Video Repeat Recognition and Mining by Visual Features

Xianfeng Yang and Qi Tian

Abstract. Repeat video clips such as program logos and commercials are widely used in video productions, and mining them is important for video content analysis and retrieval. In this chapter we present methods to identify known and unknown video repeats respectively. For known video repeat recognition, we focus on robust feature extraction and classifier learning problems. A clustering model of visual features (e.g. color, texture) is proposed to represent video clip and subspace discriminative analysis is adopted to improve classification accuracy, which results in good results for short video clip recognition. We also propose a novel method to explore statistics of video database to estimate nearest neighbor classification error rate and learn the optimal classification threshold. For unknown video repeat mining, we address robust detection, searching efficiency and learning issues. Two detectors in a cascade structure are employed to efficiently detect unknown video repeats of arbitrary length, and this approach combines video segmentation, color fingerprinting, self-similarity analysis and Locality-Sensitive Hashing (LSH) indexing. A reinforcement learning approach is also adopted to efficiently learn optimal parameters. Experiment results show that very short video repeats and long ones can be detected with high accuracy. Video structure analysis by short video repeats mining is also presented in results.

1 Introduction

Video repeats which refer to copies of a video clip ubiquitously exist in broadcast and web videos, and their distributions embed abundant structural information both in program level and web database scale. The most common video repeats are those short video clips from a few seconds to several minutes such as TV commercials, station logo or program logo, etc. To discover and locate video repeats from large video database or video streams robustly and efficiently is very

Xianfeng Yang
Faculty of Engineering, National University of Singapore, Singapore
xianfengy@gmail.com

Qi Tian
Institute for Infocomm Research, Singapore
tianqig@gmail.com

D. Schonfeld et al. (Eds.): Video Search and Mining, SCI 287, pp. 305–326.
springerlink.com

important for video content analysis and retrieval. For example, by detecting video repeats in unlabeled video data, we can find correlation of different video parts and discover structural video elements used for syntactic segmentation purpose, hence video structure model can be effectively constructed and applied to video syntactical segmentation ([1][2]). Video repeat mining also has many other prospective applications, such as commercial monitoring ([3][4][5]), video copy detection ([6][7]), web video multiplicity estimation ([8]), video content summary, personalization as well as lossless video compression ([9]).

Video repeat mining tasks can be divided into two categories: known video repeat mining and unknown video repeat mining. For known video repeat mining, we often construct a feature vector set from prototype videos and use nearest neighbor (NN) classifier to recognize copies of prototype videos from video collections or streams. In this part we focus on the feature representation and classifier learning problems. Since video copies located in different video sources have different formats, e.g. different frame sizes, frame rates as well as bitrates, so diverse distortions pose a big challenge to video copy recognition. So far many research efforts on video identification have been dedicated to extraction of distinct and robust video features from color or geometry field ([10][11][12]), called video hashing, with the aim to map video object to a unique hash code that could also be robust to kinds of distortions. However, finding a general robust yet distinct video hash code is very difficult, so the question is: if video features do not show good identification performance under certain video distortions, can they be transformed to a better one? In this chapter we examine commonly used visual features (e.g. color histogram, texture) and improve their video recognition performance under significant distortions through subspace discriminative analysis. Subspace discriminative analysis is extensively used in face recognition and text classification ([13][14][15]), and we will show it also results in very promising results in video copy recognition ([16]).

To obtain the minimum error classifier, we propose a novel method to explore statistics of prototype video database in order to estimate error rate of threshold NN classifier and learn the optimal threshold. Three types of 'sample-to-database' distances are defined, and error rate is exactly estimated from the three distance distributions. Compared to 'sample-to-sample' distance ([17]), 'sample-to-database' distance is naturally related to the feature distribution of video database, thus making database statistics and error rate estimation more reasonable.

Unknown repeat mining task is usually implemented on video collections from the same source, e.g. broadcast videos in different days, to analyze video structure, and the challenge is that prior knowledge about video repeats such as their content, length and location, is not known in advance, moreover video repeats in different locations may also have distortions, e.g. caption overlay, partial repeats. In unknown repeat mining section we will address robust detection, searching efficiency and learning issues. The approach we proposed combines video segmentation, color fingerprinting, self-similarity analysis, cascaded detection, LSH indexing and reinforcement learning. Compared to other media repeat pattern identification methods ([3][9][18]), our approach can detect very short repeats (e.g. those less than 1 second) along with long ones, and high accuracy has been

achieved in our experiments. Methods by Cheung et al. ([3]) and Herley ([18]) both use a fixed time window to do feature extraction and comparison, so those repeats significantly shorter than the window are very likely be missed. The method by Pua et al. ([9]) is able to identify repeated shots but can not identify partially repeated shots, while our approach can identify even small portion of a shot or clip by adopting segmentation with granularity smaller than the shot. Another novelty of our approach is that a reinforcement learning approach is adopted to train the video repeat detectors, and this approach demonstrates efficiency in parameter learning, which makes the repeat mining system manageable and easy to train.

The remainder of this chapter is arranged as follows: In section 2, we present the known video repeat recognition approach and results. In section 3, the unknown video repeat mining method and results are presented. In section 4, the concluding remarks are discussed.

2 Known Video Repeat Recognition

In this section we first propose a model clustering visual features to represent a video clip, and adopt Oriented PCA (OPCA) approach to transform this video feature to subspace representation in order to improve video model separability while suppressing distortions. We also propose a novel method to explore statistics of video database to estimate error rate of threshold NN classifier and learn the optimal classification threshold. Recognition performance is evaluated under significant video distortions and different video length. Results show that recognition error rate below 5% has been achieved under significant distortions, and subspace representation lead to a large reduction of error rate compared to using original feature, especially for very short video clips (e.g.5s).

2.1 Video Feature Extraction

2.1.1 Color and Texture Feature Model

Since a video clip consists of a group of images, to reduce video data and remove redundancy, the frames are sampled every half second. For each sample frame RGB color histogram and texture feature are calculated. R, G, B channels are each divided into 8 bins, thus color histogram is a 512 dimensional feature vector. Texture feature extraction adopts the statistical texture analysis method based on concurrence gray matrix ([19]). In this method four gray level concurrence matrices are first computed, which corresponds to four neighborhood directions, namely horizontal, vertical, left-down diagonal 45° and right-down diagonal 45°. Totally 13 texture components are computed from each gray level concurrence matrix, including Angular Second Moment, Contrast, Variance, Relevance Coefficient, Entropy etc. Complete computation of the 13 texture components please refer to ([19]).

