A NOVEL VIDEO MATCHING FRAMEWORK FOR COPY DETECTION [†]

Chih-Yi Chiu (邱志義), *Hsin-Min Wang* (王新民)
Institute of Information Science, Academia Sinica

ABSTRACT

The fixed-length sliding window approach is widely used in video search applications. This approach computes the similarity between a given query and each windowed sequence, and determines whether their similarity exceeds a predefined threshold. However, using the fixed window length and the predefined threshold is inflexible for detecting video copies that might have been derived by applying a variety of video transformations. To address this problem, we present a novel matching scheme, which estimates an appropriate window length and threshold for each matching by inferring the context of the query and the current window. Moreover, to make the matching more efficient, we adopt a coarse-to-fine strategy which utilizes the compact min-hashing signature for fast filtering and robust spatio-temporal analysis for detailed verification. Substantial experiments show that the proposed video matching framework yields the robust accuracy and efficient computation for handling every type of video transformation.

1. INTRODUCTION

The rapid development of multimedia technologies in recent years has spurred an enormous growth in the number of digital videos posted on the Internet. The volume of video data has led to the requirement for efficient and effective techniques of video indexing and retrieval. In particular, since digital videos can be easily duplicated, edited, and disseminated, video copying has become an increasingly serious problem. For example, news channels would like to track particular videos with respect to royalty payments and copyright infringement issues; video blog operators might wish to identify near-duplicate videos for removal from databases or for aggregation in search results. Fig. 1 shows the result of inputting the phrase "UFO Apollo 11" to YouTube, where most of the retrieved video clips are identical or near-duplicate. In such cases, it would be helpful if effective and efficient video copy detection techniques could be used to protect the source content and manage copies of it.

There are two general techniques for video copy detection: digital watermarking, which embeds hidden information in videos; and *Content-Based Copy Detection* (CBCD), which employs perceptual features of the video content as a unique signature to distinguish one video from another. Because CBCD does not destroy or damage video content, it has generated a great deal of research interest recently. In this paper, we propose a novel content-based method for detecting video copies.

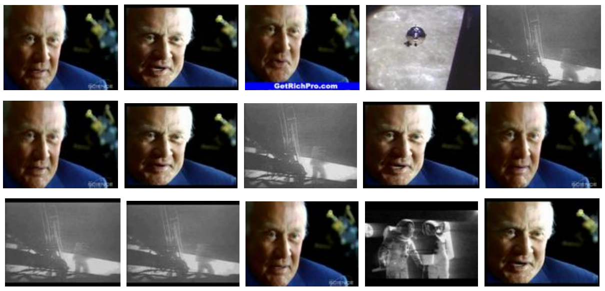


Fig. 1. The first fifteen search results retrieved by inputting the phrase "UFO Apollo 11" to YouTube.

1.1. Related Work

The development of CBCD evolved from Content-Based Video Retrieval (CBVR) research. The general framework of CBVR methods is based on the analysis and retrieval of video shots [1]. Shot boundaries are automatically detected by finding transitions (e.g., cut and fading) in video sequences. Each shot is then summarized by several key frames or feature clusters. For a given query, the framework searches the dataset for similar shots by matching their key frames or feature clusters.

However, the ultimate goal of CBVR is to find videos that are "semantically similar" to the query, while that of CBCD is to detect "perceptually similar" videos. Moreover, existing CBVR methods do not consider video transformations that might be applied to the source video. In this paper, we discuss three categories of widely used video transformations:

(1) Preserved frame region transformation. This category includes brightness enhancement, compression, noise addition, and frame resolution change, which modify the frame content while preserving the whole frame region, as shown in Fig. 2(b).

(2) Discarded frame region transformation. This category includes cropping and zooming in, which discard partial frame regions and modify the remaining content, as shown in Fig. 2(c).

[†] This research was supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under Grant: NSC 96-3113-H-001-012.

(3) Changed frame number transformation. This category includes frame rate change and video speed change (fast forward and slow motion), which increase or decrease the number of frames, as shown in Fig. 2(d).

While category (1) has been widely evaluated in previous works, categories (2) and (3) have received comparatively little attention. However, the latter two categories can be applied in many real applications. For example, frame cropping and fast forward operations can be used to generate condensed video content for skimming. Hence, we believe it is necessary to consider all three categories in CBCD.

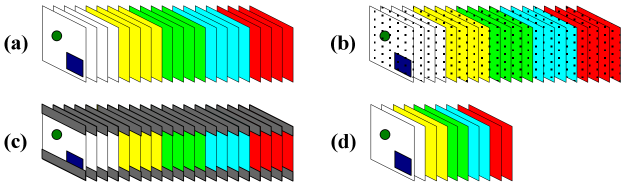


Fig. 2. (a) the source video; the transformed videos after (b) preserved frame region transformation (noise addition); (c) discarded frame region transformation (cropping); (d) changed frame number transformation (2× speed).

