Efficient and Effective Video Copy Detection Based on Spatiotemporal Analysis

Chih-Yi Chiu Cheng-Chih Yang* Chu-Song Chen
Institute of Information Science, Academia Sinica, Taiwan
**Department of Computer Science, Columbia University, USA*
*cychiu@iis.sinica.edu.tw, *cy2184@columbia.edu, song@iis.sinica.edu.tw*

Abstract

In this paper, a novel method is presented to detect video copies for a given video query. These copies and the query have identical or near-duplicate content, which might differ in their spatiotemporal structures slightly. To address both the efficient and effective issues, we conduct the bag-of-words model for video feature representation, and apply a coarse-to-fine matching scheme to analyze the video spatiotemporal structure. The proposed method can deal with various kinds of video transformations, such as cropping, zooming, speed change, and subsequence insertion/deletion, which are not well addressed in existing methods. Besides, two indexing methods are employed to speed up the matching process. Experimental results show that the proposed method can behave in an efficient and effective manner.

1. Introduction

Digital videos, which have become ubiquitous over the Internet, can be easily duplicated, edited, and redistributed. From the view of content management, it would be helpful to devise some tools to detect video copies that share the same video source. Example applications include information tracking, document clustering, copy identifying, etc. In this study, we propose a content-based technique to detect coderivative videos by matching their video contents, where no watermark is embedded for identification [4].

The problem of video copy detection is defined as follows. Given a database video D and a query video Q, we have to determine if there exist subsequences $d = \{d_i \mid i = 1, 2, ...\}$ and $q = \{q_j \mid j = 1, 2, ...\}$ so that d and q have identical or near-duplicate content, where d_i and q_j are frames in D and Q, respectively. Figure 1 gives an example for illustration. Suppose that s is the video source shared by d and q. That is, some video transformations are applied on s to produce d and q. Even the video transformations (e.g., brightness enhancement, frame cropping, and speed change) may slightly modify the spatiotemporal structures of videos, the content of s, d and q is perceptually similar. Hence, we can simply measure the content similarity between d and q for copy detection. For simplicity, we assume that d is a copy of q in this study. However, to evaluate the content similarity is still a challenging issue in terms of efficiency and effectiveness. In practice, the video copy detection method may have to process giga- or tera-size data to search for variant video copies. Existing detection methods are either time consuming or capability limited.

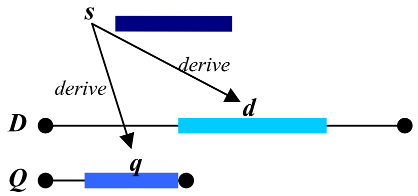


Figure 1. The relation between the source video *s* and the copy videos *d* and *q*.

To address the efficiency and effectiveness issues, in this paper, a novel method for video copy detection is proposed. The major contribution of the proposed method is threefold. First, we conduct the bag-of-words model as the feature representation. The SIFT descriptors, which are robust under minor geometric and local changes, are extracted from video frames as "words" and modeled into a histogram form. Second, we apply a coarse-to-fine matching scheme to analyze the spatiotemporal structures of videos. Our matching scheme can detect videos modified by some transformations such as cropping, zooming, speed change, and subsequence insertion/deletion, which are not well addressed in existing methods. Third, we incorporate two indexing techniques to the coarse and fine matching stages respectively. The detection process can be speeded up substantially.

0-7695-3058-3/07 $25.00 © 2007 IEEE
DOI 10.1109/ISM.2007.38

This paper is organized in the following. Section 2 reviews the related work. Section 3 describes the bag-of-words model, and Section 4 details the spatiotemporal analysis. We provide some experimental results in Section 5. Conclusions and future directions are given in Section 6.

2. Related work

The related work in video copy detection is discussed from two aspects: the feature representation and the matching scheme.

2.1. Feature representation

The ordinal measure [1] is widely used in video copy detection [3][6][7][9][12][20]. For each video frame, it is partitioned into N_x×N_y non-overlapping blocks and the average intensity of each block is computed. We then rank the blocks according to their average intensities. Consequently, the rank order of the blocks is denoted as the ordinal measure of the video frame. The merit of the ordinal signature is the robustness and simple computation. It is more insensitive to several spatial transformations (e.g., brightness adjustment, histogram equalization, and frame resizing) than general low-level image features (e.g., color histograms and texture descriptors). Besides, the representation of ordinal signature is very compact. Only a 9-dimension vector is used for a 3×3-block frame.