Texture components computed from the four gray level concurrence matrices are averaged to form one mean texture feature vector. Since texture components have different physical meanings and value ranges, each component is normalized by Gaussian normalization approach to make them equally contribute to feature distance computation. Based on the normalized color and texture features extracted from sample frames, unsupervised clustering approach (e.g. K-Means clustering) is employed to get typical feature model of the video clip. Color feature vector and texture feature vector are clustered separately, and feature distance measure adopts Euclidean distance. The number of clusters for color feature and texture feature are set as the same value, so the video clip's feature model F is as follows,

$$F = [F_c^1, ..., F_c^{K_1}; F_T^1, ..., F_T^{K_2}] \tag{1}$$

Where F_c^i represents the ith color cluster center, F_T^i represents the ith texture cluster center, K_1 and K_2 are the number of clusters. Advantage of this representation is that a video clip can be represented by a fixed dimensional feature vectors, and it is robust to feature distortion of individual frames, as well as frame dropping.

2.1.2 Subspace Discriminative Analysis by OPCA

In above feature representation, dimension of color feature is 512x K_1, and that of texture feature is 13x K_2. If video is matched in this space, computation load will be heavy, and storage need is high, moreover, prototype videos may not be well separated regarding to Euclidean distance. So it is necessary to reduce feature's dimensionality and find its optimal representation in subspace.

In our approach, video clips with different contents means different video classes, and each class is represented by one prototype video, hence. a video database with N classes consists of N prototype feature vectors represented by set $X = \{X_i\}_{i=1,...N}$, $X_i \in R^D$, where X_i is feature vector of the ith proto-video, which is treated as the signal vector, D is the dimension of original feature space. Thereafter, vector is defined as a row vector. The vectors of distorted proto-videos are included in set $\hat{X} = \{\hat{X}_1^1, \hat{X}_1^2, \cdots, \hat{X}_N^m\}$, $\hat{X}_i^k \in R^D$, where $\hat{X}_i^k$ represents the kth distorted vector of the ith proto-video. Difference vector between vectors X_i and $\hat{X}_i^k$ is $Z_i^k = X_i - \hat{X}_i^k$, which is treated as the noise vector. The set of difference vectors is denoted by $Z = \{Z_1^1, Z_1^2, \cdots, Z_N^m\}$.

Given original prototype feature set X and difference vector set Z, Oriented PCA is adopted to compute feature's optimal subspace projection with the aim to maximize signal-to-noise ratio in this subspace ([17] [20]).

Let one of the unit projection vector be denoted by $\vec{n}$, OPCA is to maximize the following generalized Rayleigh quotient:

$$q = \frac{\vec{n} C_X \vec{n}^T}{\vec{n} R_Z \vec{n}^T} \tag{2}$$

$$\text{where } C_X = E(X_i - \bar{X})^T (X_i - \bar{X})$$

$$R_Z = E(Z_i^T Z_i)$$

Where C_X is covariance matrix of feature vectors in *X*, while R_Z is correlation matrix of difference vectors in *Z*. The nominator of (2) is the variance of prototype vectors' projected values on direction $\vec{n}$, while denominator is correlation of difference vectors' projected values on the projection axis. Therefore maximizing *q* will make proto-vectors separate while making difference vectors shrink as much as possible on this projection direction. Here correlation matrix of difference vectors is computed instead of covariance matrix, because their mean value not just variance should be compressed on the projections. To compute projection directions, let $\nabla q = 0$, then the solution of $\vec{n}$ becomes solving the following generalized eigenvector problem,

$$C_X \cdot \vec{n}^T = q \cdot R_Z \cdot \vec{n}^T \tag{3}$$

If the dimension of subspace is set to D_1, then unit vectors corresponding to the D_1 largest generalized eigenvalues are used as OPCA projection directions. Solving (3) can first take Cholesky decomposition of R_Z and transform it to the normal eigenvector problem. Different from PCA, OPCA projection vectors are not necessarily orthogonal to each other, and not necessarily unit ones.

2.2 Statistical Analysis of Video Database

A video model database generally consists of a lot of prototype feature vectors, so NN classifier will be an efficient way to recognize video copies. A test video is recognized as the closest proto-video if the distance is below a threshold θ, otherwise the test video will not belong to any proto-video.

In order to estimate error rate of threshold NN classifier and obtain optimal classification threshold θ_{opt}, it is necessary to know about statistics of video model database. Since classification is based on feature discrimination, we define three types of feature distances to explore video database statistics. Distances are defined as follows:

1) The first type of distance is within-class distance d_w between distorted proto-videos and model database. Let the set of prototype vectors be denoted by $O = \{O_i\}_{i=1,\ldots,N}$, and its distorted vector set be $\hat{O} = \{\hat{O}_1^1, \hat{O}_1^2, \cdots, \hat{O}_N^m\}$, where $\hat{O}_i^k$ represents a distorted vector of the *i*th proto-video, so the within-class distance between $\hat{O}_i^k$ and *O* is defined as,

$$d_w(\hat{O}_i^k, O) = d(\hat{O}_i^k, O_i) \tag{4}$$

Where $d(\hat{O}_i^k, O_i)$ is the feature distance function.

2) The second type of distance is the minimum between-class distance between distorted proto-video and the database, denoted by d_{bi},

$$d_{bi}(\hat{O}_i^k, O) = \min_{j \neq i} d(\hat{O}_i^k, O_j) \tag{5}$$

3) The third type of distance is the minimum distance between non-prototype video and the database, denoted by d_{bo}, where non-prototype video means the video that does not belong to any class in database.

$$\text{If } Q \notin O \cup \hat{O},\ d_{bo}(Q, O) = \min_i d(Q, O_i) \tag{6}$$

Illustration of d_w, d_{bi} and d_{bo} in feature space is shown as Fig.1.

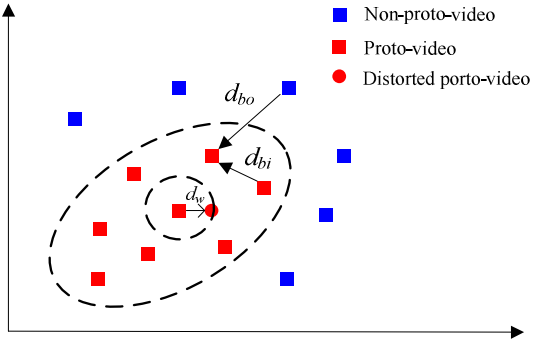

Fig. 1 Illustration of d_w, d_{bi} and d_{bo}

Distributions of d_w and d_{bi} are built-in statistics of model database, which reflect the variation between proto-videos and their distorted copies, and the discrimination between proto-videos. Distribution of d_{bo} is not only dependent on model database, but also related to distribution of non-prototype videos.

