A Segmentation and Graph-Based Video Sequence Matching Method for Video Copy Detection

Hong Liu, Hong Lu, *Member*, *IEEE*, and Xiangyang Xue, *Member*, *IEEE*

Abstract—We propose in this paper a segmentation and graph-based video sequence matching method for video copy detection. Specifically, due to the good stability and discriminative ability of local features, we use SIFT descriptor for video content description. However, matching based on SIFT descriptor is computationally expensive for large number of points and the high dimension. Thus, to reduce the computational complexity, we first use the dual-threshold method to segment the videos into segments with homogeneous content and extract keyframes from each segment. SIFT features are extracted from the keyframes of the segments. Then, we propose an SVD-based method to match two video frames with SIFT point set descriptors. To obtain the video sequence matching result, we propose a graph-based method. It can convert the video sequence matching into finding the longest path in the frame matching-result graph with time constraint. Experimental results demonstrate that the segmentation and graph-based video sequence matching method can detect video copies effectively. Also, the proposed method has advantages. Specifically, it can automatically find optimal sequence matching result from the disordered matching results based on spatial feature. It can also reduce the noise caused by spatial feature matching. And it is adaptive to video frame rate changes. Experimental results also demonstrate that the proposed method can obtain a better tradeoff between the effectiveness and the efficiency of video copy detection.

Index Terms—Video copy detection, graph, SIFT feature, dual-threshold method, SVD, graph-based matching

1 Introduction

With the rapid development and wide application of multimedia hardware and software technologies, the cost of image and video data collection, creation, and storage is becoming increasingly low. Each day tens of thousands of video data are generated and published. Among these huge volumes of videos, there exist large numbers of copies or near-duplicate videos. According to the statistics of [1], on average, there are 27 percent redundant videos that are duplicate or nearly duplicate to the most popular version of a video in the search results from Google video, YouTube, and Yahoo! video search engines. As a consequence, an *effective* and *efficient* method for video copy detection has become more and more important. A valid video copy detection method is based on the fact that "video itself is watermark" [2] and makes full use of the video content to detect copies.

To facilitate the discussion of "video copy" in this paper, we use the definition of video copy in TRECVID 2008 tasks.

Definition of copy video: A video V_1, by means of various transformations such as addition, deletion, modification (of aspect, color, contrast, encoding, and so on), camcording, and so on, is transformed into another video V_2, then video V_2 is called a copy of video V_1.

In content-based copy detection task of TRECVID 2008, 10 transformations [3] are defined. These 10 transformations are as below, see [4] for detail. And Table 1 shows five single transformations and the corresponding image examples.

T1. *Cam-cording*; **T2**. *Picture in picture*; **T3**. *Insertions of pattern: Different patterns are inserted randomly: captions, subtitles, logo, sliding captions*; **T4**. *Strong re-encoding*; **T5**. *Change of gamma*; **T6, T7**. *Decrease in quality: Blur, change of gamma (T5), frame dropping, contrast, compression (T4), ratio, white noise*; **T8, T9**. *Post production: Crop, Shift, Contrast, caption (text insertion), flip (vertical mirroring), Insertion of pattern (T3), Picture in picture (the original video is in the background)*; **T10**. *Combination of random five transformations among all the transformations described above.*

The objective of video copy detection is to decide whether a query video segment is a copy of a video from the video data set. A copy can be obtained by various transformations. If a video copy detection system finds a matching video segment, it returns the name of copy video in the video database and the time stamp where the query was copied from.

Fig. 1 shows the framework of content-based video copy detection. It is composed of two parts:

1) *An offline step*. Keyframes are extracted from the reference video database and features are extracted from these keyframes. The extracted features should be robust and effective to transformations by which the video may undergo. Also, the features can be stored in an indexing structure to make similarity comparison efficient.

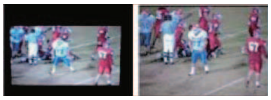
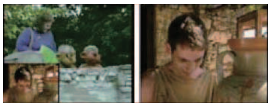
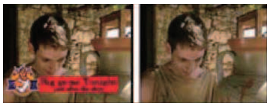
2) *An online step*. Query videos are analyzed. Features are extracted from these videos and compared to those stored in the reference database. The matching results are then analyzed and the detection results are returned.

• *The authors are with the Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Building of Computer Science, 825 Zhangheng Road, Shanghai 201203, China. E-mail: {liuhong2007, honglu, xyxue}@fudan.edu.cn.*

Manuscript received 26 Oct. 2011; revised 4 Mar. 2012; accepted 6 Apr. 2012; published online 19 Apr. 2012.
Recommended for acceptance by B.C. Ooi.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-10-0658.
Digital Object Identifier no. 10.1109/TKDE.2012.92.

1041-4347/13/$31.00 © 2013 IEEE Published by the IEEE Computer Society

TABLE 1
Examples of Five Single Transformations

Type	Example *
T1-Cam-cording: this transformation is done manually by filming a movie on a screen.	
T2-Picture in picture: a video is inserted in another video, the scale and spatial location of the inserted video can be changed.	
T3-Insertion of patterns: different patterns are inserted randomly: captions, subtitles, logo, sliding captions.	
T4-Strong re-encoding: the resolution of the video is reduced, the bit rate is changed and the video can be also encoded with a different codec.	
T5-Change of gamma: the gamma value for each color is changed randomly.	

* The right example image is the original video frame and the left example image is the transformed video frame. These video frames come from MUSCLE-VCD-2007 and TRECVID 2008.

Based on the study, in these transformations, picture in picture is especially difficult to be detected [5], [6], [7]. And for detecting this kind of video copies, local feature of SIFT is normally valid. However, matching based on local features of each frames in two videos is in high computational complexity. In this paper, we focus on detecting picture in picture and propose a twin-threshold segmentation, feature set matching, and graph-based sequence matching method.

2 Related Work

As reviewed in [8], many content-based video copy detection methods have been proposed. Furthermore, *copy* is a subset of *near duplicate*. Copies have an origin, while near-duplicates may not. Specifically, two news videos on the same event from two broadcasting corporations are not copies, but near duplicates since they deliver the same information to audience, although some variations on the scenes may exist. Also, there are many methods proposed on near-duplicate detection. The methods on *copy* and *near-duplicate* detection can be grouped into two types.

One type of copy detection methods use global descriptor. Specifically, Hampapur et al. compared distance measures and video sequence matching methods for video copy detection [2], [9]. They employed convolution for motion direction feature, $L1$ distance for ordinal intensity signature (OIS), and histogram intersection for color histogram feature. The results show that the method using OIS performs better. Yuan et al. combined OIS with color histogram feature as a tool for describing video sequence [10]. The work in [11] and [12], with [9] as the basis, designed region intensity rank signature along time sequence. Specifically, they divided each video frames along the time sequence into several blocks and proposed average gray values for each block. Then, they linked gray values of these divided blocks separately along the time direction before they use those sequence information to describe the video content. Shen et al. [13], [14] introduced a real-time near-duplicate video detection system, UQLIPS, which globally summarized each video to a single vector. Huang et al. [15] used global image feature such as color histogram and texture to represent each video frame. Wu et al. [1] adopted the color histogram in HSV color space to detect and remove the majority of duplicates of web videos.

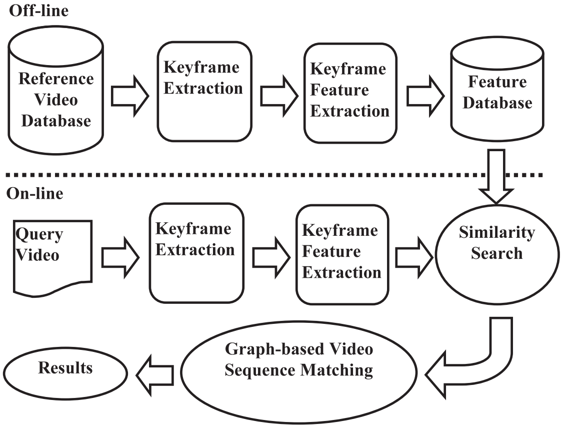
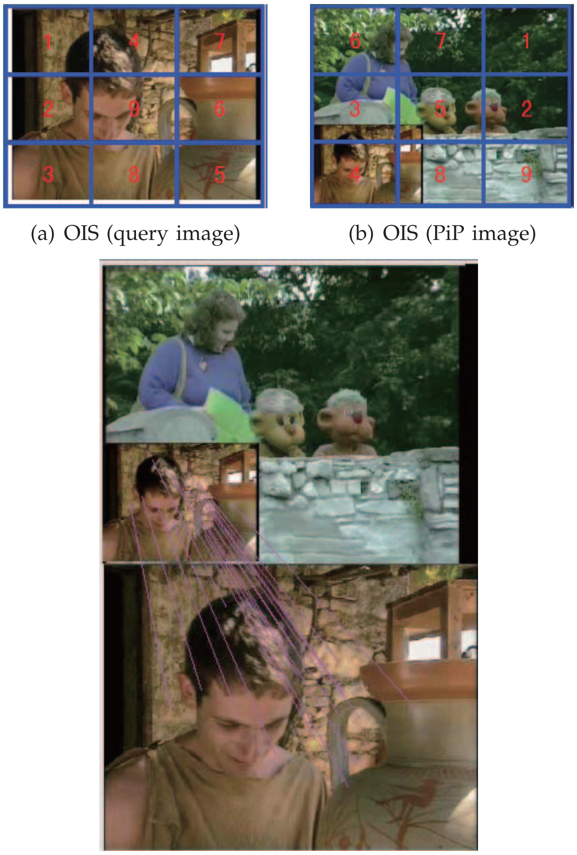


Fig. 1. A framework of video copy detection system.