In addition to the ordinal measure, Hoad and Zobel [8] proposed a compact video representation composed of color-shift and centroid-based signatures. The color-shift method uses the temporal change of color distributions, while the centroid-based signature computes the spatial movement of the lightest and darkest pixels. Subsequently, each frame is represented as a 2-dimension vector, which is more compact than that of the ordinal signature. Experimental results show the strong robustness of the constituent signatures.

Both the above-mentioned features are extracted based on the whole frame. That is, they take the whole region of a frame to compute a global descriptor as the feature representation. However, if we apply cropping or zooming to the frame, only a part of region is modified and the discriminative power of the local descriptor is degraded. Local descriptors, such as the Harris descriptor [17] or SIFT [15], have shown the robustness under minor geometric and local changes. Joly et al. [10] and Law-To et al. [13] used the Harris detector to detect points of interest as local descriptors. Law-To et al. further connected those local descriptors similar in consecutive frames as a trajectory. Such a representation is feasible to match partial region of the image content.

2.2. Matching scheme

Shot-based video retrieval (SBVR) is a well-known technique for content-based video search. In SBVR, video shot boundaries are automatically detected by finding both abrupt transitions (e.g., cut) and gradual transitions (e.g., fade-in/out, dissolve, wipe). Key frames are extracted from shots as the shot representations. When a query clip is given, the system searches for shots in the database whose key frames are similar (perceptually or semantically) to the query. Cheung and Zakhor [2] described an alternative approach to extract key frames based on the model of Voronoi video similarity. However, the above methods are not suitable for video copy detection; they are limited to the comparisons of whole clips, and can not detect similar subsequences.

Searching by a fixed-length sliding window [6][7][9][11][12] is a very popular method due to its simple and fast computation. Further, Kashino et al. [11] proposed the histogram pruning method to scan videos. It is similar to quick string-matching to avoid many of the unnecessary matches according to the difference between the sliding window similarity and the predefined threshold. However, the fixed-length sliding window can not handle temporal transformations like speed change and cut insertion/deletion. Hoad and Zobel [8] and Chiu et al. [3] used approximated string matching and dynamic time warping, respectively, to deal with some temporal transformations. However, their methods still can not deal with cut insertion/deletion.

Some index techniques are proposed to speed up the detection process. Yuan et al. [20] constructed a multi-resolution *kd*-tree to complete exact *k*-NN query and range query for searching short video segments. Joly et al. [10] proposed an approximate search paradigm called statistical similarity search. A probability model is generated based on the transformation distortion to process the statistical queries, rather than the classical range queries. Law-To et al. [13] proposed a voting function to evaluate similarities between the query frames and database trajectories. They used the local descriptor information for candidate selection, and then employed the trajectory information for spatio-temporal registration. The major problem of these methods is the same to the SBVR: they can not detect similar subsequences. Besides, their matching schemes do not take the temporal transformations into consideration.

3. Bag-of-words modeling

The *bag-of-words* model is widely used in text document analysis. This model simply uses all words in a document as the features. The feature dimension is equal to the number of distinct words in the document database. Since the dimensionality is very high, we usually group words into categories by some preprocesses like stop word removal, stemming, case folding, etc. For multimedia documents, we can extract low-level features from images/videos as "words," and apply vector quantization to cluster these low-level features. In this study, we use the SIFT descriptors as video words and the LBG algorithm for vector quantization.

The SIFT descriptor [15] can provide robust matching across affine distortion, noise addition, and illumination change. It is widely applied for object recognition [5][14][18], and shows the best evaluation performance compared with other local descriptors [16]. However, the SIFT descriptor involves a high dimensional feature set (128 dimensions for a point of interest). It will generate a tremendous size of feature data for videos of several hours, if each frame contains dozens of SIFT descriptors. In this paper, we model SIFT descriptors into a bag-of-words representation to not only reduce the data size of the video representation, but preserve the robustness of the SIFT descriptor.

Let D be the video database and d_i be the i-th frame in D. For each frame d_i, we extract the SIFT descriptors by finding local extrema of the DOG scale space that are above a given threshold $\theta_{\mathbf{SD}}$. The use of $\theta_{\mathbf{SD}}$ is to control the number of SIFT descriptors extracted from a frame. A higher $\theta_{\mathbf{SD}}$ decreases the number of SIFT descriptors. Since these SIFT descriptors are not uniformly distributed in the feature space, the feature vector density should be considered in the classification process for better discrimination. In this study, we use the Linde-Buzo-Gray (LBG) algorithm to create a quantization codebook. The LBG algorithm enables the number of codes assigned in the feature space to reflect the density of feature vectors. Then, in the quantization stage, every descriptor is classified into the nearest code in the codebook.