If we get the distributions of the above three distances, error rate of threshold NN classifier can be exactly computed. When the number of video models is greater than 2, recognition error comes from the following three sources: ① distorted proto-video is classified as non-prototype video; ② distorted copy of one proto-video is recognized as another proto-video; ③ non-prototype video is recognized as a proto-video.

If proto-video vector set is denoted as O, and video class label set be denoted by $\tilde{O}$, prototype vectors and their distorted vectors are included in set $\bar{O} = O \cup \hat{O}$, q is test video, $r(q)$ is the class label recognized, while $\bar{r}(q)$ is its

true class label, then the probability that q be wrongly classified is computed as follows:

$$\begin{aligned} P_e &= P(q \in \bar{O}, r(q) \notin \tilde{O}) \\ &\quad + P(q \in \bar{O}, r(q) \in \tilde{O}, r(q) \neq \bar{r}(q)) + P(q \notin \bar{O}, \bar{r}(q) \in \tilde{O}) \end{aligned} \tag{7}$$

If a unified threshold θ is adopted, (7) will become as,

$$\begin{aligned} P_e &= P(q \in \bar{O}) \cdot P(d_w > \theta, d_{bi} > \theta \mid q \in \bar{O}) \\ &\quad + P(q \in \bar{O}) \cdot P(d_{bi} \le \theta, d_w > d_{bi} \mid q \in \bar{O}) \\ &\quad + P(q \notin \bar{O}) \cdot P(d_{bo} \le \theta \mid q \notin \bar{O}) \end{aligned} \tag{8}$$

Suppose normalized distance is continuous in [0,1], and the density functions of d_w, d_{bi} and d_{bo} are $p_1(x)$, $p_2(x)$ and $p_3(x)$ respectively. It is also assumed that random variables d_w, d_{bi} are independent to each other, which is reasonable because for one feature point d_w, d_{bi} are computed with reference to two non-overlapped subsets of proto-vectors. Given assumption above, if the prior probability of proto-videos and their distorted copies is $P(q \in \bar{O}) = \eta$, and the prior for non-prototype videos is $P(q \notin \bar{O}) = 1 - \eta$, then (8) will become as,

$$P_e(\theta) = \eta \cdot (\int_{\theta}^{1} p_1(x) \cdot dx \int_{\theta}^{1} p_2(y) \cdot dy + \int_{0}^{\theta} p_2(x) \cdot dx \int_{x}^{1} p_1(y) \cdot dy) + (1-\eta) \cdot \int_{0}^{\theta} p_3(x) \cdot dx$$

Since it is a function of threshold θ, its minimum value can be obtained by setting $P_e'(\theta) = 0$, which is,

$$P_e'(\theta) = -\eta \cdot p_1(\theta) \int_{\theta}^{1} p_2(x) dx + (1-\eta) p_3(\theta) = 0$$

If $\eta = 0.5$,

$$p_1(\theta) \int_{\theta}^{1} p_2(x) dx = p_3(\theta) \tag{9}$$

Since $\int_{\theta}^{1} p_2(x) dx \le 1$, so $p_1(\theta) \ge p_3(\theta)$, the optimal threshold lies on the left of the intersection point of $p_1(x)$ and $p_3(x)$.

2.3 Results

In experiment we built a prototype video database which consists of 1000 short video clips with length from 15 to 90s, most of which are commercials and film trailers. Video format is: frame size 720x576, 25fps. Distorted copies of these

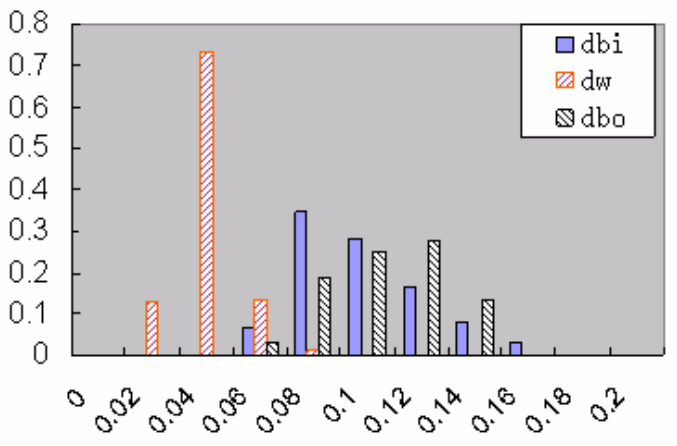

Fig. 2 Histograms of d_w, d_{bi} and d_{bo} (350 models)

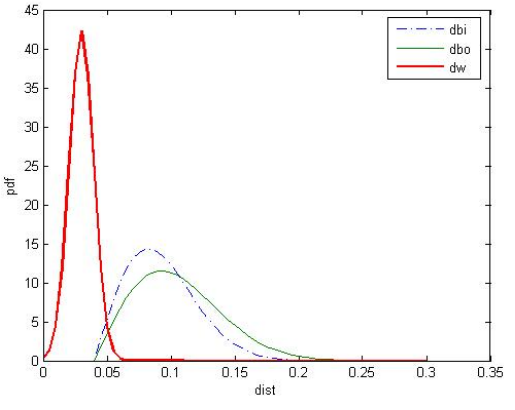

Fig. 3 Density functions of d_w, d_{bi} and d_{bo} (350 models)

proto-videos are produced by transcoding operations combining frame downsizing from 720x576 to 352x288 with frame rate reduction from 25fps to 15fps, which is the common distortion lying between broadcast video and web video copies. Video length is set to 10s when computing the feature vectors. The number of texture feature clusters is 5 while that of color feature is 1. Then OPCA is adopted to compute the 64 subspace projections.

This method is evaluated under two database sizes, one has all the 1000 proto-videos, and the other has 350 randomly chosen proto-videos. Density functions of d_w and d_{bi} are estimated from proto-vectors and their distorted vectors in subspace, and another 670 non-prototype videos are used to compute distribution of d_{bo}. Histograms of the three distances for 350 models are shown as Fig.2, and

their density functions are shown as Fig.3, where d_w is approximated by normal distribution, while d_{bi}, d_{bo} are approximated by Rayleigh distribution.

From Fig.2, 3 we can see that distributions of d_w and d_{bi} or d_{bo} are well separated in subspace. Optimal threshold θ_{opt} is chosen as the intersection point of d_w and d_{bi} which is about 0.05, and corresponding training error rate is a quite low value 1.63%, as shown in Table 1.

Table 1 Minimum training error rates

Error rate / Method	*L*=5s	*L*=10s	*L*=15s
Subspace feature (350 models)	2.9 %	1.63 %	1.57 %
Original feature (350 models)	23.7 %	20.7 %	5.8 %
Subspace feature (1000 models)	7.38 %	4.27 %	5.37 %
Original feature (1000 models)	26.25 %	21 %	8 %

The projection matrix computed from 10s clips is applied on 5s and 15s clips to calculate subspace features, and quite low error rates are also achieved. When the number of models increase from 350 to 1000, error rates increase correspondingly, but are still very low, the error rate for 5s clips is below 8%.