Another type of methods are based on local descriptors. The local descriptors on points, lines, and shape play an important role in image and video copy detection. Among them, descriptors on points are widely used. Specifically, spatiotemporal interest points were employed to classify human actions and to detect periodic movement [16]. Willems et al. [17] presented a robust content-based video copy detection method based on local spatiotemporal features. Ke et al. used local point features for near-duplicate image detection and subimage detection [18]. Law-To et al. [19] and Joly et al. [20] adopted Harris corner points [21] as feature points in video frames. And the difference of their methods lies in how to describe the feature points. Specifically, Law-To et al. [19] selected four different locations at the space around interest points (i.e., these four locations are in the same frame) when they describe the feature points, while Joly et al. [20] selected four different locations around interest points in both time and spatial domain. Besides, Law-To et al. [19] also described the trajectory characteristic of feature points and used labels (such as "background" or "movement") to label some feature points. This method can effectively improve the robustness and discriminative ability of video signature. Similarly, Satoh et al. [22] detected duplicate scenes by using the trajectory characteristic of the feature points. Zhou et al. [23] proposed a shot-based interest point selection approach for near-duplicate search.

Methods based on global descriptor are carried out primarily by using spatiotemporal low-level features of the whole image. The features used include color histogram,



(a) OIS (query image)

(b) OIS (PiP image)

(c) The matching result using SIFT

Fig. 2. Detection effect comparison of a global descriptor with a local descriptor.

color layout descriptor, ordinal intensity signature, and so on. The advantages lie in simple computation, capable of dealing with copies with relatively small changes, and so on. However, the performance on detecting copies with complicated transformations is not satisfactory. The methods based on local descriptor must first detect local spatiotemporal feature points on the video sequence, i.e., interest points or key points, and then use the content around the feature points to them. Mikolajczyk and Schmid [24] made a comparative study on many local descriptors. The study showed that the SIFT descriptor performs better in identifying the objects. It not only has good tolerance to scale changes, illumination variations, and image rotations, but also is robust to affine distortion, change of viewpoints, and additive noise. Compared with methods based on global descriptor, the methods based on local descriptor have a better detection performance on logo insertion, shifting or cropping, complicated edit. However, its disadvantage is the high computational cost in matching.

Fig. 2 illustrates the matching of two images using global features and local features. In Fig. 2a, the feature of the query image is (1, 4, 7, 2, 9, 6, 3, 8, 5). And the feature of the Picture in Picture (PiP) image in Fig. 2b is (6, 7, 1, 3, 5, 2, 4, 8, 9). Thus, it is difficult to use global descriptor (such as OIS) to detect video copies with complicated transformation (such as PiP). Fig. 2c shows that the SIFT feature has good matching results for PiP. However, the matching time cost for images in Figs. 2a and 2b with SIFT feature takes about 1,735 ms, which is much more than that using OIS feature, which takes 1 ms.

As SIFT descriptor has good stability and discriminative ability, we choose SIFT descriptor to describe video characteristics. Meanwhile, we suggest two solutions to the lack of high computational cost in the process of copy detection: 1) dual-threshold method to eliminate video redundant frames; 2) using singular value decomposition (SVD) for matching two feature sets of SIFT features on key points.

Another major task of the video copy detection is the video subsequence matching by using the video's temporal information [19], [16], [22], [17], [25]. When we input a query video, the objective of the video copy detection is to find whether the copy sequence exists in the target video or not. There are much uncertainty in the process of video copy detection, for example, whether there exists a copy in the video, what is the length of copy clip, and where is start and end position. Therefore, it is difficult for video copy detection to employ some supervised learning methods, which makes video copy detection more difficult than the ordinary video retrieval. The video sequence matching algorithms for video copy detection task now have two weak points. First, in matching, a hard threshold needs to be determined first to obtain the matching results. However, it is difficult to determine a generic threshold for matching different video clips. Second, an exhaustive search method will involve high computational cost to detect all possible video copies with various lengths and locations. To resolve this problem, we propose a graph-based video sequence matching method in this paper. This method has the advantages of high accuracy in locating copies, being able to compensate the deficiency in description of image low-level features, reducing detection time costs, and being able to simultaneously locate more than one copy in two comparing video sequences.

As to the preliminary version of this paper [26], the improvements of this paper lie in both the technical and evaluation parts. Specifically, in [26], the SVD method is proposed to match two images with SIFT feature point sets. And it is a step for comparing the similarity of two keyframes in the whole framework in this paper. This paper also proposes the dual threshold method to segment the video into segments and extract keyframes from each segment. Also, after obtaining the similarity for segments, considering the temporal information of video, a graph-based video sequence matching method is proposed for video copy detection. Furthermore, in the experimental part, Liu et al. [26] compare the proposed SVD-SIFT method with the frame similarity comparison by global feature, FULL-SIFT method. And in this paper, we also compare the method by performing the graph-based video sequence matching method. Finally, since the final target is for video copy detection, we also compare the method with the two runs submitted by INRIA-LEAR in TRECVID 2008.

The rest of the paper is organized as follows: The proposed auto dual-threshold method for eliminating redundant frames is presented in Section 3. And based on the extracted SIFT features for two keyframes, the feature

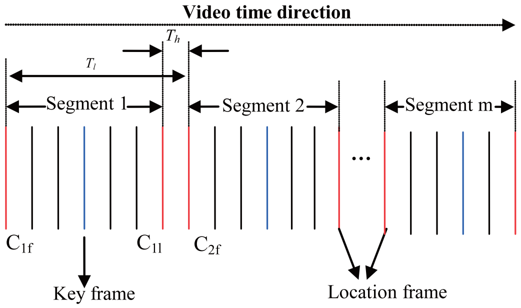


Fig. 3. Auto dual-threshold method to eliminate redundant video frames, select keyframes. C_{1f} denotes the first frame of Segment 1, C_{1l} the last frame of the Segment 1; C_{2f} denotes the first frame of the Segment 2.

matching on SVD is introduced in Section 4. Then, a graph-based video matching method for video sequence matching is proposed in Section 5. Experiments are presented in Section 6. And we conclude our work in Section 7.

3 Auto Dual-Threshold Method to Eliminate Redundant Video Frames

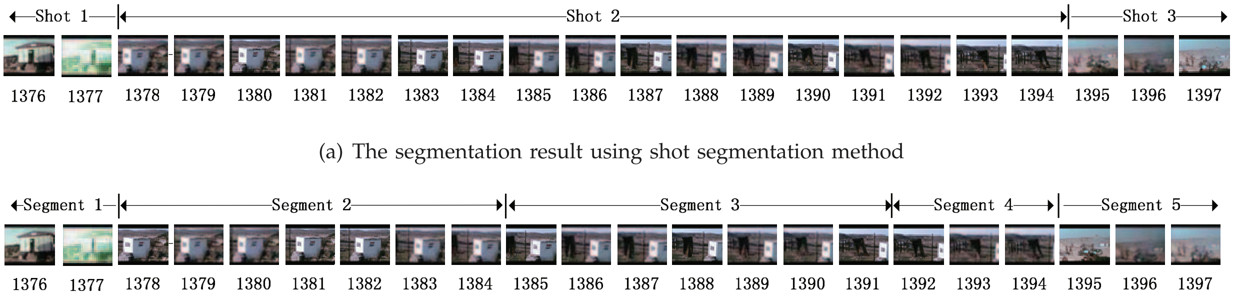
Normally, visual information of video frames is temporally redundant. So, video sequence matching is not necessarily to be carried out using all the video frames. An effective way of reducing nonnecessary matching is to extract certain keyframes to represent the video content [27], [28], [29]. And the matching of two video sequences can be first performed by matching the keyframes. Specifically, Guil et al. proposed to cluster video frames by computing the similarity between neighboring frames and choose a keyframe from each cluster to represent it [30]. However, the extracted keyframes cannot represent the temporal information among frames. On the other hand, some methods were proposed to detect video shots and extract keyframes from each shot to represent the video content [31], [32], [33]. Since there are some camera motion and object motion, the content within one shot will still has much variance. Then, we propose to detect video segments, which is an intermediate representation between video frames and video shots. Furthermore, matching two video sequences based on extracted keyframes from the segments can meet the requirement of two videos being in different frame rates.

In our framework, we use an auto dual-threshold method to eliminate redundant video frames. This method cuts continuous video frames into video segments by eliminating temporal redundancy of the visual information of continuous video frames. This method has the following two characteristics. First, two thresholds are used. Specifically, one threshold is used for detecting abrupt changes of visual information of frames and another for gradual changes. Second, the values of two thresholds are determined adaptively according to video content. Specifically, $T_h = \mu + \alpha\sigma$, where μ and σ are the mean and standard deviation of difference values between consecutive frames and α is suggested to be between 5 and 6 according to empirical study [34]. And the low threshold T_l is set to $b \times T_h$, where b is selected from the range 0.1-0.5 [31]. The auto dual-threshold method to eliminate the redundant frames is shown in Fig. 3.

Fig. 4b shows an example video segmented by using the shot boundary detection and auto dual-threshold method. It can be observed from Fig. 4b, the video shots can be obtained by the shot boundary detection method. Also, the shot with content change within it. For example, Shot 2 as shown in the figure, can be further segmented into Segments 2, 3, and 4 by the auto dual-threshold method.

Our method aims to eliminate the near-duplicate frames along the video time direction; it does not take into account the concept of the shot, also does not require postprocessing to obtain the actual shots [32]. Therefore, the particle size of our segmentation results is smaller than the shot, and is continuous in time, unlike the "shot" concept, but are subshots or segments.