We generate a SIFT histogram for each database frame. The histogram is a frequency distribution of the SIFT descriptors extracted in that frame. Denote dh_i as the SIFT histogram of the frame sequence d_i:

$$dh_i = \{dh_{i,1}, dh_{i,2}, \ldots , dh_{i,l}, \ldots , dh_{i,L}\}. \tag{1}$$

L is the number of histogram bins, i.e., the codebook size, and $dh_{i,l}$ is the number of SIFT descriptors classified into the l-th quantization code observed in frame d_i. Then we normalize dh_i so that the summarization of all histogram bins is 1:

$$dh_{i,l} = \frac{dh_{i,l}}{\sum_{l=1}^{L} dh_{i,l}}. \tag{2}$$

Now $dh_{i,l} \in [0, 1]$ is served as the weight of the l-th bin of frame d_i.

4. Spatiotemporal analysis

Given a query video Q, we want to find its video copies from the database D. The matching between Q and D is performed through a coarse-to-fine scheme. First, at the coarse matching stage, we employ a sliding window to evaluate the similarity between Q and the windowed sequence in D. The histogram pruning algorithm is applied to ignore unnecessary sequence matching from D. Those windowed sequences with high similarities are selected as the candidate sequences. At the fine matching stage, we analyze the similarity between each frame in the candidate sequence and Q. An invert indexing method is used to quickly build up a frame-pair similarity matrix. Our method is based on the coarse-to-fine scale analysis considering the spatial and temporal dimensions. It addresses both the efficiency and effectiveness issues well.

4.1. Coarse matching: window similarity

Let $Q = \{q_k \mid k = 1, 2, \ldots , n\}$ be an n-frame sequence. For each query frame q_k, we extract its SIFT descriptors and generate the corresponding SIFT histogram qh_k, as we depicted in Section 3. We employ a window of length n to slide over database D. Inside the window the similarity between the two video sequences $C_j = \{d_j , d_{j+1}, \ldots , d_{j+n-1}\}$ and Q is defined as follows:

$$\mathbf{WS}(C_j, Q) = \frac{1}{n}\sum_{l=1}^{L} \min\left(\sum_{k=1}^{n} dh_{j+k-1,l}\, , \sum_{k=1}^{n} qh_{k,l}\right). \tag{3}$$

The summation term inside the minimum function is to count the total weight of the l-th histogram bins from all frames. Then $\mathbf{WS}(C_i, Q) \in [0, 1]$ serves as the *window similarity* at the coarse matching stage.

Speed-up

Even though the calculation of the window similarity is simple, searching by the window sliding over the database frame-by-frame is impractical in terms of computing time. We employ a histogram pruning method proposed by Kashino et al. [11] to accelerate the sliding window search. The basic idea is to skip unnecessary database frames for matching according to the difference between the window similarity and the predefined threshold. As the window shifting forward to the next frame, the maximum increment from $\mathbf{WS}(C_i, Q)$ to $\mathbf{WS}(C_{i+1}, Q)$ is $1/n$ for all index i. Therefore, we can derive the number of frames to be skipped for the window, denoted as w:

$$w = \begin{cases} \lfloor n(\theta_{\mathbf{WS}} - \mathbf{WS}) \rfloor + 1, & \text{if } \mathbf{WS} < \theta_{\mathbf{WS}}, \\ 1, & \text{otherwise,} \end{cases} \tag{4}$$

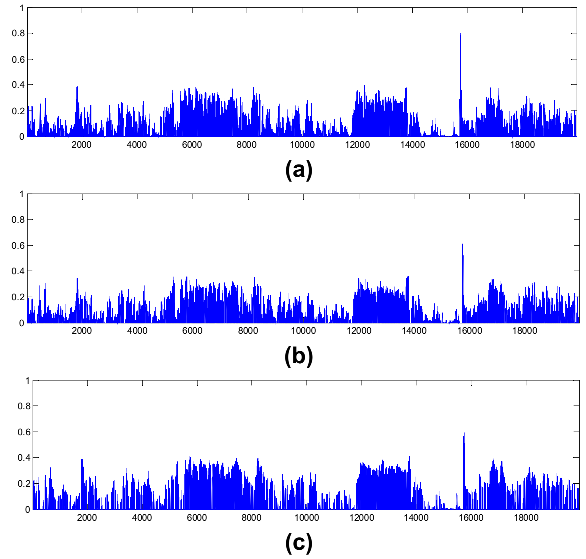
where $\lfloor x \rfloor$ rounds x to the nearest integers greater than or equal to x and $\theta_{\mathbf{WS}}$ is the detection threshold for the window similarity matching. It is guaranteed that no sequence whose window similarity is greater than $\theta_{\mathbf{WS}}$ is missed, even if we skip w frames given by Eq. (4).