By comparison, error rate is also tested using original video feature. As is shown in Table I, longer clips show better robustness to significant distortion by frame dropping and downsizing, since more feature points join clustering process, so the effect of frame distortion and dropping can be better counteracted. However, by subspace feature transformation shorter clips (e.g. below 10s) can result in the same performance with that of longer clips (e.g. 15s).

For testing, those 1000 prototype videos are transcoded by frame downsizing to CIF or frame rate reduction to 15fps alone, and these distorted videos plus other 1000 non-prototype videos are used to test the trained subspace classifier under 1000 prototypes with length set to 10s. False negative error rate is zero, and total error rate is 2.8%. This result shows that since composite distortions by downsizing and frame dropping are maximally compressed in subspace projections, slighter distortion by downsizing or frame dropping alone can also be maximally compressed.

We also tested that when PCA is applied to 10s videos using 64 eigenvector projections corresponding to the largest eigenvalues, minimum error rate is 19.8% in case of 350 models, nearly the same performance with original feature. This result explains that PCA is built for reconstruction and compression, while OPCA is good for classification.

3 Unknown Video Repeat Mining

In this section we propose a novel approach for unknown repeat repeats mining. Two detectors in a cascade structure are employed to achieve fast and accurate detection, and a reinforcement learning approach is adopted to efficiently maximize detection accuracy. In this approach very short video repeats (< 1s) and long ones can be detected by a single process, while overall accuracy remains high. Since video segmentation is essential for repeat detection, performance analysis is also conducted for several segmentation methods. Results of video structure analysis by video repeat mining are also presented.

3.1 Framework

The proposed framework is shown in Fig. 4. We employ two cascade detectors to identify repeated clips, with the first detector discovering potential repeated clips, and the second one improving accuracy.

The first detector includes three temporal level video representations, namely video units (VU), video segments (VS) and video clips (VC), as well as corresponding video similarity measures. The first step is content based video segmentation. Video stream is partitioned into basic video units (VU). The second step is self-similarity analysis. Video units are grouped by a window size W, e.g. two units as one group, to form bigger size video segments (VS), then they are compared with each other to produce similarity matrix S. By similarity measure f_1, two segments will be judged as either identical or non-identical, so S is a binary matrix

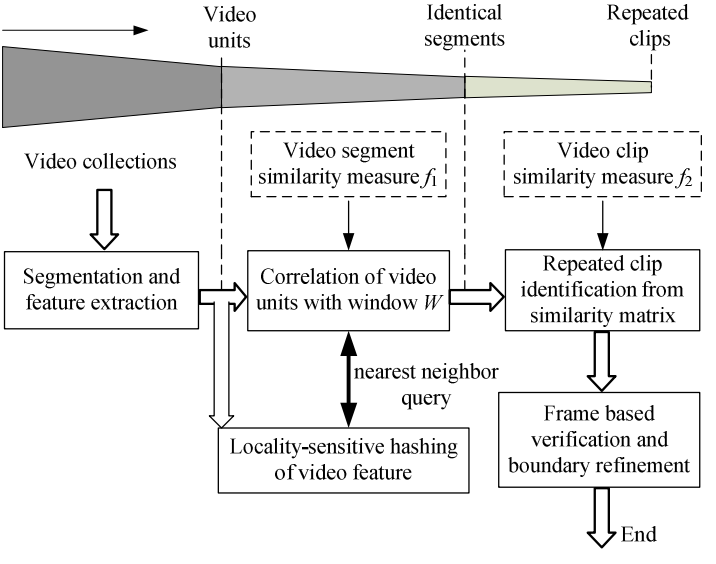

Fig. 4 Framework for repeat video clip identification

which is generally a sparse one that can be compactly represented to save storage. Here *locality sensitive hashing* (*LSH*) is adopted to reduce correlation complexity. The third step is to identify repeat clips from similarity matrix S. Basically repeated clips can be identified from diagonals, which is controlled by similarity measure f_2.

The second detector adopts frame based matching to verify candidate repeated clips for accuracy improvement. After that a boundary refinement step is employed to extend repeated clips' boundaries close to their maximum ones as possible. The last step is repeated clip labeling. Repeated instances will be extracted from repeated clip pairs and grouped into multiple categories. Each category represents a unique repeat pattern.

3.2 Video Representation and Feature Extraction

3.2.1 Video Segmentation and Three Level Representation

In our method video stream is segmented by content based keyframes, and interval between two consecutive keyframes is treated as the basic video unit (VU). Keyframe selection is based on color histogram difference. Suppose H_1 and H_o are color histograms of current frame and the last keyframe respectively, then current frame is selected as new keyframe if the following condition is satisfied,

$$\left|1-\text{inter}(H_1, H_0)\right| > \eta \tag{10}$$

where $\text{inter}(H_1, H_0)$ is intersection of two color histograms, η is threshold.

This representation is a seamless video segmentation without temporal data loss, which is similar to shot segmentation, but its granularity is smaller than shot. Its advantages lie in: First it is robust to boundary shift of repeat clips. Generally shift error can be corrected after a shot cut. Secondly it can reduce correlation between adjacent video units, so diagonal pattern will be sharper and easier be identified. The third advantage is that temporal length of video unit can be added to increase feature discrimination.

The second level video representation (VS) is formed by grouping two neighbor units (W=2). Compared to the first level, the second level has almost the same number of samples, but the discrimination ability will improve a lot, thus providing a less noisy output to build a higher level of video repeat clips.

3.2.2 Video Features

Two types of video features are extracted. The first one is video unit (VU) feature used in the first detector, and the other one is frame feature used in the second detector.

1) *Video unit feature*

Video unit feature includes interval length and color fingerprint proposed by Yang et. al ([12]). A video unit is partitioned into K sub-intervals, and represented by K blending images formed by averaging frames within each sub-interval along

time direction. Each blending image is then divided into $M \times N$ equal size blocks each of which is represented by the major and minor color components among RGB, as illustrated in Fig. 5. Color fingerprint is the ordered catenation of these block features. If $\overline{R}$, $\overline{G}$, $\overline{B}$ are the average color values of a block, and their descending order is (V_1, V_2, V_3), then the major color and minor color are determined by the following rules:

Rule 1: if $V_1 > V_3$,

$$\text{Major Color} = \begin{cases} \arg\max(\overline{R}, \overline{G}, \overline{B}) & \text{if } (V_1 - V_3) > \tau \\ \text{Uncertain} & \text{if } (V_1 - V_3) \le \tau \end{cases}$$

$$\text{Minor Color} = \begin{cases} \arg\min(\overline{R}, \overline{G}, \overline{B}) & \text{if } (V_2 - V_3) > \tau \\ \text{Uncertain} & \text{if } (V_2 - V_3) \le \tau \end{cases}$$

Where τ is the parameter that controls the robustness to color distortion and discriminative ability of this feature.