In contrast to CBVR, which seeks high-level semantic features to represent video content, CBCD explores low-level image features that are compact and robust to a variety of video transformations. *Global descriptors*, e.g., the ordinal measure [2][6][8][11][16] and the color-shift/centroid-based signature [7], try to model the properties of the entire frame region. They have proven robust against the preserved frame region transformation in many studies. However, if partial frame regions are discarded, the global descriptor of the modified frame might be totally different from that of the source frame. Therefore, some researchers have investigated an alternative feature representation, *local descriptors*, which model the local region properties of the points of interest in a frame. Examples include the Harris descriptor [9][12], the SIFT descriptor [3][15], and the color/motion volume [5]. In this paper, we exploit the SIFT-based feature to represent video content.

A number of matching schemes have been employed in the CBCD task. *Clip-level matching* is derived from the CBVR framework [5][9][12][14][15][16]. With this scheme, the given query and the video dataset must be at the same granularity level (e.g., shots or trajectories). Thus, searching a dataset of video shots would be problematic if the given query contains partial content of a shot. This scheme also lacks the ability to locate the exact time position of the copied segment in a video clip. *Fixed-length window sliding* [3][6][7][8][10][11], on the other hand, does not suffer from the above limitations because it uses a sliding window to scan a video sequence. The similarity between the given query and the windowed sequence is computed to determine whether it exceeds a predefined threshold. Since the window length is subject to the number of query frames, the comparison is not limited to a certain granularity level. In addition, the sliding window can indicate the definite time position of a detected copy.

However, the fixed-length sliding window approach confronts some problems in CBCD, since the fixed window length and the predefined threshold are inflexible for handling the various video transformations. For example, let us compare a source video in Fig. 2(a), and its copy derived by applying fast forward transformation in Fig. 2(d). It is clear that, with a fixed-length window, the contents of the source and the copy do not synchronize; hence, the similarity between the two sequences might be very low. Another example of the inflexibility is shown by the copy derived through cropping transformation, as shown in Fig. 2(c). Although the copy contains the essential content of the source video, the discarded parts might reduce the similarity of the two sequences. The reduced similarity scores in the above examples might be below the predefined threshold, and thus induce possible *false negatives*, i.e., the real copies cannot be detected.

1.2. Framework Overview

In this paper, we present a novel matching scheme that follows the window sliding paradigm, but tries to alleviate the problems posed by the conventional fixed-length sliding window approach. Although we do not know what types of video transformation have been applied in the copy in advance, some relationships between the source and the copy can be exploited to assist in similarity measurement. Based on this concept, we exploit the context of the query and the current window, and propose using *varied-length window sliding* and *adaptive thresholding* to estimate, respectively, the appropriate window length and threshold for effective matching. In addition, to make the matching more efficient, we integrate the proposed matching scheme into a coarse-to-fine framework. An overview of the integrated framework is illustrated in Fig. 3 and described in the following.

Given a query sequence and a target sequence, we assume that the query is an original video source and the target is a suspect video stream derived through web broadcasting. We use a sliding window to scan the target sequence in order to detect copies derived from the query. In the coarse stage, we first estimate a suitable length for the current sliding window. Then, the windowed sequence, as well as the query sequence, is represented by a compact feature called the *min-hashing signature*. In our experience, a thirty-dimension min-hashing signature is sufficient to represent a sixty-frame video sequence; hence, computing the similarity of two min-hashing signatures is very efficient. However, since the min-hashing similarity does not faithfully reflect the "containing relation" of the previous examples in Fig. 2, we infer a reasonable threshold to reflect this relation for each similarity measurement. If the similarity exceeds the adaptive threshold, the two matching sequences require an additional examination. This is performed in the fine stage through *spatio-temporal analysis*, which verifies the two sequences from spatial and temporal aspects. Then, the window moves forward to the next target frame. The whole process is repeated until the window reaches the end of the target sequence.

We implement several methods with different feature descriptors and matching schemes for the performance comparison. The experiment results demonstrate that the proposed framework yields a relatively excellent and stable accuracy among these methods on various video transformation types. The coarse-to-fine framework is efficient, as the excellent result can be obtained with approximately 0.4 seconds to search for copies of a thirty-second query sequence in a six-hour video sequence using a PC with 1.8 GHZ CPU and 2GB RAM. The successful integration of the coarse-to-fine matching scheme ensures that the proposed framework is fast and robust.

The remainder of this paper is organized as follows. In Section 2, we describe the components of the proposed framework. In Section 3, we discuss the extensive experiments conducted for the performance evaluation. Then, in Section 4, we summarize our conclusions.

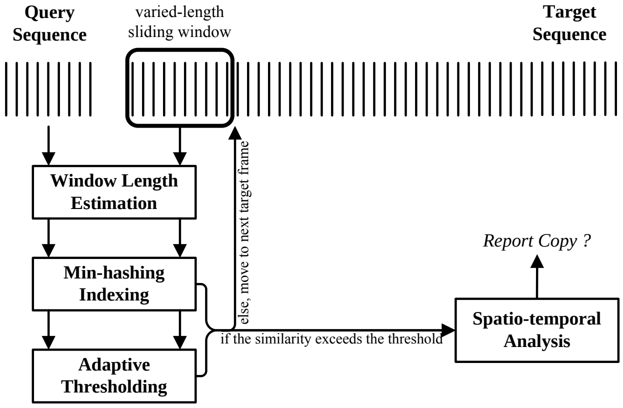


Fig. 3. An overview of the proposed framework

2. THE PROPOSED FRAMEWORK

The CBCD problem in this study is defined as follows. Let $Q = \{q_i \mid i = 1, 2, \ldots , n\}$ be a query sequence with n frames, where q_i is the i-th query frame; and let $T = \{t_j \mid j = 1, 2, \ldots , m\}$ be the target sequence with m frames, where t_j is the j-th target frame, and $n << m$. Suppose there exists a subsequence C in T, which is a copy derived by applying video transformation to Q. The goal is to quickly and accurately locate C from T for the given Q.