The determination of $\theta_{\mathbf{WS}}$ becomes the tradeoff between the matching efficiency and effectiveness. A higher value of $\theta_{\mathbf{WS}}$ can skip more number of frames according to Eq. (4), but degrade the accuracy for identifying video copies. In case to detect video copies with partial content modified, we have to consider the ratio of the partial content to its original source. For example, to detect the video copy whose frames are cropped a half of the original region, $\theta_{\mathbf{WS}}$ should be set to less than 0.5.

Figure 2 plots a part of the window similarities for some coderivative query videos: enhance 20% brightness (Figure 2a), crop a half of frame region (Figure 2b), and speed 0.5× (Figure 2c). The X-axis is the frame index in the database, and the Y-axis is the window similarity. The similar distributions of these window similarities manifest the robustness of the bag-of-words model under these spatial and temporal transformations. A sequence C_i is denoted as the candidate for further fine matching, if its window similarities $\mathbf{WS}$ are local maxima and greater than $\theta_{\mathbf{WS}}$.

4.2. Fine matching: pairwise similarity

Since the coarse matching is based on the histogram representation of feature vectors over the window, it does not reflect the time relationship of feature vectors. This will increase false positives at the coarse matching stage. That is, some candidates found in Section 4.1 may not be the copies of the query. Therefore, we propose a novel method to determine the similarity between the query and the candidate by fine matching.



(a)

(b)

(c)

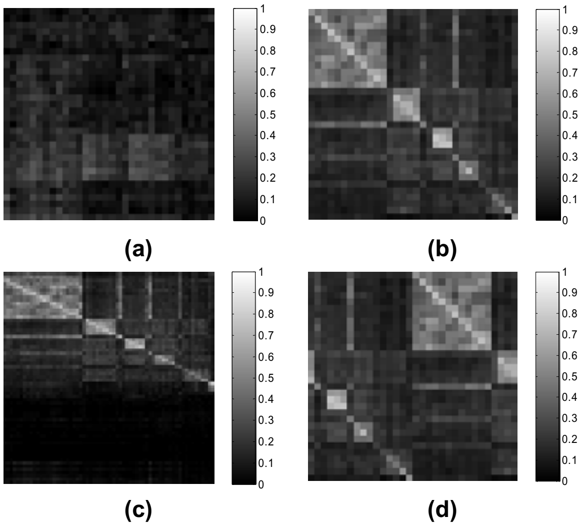
Figure 2. The window similarity for coderivative query videos: **(a)** enhance 20% brightness; **(b)** crop a half of frame region; **(c)** speed 0.5×.

For easy exposition, the candidate C is denoted as $C = \{d'_1, d'_2, \ldots d'_j, \ldots, d'_n\}$ and dh'_j is the SIFT histogram of d'_j. Given the candidate C and query Q, we define the *pairwise similarity*, which is an $n \times n$ matrix representing all-pairs frame similarities:

$$\mathbf{PS}(C, Q) = \{ m_{jk} \mid m_{jk} = \sum_{l=1}^{L} \min(dh'_{j,l}, qh_{k,l}) \}, \tag{5}$$

for j and $k = 1, 2, \ldots, n$. The (j, k)-th element $m_{jk} \in [0, 1]$ stores the frame similarity between the j-th candidate frame and the k-th query frame. The frame similarity is calculated by the SIFT histogram intersection of the two frames.

The pairwise similarity $\mathbf{PS}$ can be visualized by plotting the similarity values stored in the matrix. Let us take the example in Figure 3 for illustration. Figure 3 shows four pairwise similarities, where the X-axis and the Y-axis indicate the query frame index and the database frame index, respectively. From the intensity distribution of the pairwise similarity, we can investigate the time relationship between the candidate and the query. For example, given a query Q, we find two candidates C_1 and C_2 from the database. We plot the pairwise similarities $\mathbf{PS}(C_1, Q)$ and $\mathbf{PS}(C_2, Q)$ in Figures 3a and 3b, respectively. Actually, C_2 is a copy of Q, but C_1 is not. It is clear to see a high intensity distribution along the main diagonal of the matrix in Figure 3b, while the intensity distribution is much scattered in Figure 3a. Such the slant line-wise distribution shows the continuity of high frame similarities in a certain time period. Another two examples shown in Figures 3c and 3d are the matrices of **PS**(C_3, Q) and **PS**(C_4, Q), respectively. Here C_3 is a slow motion version of C_2 by speeding 0.5×, and C_4 is a cut editing version of C_2 by swapping the first-half part and the second-half part. Both two matrices also manifest the slant line-wise distributions.