Rule 2: if $V_1 = V_3$ (gray image),

$$\text{Major Color} = \text{Minor Color} = \begin{cases} \text{bright} & \text{if } V_1 > \tau_1 \\ \text{dark} & \text{if } V_1 \le \tau_1 \end{cases}$$

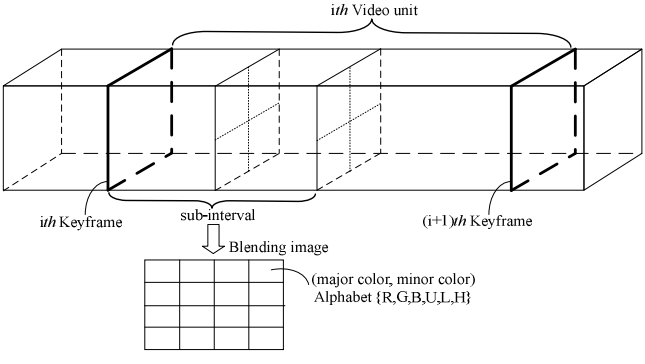

Fig. 5 Illustration of video segmentation and feature extraction

Major and minor color patterns have six possible symbol values from alphabet {R, G, B, U, L, H}, where U, L and H stand for uncertain, dark and bright respectively. In this work one blending image (K=1) is used for each unit, and divided into 8x8 blocks (M=N=8), thus the color feature is a 128 dimensional symbol vector. We also apply LSH indexing on this color fingerprint, and its string representation can be easily transformed to a bit string required by LSH algorithm ([21]) without incurring extra errors. By LSH and unit length filtering, complexity of searching identical video units can be reduced by hundreds of times.

2) *Frame feature*

Each frame is divided into 4 sub-frames, and RGB color histogram(8x8x8 bins) of each sub-frame is quantized to a symbol by VQ, so each frame is represented by 4 symbols.

3.3 Video Similarity Measures

Video similarity measures are conducted at several levels to ensure efficient and robust video repeat discovering: Video Unit, Video Segment, and Video Clip.

3.3.1 Video Segment Similarity Measure

Given two video units vu_i and vu_j, their distance $D(vu_i, vu_j)$ is defined as:

$$D(vu_i, vu_j) = \sqrt{d^2(F_i, F_j) + \left[len(vu_i) - len(vu_j)\right]^2} \tag{11}$$

where F_i, F_j are color fingerprint vectors of vu_i and vu_j, $d(F_i, F_j)$ is color fingerprint distance function ([12]), len(·) is length feature. If VS consists of *W* video units, similarity measure f_1 between the *i*th segment and *j*th segment $VS_j : \{vu_j, vu_{j+1}\}$ is defined as:

$$f_1(VS_i, VS_j) = \begin{cases} 1 & if\ D(vu_i, vu_j) < \varepsilon_1, \cdots, D(vu_{i+W-1}, vu_{j+W-1}) < \varepsilon_1 \\ 0 & otherwise \end{cases} \tag{12}$$

where ε_1 is distance threshold.

3.3.2 Clip Level Aggregation

Repeat clips will appear as diagonals in similarity matrix. However, due to segmentation errors, the line will not be the integrated one. Moreover those line

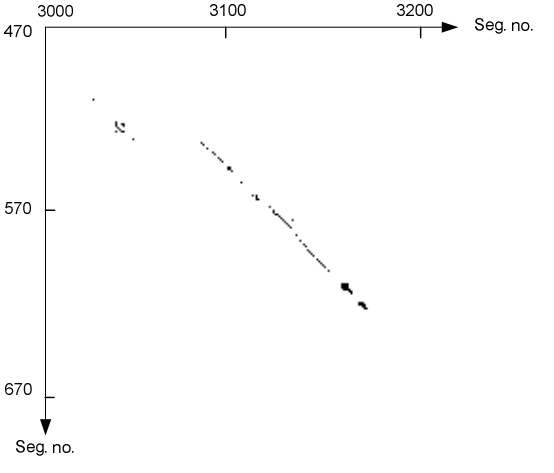

Fig. 6 Example of diagonal tracks for repeat sequences

fragments will not be collinear if non-uniform partition is used. Fig. 6 shows part of a similarity matrix computed in our experiment. As we can see, diagonal tracks are fragmented and contaminated by noises. To get the whole repeat clip correctly we design a hierarchical aggregation algorithm purely based on temporal boundaries of repeat segments.

This algorithm is described as follows:

Step 1: First link strong diagonal tracks whose length exceeds one. The start and end time of two pairs of repeat sequences (I,I') and (II,II') corresponding to two diagonal lines are represented by ($T1_{start}$,$T1_{end}$), ($T1'_{start}$,$T1'_{end}$) and ($T2_{start}$,$T2_{end}$), ($T2'_{start}$,$T2'_{end}$) respectively, which is illustrated in Fig. 7.

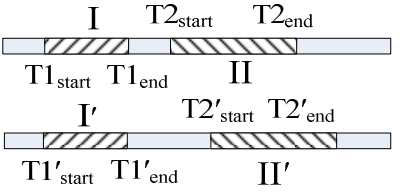

Fig. 7 Illustration of two pairs of adjacent repeat segments

If one of the two conditions in (13) is satisfied, (I,I') and (II,II') will be merged into one repeat pair.

a. Overlap: $T1_{start} \le T2_{start} \le T1_{end}$, $T1'_{start} \le T2'_{start} \le T1'_{end}$

b. Adjacency: $|T2_{start} - T1_{end}| < \mu_1$, $|T2'_{start} - T1'_{end}| < \mu_1$, $|(T2_{start} - T1_{end}) - (T2'_{start} - T1'_{end})| < \varepsilon_2$ (13)

where μ_1 defines neighborhood distance, ε_2 is displacement allowed for neighbor repeat segments, thus controls temporal variations of the whole repeat clip.

Boundaries of merged repeat pair are computed as:

$T_{start}=\min(T1_{start},T2_{start})$, $T_{end}=\max(T1_{end},T2_{end})$;

$T'_{start}=\min(T1'_{start},T2'_{start})$, $T'_{end}=\max(T1'_{end}, T2'_{end})$.

This new repeat pair will be put into the repeats list to replace originals, and the above process is iterated till no change of the list.