2.1. Feature Representation

We extract SIFT descriptors [13] to build the histogram for each video frame. The SIFT histogram, which is a typical "bag-of-words" model, can be constructed as follows. Given a frame q_i, we locate points of interest and compute their gradient orientation histograms, known as the SIFT descriptors. A training dataset collected from another video collection is used to generate a codebook with L codewords. Based on the codebook, each SIFT descriptor is quantized to the nearest codeword. Then frame q_i is represented by a histogram with L bins as $qH^{(i)} = \{qh_1^{(i)}, qh_2^{(i)}, \ldots, qh_l^{(i)}, \ldots, qh_L^{(i)}\}$, where $qh_l^{(i)}$ is the number of SIFT descriptors classified into the l-th cluster (bin). We then obtain the histogram for query sequence Q, denoted as $QH = \{qh_1, qh_2, \ldots , qh_l, \ldots , qh_L\}$, by aggregating all frames' histograms:

$$qh_l = \sum_{i=1}^{n} qh_l^{(i)} \cdot \qquad (1)$$

For t_j, the j-th frame in the target sequence, the histogram $tH^{(j)} = \{th_1^{(j)}, th_2^{(j)}, \ldots, th_l^{(j)}, \ldots, th_L^{(j)}\}$ can be constructed in the same manner.

2.2. Varied-length Window Sliding

The varied-length sliding window approach operates as follows. Let W be the sliding window used to scan T; and let $p \in [1, m]$ be the index indicating W's head location in T, and n' be the length of W. The windowed sequence C_p is thus represented as $C_p = \{t_j \mid j = p, p+1, \ldots , p+n'-1\}$. We compute the similarity between Q and C_p and shift W forward one frame for the next similarity computation. Unlike the fixed-length sliding window, the length of which is always equal to the query length n in the scanning process, W needs an appropriate length n' in each similarity computation. In this study, a fast estimation method is proposed to infer n'.

We use a distance feature to model the frame's motion energy. The distance feature, devised by Hoad and Zobel [7], is insensitive to several video transformation types and incurs a lower computational cost than existing motion estimation algorithms. To extract the distance feature, we identify the locations of the lightest (darkest) 5% of pixels in a frame and compute their average coordinate. Then, we calculate the Euclidean distance between the frame's average coordinate and the previous frame's average coordinate, and normalize it by their frame size. The normalized distance is denoted as the distance feature of the frame. For the sake of robustness, we use the sum of the lightest and darkest distances in this study.

The proposed estimation method for n' is described as follows. Let d_{q_i} be the distance features of a query frame q_i, $i = 1, 2, \ldots , n$. For the query Q, we compute the *accumulated distance* ad_Q by summing all of its distance features as follows:

$$ad_Q = \sum_{i=1}^{n} d_{q_i} \cdot \qquad (2)$$

Intuitively, the accumulated distance, which represents the movement of the lightest (or darkest) object, should remain consistent for the same video content despite video transformation. Based on this assumption, we take a video snippet S that starts from index p with length e from the target sequence T, i.e., $S = \{s_1 , \ldots s_j, \ldots s_e \mid s_j = t_{p+j-1}\}$. If S is part of the copy derived from Q, the following expression will hold:

$$e : ad_S = n' : ad_Q \, , \qquad (3)$$

where ad_S is the accumulated distance of S. That is, from the snippet's movement, we can infer the length of the complete copy sequence n' as $e \cdot ad_Q / ad_S$; thus, the interval of the windowed sequence C_p is identified.

The estimation of n' is very fast. For each similarity computation, it is only necessary to calculate the accumulated distance of S online by summing the distance features of totally e frames. We tried various numbers of e to estimate the window length and found that the estimation is not very sensitive to the choice of e. This point is discussed in more detailed in the experiment section.

2.3. Min-hashing Indexing

For a windowed sequence C_p, we construct a histogram $CH^{(p)} = \{ch_1^{(p)}, ch_2^{(p)}, \ldots, ch_l^{(p)}, \ldots, ch_L^{(p)}\}$, A conventional similarity between Q and C_p can be computed by the *Jaccard coefficient*:

$$J(Q, C_p) = \frac{|Q \cap C_p|}{|Q \cup C_p|} = \frac{|QH \cap CH^{(p)}|}{|QH \cup CH^{(p)}|}. \tag{4}$$

The Jaccard similarity $J(Q, C_p) \in [0, 1]$ is defined as the histogram intersection divided by the histogram union with respect to Q and C_p. If $J(Q, C_p) \geq \theta$, where θ is a predefined threshold, we take C_p as a candidate for later verification.