By using the auto dual-threshold segmenting method, continuous video frames can be segmented into temporally continuous and visually similar video segments. Three frames are extracted from each video segment, which are the first frame, the keyframe and the last frame of this segment (as shown in Fig. 3). The keyframe is determined by the frame that is the most similar to the average frame (i.e., the average feature value of all the frames within the segment). The keyframe is used for video sequence matching, while the first and the last frames for accurately determining the segment location for copy detection and assisting matching. Each segment is assigned a continuous ID number along the time direction. We also make the



(a) The segmentation result using shot segmentation method

(b) The segmentation result using auto dual-threshold method

Fig. 4. The Difference between the auto dual-threshold method and the shot segmentation method. In Fig. 3, the distance between the last frame (Frame 1,377) of video Segment 1 and the first frame (Frame 1,378) of video Segment 2 is larger than the threshold value of sharp transition (T_h), the distance between the first frame (Frame 1,378) of video Segment 2 and the first frame (Frame 1,385) of video Segment 3 is larger than the threshold value of gradual transition (T_l).

TABLE 2
Notations

$V_f = \{f_1, f_2, f_3, ..., f_n\}$	contiguous frames of the video
$V_{f-time} = \{t_1, t_2, t_3, ..., t_n\}$	time of contiguous frames of the video, t_n is the time of f_n, $t_1 < ... < t_n$
$V_s = \{S_1, S_2, S_3, ..., S_n\}$	frame features of the video
$Vc = \{C_1, C_2, C_3, ..., C_m\}$	m segments of the video
$C_k = i, j, S_l$	index of video segment C_k. i and j, the first and last frame number of segment C_k, S_l the feature of the key frame of segment C_k
$sim(C_i^Q, C_j^T)$	similarity between the i^{th} query video segment and the j^{th} target video segment. It is measured by features of the keyframes.

statistical analysis on the time length of the video segments obtained by the auto dual-threshold segmenting method (see Fig. 10 in Section 6).

For easy reference, the notations used in the paper are listed in Table 2.

4 Matching SIFT Feature Point Sets Based on SVD

To better represent the local content of video frames, we choose SIFT descriptors to present the video sequences. On the other hand, since the number of SIFT feature points in video sequences is large, it thus exists high computational cost for video copy detection. Fig. 2 shows that matching the SIFT feature points in two frames with the *BBF-Tree* method [35] needs about several seconds. And the computational cost for matching the whole video sequences is high. Thus, many methods, such as *bag of features* (BoFs) [36] or *visual word for video copy detection* [37], *locality sensitive hashing* (LSH) [38], *hierarchical indexing structure* for efficient video retrieval [39], and so on, have been proposed for efficient video search. However, by using these indexing methods, the temporal information of the SIFT feature points in different frames will be lost. Thus, our motivation is to match the two SIFT feature sets in two video frames and make use of the temporal information of video frames.

The SVD method has been widely used in pattern recognition, data compression, signal processing, and other fields [40], [41], [42], [43]. It needs to be noted that, although Delponte et al. [43] also used the SVD method to match SIFT features, the motivation of our method is essentially different from their method. Specifically, the goal of [43] is to use the SVD method to reduce and correct the wrong match between the two points in two SIFT feature sets. The method focused on the "one point-to-one point" correspondence. However, we use the SVD method to measure the similarity between two SIFT feature point sets, and emphasize the similarity of "frame-to-frame."

The matrix SVD theorem can be described as follows:

If $A \in R^{m \times n}$ (based $m > n$), $ran(A) = r$, then there exists two orthogonal matrices U, V, and a diagonal matrix makes the establishment of the following equation:

$$A = U \Lambda V^T, \tag{1}$$

where

$$U = [u_1, u_2, u_3, \ldots, u_m] \in R^{m \times m}, UU^T = I;$$
$$V = [v_1, v_2, v_3, \ldots, v_n] \in R^{n \times n}, VV^T = I;$$
$$\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_r, 0, \ldots, 0] \in R^{m \times n}, \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r.$$

The matrix singular value has the following characteristics:

Characteristic 1: transposition and replacement invariance. That is to say, after transposition or row-column replacement operation of the matrix, its singular value remains unchanged. This characteristic can be directly proved according to the definition of singular value and the characteristic of elementary matrix [44].

Characteristic 2: energy concentricity. The matrix A can be approximately restructured by the first k largest singular values of A. It can be proved that the matrix corresponding to the first k largest singular values of A is the closest to matrix A under the $Frobenius$ norm.

There exist some effective methods on set matching [45], [46], [47], [48]. These methods were mainly applied for image classification and object recognition. We aim to compute the similarity between two SIFT feature point sets that are extracted from keyframes of videos. Accordingly, we propose an SVD-based method to match the SIFT feature point sets. The image SIFT feature contains many local feature points, each feature point is described by a *128*D vector. Then, the SIFT feature points set of one image can be represented as a matrix. It can be known from *Characteristic 1* that the singular value of the image SIFT feature matrix is not related to the position of SIFT feature point. According to *Characteristic 2*, we can use the energy concentricity of the first k largest singular values of the image SIFT feature matrix to greatly reduce the matching cost. In this paper, we use these characteristics of the image SIFT feature matrix and its singular value to match the SIFT feature point sets of images. Suppose that A and B represent two images containing m and n SIFT feature points, respectively, the objective of the algorithm is to match two point sets and compute the similarity between two images. Our proposed method consists of three steps.

Step 1: Matrix $A^{N \times m} = (A_1, A_2, \ldots, A_m)$ represents the feature point set of image A and matrix $B^{N \times n} = (B_1, B_2, \ldots, B_n)$ represents the feature points set of image B, respectively. In another word, $A_i (i = 1, \ldots, m)$ and $B_j (i = 1, \ldots, n)$ represent SIFT feature points in image A and B, respectively. The dimension of A_i and B_j is $N (N = 128)$.

Step 2: A d-dimensional linear subspace of A and B is represented by an orthonormal basis matrix $P_A \in A^{N \times d}$ and $P_B \in B^{N \times d}$, respectively, s.t. $AA^T \cong P_A \Lambda_A P_A^T$ and $BB^T \cong P_B \Lambda_B P_B^T$, where Λ_A and Λ_B are the eigenvalue dialog matrices of the d largest eigenvalues, P_A and P_B the eigenvector matrices of the d largest eigenvalues.

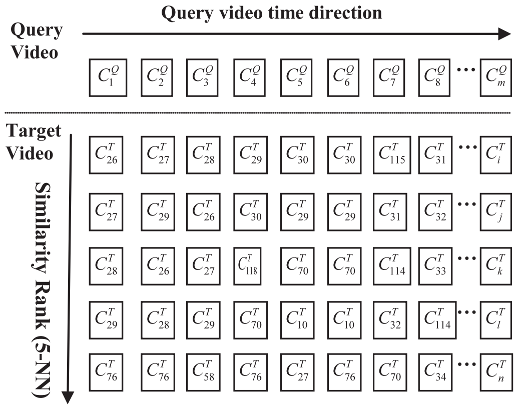

Query video time direction
Query Video
C^Q_1 C^Q_2 C^Q_3 C^Q_4 C^Q_5 C^Q_6 C^Q_7 C^Q_8 ··· C^Q_m
Target Video
C^T_{26} C^T_{27} C^T_{28} C^T_{29} C^T_{30} C^T_{30} C^T_{115} C^T_{31} ··· C^T_i
C^T_{27} C^T_{29} C^T_{26} C^T_{30} C^T_{29} C^T_{29} C^T_{31} C^T_{32} ··· C^T_j
C^T_{28} C^T_{26} C^T_{27} C^T_{118} C^T_{70} C^T_{70} C^T_{114} C^T_{33} ··· C^T_k
C^T_{29} C^T_{28} C^T_{29} C^T_{70} C^T_{10} C^T_{10} C^T_{32} C^T_{114} ··· C^T_l
C^T_{76} C^T_{76} C^T_{58} C^T_{76} C^T_{27} C^T_{76} C^T_{70} C^T_{34} ··· C^T_n
Similarity Rank (5-NN)

Fig. 5. Matching results between query video and target video.

Step 3: Make the singular value decomposition for $P_A^T P_B \in R^{d\times d}$, i.e., $P_A^T P_B = USV^T$, so the similarity between A and B is $sim(A, B) = trace(S)$.

The experimental results in Section 6 show that the proposed SVD-based SIFT feature point set matching method can obtain a better tradeoff between the detection effectiveness and detection time cost.

5 Graph-Based Video Sequence Matching Method for Video Copy Detection

Sections 3 and 4 focus on using the video's spatial characteristics to solve the problem of the efficiency of video copy detection. Meanwhile, video has inherent temporal characteristics that can also be used for video copy detection [19], [16], [22], [17], [25]. In this paper, we propose a new graph-based video sequence matching method that reasonably utilize the video's temporal characteristics. This section will describe the proposed graph-based video sequence matching method for video copy detection. The method is presented as follows:

Step 1: Segment the video frames and extract features of the keyframes. According to the method described in Section 3, we perform the dual-threshold method to segment the video sequences, and then extract SIFT features of the keyframes.

Step 2: Match the query video and target video. Assume that $Q_c = \{C_1^Q, C_2^Q, C_3^Q, \ldots, C_m^Q\}$ and $T_c = \{C_1^T, C_2^T, C_3^T, \ldots, C_n^T\}$ are the segment sets of the query video and target video from Step 1, respectively. For each C_i^Q in the query video, compute the similarity $sim(C_i^Q, C_j^T)$, and return k largest matching results. $K = \alpha n$, where n is the number of segments in the target set, and α is set to 0.05 based on our empirical study.