Step 2: Connecting single dots based on results of step 1 with the same merging criterion as step 1.

Step 3: The connected sequences after above two steps are further connected and merged until there is no change.

By the above aggregation algorithm the whole image of repeat clips can be well constructed from their local repeat segments, thus providing good foundation for further similarity analysis and boundary refinement. Moreover, this algorithm only needs to store boundaries of repeat segments but not similarity matrix, which can have efficient implementation for even large video data mining.

3.3.3 Second Stage Matching

The second detector adopts frame by frame matching. The total number of identical frames is normalized by the average sequence length to get the similarity score. A repeat pair is judged as true one if the following condition is satisfied,

$$score > (1 + e^{-L})\varepsilon_3 \tag{14}$$

where *score* is the similarity value, *L* is the minimum length of the two clips in seconds, and ε_3 is threshold. This decision rule uses soft thresholds for different length sequences. Since shorter sequences are assumed less reliable ones, they should satisfy more stringent condition to pass through verification. Once a repeat pair is verified, their boundaries are extended frame by frame until dissimilar frames are encountered.

3.4 Reinforcement Learning of Detectors

The two cascade detectors contain several parameters, like distance thresholds, LSH parameters etc., but the intrinsic and crucial ones that affect detection accuracy are ε_1, ε_2 and ε_3 in (12)(13)(14) respectively. Tuning these three parameters can significantly change detection results. The three parameters have clear physical meanings. ε_1 reflects feature distortion of identical video units for certain video data and feature extraction; ε_2 that defines maximum temporal displacement between neighbor repeat segment pairs in clip aggregation function is related to video unit granularity and temporal variation allowed for the whole repeat clips. ε_3 in the second detector balances recall and precision. Parameter μ_1 in (13) defining neighborhood of repeat segments is not crucial for final results as long as it is

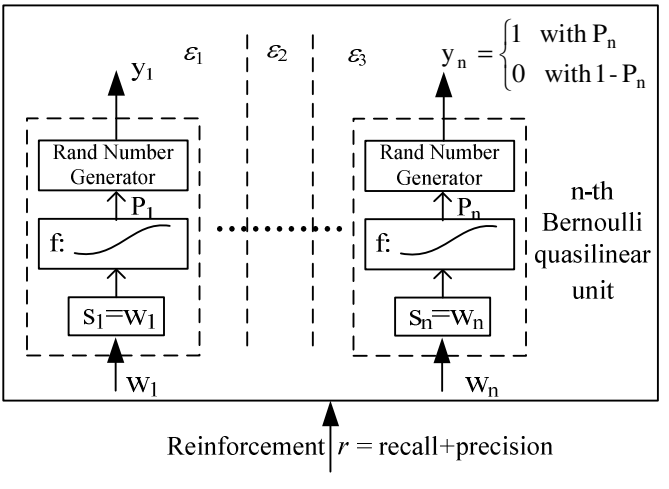

Fig. 8 Connectionist reinforcement learning network

in a range, e.g. 10s ~ 20s. Segmentation related parameter is important for final results, but it is not intrinsic to the detector. Different segmentation methods may have different types of parameters or no parameter at all.

In the following we will propose a method to learn appropriate values of ε_1, ε_2 and ε_3 in order to achieve optimal performance on selected video data. Given certain segmentation and feature extraction, the three parameters in the two detectors are trained together by reinforcement learning in a non-associative paradigm. Given an input, the learning network produces the three parameters, then a scalar indicating "goodness" of detection results under these parameters is immediately used as a reinforcement for the learning network. In our approach the sum of recall and precision is taken as the reinforcement factor. We also adopt the connectionist REINFORCE algorithm ([22]) in which the units of network are Bernoulli quasilinear units whose output is 0 or 1, statistically determined by Bernoulli distribution with parameter $p = f(s) = 1/(1+\exp(-s))$, which is shown in Fig.8. Each Bernoulli quasilinear unit has one input weight, and the three parameters are encoded by gray codes corresponding to the outputs of n Bernoulli quasilinear units. After receiving a reinforcement r, the weights of Bernoulli quasilinear units are updated by (15).

$$\Delta w_i = \alpha(r-b)(y_i - p_i) \tag{15}$$

where α is a positive learning rate, b serves as a reinforcement baseline, y_i is the output of the ith Bernoulli quasilinear unit, and p_i is the Bernoulli distribution parameter. It has been shown by Williams ([22]) that this learning algorithm statistically climbs the gradient of expected reinforcement in weight space, which means that the detector parameters will change in the direction along which the sum of recall and precision increases.

3.5 Results

For news video we chose half-hour CNN and ABC news videos from TRECVID data to form two video collections, each of which contains 12 day programs with 6 hours around. By manually searching short repeat clips including program logos and commercials, but neglecting other repeat scenes, i.e. anchor persons, 34 kinds of repeat clips with totally 186 instances are found from CNN collection, while 35 kinds with totally 116 instances found from ABC collection. In addition broadcast videos of Channel News Asia (CNA) are also used for structure analysis.

3.5.1 Detector Training

Parameters of the two detectors are learned by the approach presented in section 3.4. Three hour CNN news videos are randomly chosen for training. Videos are segmented by content based keyframes.

The reinforcement learning rate α in (15) is set to 0.01 and reinforcement baseline b set to 0.7. Parameters ε_1, ε_2 and ε_3 are each encoded by 5 bit gray code, so

there are totally 15 Bernoulli units in this network. Parameter value range is set to [0,1]. Initial parameters are set to empirical values, and initial weights are all zeros. During each learning round we manually check the detection results to compute recall and precision, then feed their sum as reinforcement of the learning network. Recall and precision are calculated as (16).

$$\text{recall} = \frac{\text{number of correct repeat instances}}{\text{number of all true repeat instances}}$$
$$\text{precision} = \frac{\text{number of correct repeat instances}}{\text{number of all detected instances}} \tag{16}$$

In experiment recall and precision in the first round learning are 74% and 100%, but after ten rounds of learning, recall and precision already climb to 94.2% and 96% respectively. Since the next several rounds of learning do not lead to reinforcement increase, we then stop the learning.

3.5.2 Testing Accuracy

The trained detectors are tested on the rest 3 hour CNN videos and 6 hour ABC videos. Recall and precision on CNN videos are 92.3% and 96%, while 90.1% and 90% those for ABC videos. This accuracy is obtained without setting a minimum sequence length to filter errors, so most of the errors come from those very short clips. The shortest correct repeat detected is just 0.26s (partial of "play of the day" logo in CNN video), while the longest one is 75 seconds long.