The computational cost of computing the Jaccard similarity comprises O(nL) for constructing $CH^{(p)}$ by summing n histograms of L dimensions, as well as O(L) for calculating the histogram intersection and the union between Q and C_p, which is O((n+1)L) in total. Note that, for simplicity, we assume $n = n'$ in the complexity analysis. Next, we present an approximate similarity measurement for the Jaccard coefficient, called *min-hashing indexing*, to reduce the computation cost.

Min-hashing, a kind of Locality Sensitive Hashing (LSH) function, is introduced to solve the nearest neighbor search problem efficiently. LSH will hash an input item multiple times, and the probability of collision is much higher for similar items than for dissimilar items. Several LSH measures have been derived based on the distance or similarity functions, e.g., the Hamming norm, Lp norms, the cosine distance, the Earth Mover's Distance, and the Jaccard coefficient. Here we employ the Jaccard coefficient-based LSH, i.e., min-hashing [4].

Basically, min-hashing associates each element of a feature vector with a hash value, which is a number generated independently and uniformly at random from a range of values. We can pick k minimum hash values, $k \geq 1$, as the min-hashing signature of the feature vector. The probability of two feature vectors having the same signature is proportional to their similarity. Cohen *et al.* [4] stated that, with a suitable choice of k, the number of false positives is fairly small and the number of false negatives is essentially zero. This is a desirable characteristic for copy detection applications.

To ensure the hashing process is efficient, we employ the histogram-based feature representation described in Section 2.1 and use the index of the histogram bin as the hash value. Let Ω be the set of indices of histogram bins with nonzero values, i.e., $\Omega = \{l \mid h_l > 0\}$. The elements in Ω are ranked in an ascending-order sequence $l(1)$, $l(2)$, ... $l(p)$, ... $l(|\Omega|)$, where $|\Omega|$ is the cardinality of Ω, and $l(p) \in \Omega$ is the p-th smallest index in Ω. A k-min-hashing signature SIG is defined as a sequence whose length is not larger than k:

$$SIG = \{l(r), r = 1,2,\ldots,\min(|\Omega|, k)\}, \tag{5}$$

In an ideal case, every histogram would be generated with an equal probability; and the cardinality of Ω would be much smaller than L, so that the proposed hashing would act approximately randomly.

We denote the min-hashing signatures of Q and C_p as SIG_Q and SIG_{C_p}, respectively. Then, their min-hashing similarity can be estimated through the expression proposed by Cohen *et al.* [4]:

$$M(Q, C_p) = \frac{|SIG_Q \cap SIG_{C_p}|}{\min(k, |SIG_Q \cup SIG_{C_p}|)}. \tag{6}$$

Cohen *et al.* showed that if $J(Q, C_p) \geq \theta$, then $M(Q, C_p) \geq \delta \cdot \theta$ with a probability $\geq \varepsilon$, where $\delta \in [0, 1]$; and ε will be very close to 1 if the choice of k is suitably large. That is, the min-hashing similarity M is proportional to the Jaccard similarity J. Therefore, we modify the candidate selection criterion as follows: if $M(Q, C_p) \geq \delta \cdot \theta$, C_p is considered a candidate for later verification.

However, the computational cost of the proposed min-hashing method is still O(nL) due to the time spent on histogram construction. Hence, we introduce a fast approximation technique to extract the min-hashing signature without histogram construction. We maintain a min-hashing signature with maximal g min-hashing values for each target frame t_j:

$$sig_j = \{l(p), p = 1,2,\ldots,\min(|\Psi|, g)\}, \tag{7}$$

where $\Psi = \{l \mid th_l^{(j)} > 0\}$ is an ascending sequence for t_j, as described in Ω. The signature dimension of a frame is usually much smaller than that of a sequence, i.e., $g << k$. Hence, C_p's min-hashing signature can be approximated by sorting all of C_p's frame signatures as follows:

$$SIG_{C_p} \approx SIG_{C_p}^{*} = \min_k(\{sig_p, sig_{p+1}, \ldots, sig_{p+n-1}\}). \tag{8}$$

where $\min_k(A)$ returns the k smallest elements of the set A in ascending order. If the cardinality of A is smaller than k, $\min_k(A)$ returns A in ascending order. By applying Eq. (8), we do not need to construct C_p's histogram. The computation time required to sort the k smallest signatures of C_p thus becomes $\mathrm{O}(ng \cdot \lg k)$.

Finally, the total time required to compute the min-hashing similarity between two sequences is comprised of $\mathrm{O}(ng \cdot \lg k)$ for generating a signature and O(k) for calculating the intersection and union between signatures[1], i.e., $\mathrm{O}(k + ng \cdot \lg k)$ in total. Compared with the Jaccard coefficient, whose time complexity is O((n+1)L), the min-hashing similarity can be computed more efficiently when $g << L$.

[1] Since the elements in the signature are presented in ascending order, the computational cost of signature intersection and union is O(k).