As an example, the matching results in Fig. 5 can be converted into a matching result graph in Fig. 6. Obviously, the matching result graph is a directed acyclic graph.

Step 3: Generate the matching result graph according to the matching results. In the matching result graph, the vertex M_{ij} represents a match between C_i^Q and C_j^T. To determine whether there exists an edge between two vertexes, two measures are evaluated.

Time direction consistency: For M_{ij} and M_{lm}, if there exists $(i-l)*(j-m) > 0$, then M_{ij} and M_{lm} satisfy the time direction consistency.

Time jump degree: For M_{ij} and M_{lm}, the time jump degree between them is defined as

$$\triangle t_{lm}^{ij} = max(|t_i - t_l|, |t_j - t_m|). \quad (2)$$

If the following two conditions are satisfied, there exists an edge between two vertexes:

1. The two vertexes should satisfy time direction consistency.
2. The time jump degree $\triangle t < \tau$ (τ is a preset threshold based on our empirical study).

Condition 1 indicates that if the query video is a copy deriving from the target video, then the video subsequence temporal order between query video and target video must be consistent, which is reasonable in real application. If Condition 1 is satisfied, Condition 2 is used to constrain the time span of two matching results between the query video and the target video. If the time span exceeds a certain threshold, it is considered that there does not exist certain correlation between the two matching results. This method is similar to the probability model in [49].

Also, as an example, the matching results in Fig. 5 can be converted into a matching result graph in Fig. 6. Obviously, the matching result is a directed acyclic graph. In the graph, in *Case 1*, because of violating the condition of time direction consistency, it does not exist an edge between $M_{2,29}$ and $M_{3,26}$. For *Case* 2, although it meets time direction consistency, the time jump between $M_{4,30}$ and $M_{5,70}$ exceeds the threshold, so it also does not exist an edge between $M_{4,30}$ and $M_{5,70}$. For each vertex of the matching result graph, it may have more than one path or no path. For example, for vertex $M_{1,29}$, $M_{1,76}$, $M_{2,76}$, it has not any path to other

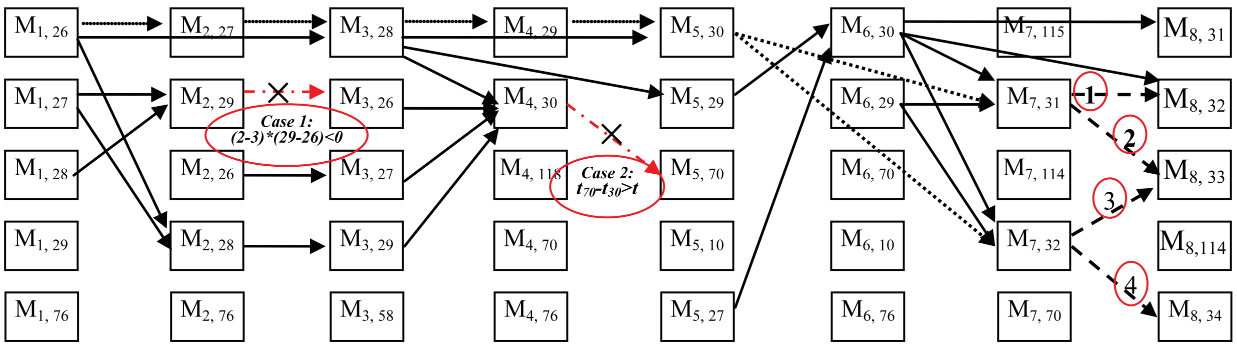

$M_{1,26}$ $M_{2,27}$ $M_{3,28}$ $M_{4,29}$ $M_{5,30}$ $M_{6,30}$ $M_{7,115}$ $M_{8,31}$
$M_{1,27}$ $M_{2,29}$ $M_{3,26}$ $M_{4,30}$ $M_{5,29}$ $M_{6,29}$ $M_{7,31}$ $M_{8,32}$
$M_{1,28}$ $M_{2,26}$ $M_{3,27}$ $M_{4,118}$ $M_{5,70}$ $M_{6,70}$ $M_{7,114}$ $M_{8,33}$
$M_{1,29}$ $M_{2,28}$ $M_{3,29}$ $M_{4,70}$ $M_{5,10}$ $M_{6,10}$ $M_{7,32}$ $M_{8,114}$
$M_{1,76}$ $M_{2,76}$ $M_{3,58}$ $M_{4,76}$ $M_{5,27}$ $M_{6,76}$ $M_{7,70}$ $M_{8,34}$
Case 1: (2-3)*(29-26)<0
Case 2: t_{70}-t_{30}>t
1
2
3
4

Fig. 6. Matching results graph obtained from matching results between query video and target video in Fig. 5.

vertexes (or say the path is the vertex itself). However, for $M_{1,26}$, four paths are available as follows:

1. $M_{1,26} \rightarrow M_{2,27} \rightarrow M_{3,28} \rightarrow M_{4,29} \rightarrow M_{5,30} \rightarrow M_{7,31} \rightarrow M_{8,32}$;
2. $M_{1,26} \rightarrow M_{2,27} \rightarrow M_{3,28} \rightarrow M_{4,29} \rightarrow M_{5,30} \rightarrow M_{7,31} \rightarrow M_{8,33}$;
3. $M_{1,26} \rightarrow M_{2,27} \rightarrow M_{3,28} \rightarrow M_{4,29} \rightarrow M_{5,30} \rightarrow M_{7,32} \rightarrow M_{8,33}$;
4. $M_{1,26} \rightarrow M_{2,27} \rightarrow M_{3,28} \rightarrow M_{4,29} \rightarrow M_{5,30} \rightarrow M_{7,32} \rightarrow M_{8,34}$.

Step 4: Search the longest path in the matching result graph. The problem of searching copy video sequences is now converted into a problem of searching some longest paths in the matching result graph. The dynamic programming method is used in this paper. The method can search the longest path between two arbitrary vertexes in the matching result graph. These longest paths can determine not only the location of the video copies but also the time length of the video copies.

Step 5: Output the result of detection. For each vertex of the matching result graph, it has more than one path or no path.

As in Fig. 6, for the vertexes $M_{1,29}$, $M_{1,76}$, and $M_{2,76}$, they have no path to other vertexes, or only have path to the vertex itself. For $M_{1,26}$, four paths are available. Accordingly, we need to combine these paths that overlap on time. Then, we can get some discrete paths from the matching result graph; it is thus easy to detect more than one copy segments by using this method. For each path, we use (3) to compute the similarity of the video sequences:

$$sim(path) = \frac{\sum_{k=1}^{m} sim_k(M_{ij})}{m} \log(1+m), \qquad (3)$$

where m is the number of vertexes of the path, M_{ij} is the vertex in the path, $sim(M_{ij}) = sim(C_i^Q, C_j^T)$. According to the start point and end point of the path, we can obtain the time stamp of the two copies.

6 Experiments

In this section, we present our experimental results of the proposed graph-based video sequence matching method for video copy detection. Two key techniques will be evaluated in our experiments. The first experiment is to examine the effectiveness of the SIFT feature point set matching method based on SVD. Second, we examine the effectiveness of the proposed graph-based video sequence matching method, compared with the traditional sequence matching method. Furthermore, we study how to determine an optimal time jump threshold.

6.1 Experiment Setting

We use the TRECVID 2008 data set for evaluation [4], which has been widely used in video copy detection. The video data set includes 438 video files, about 200 GB data. The query videos are provided which are generated using the method in [50], [51]. Specifically, each query is constructed by taking a segment of variable length from the test video data set. And the segment is embedded into a video which is not in the test data set. Then, one or more transformations are applied to the entire query video segment. In the obtained query videos, some may contain no test segment and others may contain entire segment of one test video segment.

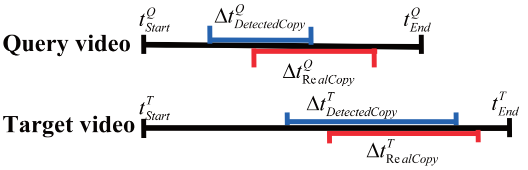


Fig. 7. Copy time stamp accuracy.

In TRECVID 2008 data set, there are 2,010 query videos, which are generated by extracting 201 video segments and applying 10 transformations on each video segment. In the experiment, we evaluate the detection performance for 10 transformations. In our experiment, we use an Intel Core 2 Duo 2.53 GHz PC with 1 G memory.

6.2 Evaluation Criteria

To evaluate the performance of video copy detection, we use three criteria [52] which have been defined by TRECVID organization committee. These criteria are minimal normalized detection cost rate (MinNDCR), copy location accuracy, computational time cost, recall, and precision. Also, to intuitively measure the copy localization accuracy, we define another criterion, copy time stamp accuracy (CTSA), as in (5).

MinNDCR: This measure is a tradeoff between the cost of missing a true positive and the cost of dealing with false positives. NDCR is defined as follows:

$$NDCR = P_{Miss} + \beta R_{FA}, \qquad (4)$$

where P_{Miss} and R_{FA} are the conditional probability of a missed copy and the false alarm rate, respectively [52]. NDCR allows to measure the ability to detect true copies as well as to avoid false alarms. The smaller the NDCR is, the better the detection performance is. In the experiment, NDCRs for different decision thresholds are computed and minimal NDCR, i.e., MinNDCR, is obtained for each transformation.

Copy location accuracy: This measure aims to assess the accuracy of finding the exact extent of the copy in the reference video. It is measured by the F1 measure which is defined as the harmonic mean of precision and recall.