(a)

(b)

(c)

(d)

Figure 3. The pairwise similarity matrices. **(a)** **PS**(C_1, Q); **(b)** **PS**(C_2, Q); **(c)** **PS**(C_3, Q); **(d)** **PS**(C_4, Q). Except for C_1, all other sequences are copies of Q.

Based on the above observation, we deal with the fine matching process for C and Q by detecting slant line-wise distributions on **PS**. Note that there may be multiple slant line-wise distributions with various directions and positions on **PS**. The detection should be able to find all of these distributions. Therefore, we accomplish the detection by the use of the *Hough transform* [19]. The Hough transform can detect multiple objects in an image, even these objects are fractional. Basically, the Hough transform is a voting algorithm for an analytic equation of an object contour. In this study, the objects to be detected on **PS** are slanted lines.

First, we apply an edge detection method on **PS**. The Sobel edge detection method is employed in our work. The Canny method is not used because its noise suppression and hysteresis thresholding will impede the analysis of intensity distributions. At each point on **PS**, we calculate its edge magnitude and direction.

Next, we build a two-dimension accumulator matrix M to detect the existence of a line $s = x\cos\rho + y\sin\rho$, where the parameters s and ρ are the two coefficients of M. Points in **PS** can be considered as potential line points if they satisfy the following two conditions: (1) the edge magnitudes exceed a given threshold $\theta_{\mathbf{EM}}$, and (2) the edge directions are within a certain range $\Theta_{\mathbf{ED}}$. These potential line points are counted to accumulate the numbers in their corresponding elements in M. While $\theta_{\mathbf{EM}}$ is usually decided empirically, $\Theta_{\mathbf{ED}}$ can be determined by the video speed range [ρ_L, ρ_H], where ρ_L and ρ_H are the lowest and highest video speed boundaries to be detected, respectively:

$$\Theta_{\mathbf{ED}} = [\ \arctan(\rho_L), \arctan(\rho_H)\ ]. \qquad (6)$$

This is because the speed range will bound the slant line-wise distributions in **PS** (see the query video with 0.5× speed in Figure 3c). For example, to detect video copies with the speed range from 0.5× to 2×, we can use $\Theta_{\mathbf{ED}} = [26.57° - \varepsilon\ , 63.43° + \varepsilon]$, where ε is a small tolerance value.

Finally, we smooth M by a Gaussian filter, and find the local maxima in M. The local maxima greater than a given threshold $\theta_{\mathbf{LM}}$ are regarded as the detected lines, corresponding to the diagonal-wise distributions on **PS**. $\theta_{\mathbf{LM}}$ can be determined by a proportion to the number of query frames. Figures 4 and 5 give two examples in association with Figure 3c and 3d for the slant line-wise detection, respectively: Part (a) are the pairwise similarities; Part (b) are the edges satisfying the threshold conditions of $\theta_{\mathbf{EM}}$ and $\Theta_{\mathbf{ED}}$; Part (c) are the Hough spaces of the accumulator matrix M, where the rectangles label the local maxima on M; Part (d) show the detection results as the dash lines. The two examples show the proposed detection method is able to deal with multiple diagonal-wise distributions with various directions and positions.

Speed-up

To construct the pairwise similarity **PS**, a naïve way is computing all-pairs frame similarities according to Eq. (5), and the time complexity is $L{\cdot}n^2$. Actually, **PS** is a sparse matrix that can be constructed by counting only a few frame pairs. Here we introduce an invert indexing method to reduce the computation cost for the construction of **PS**.

Let T be the invert table containing L cells, in which the frame indices will be stored. For each candidate frame d'_j, if the l-th bin ($l \in [1, L]$) of its SIFT histogram $dh'_{j,l} > 0$, d'_j is inserted to the l-th cell of T, denoted as $T(l)$. We do the same step to insert each query frame q_k into the associated cells of T. For each frame pair (d'_j, q_k) found in the l-th cell of T, we increment the (j, k)-th element m_{jk} in **PS** by $\min(dh'_{j,l}\ , qh_{k,l})$.