To summarize, we present a min-hashing-based method to represent and match video sequences. Because the min-hashing signature is very compact and the approximated form is calculated very rapidly, the video sequences can be matched efficiently.

2.4. Adaptive Thresholding

The adaptive thresholding technique is widely used in image segmentation, where the threshold varies over the image to reflect the image's local characteristics. In this subsection, we apply a similar concept in video matching. As mentioned earlier, the min-hashing similarity M, which is based on the Jaccard coefficient J, might not faithfully reflect the "containing relation" in the similarity score. Therefore, M might be inappropriate for handling some types of video transformation, such as cropping and slow motion. In contrast, the *overlap coefficient* can capture the containing relation for two video sequences. The overlap coefficient between a query Q and a windowed sequence C_p is defined as

$$O(Q,C_p)=\frac{|Q\cap C_p|}{\min(|Q|,|C_p|)}. \tag{9}$$

Similar to the relation between the Jaccard coefficient J in Eq. (4) and the min-hashing similarity M in Eq. (6), we derive the relation between the overlap coefficient O in Eq. (9) and the min-hashing similarity M in Eq. (6) as follows.

First, we apply the inclusion-exclusion principle,

$$|Q\cup C_p|=|Q|+|C_p|-|Q\cap C_p|, \tag{10}$$

to replace the denominator in Eq. (4), and obtain:

$$|Q\cap C_p|=\frac{(|Q|+|C_p|)\cdot J}{1+J}. \tag{11}$$

Therefore, Eq. (9) can be rewritten as

$$O=\frac{(|Q|+|C_p|)}{\min(|Q|,|C_p|)}\cdot\frac{J}{(1+J)}. \tag{12}$$

Note that, for simplicity, we denote $O(Q, C_p)$ and $J(Q, C_p)$ by O and J, respectively. Eq. (12) shows the relation between the overlap coefficient O and the Jaccard coefficient J. It is clear that $J \geq \theta$ if and only if $\frac{J}{1+J}\geq\frac{\theta}{1+\theta}$. Therefore, we have the following inequality condition:

$$J\geq\theta \;\Leftrightarrow\; O\geq\frac{(|Q|+|C_p|)}{\min(|Q|,|C_p|)}\cdot\frac{\theta}{(1+\theta)} \tag{13}$$

Recall that if $J \geq \theta$, then $M \geq \delta\cdot\theta$ with a probability $\geq \varepsilon$. Let $\theta'=\frac{(|Q|+|C_p|)}{\min(|Q|,|C_p|)}\cdot\frac{\theta}{(1+\theta)}$ in Eq. (13). The relation between O and M can then be formulated as follows:

if $O \geq \theta'$, then

$$M\geq\delta\cdot\frac{\theta'\cdot\min(|Q|,|C_p|)}{|Q|+|C_p|-\theta'\cdot\min(|Q|,|C_p|)}. \tag{14}$$

The detection criterion can thus be modified as: if the min-hashing similarity $M(Q, C_p)$ exceeds the threshold given in Eq. (14), C_p is considered a candidate.

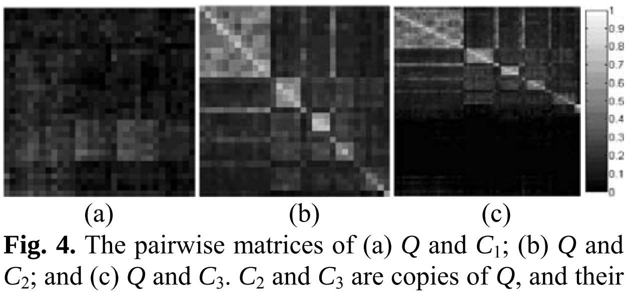
It is clear that the threshold in Eq. (14) is adaptable subject to the cardinalities of the query and windowed sequences for each similarity computation. In addition, the estimation of the threshold is very efficient, as only the cardinality of the windowed sequence has to be calculated online. Its effectiveness will be discussed later.

2.5. Spatio-temporal Analysis

As the histogram-based feature representation does not model the time relationship between frames, some candidates found in the coarse stage might be *false positives*, i.e., they might not be real copies of the query sequence. In fact, because of the data quantization error, a high similarity score between two frames does not necessarily mean their contents are near-duplicates. On the other hand, continuously high similarity scores between sequence frame pairs provide further evidence of the strength in their copy relation. Therefore, the similarity measurement should further integrate the information from the temporal aspect for the copy detection task.

To this end, we propose a verification method called *spatio-temporal analysis*, which dissects two matching sequences by compiling the spatio-temporal information with respect to each frame pair. For further details, please refer to our previous work [3]. In the following, we provide a brief overview of spatio-temporal analysis.