CTSA: This measure aims to show the registration level between detected copy location and the true copy location. CTSA is defined as below and Fig. 7 illustrates the measure

$$\begin{aligned} CTSA = \frac{1}{2}\Bigg(& \frac{|\triangle t^Q_{DetectedCopy} \cap \triangle t^Q_{RealCopy}|}{max\left(|\triangle t^Q_{DetectedCopy}|, |\triangle t^Q_{RealCopy}|\right)} \\ & + \frac{|\triangle t^T_{DetectedCopy} \cap \triangle t^T_{RealCopy}|}{max\left(|\triangle t^T_{DetectedCopy}|, |\triangle t^T_{RealCopy}|\right)}\Bigg), \end{aligned} \qquad (5)$$

where $\triangle t$ indicates the time range and $|\triangle t|$ the time length.

Computational time cost: It is measured by the time used for detection. The lower the computational time, the more efficient the method.

Recall and precision: Meanwhile, we also use the standard *precision* and *recall* measures to compare the effectiveness of our proposed method with the state-of-the-art duplicate detection approaches.

TABLE 3
Some Parameters of Eight Copy Detection Methods

Parameters / Methods	Feature Descriptor	Feature comparison	Sequence Matching	Nb of reference keyframes	Nb of reference features	Nb of feature dimensions
OIS	OIS	L1	Window	78,225	78,225	9(3*3)
FULL-SIFT	SIFT	BBF-Tree	Window	78,225	26,987,625	128
SVD-SIFT	SIFT	SVD	Window	78,225	26,987,625	128
OIS-GRAPH	OIS	L1	Graph	78,225	78,225	9(3*3)
FULL-SIFT-GRAPH	SIFT	BBF-Tree	Graph	78,225	26,987,625	128
SVD-SIFT-GRAPH	SIFT	SVD	Graph	78,225	26,987,625	128
INRIA-LEAR.v.KeysAdves	SIFT	Hamming Embedding	Frame grouping &Geometrical verification	95,411	39,112,273	128
INRIA-LEAR.v.Strict	SIFT	Hamming Embedding	Frame grouping &Geometrical verification	2,080,446	874,697,777	128

[a] **Note:** *INRIA-LEAR.v.KeysAdves* and *INRIA-LEAR.v.Strict* are two runs submitted by *INRIA-LEAR* in *TRECVID 2008* [7], *INRIA-LEAR.v.Strict* can obtain the best performance in all the submitted runs, *INRIA-LEAR.v.KeysAdves* obtain the fourth performance.

6.3 Experimental Results with *TRECVID 2008* Evaluation Metric

In this test, we evaluate eight copy detection methods for 10 copy types (T1-T10) with *TRECVID 2008* evaluation metric. These 10 transformations were briefly described in Section 1.

Table 3 shows the parameters and settings of eight copy detection methods. Specifically, in feature comparison, Hamming embedding and geometrical verification are described in [53]. *BBF-Tree* is the SIFT feature matching algorithm described in [35]. SVD is the proposed SVD-based SIFT feature matching method. Window is the video sequence matching method based on sliding window. In sequence matching, graph is the proposed graph-based video sequence matching method. For the method based on global descriptors (OIS and *OIS-Graph*), the number of features equals the number of keyframes. And for the method based on SIFT local descriptors (*Full-SIFT, Full-SIFT-Graph, SVD-SIFT, SVD-SIFT-Graph, INRIA-LEAR.v.KeysAdves, INRIA-LEAR.v.Strict*), the number of features equals the sum of SIFT descriptors included in all keyframes.

And the experimental results are illustrated in Fig. 8. It can be observed from Figs. 8a and 8b (the experimental data of *INRIA-LEAR.v.KeysAdves* and *INRIA-LEAR.v.Strict* from [7]) that the performance of proposed SVD-SIFT-GRAPH is close to the performance of INRIA-LEAR.v.Strict. And from Table 3, we can find the number of keyframes and number of features of the proposed SVD-SIFT-GRAPH method are much smaller than that of INRIA-LEAR.v.Strict. Meanwhile, the experimental results of Fig. 8 show the SVD-based SIFT feature point set matching method can obtain a good tradeoff between detection effectiveness and time cost.

Regarding the effectiveness, SVD-SIFT preserves the advantage of the good stability and discriminative ability of local descriptors. The proposed SVD-SIFT applies the SVD method to match two SIFT feature point sets extracted from two video frames. According to the transposition and replacement invariance of matrix singular value, it can be known that the singular value of image SIFT feature matrix is not related with the position of SIFT feature point. Then, we do not need to care about the position of the SIFT point matching actually occurs in the image. According to energy concentricity of matrix singular value, the first k largest singular values of image SIFT feature matrix maintains the original characteristic of the matrix well, which can be used to represent the image's characteristic.

Regarding the efficiency, SVD-SIFT overcomes the disadvantages of high computation cost of local descriptors. In the step of online feature similarity matching, the SVD-based method only requires SVD of a $k \times k$-dimensional matrix (the other steps of the SVD-based method can be implemented by offline processing), its matching complexity is $O(k^3)$, where k is the subspace dimension of each set. And the complexity is much lower than that of the matching methods based on single point in nearest neighbor matching, $O(mnN)$, where m, n is the number of SIFT feature points of two comparing sets and N is the dimensionality of SIFT feature points, since $k \ll m, n, N$. In the video copy detection experiments, the averaging matching time of comparing the two video frame sets which contain about 500 frames is 0.03 second for the SVD-SIFT-based method and 5.82 seconds for the kNN method. Comparing with INRIA methods, the SVD-SIFT method does not need "Frame grouping" described in [7], because the SVD-SIFT method directly measures the similarity of "frame-to-frame."

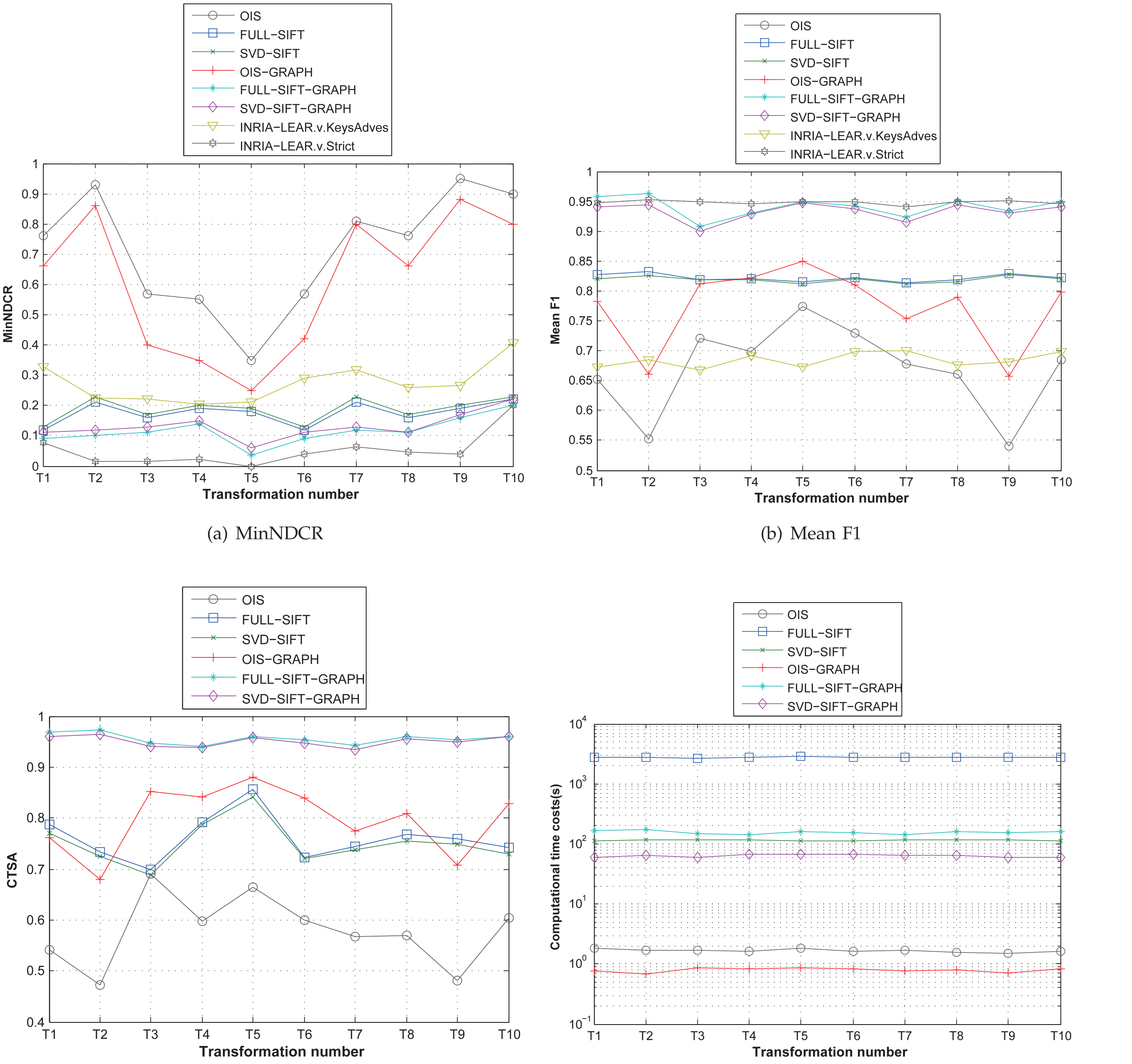
So, it can be observed from the experimental results, the detection methods based on local descriptors perform much better than the methods based on global descriptors. Especially, for "Picture in Picture" copy type (T2 as illustrated in Table 1), the detection methods based on global descriptors can hardly detect the copies. On the other hand, it can be observed that under the same condition, the graph-based methods (OIS-GRAPH, FULL-SIFT- GRAPH, SVD-SIFT-GRAPH) perform better than the window-based methods (OIS, FULL-SIFT, SVD-SIFT) in all four evaluation criteria.