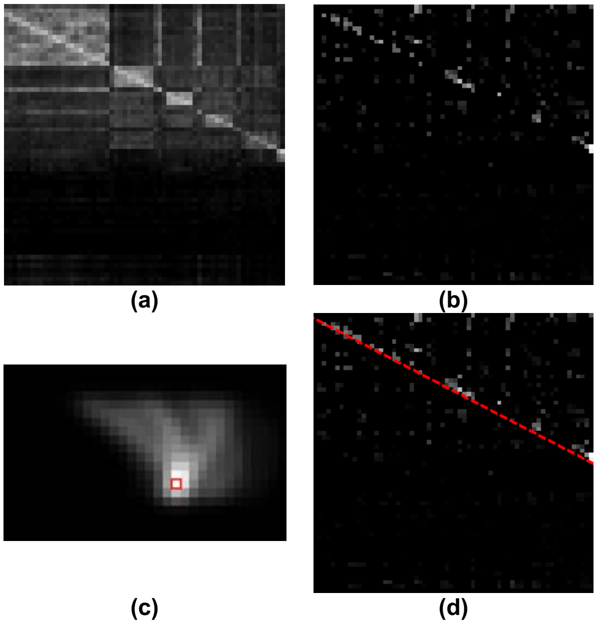


Figure 4. The slant line-wise detection for **PS**(C_3, Q). **(a)** the pairwise similarity matrix; **(b)** the edge map; **(c)** the Hough space; **(d)** the detected line.

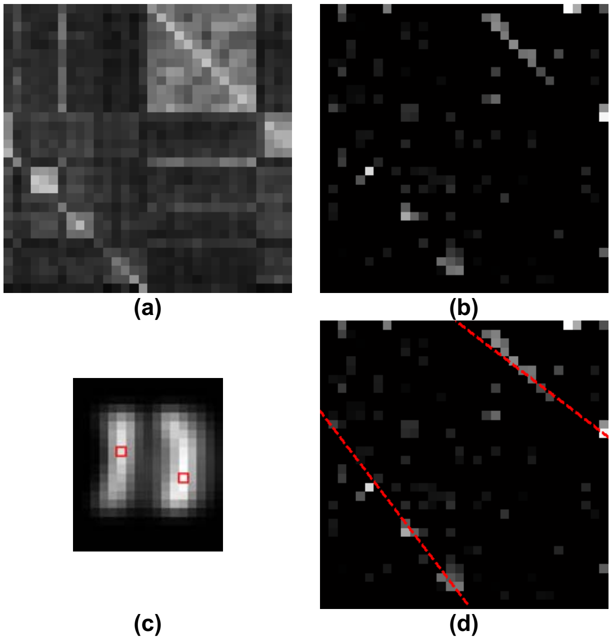


Figure 5. The slant line-wise detection for **PS**(C_4, Q). **(a)** the pairwise similarity matrix; **(b)** the edge map; **(c)** the Hough space; **(d)** the detected lines.

While there are few frame pairs intersecting with some histogram bins, the use of the invert table can reduce a lot of computation cost. The computation cost is $2 \cdot L \cdot n + NFP$, where NFP is the total number of frame pairs found in T. The former cost is the time for inserting the candidate and query frames in T, and the latter cost is the time for counting all frame pairs. More discussion about the computation cost are given in the experiment section.

5. Experimental results

A video database containing 6.1 hours is collected from the Open Video Project and the MPEG-7 collection. Its contents include sports, news, documentaries, landscapes, and so on. We transform these video data to the following format: MPEG-1, 320×240 pixels, and 30 frames per second (fps). With the database, we design several experiments to evaluate the efficiency and effectiveness of the proposed method.

5.1. Environment configuration

Since a continuous video sequence contains many identical or near-duplicate frames, it is not necessary to use every frame in the sequence for matching, in terms of the efficiency. Here we select every 15 frames in the database as the key frames. The key frame sequence in association with 2 fps is used for matching.

We take the brightness value (i.e., the Y component of the YCbCr color space) of every frame pixel for image processing. To extract the SIFT descriptors, we set the threshold $\theta_{\mathbf{SD}} = 0.05$, and the average number of SIFT descriptors in a frame is 18.14. In the LBG algorithm, a quantization codebook is created with the size 1024. Therefore, the number of bins of the SIFT histogram L = 1024. The other threshold parameters are configured as follows. At the coarse matching stage, $\theta_{\mathbf{WS}} = 0.4$. At the fine matching stage, $\theta_{\mathbf{EM}} = G/5$ (G is the maximum of the edge magnitudes), $\Theta_{\mathbf{ED}} = [20°, 70°]$, and $\theta_{\mathbf{LM}} = n/4$ (recall that n is the number of the query frames).