Boundary accuracy of repeat pairs is also measured. We selected 300 repeated pairs that cover almost all repeat patterns and checked their boundary shift before boundary refinement. The smallest shift is 0 s, while the largest one is 16.4s, and the average shift is 0.47s. Around 80% of the shifts are within 0.2 seconds. After frame by frame boundary refinement those large shifts can be effectively reduced to 0~1 second.

3.5.3 Performance Analysis of Segmentation Methods

Video segmentation is essential for this approach, so experiments are conducted to compare performances by proposed keyframe based segmentation, uniform segmentation and shot segmentation. The video data are 3 hour CNN videos used in Section 3.5.1. Two keyframe based segmentations are implemented with η =0.15, 0.30 respectively. Uniform segmentation utilizes I frames (every 12 frames). Shot detection includes cuts, fades in-outs and dissolves. Video unit features for all segmentations are color fingerprint and length. The video segment (VS) size W for shot segmentation is set to 1, and the minimum number of diagonal points in repeat aggregation is also set to 1. Thus this method can detect not only single repeat shots, but also repeat clips beyond shots. Detectors are separately trained for each segmentation strategy to achieve their nearly optimal performance, and training results are shown in Table 2.

Table 2 Performance comparison of video segmentation methods

	Uniform sampling	Keyframe (η =0.15)	Keyframe (η =0.30)	Shot based
recall	87.8%	94.2%	90.7%	66.7%
precision	95.9%	96.0%	86.0%	84.7%
Video units	26344	14872	6316	1911

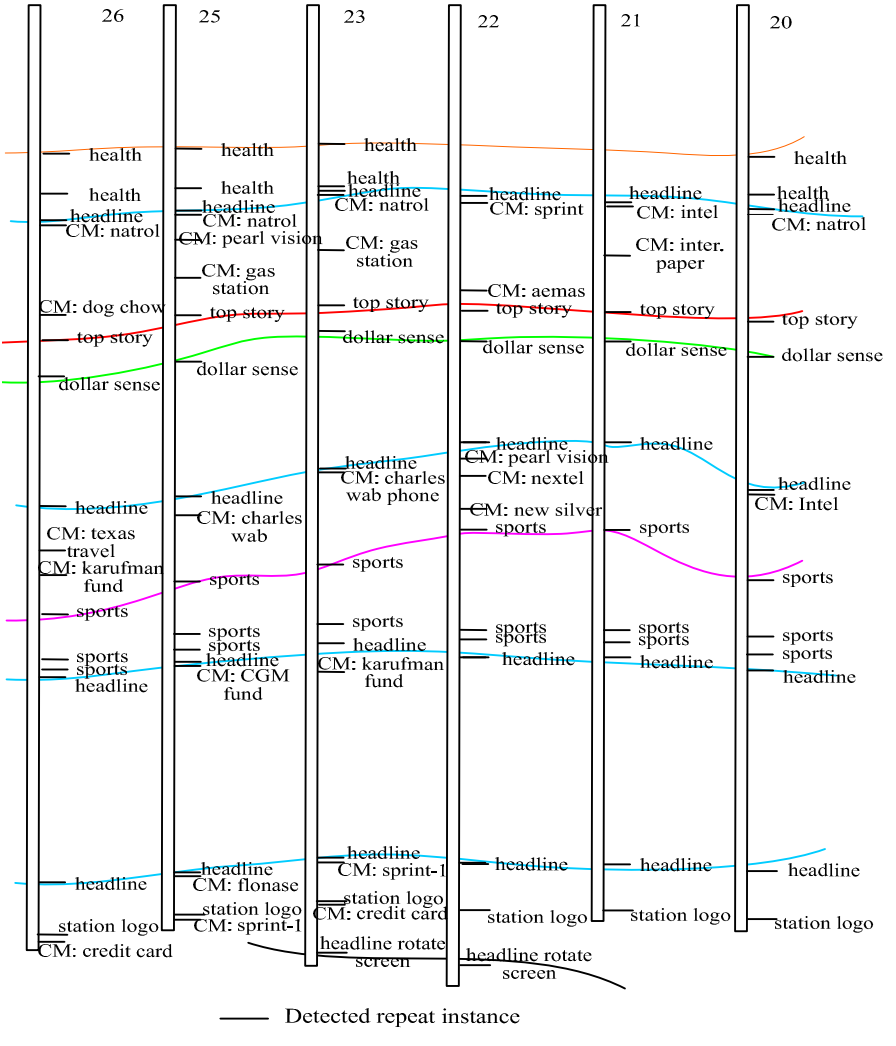

Fig. 9 CNN news video structure analysis by video repeats

From Table 2 we know that keyframe based segmentation achieves best performance. The uniform segmentation results in several times more video units than keyframes, but still gets lower recall on program logos and commercials. Uniform segmentation also detects quite many stationary scenes, such as anchor shots and black frames which occupy nearly 74% of the whole detected repeat clips pool thus overwhelm other interesting repeat patterns like program logos and commercials. Under keyframe based segmentation these still scenes are all filtered, program logos and commercials are main body of detected repeats. Shot based segmentation results in much fewer video units, but its total accuracy is much lower and many fast changing program logos are missed. When granularity of keyframe based segmentation becomes bigger, its performance will also drop because of heavier data loss.

3.5.4 Searching Efficiency Evaluation

By LSH indexing on color fingerprint, the average number of retrieved units for a query unit of CNN collection (totally 629,380frames and 31496 units) is 320, and the number of color feature comparisons is further reduced to 20 by pre-filtering one dimension length feature at trained distance threshold ε_1 =0.1, thus speedup factor is about 1575 compared to pair-wise searching. For ABC collections (totally 616,780 frames and 29838 units), the average number of retrieved units for a query unit is 1026, and further reduced to 56 by length filtering, thus speedup factor is 533. On PC with Pentium-4 2.5GHz processor the two stage detections on 6 hour CNN videos can be finished in 22 seconds, while 40 seconds for ABC videos.

3.5.5 Video Structure Discovery Results

Fig. 9 shows temporal distribution of short video repeats identified from CNN news videos of six days. Those repeat instances linked by curves are chosen as structural video elements (SVE). From this map we can clearly see that the whole program is segmented by SVEs into several layers each of which contains certain

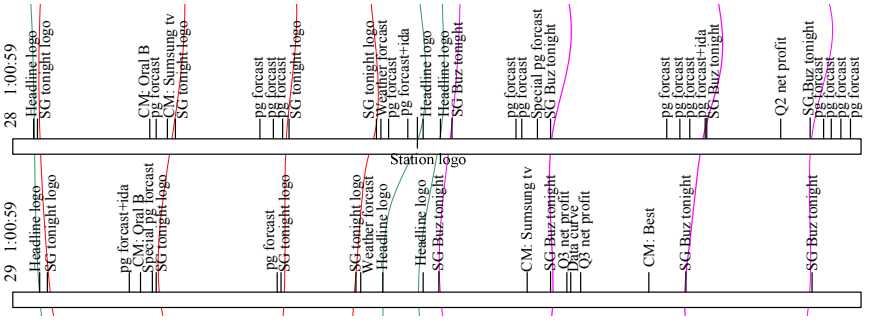

Fig. 10 CNA news video structure analysis by video repeats

topics, such as health program, top stories, financial news, sports news, commercials *et al*. Similar results are also achieved for CNA video structure discovery. Fig. 10 shows distribution of short video repeats identified from CNA one hour news videos in two days.