To determine an optimal time jump threshold (as described in Section 5) for a graph-based video sequence matching method, we make a comparative experiment for 10 copy types (T1-T10). The experimental result in Fig. 9 shows the time jump threshold can be determined in a small range of 10 to 29 seconds.

On the other hand, an auto dual-threshold method (as described in Section 3) aims to eliminate redundant video frames and partition the video into segments. Fig. 10 shows that the time length of the majority of video segments lies within a range between 10 to 30 seconds. It also validates the time jump threshold suggested from Fig. 9.

6.4 Comparing with Existing Near-Duplicate Approaches

The purpose of this study is to detect the *copy* videos in large video data set. On the other hand, as we have mentioned in Section 2, *copy* is a subset of *near duplicate*. And many near-duplicate video detection approaches have been proposed recently [1], [13], [14], [15], [23]. In this test, we compare the proposed approach (*SVD-SIFT-GRAPH*) with the state-of-the-art near-duplicate detection approaches, LSF [15], *CONT-CONX* [1], and MRF [23]. Specifically, Huang et al. [15] proposed a sequence of compact signatures called linear smoothing functions (LSFs) for online near-duplicate subsequence detection. LSF transformed a video stream into a 1D video distance trajectory monitoring the continuous changes of consecutive frames with respect to a reference point. LSF used global image feature such as color histogram and texture to represent each video frame. The 1D video distance trajectory is further segmented and represented by a sequence of compact signatures called linear smoothing functions (LSFs). Wu et al. [1] integrated content and context (*CONT-CONX*) to rapidly detect and remove the



(a) MinNDCR

(b) Mean F1

(c) CTSA

(d) Computational Time Cost

Fig. 8. Experimental results.

majority of duplicates of web videos from the top rankings. *CONTENT* adopted the color histogram in HSV color space. *CONTEXT* included *time duration*, *thumbnail images*, and *view counts* contextual cues. Zhou et al. [23] proposed a shot-based interest point selection approach and an adaptive frame selection strategy called furthest point voronoi (FPV) for near-duplicate search.

We compare the effectiveness of our proposed *SVD-SIFT-GRAPH* with the existing methods by the standard *precision* and *recall* measures. Fig. 11 shows the recall for 10 transformations (T1-T10) when precision is 0.8. From Fig. 11, we can see that the proposed *SVD-SIFT-GRAPH* can achieve better performance compared with the other methods in several copy types, i.e., T2, T6, T8, T9, and T10. In rest copy types, *SVD-SIFT-GRAPH* also obtain the close performance to the compared methods. From the view of video signature, *LSF* and *CONT-CONX* use the global signature, and *SVD-SIFT-GRAPH* and *MRF* use the local signature. Although the methods based on global signature can detect copy videos to some extent, its capability to identify the copy videos with complex transformations (such as T2, T8, and T9) is limited. Thus, in effectiveness, the methods based on local signature perform better than the ones based on global signature. On the other hand, with respect to efficiency, the methods based on global signature (*LSF*, *CONT-CONX*) have more advantages over the ones based on local signature (*SVD-SIFT-GRAPH*, *MRF*). Fig. 12 illustrates the average detection time cost for each query over the TRECVID 2008 data set. In summary, the proposed *SVD-SIFT-GRAPH* can obtain a better tradeoff between the effectiveness and the efficiency.

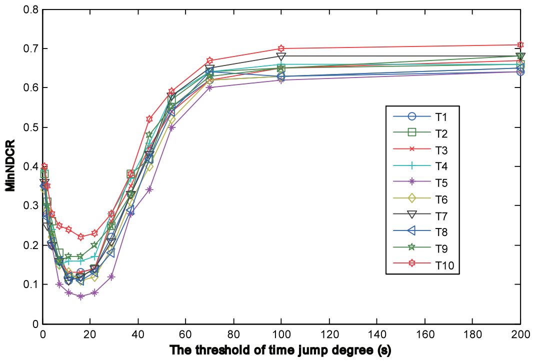


Fig. 9. Time jump threshold selection.

6.5 Advantages of the Graph-Based Video Sequence Matching Method

Since the matching results based on visual features of the video frames do not incorporate the videos' temporal characteristics, the goal of the proposed graph-based video sequence matching method is to refine and order the segment matching results by incorporating the temporal information. The proposed method demonstrate the following advantages:

1. *It can automatically find optimal sequence matching result.* Fig. 13 uses the most challenging "Picture in Picture" copy type to illustrate some advantages of our proposed approach. The dashed box A represents the query video frames (in temporal order). The dashed box B represents query matching results based on a similarity threshold. Specifically, each column in B corresponds to the similarity matching results of each query frame (order by similarity). Then, the graph-based video sequence matching method can automatically find an optimal path to describe the sequence matching result. In Fig. 13, the red path represents the final matching results by using our proposed approach.
2. *It can automatically remove the noise caused by visual feature matching*. In the detection process, there are some noise in the matching results based on the spatial feature of video frame (such as *noise 1, 2, and 3* in Fig. 13). The graph-based matching method can use two constraints (as described in Section 5) to remove the noise.

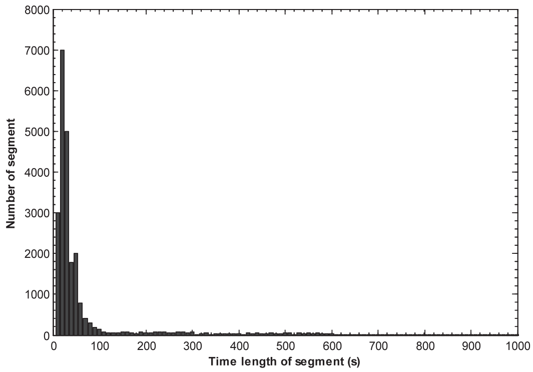


Fig. 10. The time length distribution of the video segments.

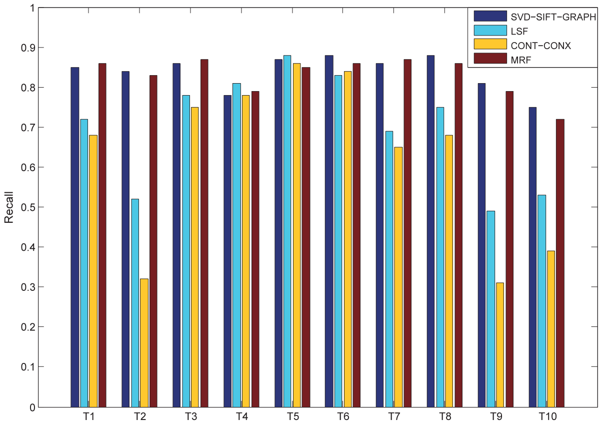


Fig. 11. Recall of 10 different copy types (T1-T10) when precision is 0.8.

3. *It is adaptive to video frame rate change.* The query video's frame rate in Fig. 14 is twice the query video's frame rate in Fig. 13. Experimental results demonstrate that we can still detect the video copies at different frame rate.
4. *It can detect multiple copies existed in the detected video*. In real application, the video copies can appear more than once. For example, Fig. 15 shows that the same advertisement is broadcasted twice at different time slots. Experimental results in Fig. 16 show that our proposed method can detect the multiple copies.

For computational cost, the proposed graph-based method has no complex floating-point calculation and has small computational cost. In real application, the query video is normally short. And even for a full copy with 1 hour length, the number of nodes in the matching result graph is generally smaller than 2,000. Then, by using a PC for computation, our proposed approach can take tens of milliseconds.

Finally, the graph-based video sequence matching method has good scalability to the application based on sequence matching. And the matching methods on visual features will not affect the graph-based sequence matching method. Thus, we can use a variety of methods to obtain the matching results based on visual features, such as the proposed SVD-based method, LSH, BOF, and so on.

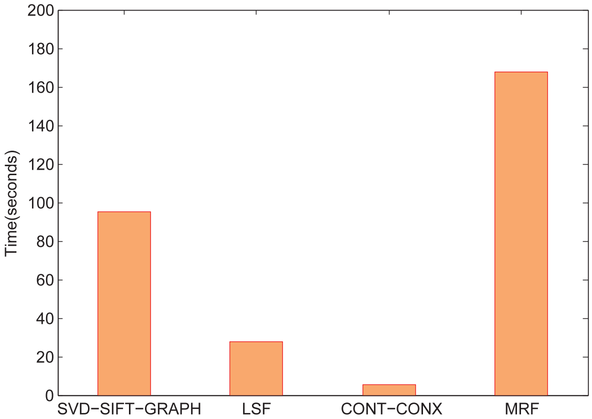


Fig. 12. Computational time comparison.