From the database, we randomly select 31 segments, each of which is of thirty seconds long. Then every segment is used to generate eleven video copies by the following modifications: (1) brightness enhancement 20%, (2) histogram equalization, (3) random noise addition 10%, (4) compression with quality 50% (by IndeoR 5.10 compressor), (5) frame rate change to 15 fps, (6) frame resolution change to 240×180 pixels, (7) cropping a half of frame region (replaced with black regions), (8) zoom in 1.33×, (9) speed 0.5×, (10) speed 2×, and (11) subsequence reordering (by swapping the first-half part and the second-half part). These videos serve as the queries, each of which is submitted to detect the corresponding segments in the database. Note that before the detection process, we have to determine the frame rate of the query, which is available in the video file header. Then the query video is re-sampled so that its frame rate is synchronized with that of the database video.

5.2. Detection effectiveness

We consider a detection result as correct if there is any overlap with the region from which the query was extracted. The metrics of *precision*, *recall* and F_1-*measure* are used for the accuracy evaluation:

$$\textit{Precision} = \mathrm{TP} / (\mathrm{TP} + \mathrm{FP}),$$
$$\textit{Recall} = \mathrm{TP} / (\mathrm{TP} + \mathrm{FN}),$$
$$F_1 = (2 \times \textit{Recall} \times \textit{Precison}) / (\textit{Recall} + \textit{Precision}). \quad (8)$$

Positives (TP) are positive examples correctly labeled as positives. False Positives (FP) refer to negative examples incorrectly labeled as positive. False Negatives (FN) refer to positive examples incorrectly labeled as negative.

For the sake of comparison, we choose the ordinal-based method as the baseline, and implement it in a common sense manner. A 9-dimension ordinal signature is used as a frame feature, and a fixed-length sliding window is applied to scan frames one-by-one over the database. The similarity judgment is to measure the Euclidean distances of ordinal signatures of frames. Here we consider a sequence a copy if its average Euclidean distance to the source is less than 7.

Table 1 shows the detection accuracy of the baseline and proposed methods, with respect to eleven spatial and temporal transformations. The baseline method performs very well for the whole-region spatial transformations (case (1)-(6)), while degrades severely for the partial-region spatial and temporal transformations (case (7)-(11)). This inefficiency mainly comes from two factors. One is the ordinal signature representation, which is a global descriptor for a video frame, will cause a great discrepancy if only parts of the frame region are modified. The other factor is only applying the sliding window is not enough to deal with the temporal transformations. A further analysis is required after scanning over the database.

Compared with the baseline method, the proposed method obtains the better performance for the partial-region spatial and temporal transformations. However, we find the proposed method behaves much worse in case (2) histogram equalization. This is because histogram equalization tries to equalize the intensity distribution by adjusting each pixel individually. Therefore, the gradient distribution of a SIFT descriptor at the same location may alter a lot. We also note in cases (6)-(8), the modifications of the frame scale usually influences the number of SIFT descriptors to be extracted, and thus impairs the similarity computation in our matching scheme. This might be improved by amending the normalization of the SIFT histogram in Eq. (2).

5.3. Detection efficiency

Due to the space limitation, we only discuss the efficiency of online matching. Given a thirty-second query to search over the 6.1 hours video database, the proposed method has to proceed through the coarse-to-fine matching. At the coarse matching stage, the use of the histogram pruning algorithm reduces the matching calculations to 2.51% averagely, compared with scanning frames one-by-one. At the fine matching stage, the use of the invert indexing method reduces the matching calculations to 0.11% averagely, compared with all-pairs frame similarity computation. The proposed method takes 781 ms to complete the matching, while the baseline method takes 62 ms, measured in a computer with 2.6GHz CPU and 4GB ram. Note the feature dimensions of a frame are 1024 and 9 for the proposed and baseline methods, respectively. Although the feature dimension of our method is 114 times larger than that of the baseline method, the time spent in our method is only 13 times greater than that in the baseline method. This observation reveals the proposed speedup method actually improve the detection efficiency of our method, which is much superior to the baseline method in detecting partial-region spatial and temporal transformations.