4 Concluding Remarks

In this chapter we frame known video repeat recognition as a standard pattern recognition problem, and take advantage of the techniques successfully applied to other classical pattern recognition problems such as face recognition, speech recognition and OCR. For example, subspace discriminative analysis is used to optimize video feature representation. Video feature model adopts sampling and clustering strategy to capture typical color and texture features of a video clip, and it shows good robustness to frame distortion and dropping. Video feature's subspace representation computed by OPCA leads to a significant improvement of recognition performance especially for very short video clips (e.g. below 10s). Compared with other robust image descriptors that require high computational complexity, e.g. SIFT ([10]), RGB color histogram and texture feature adopted in this approach can also achieve robustness to common distortions through appropriate coordinate transformation, while computation is much simpler. The proposed statistical analysis method reflects the distribution of video database in feature space by three distance distributions from which nearest neighbor classifier's error probability can be exactly estimated, and optimal classification threshold can be theoretically computed. Classification accuracy is evaluated under 1000 video models, which is a reasonable database size for some real applications, e.g. TV commercial monitoring, and very low error rate is obtained.

In unknown video repeat mining approach, we do not make feature optimization according to specific video database, but rely on self-similarity rule to discover all possible video repeats in an unsupervised way. However, in order to achieve high mining accuracy for given video collections, we adopt a supervised approach to tune the mining parameters, so this approach can be regarded as a mixture of unsupervised discovery and supervised learning. This method achieves robust detection of arbitrary length video repeats by cascaded detectors that employ different features and similarity measures. Quite short repeats (e.g. those less than 1 second) along with long ones can be detected with high accuracy, which is the strength of our approach compared to previous work ([3][9]). Similarity searching complexity of the first detector is reduced hundreds of times through LSH indexing and length filtering. Here color fingerprint is used as video unit feature, for its discrete values are naturally suitable for LSH indexing, and its combination with unit length can give discriminative representation of video units. As a comparison, known video repeat recognition approach adopts color histogram and texture as feature, for they have continuous values that are suitable for statistical analysis and subspace feature transformation. By analyzing detection performance under several video segmentation strategies, we know that video segmentation

utilizing content-based keyframes achieves best balance between detection accuracy and efficiency on short video repeats mining compared to uniform and shot based segmentation. Parameters of detectors can be efficiently optimized in a few rounds of reinforcement learning without knowing statistics of large volume of video data, which makes our approach easily adapt to different video sources. Results also show that short video repeats mining is an effective way to discover syntactic structure of news videos.

References

[1] Yang, X., Xue, P., Tian, Q.: A repeated video clip identification system. In: Proc. ACM Multimedia, Singapore (2005)

[2] Yang, X., Tian, Q., Xue, P.: Efficient Short Video Repeat Identification With Application to News Video Structure Analysis. IEEE Trans. Multimedia 9, 600–609 (2007)

[3] Cheung, S.-C., Ngueyen, T.P.: Mining Arbitrary-length Repeated Patterns in Television Broadcast. In: Proc. IEEE Int. Conf. on Image Processing (2005)

[4] Lienhart, R., Kuhmunch, C., Effelsberg, W.: On the detection and Recognition of Television Commercials. In: Proc. IEEE Int. Conf. Multimedia Computing and Systems (1997)

[5] Snchez, J.M., Binefa, X., Vitri, J.: Shot partitioning based recognition of TV commercials. Multimedia Tools and Applications 18, 233–247 (2002)

[6] Kashino, K., Kurozumi, T., Murase, H.: A quick search method for audio and video signals based on histogram pruning. IEEE Trans. Multimedia 5(3), 348–357 (2003)

[7] Yuan, J., Duan, L.-Y., Tian, Q., Xu, C.: Fast and robust short video clip search using an index structure. In: Proc. ACM Multimedia's Multimedia Information Retrieval Workshop (2004)

[8] Cheung, S.-C., Zakhor, A.: Estimation of web video multiplicity. In: Proc. SPIE, vol. 3964, pp. 34–46 (2000)

[9] Pua, K.M., Gauch, J.M.: Real time repeated video sequence identification. Computer Vision and Image Understanding 93(3), 310–327 (2004)

[10] Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–120 (2004)

[11] Oostveen, J.C., Kalker, A.A.C., Haitsma, J.A.: Visual hashing of digital Video: applications and techniques. In: SPIE applications of digital image processing XXIV, San Diego, pp. 121–131 (2001)

[12] Yang, X., Tian, Q., Chang, E.C.: A Color Fingerprint of Video Shot for Content Identification. In: Proc. ACM Multimedia, NY, USA (2004)

[13] Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: Discriminant Common Vectors for Face Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 27(1), 4–13 (2005)

[14] Chakrabarti, S., Roy, S., Soundalgekar, M.: Fast and Accurate Text Classification via Multiple Linear Discriminant Projections. In: Proc. Int'l. Conf. Very Large Data Bases, pp. 658–669 (2002)

[15] Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces versus Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Analysis and Machine Intelligence 19(7), 711–720 (1997)

[16] Yang, X., Yuan, M.: Video Copy Recognition By Oriented PCA and Statistical Analysis. In: Proc. IEEE Int. Conf. on Image Processing, Cario, Egypt (2009)
[17] Burges, C.J.C., Platt, J.C., Jana, S.: Distortion Discriminant Analysis for Audio Fingerprinting. IEEE Trans. Speech and Audio Processing 11(3), 165–174 (2003)
[18] Herley, C.: ARGOS: Automatically Extracting Repeating Objects From Multimedia Streams. IEEE Trans. Multimedia 8(1), 113–129 (2006)
[19] Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics 3(6), 610–621 (1973)
[20] Diamantaras, K., Kung, S.: Principal Component Neural Networks. John Wiley, Chichester (1996)
[21] Gionis, A., Indyky, P., Motwaniz, R.: Similarity Search in High Dimensions via Hashing. In: Proc. Int. Conf. Very Large Data Bases, pp. 518–529 (1999)
[22] Williams, R.J.: Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. Machine Learning 8, 229–256 (1992)