Software, 13*(08), 1577–1585.

Ma, Y. F., Lu, L., Zhang, H. J., & Li, M. J. (2002). A user attention model for video summarization. In *Proceedings of the 10th ACM international conference on multimedia* (pp. 533–542).

Shao, J., Huang, Z., Shen, H. T., Zhou, X. F., Lim, E. P., & Li, Y. J. (2008). Batch nearest neighbor search for video retrieval. *IEEE Transactions on Multimedia, 10*(3), 409–420.

Snoek, C. G. M., Huurnink, B., Hollink, L., de Rijke, M., Schreiber, G., & Worring, M. (2007). Adding semantics to detectors for video retrieval. *IEEE Transactions on Multimedia, 9*(5), 975–986.

Song, X. M., & Fan, G. L. (2005). Joint key-frame extraction and object-based video segmentation. In *Proceedings of IEEE workshop on motion and video computing, MOTION 2005* (Vol. 2, pp. 126–131).

Wang, T., Wu, Y., & Chen, L. (2007). An approach to video key-frame extraction based on rough set. In *Proceedings of international conference on multimedia and ubiquitous engineering* (pp. 590–596).

Wikipedia. Co-occurrence matrix, http://en.wikipedia.org/wiki/Co-occurrence_matrix.

Wlof, W. (1996). Key frame selection by motion analysis. In *Proceedings of 1996 IEEE international conference the acoustics, speech, and signal* (pp. 1228–1231). Washington, DC: IEEE Computer Society.

Zhao, L., & Wei, Q. (2000). Key-frame extraction and shot retrieval using nearest feature line. *Chinese Journal of Computers, 23*(12), 1292–1298.

Zhou, S. K., & Chellappa, R. (2006). From sample similarity to ensemble similarity: Probabilistic distance measure in reproducing Kernel Hilbert space. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(6), 917–929.

Zhuang, Y., Rui, Y., Huang, T. S., & Mehrotra, S. (1998). Adaptive key frame extraction using unsupervised clustering. *Proceedings of IEEE international conference on image processing* (Vol. 1, pp. 866–870). Los Alamitos, CA: IEEE Computer Society.