6. Conclusions and future work

In this paper, a novel video copy detection method is presented. We conduct the bag-of-words model as the feature representation, and a coarse-to-fine matching scheme for spatiotemporal analysis. The proposed method can detect video copies modified by partial-region spatial and temporal transformations, which are not well addressed in existing methods. Besides, we incorporated the histogram pruning and invert indexing methods to speed up the matching process. The future work will focus on the improvement of the detection accuracy, especially the recall rate.

7. Acknowledgements

This research was supported in part by National Digital Archives Program (NDAP, Taiwan) sponsored by the National Science Council of Taiwan under NSC Grants: NSC96-2422-H-001-001-.

Table 1. Detection effectiveness.

	Baseline method			**Proposed method**		
	Precision	*Recall*	*F₁*	*Precision*	*Recall*	*F₁*
(1) Brightness enhancement	0.9688	0.9688	0.9688	1.0000	0.9375	0.9677
(2) Histogram equalization	1.0000	0.8750	0.9333	1.0000	0.2813	0.4391
(3) Random noise	0.9688	0.9688	0.9688	1.0000	0.8438	0.9153
(4) Compression	0.9688	0.9688	0.9688	1.0000	0.8750	0.9333
(5) 15 fps	0.9394	0.9688	0.9539	1.0000	0.9375	0.9677
(6) Frame resizing	0.9688	0.9688	0.9688	1.0000	0.5938	0.7451
(7) Cropping	0.0000	0.0000	0.0000	1.0000	0.5000	0.6667
(8) Zooming in	0.2500	0.1563	0.1923	1.0000	0.3750	0.5455
(9) 0.5× speed	1.0000	0.0938	0.1715	1.0000	0.7813	0.8772
(10) 2× speed	0.2340	0.3438	0.2785	1.0000	0.7813	0.8772
(11) Cut editing	1.0000	0.4688	0.6383	1.0000	0.8438	0.9153

8. References

[1] D. N. Bhat and S. K. Nayar, "Ordinal measures for image correspondence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 4, pp. 415-423, 1998.

[2] S. C. Cheung and A. Zakhor, "Efficient video similarity measurement with video signature," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 1, pp. 59-74, 2003.

[3] C. Y. Chiu, C. S. Chen, and L. F. Chien, "A framework for handling spatiotemporal variations in video copy detection," *IEEE Transactions on Circuits and Systems for Video Technology*, accepted.

[4] I. J. Cox, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, Vol. 6, No. 12, pp. 1673-1687, 1997.

[5] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka, "Visual categorization with bags of keypoints," *International Workshop on Statistical Learning in Computer Vision*, 2004.

[6] A. Hampapur and R. M. Bolle, "Comparison of distance measures for video copy detection," *IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, Aug. 22-25, 2001.

[7] A. Hampapur, K.-H. Hyun, and R. M. Bolle, "Comparison of sequence matching techniques for video copy detection," *SPIE Conference on Storage and Retrieval for Media Databases*, 2002.

[8] T. C. Hoad and J. Zobel, "Detection of video sequence using compact signatures," *ACM Transactions on Information System*, Vol. 24, No. 1, pp. 1-50, 2006.

[9] X. S. Hua, X. Chen, and H. J. Zhang, "Robust video signature based on ordinal measure," *IEEE International Conference on Image Processing*, Singapore, Oct. 24-27, 2004.

[10] A. Joly, C. Frélicot, and O. Buisson, "Content-based video copy detection in large databases: a local fingerprints statistical similarity search approach," *IEEE International Conference on Image Processing*, Genova, Sep. 11-14, 2005.

[11] K. Kashino, T. Kurozumi, and H. Murase, "A quick search method for audio and video signals based on histogram pruning, " *IEEE Transactions on Multimedia*, Vol. 5, No. 3, pp. 348-357, 2003.

[12] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, No. 1, pp. 127-132, 2005.

[13] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa, "Robust voting algorithm based on labels of behavior for video copy detection," *ACM International Conference on Multimedia*, 2006.

[14] F. Li, and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[16] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp. 1615-1630, 2005.

[17] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 5, pp. 530-535, 1997.

[18] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering objects and their location in images," *IEEE International Conference on Computer Vision*, 2005.

[19] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, Brooks/Cole Publishing, 1999.

[20] J. Yuan, L. Y. Duan, Q. Tian, and C. Xu, "Fast and robust search short video clip search using an index structure," *ACM International Workshop on Multimedia Information Retrieval*, New York, USA, Oct. 15-16, 2004.