Given the query and the candidate, we construct their *pairwise matrix* to represent all-pair frame similarities. The pairwise matrix can be visualized by plotting its frame similarities as gray-level intensities. Figs. 4(a)-(c) illustrate some examples with query Q and three candidates C_1, C_2, and C_3, where the X-axis and the Y-axis indicate the candidate frame index and the query frame index, respectively. Except for candidate C_1, the other candidates are copies of Q: C_2 is a brightness enhanced copy and C_3 is a slow motion copy. We observe that the intensity distribution in Fig. 4(a) is very scattered, whereas slant line patterns appear clearly in Figs. 4(b)-(c). The slant line pattern, which manifests a set of consecutive frame pairs with high similarity scores, indicates a possible copy relation in that portion. Based on this observation, the task of spatio-temporal matching involves detecting slant line patterns on the pairwise matrix. We perform the detection process by using the *Hough transform* algorithm, a well-known technique for detecting objects in an image. With this algorithm, the candidate that does not have any slant line pattern in its corresponding pairwise matrix, i.e., it is not a real copy, can be removed effectively.



(a) (b) (c)

Fig. 4. The pairwise matrices of (a) Q and C_1; (b) Q and C_2; and (c) Q and C_3. C_2 and C_3 are copies of Q, and their pairwise matrices exhibit slant line patterns clearly.

3. EXPERIMENTS

We compiled a 6.1-hour video dataset from the Open Video Project for evaluations. The video clips were transformed into a uniform format, namely MPEG-1, 320×240 pixels, and 30 frames per second (fps), and then concatenated into a single sequence that served as the target sequence.

From the target sequence, we randomly extracted 31 sequences, each of thirty seconds duration. Each sequence was used as a source to derive six video copies, including compression 50%, noise addition 10% (preserved frame region transformation), cropping 20%, zooming in 10% (discarded frame region transformation), Halve the video speed and double the video speed (changed frame number transformation). This yielded a total of 186 (31×6) video copies, which served as the queries. We then used each query to detect the corresponding subsequences in the target sequence.

Since a continuous video sequence contains many identical or near-duplicate frames, for the sake of efficiency, it is not necessary to use every frame in the sequence for matching. Therefore, we selected a key frame every 15 frames of the target sequence. In other words, the frame rate of the target sequence was 2 fps. In addition, before starting the detection process, we had to determine the frame rate of the query video, which is commonly available from the file header. The query clip was re-sampled so that its frame rate synchronized with that of the target sequence. For example, a 30-second query video with 30fps will be a 60-frame sequence after re-sampling.

We extracted the SIFT descriptors for the query and target sequences. On average, each frame of the sequence in our dataset contains 22.84 SIFT descriptors. The LBG (Linde-Buzo-Gray) clustering algorithm was used to generate a codebook of size L = 1024; hence, the number of histogram bins was set at 1024.

3.1. Methods Evaluated

We implemented the following five methods to compare their performances:

(1) Hoad and Zobel's method [7] (abbreviated as "HZ").
(2) The coarse-to-fine framework, i.e., min-hashing indexing and spatio-temporal analysis ("CF").
(3) The coarse-to-fine framework + varied-length window sliding ("CF + VL").
(4) The coarse-to-fine framework + adaptive thresholding ("CF + AT").
(5) The coarse-to-fine framework + varied-length window sliding + adaptive thresholding ("CF + VL + AT").

Hoad and Zobel's method is one of the state of the art methods that uses fixed-length window sliding. We implemented it as follows. From each frame, we extracted the color-shift signature by using 16 bins for each of the three color channels in YCbCr, and the centroid-based signature described for the distance feature in Section 2.2. The two signatures were then combined into a two-dimensional feature vector, and an approximate string matching algorithm was applied to search video. This method provides a perspective of the typical fixed-length sliding window approach.

We define the following evaluation criterion: a detection result is considered correct if it has any overlap with the region from which the query was extracted. The *recall* and *precision* rates are used to evaluate the accuracy of the detection results:

$$recall = TP / (TP + FN), \quad (15)$$

$$precision = TP / (TP + FP), \quad (16)$$

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

3.2. Detection Accuracy

Recall that k and g, the two min-hashing parameters, are used to control the lengths of the signature dimensions of a sequence and a frame, respectively. For ease of presentation, we empirically used the following representative (k, g) pairs, (10, 4), (20, 4), (30, 5), (40, 5), (50, 6), (60, 6), (70, 7), (80, 7), (90, 7), and (100, 8), instead of all possible (k, g) combinations, to plot the PR graphs for illustrating the detection accuracy. Different values of θ in Methods (2)-(3) and θ' in Methods (4)-(5) were set for extensive observation. The related parameters were configured as follows: θ and θ' = {0.5, 0.6, 0.7, 0.8}, and δ = 0.5. Note that since Hoad and Zobel's method does not involve k and g, we evaluate its performance by simply measuring the recall and precision rates after x results have been obtained, where x is the number of positives that exist in the target sequence.

3.2.1. Preserved Frame Region Transformation

This category includes compression and noise addition. They have already been widely tested by existing methods. The results are shown in Figs. 5(a)-(b). Hoad and Zobel's method generally performs well under this transformation category. The proposed methods, i.e., Methods (2)-(5), also perform well overall. Intuitively, global descriptors (e.g., the color-shift/centroid-based signature) are superior to local descriptors (e.g., the SIFT descriptor) for this transformation category. However, the proposed methods demonstrate robust performances and are comparable with Hoad and Zobel's method.