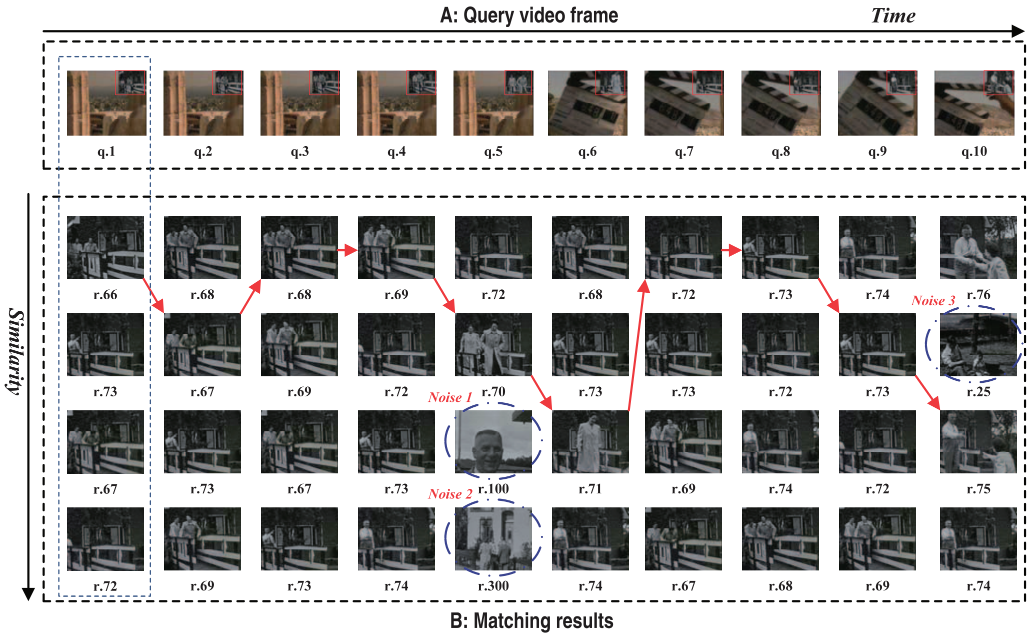


Fig. 13. Graph-based video sequence matching method can automatically find an optimal path to describe the sequence matching result.

7 Conclusions

This paper first analyzes different video copy types and the features used for copy detection. Based on the analysis, we use local feature of SIFT to describe video frames. Since the number of SIFT points extracted from a video is large, so the copy detection using SIFT features has high computational cost. Then, we use a dual-threshold method to eliminate redundant video frames and use the SVD-based method to compute the similarity of two SIFT feature point sets. Experimental results show that this method can obtain a better tradeoff between the detection effectiveness and time cost.

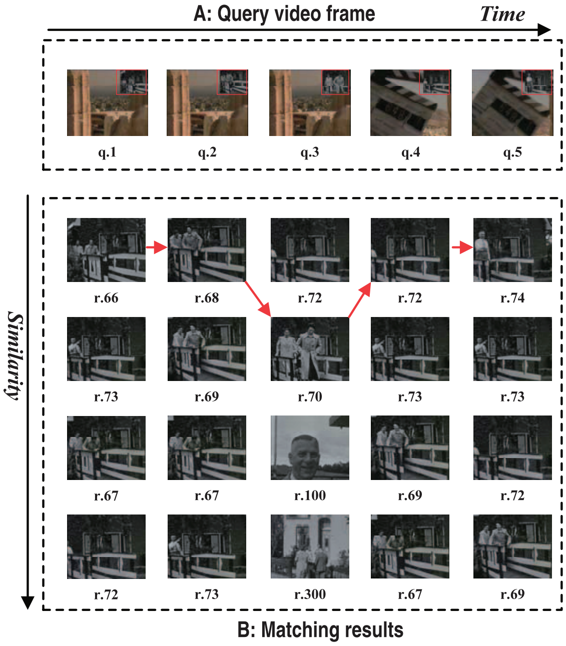


Fig. 14. Graph-based video sequence matching method is adaptive to video frame rate changes.

Furthermore, for video sequence matching, we propose a graph-based video sequence matching method. It skillfully converts the video sequence matching result to a matching result graph. Thus, detecting the copy video becomes finding the longest path in the matching result graph. Experimental results show that the proposed graph-based video sequence matching method has several advantages:

1. The graph-based method can find the best matching sequence in many messy match results, which effectively excludes false “high similarity” noise and compensate the limited description of image low-level visual features.
2. The graph-based method takes fully into account the spatiotemporal characteristic of video sequence, and has high copy location accuracy.
3. The graph-based sequence matching method can automatically detect the discrete paths in the matching result graph. Thus, it can detect more than one copies.
4. Compared to exhaustive search method, graph-based method can also reduce detection time.

Acknowledgments

This work was supported in part by 973 Program (Project No. 2010CB327900), Natural Science Foundation of China

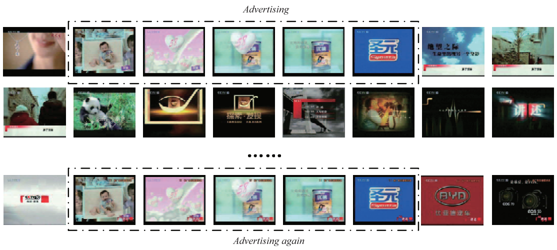


Fig. 15. The same advertisement is broadcasted twice at different time slots.

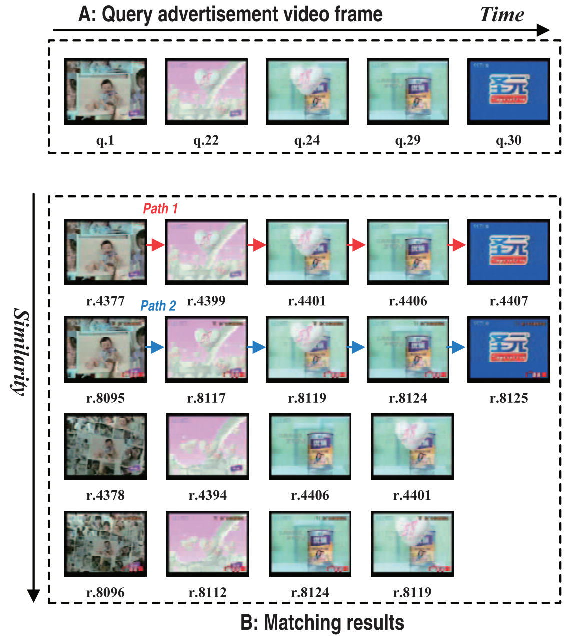


Fig. 16. Graph-based video sequence matching method can detect multiple copies. In the figure, Paths 1 and 2 represent the detected same advertisement which is broadcasted at different time slots.

(No. 61170094), Shanghai Committee of Science and Technology, China Grant No. 11ZR1403400, and Shanghai Leading Academic Discipline Project, Project Number B114.

References

[1] X. Wu, C.-W. Ngo, A. Hauptmann, and H.-K. Tan, "Real-Time Near-Duplicate Elimination for Web Video Search with Content and Context," *IEEE Trans. Multimedia,* vol. 11, no. 2, pp. 196-207, Feb. 2009.

[2] A. Hampapur and R. Bolle, "Comparison of Distance Measures for Video Copy Detection," *Proc. IEEE Int'l Conf. Multimedia and Expo (ICME),* pp. 188-192, 2001.

[3] *TRECVID 2008 Final List of Transformations,* http://www-nlpir.nist.gov/projects/tv2008/active/copy.detection/final.cbcd.video.transformations.pdf, 2008.

[4] *Final CBCD Evaluation Plan TRECVID 2008 (v1.3),* http://www-nlpir.nist.gov/projects/tv2008/Evaluation-cbcd-v1.3.htm, 2008.

[5] O. Kücüktunc, M. Bastan, U. Güdükbay, and Ö. Ulusoy, "Video Copy Detection Using Multiple Visual Cues and MPEG-7 Descriptors," *J. Visual Comm. Image Representation,* vol. 21, pp. 838-849, 2010.

[6] M. Douze, H. Jégou, and C. Schmid, "An Image-Based Approach to Video Copy Detection with Spatio-Temporal Post-Filtering," *IEEE Trans. Multimedia,* vol. 12, no. 4, pp. 257-266, June 2010.

[7] M. Douze, A. Gaidon, H. Jegou, M. Marszalek, and C. Schmid, *TREC Video Retrieval Evaluation Notebook Papers and Slides: INRIA-LEAR's Video Copy Detection System,* http://www-nlpir.nist.gov/projects/tvpubs/tv8.papers/inria-lear.pdf, 2008.

[8] J. Law-To, C. Li, and A. Joly, "Video Copy Detection: A Comparative Study," *Proc. ACM Int'l Conf. Image and Video Retrieval,* pp. 371-378, July 2007.

[9] A. Hampapur, K. Hyun, and R. Bolle, "Comparison of Sequence Matching Techniques for Video Copy Detection," *Proc. SPIE, Storage and Retrieval for Media Databases,* vol. 4676, pp. 194-201, Jan. 2002.

[10] J. Yuan, L.-Y. Duan, Q. Tian, S. Ranganath, and C. Xu, "Fast and Robust Short Video Clip Search for Copy Detection," *Proc. Pacific Rim Conf. Multimedia (PCM),* 2004.

[11] C. Kim and B. Vasudev, "Spatiotemporal Sequence Matching for Efficient Video Copy Detection," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 15, no. 1, pp. 127-132, Jan. 2005.

[12] L. Chen and F.W.M. Stentiford, "Video Sequence Matching Based on Temporal Ordinal Measurement," *Pattern Recognition Letters,* vol. 29, no. 13, pp. 1824-1831, Oct. 2008.

[13] H.T. Shen, X. Zhou, Z. Huang, J. Shao, and X. Zhou, "Uqlips: A Real-Time Near-Duplicate Video Clip Detection System," *Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB),* pp. 1374-1377, 2007.

[14] R. Cheng, Z. Huang, H.T. Shen, and X. Zhou, "Interactive Near-Duplicate Video Retrieval and Detection," *Proc. ACM Int'l Conf. Multimedia,* pp. 1001-1002, 2009.

[15] Z. Huang, H.T. Shen, J. Shao, B. Cui, and X. Zhou, "Practical Online Near-Duplicate Subsequence Detection for Continuous Video Streams," *IEEE Trans. Multimedia,* vol. 12, no. 5, pp. 386-397, Aug. 2010.

[16] I. Laptev and T. Lindeberg, "Space-Time Interest Points," *Proc. Int'l Conf. Computer Vision,* pp. 432-439, 2003.

[17] G. Willems, T. Tuytelaars, and L.V. Gool, "Spatio-Temporal Features for Robust Content-Based Video Copy Detection," *Proc. ACM Int'l Conf. Multimedia Information Retrieval (MIR),* pp. 283-290, 2008.

[18] Y. Ke, R. Sukthankar, and L. Huston, "Efficient Near-Duplicate Detection and Sub-Image Retrieval," *Proc. Ann. ACM Int'l Conf. Multimedia,* pp. 869-876, 2004.

[19] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa, "Robust Voting Algorithm Based on Labels of Behavior for Video Copy Detection," *Proc. ACM Int'l Conf. Multimedia,* pp. 835-844, 2006.

[20] A. Joly, O. Buisson, and C. Frelicot, "Content-Based Copy Retrieval Using Distortion-Based Probabilistic Similarity Search," *IEEE Trans. Multimedia,* vol. 9, no. 2, pp. 293-306, Feb. 2007.

[21] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Proc. Fourth Alvey Vision Conf.,* pp. 147-151, 1988.

[22] S. Satoh, M. Takimoto, and J. Adachi, "Scene Duplicate Detection from Videos Based on Trajectories of Feature Points," *Proc. ACM Int'l Workshop Workshop Multimedia Information Retrieval,* Sept. 2007.

[23] X. Zhou, X. Zhou, L. Chen, A. Bouguettaya, N. Xiao, and J.A. Taylor, "An Efficient Near-Duplicate Video Shot Detection Method Using Shot-Based Interest Points," *IEEE Trans. Multimedia,* vol. 11, no. 5, pp. 879-891, Aug. 2009.

[24] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 10, pp. 1615-1630, Oct. 2005.

[25] H.T. Shen, J. Shao, Z. Huang, and X. Zhou, "Effective and Efficient Query Processing for Video Subsequence Identification," *IEEE Trans. Knowledge and Data Eng.,* vol. 21, no. 3, pp. 321-334, Mar. 2009.

[26] H. Liu, H. Lu, and X. Xue, "SVD-SIFT for Web Near-Duplicate Image Detection," *Proc. IEEE Int'l Conf. Image Processing (ICIP '10),* pp. 1445-1448, 2010.

[27] D. Gibbon, "Automatic Generation of Pictorial Transcripts of Video Programs," *Multimedia Computing and Networking,* vol. 2417, pp. 512-518, 1995.

[28] F. Dufaux, "Key Frame Selection to Represent a Video," *Proc. IEEE Int'l Conf. Image Processing,* vol. 2, pp. 275-278, 2000.

[29] K. Sze, K. Lam, and G. Qiu, "A New Key Frame Representation for Video Segment Retrieval," *IEEE Trans. Circuits and Systems Video Technology,* vol. 15, no. 9, pp. 1148-1155, Sept. 2005.

[30] N. Guil, J.M. González-Linares, J.R. Cózar, and E.L. Zapata, "A Clustering Technique for Video Copy Detection," *Proc. Third Iberian Conf. Pattern Recognition and Image Analysis, Part I,* pp. 452-458, June 2007.

[31] H. Zhang, J. Wu, and S. Smoliar, *System for Automatic Video Segmentation and Key Frame Extraction for Video Sequences Having Both Sharp and Gradual Transitions,* US Patent 5,635,982, June 1997.

[32] H. Lu and Y.-P. Tan, "An Effective Post-Refinement Method for Shot Boundary Detection," *IEEE Trans. Circuits and Systems Video Technology,* vol. 15, no. 11, pp. 1407-1421, Nov. 2005.

[33] O. Kucuktunc, U. Gudukbay, and O. Ulusoya, "Fuzzy Color Histogram-Based Video Segmentation," *Computer Vision and Image Understanding,* vol. 114, no. 1, pp. 125-134, 2010.

[34] B. Furht, S.W. Smoliar, and H.J. Zhang, *Video and Image Processing in Multimedia Systems.* Kluwer Academic Publisher, 1995.

[35] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[36] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1470-1477, 2003.

[37] X. Zhou, L. Chen, A. Bouguettaya, Y. Shu, X. Zhou, and J.A. Taylor, "Adaptive Subspace Symbolization for Content-Based Video Detection," *IEEE Trans. Knowledge and Data Eng.*, vol. 22, no. 10, pp. 1372-1387, Oct. 2010.

[38] P. Indyk and R. Motwani, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality," *Proc. ACM Symp. Theory of Computing*, pp. 604-613, 1998.

[39] H. Lu, B.C. Ooi, H.T. Shen, and X. Xue, "Hierarchical Indexing Structure for Efficient Similarity Search in Video Retrieval," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 11, pp. 1544-1559, Nov. 2006.

[40] V. Klema and A. Laub, "The Singular Value Decomposition: Its Computation and Some Applications," *IEEE Trans. Automatic Control*, vol. AC-25, no. 2, pp. 164-176, Apr. 1980.

[41] Q. Tian, Y. Fainman, and S.H. Lee, "Comparison of Statistical Pattern Recognition Algorithms for Hybrid Processing. II. Eigenvector-Based Algorithms," *J. Optical Soc. of Am.*, vol. 5, no. 10, pp. 1670-1672, 1988.

[42] Z. Hong, "Algebraic Feature Extraction of Image Recognition," *Pattern Recognition*, vol. 24, no. 3, pp. 21l-219, 1991.

[43] E. Delponte, F. Isgrò, F. Odone, and A. Verri, "SVD-Matching Using Sift Features," *Graphical Models*, vol. 68, no. 5, pp. 415-431, 2006.

[44] D.C. Lay, *Linear Algebra and Its Applications.* Univ. of Maryland - College Park, 2003.

[45] L. Wolf and A. Shashua, "Learning over Sets Using Kernel Principal Angles," *J. Machine Learning Research*, vol. 4, no. 10, pp. 913-931, 2003.

[46] T.-K. Kim, O. Arandjelovic, and R. Cipolla, "Learning over Sets Using Boosted Manifold Principal Angles (BoMPA)," *Proc. British Machine Vision Conf.*, pp. 779-788, 2005.

[47] K. Fukui, B. Stenger, and O. Yamaguchi, "A Framework for 3D Object Recognition Using the Kernel Constrained Mutual Subspace Method," *Proc. Asian Conf. Computer Vision*, pp. 315-324, 2006.

[48] T.-K. Kim, J. Kittler, and R. Cipolla, "Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1005-1018, June 2007.

[49] N. Gengembre and S.-A. Berrani, "A Probabilistic Framework for Fusing Frame-Based Searches within a Video Copy Detection System," *Proc. ACM Int'l Conf. Image and Video Retrieval*, July 2008.

[50] A.F. Smeaton, P. Over, and W. Kraaij, "Evaluation Campaigns and TRECVid," *Proc. Eighth ACM Int'l Workshop Multimedia Information Retrieval (MIR '06)*, pp. 321-330, 2006.

[51] *TREC Video Retrieval Evaluation*, http://www-nlpir.nist.gov/projects/t01v/, 2006.

[52] *Final CBCD Evaluation Plan TRECVID 2010 (V2)*, http://www-nlpir.nist.gov/projects/tv2010/Evaluation-cbcd-v1.3.htm#eval, 2010.

[53] H. Jegou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," *Proc. European Conf. Computer Vision*, 2008.

Hong Liu received the BS degree from the Department of Physics, Hunan Normal University, Hunan, China, in 1998, and the MS degree from the Department of Information Management, Shanghai Branch of Nanjing Political Institute, Shanghai, China, in 2004. Since 2007, he has been working toward the PhD degree in the School of Computer Science, Fudan University, Shanghai, China. From 1998 to 2001, he was a lecturer and researcher with Lianyungang Normal Faculty. Since 2004, he has been an engineer with the Information Center of Second Military Medical University, Shanghai, China, where he is currently a senior engineer. His current research interests include multimedia information processing and retrieval, pattern recognition, and machine learning.

Hong Lu (S'01-M'04) received the BEng and MEng degrees in computer science and technology from Xidian University, Xi'an, China, in 1993 and 1998, respectively, and the PhD degree from Nanyang Technological University, Singapore, in 2005. From 1993 to 2000, she was a lecturer and researcher in the School of Computer Science and Technology, Xidian University. From 2000 to 2003, she was a research student in the School of Electrical and Electronic Engineering, Nanyang Technological University. Since 2004, she has been with the School of Computer Science, Fudan University, Shanghai, China, where she is currently an associate professor. Her current research interests include image and video processing, computer vision, machine learning, and pattern recognition. She is a member of the IEEE.

Xiangyang Xue (M'05) received the BS, MS, and PhD degrees in communication engineering from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively. He joined the School of Computer Science, Fudan University, Shanghai, in May 1995. Since 2000, he has been a full professor. His research interests include multimedia information processing and retrieval, pattern recognition, and machine learning, and so on. He has published more than 100 research papers in journals or conference proceedings. He is the associate editor of the *IEEE Transactions on Autonomous Mental Development*, and the *Journal of Computer Research and Development.* He is a member of the IEEE.