3.2.2. Discarded Frame Region Transformation

This category includes cropping and zooming in. The results are shown in Figs. 5(c)-(d). Hoad and Zobel's method does not perform well in this category because, as mentioned earlier, global descriptors are not suitable for handling discarded frame region transformation. Interestingly, their method performs poorly on cropping, but quite well on zooming in transformation. In contrast, the SIFT descriptor used in our methods is less affected in this category.

The PR graphs demonstrate that the methods that use adaptive thresholding, i.e., Methods (4)-(5), achieve better recall rates. In other words, the proposed adaptive thresholding technique does reflect the "containing relation" between video sequences. Hence, some copies with lower similarity scores that are apt to be filtered out can be recovered from the false negative set.

3.2.3. Changed Frame Number Transformation

This category includes slow motion and fast forward. The results are shown in Figs. 5(e)-(f). Hoad and Zobel's method performs poorly in this category because of the content synchronization problem in the fixed-length sliding window approach. Even the approximate string matching scheme can not compensate for the large discrepancy between the query and windowed contents. Another reason is that the magnitude of its color-shift/centroid-based signature is conceptually amortized in neighboring frames. Hence, if the number of frames increases or decreases substantially, the method might produce a very different signature pattern from the original one.

The proposed methods perform well in this category because our spatio-temporal analysis effectively compiles the spatial and temporal information to find the copy relation between the query and windowed sequences. In addition, the methods that use the varied-length sliding window, i.e., Methods (3) and (5), achieve better performances. This shows that the proposed window length estimation technique effectively alleviates the content synchronization problem.

3.2.4. Summary

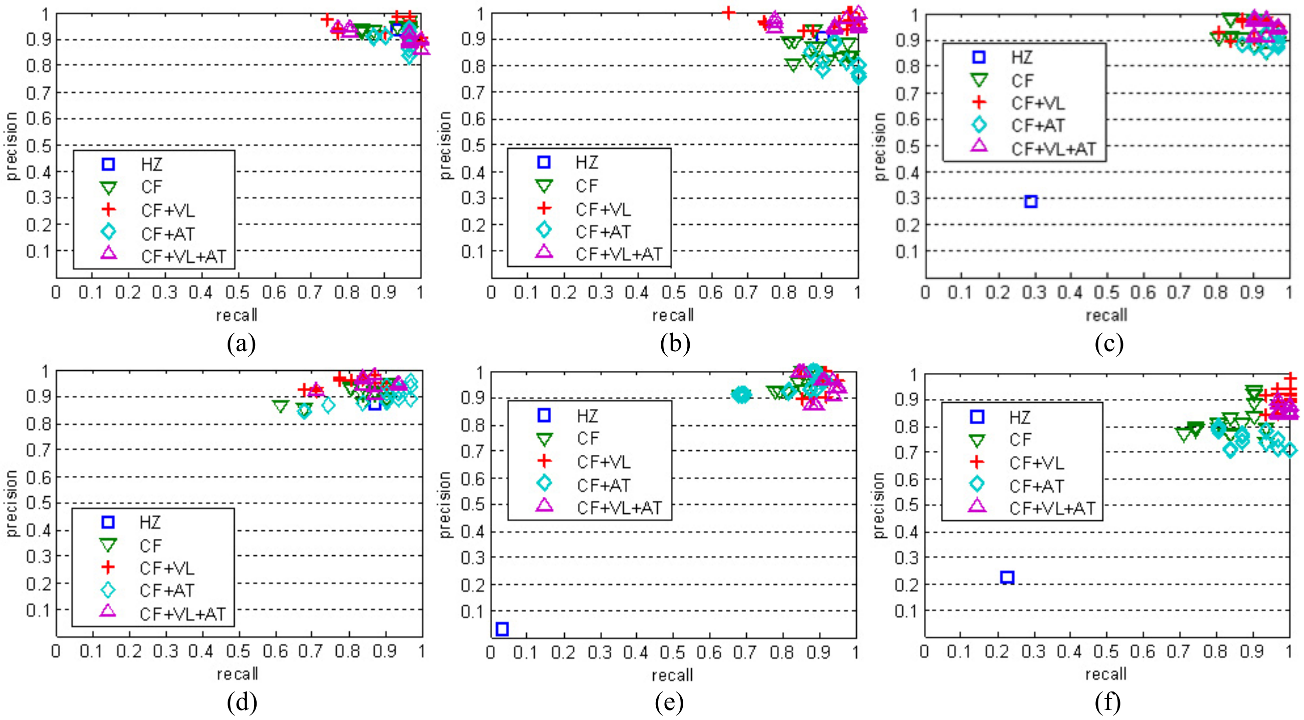
The above experiments demonstrate the proposed methods achieve excellent accuracy with very high recall and precision rates. In particular, Method (5), which integrates all the proposed techniques, yields a consistently robust performance for every type of video transformation. Moreover, the distribution of precision and recall rates are very compact in Method (5), showing its insensitivity to k and θ'. In other words, the stable and effective performance can be achieved without paying much attention to tune the signature length and threshold.

3.3. Computation Cost

The computation cost was evaluated in an environment where all the feature data of the video dataset was extracted and loaded into the memory. We take the performance of brightness enhancement transformation as example, where θ and θ' are set to 0.7.

First, we define the *candidate ratio* metric as:

$$candidate\ ratio = \frac{the\ number\ \ of\ \ frames\ of\ all\ candidates}{the\ number\ \ of\ \ frames\ in\ the\ target\ sequence}. \qquad (17)$$

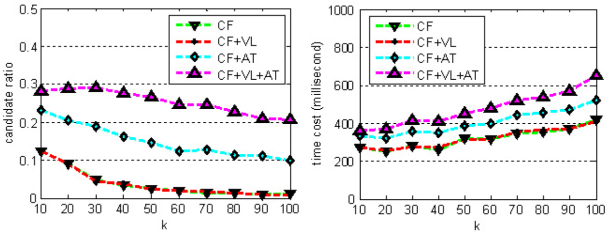


(a) (b) (c)

(d) (e) (f)

Fig. 5. PR graphs for video transformation: (a) compression; (b) noise addition; (c) cropping; (d) zooming in; (e) slow motion; and (f) fast forward

This metric, which calculates how many candidates are selected from the target sequence, expresses the efficiency of the proposed coarse-to-fine framework. A lower ratio means that fewer candidates are selected. The candidate ratios versus *k* for Methods (2)-(5) are shown in Fig. 6(a). We also show the execution time in Fig. 6(b). It is clear that, the rise of *k* usually reduces the number of candidates that fulfill the threshold criterion, while increases the computation cost in similarity measurement. Applying adaptive thresholding also increases the number of candidates. Since the candidate ratio of the proposed method does not exceed 0.3, it means at least 70% computation cost can be reduced by utilizing the proposed coarse-to-fine framework. According to our experience, *k* = 30 that requires 416 milliseconds to search in a 6.1 hour video sequence is sufficient to obtain a satisfactory accuracy rate.



(a) (b)

Fig. 6. The computation cost for Methods (2)-(5): (a) The candidate ratio versus *k*; and (b) the time cost versus *k*

4. CONCLUSIONS

To achieve efficient and effective detection of various video copies, we propose a novel video matching framework that integrates a coarse-to-fine strategy with varied-length window sliding and adaptive thresholding. Unlike conventional fixed-length window sliding, this framework estimates the appropriate window length and threshold for each similarity measurement to deal with a variety of video transformation types. In addition, to reduce the computation cost, we perform coarse-to-fine filtering by utilizing min-hashing indexing in the coarse stage and spatio-temporal analysis in the fine stage. The results of extensive experiments demonstrate that the successful integration of these components makes the proposed framework fast and robust.

REFERENCES

[1] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 602-615, 1998.

[2] 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*, Vol. 18, No. 3, pp. 412-417, 2008.

[3] C. Y. Chiu, C. C. Yang, and C. S. Chen, "Efficient and effective video copy detection based on spatiotemporal analysis," In *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, Taichung, Taiwan, Dec. 10-12, pp. 202-209, 2007.

[4] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. Ullman, and C. Yang, "Finding interesting associations without support pruning," In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, San Diego, USA, Feb. 28-Mar. 3, pp. 489-500, 2000.

[5] D. Dementhon and D. Doeramann, "Video retrieval of near-duplicates using k-nearest neighbor retrieval of spatio-temporal descriptors," *Multimedia Tools and Applications*, Vol. 30, No. 3, pp. 229-253, 2006.

[6] A. Hampapur, K. H. Hyun, and R. M. Bolle, "Comparison of sequence matching techniques for video copy detection," In *Proceedings of the SPIE Conference on Storage and Retrieval for Media Databases*, San Jose, CA, USA, pp. 194-201, 2002.

[7] 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.

[8] X. S. Hua, X. Chen, and H. J. Zhang, "Robust video signature based on ordinal measure," In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Singapore, Oct. 24-27, Vol. 1, pp. 685-688, 2004.

[9] A. Joly, C. Frelicot, and O. Buisson, "Content-based video copy detection in large databases: a local fingerprints statistical similarity search approach," In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Genova, Italy, Sep. 11-14, 2005.

[10] 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.

[11] 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.

[12] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa, "Robust voting algorithm based on labels of behavior for video copy detection," In *Proceedings of the ACM International Conference on Multimedia (MM)*, Santa Barbara, CA, USA, Oct. 23-27, pp. 835-844, 2006.

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

[14] H. T. Shen, B. C. Ooi, X. Zhou, and Z. Huang, "Towards effective indexing for very large video sequence database," In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, Baltimore, Maryland, USA, Jun. 14-16, pp. 730-741, 2005.

[15] X. Wu, A. G. Hauptmann, and C. W. Ngo, "Practical elimination of near-duplicates from web video search," In *Proceedings of the ACM International Conference on Multimedia (MM)*, Augsburg, Bavaria, Germany, Sep. 23-28, pp. 218-227, 2007.

[16] J. Yuan, L. Y. Duan, Q. Tian, and C. Xu, "Fast and robust search short video clip search using an index structure," In *Proceedings of the ACM International Workshop on Multimedia Information Retrieval (MIR)*, New York, USA, Oct. 15-16, pp. 61-68, 2004.