Frame Fusion for Video Copy Detection

Shikui Wei, Yao Zhao, *Member, IEEE,* Ce Zhu, *Senior Member, IEEE,*
Changsheng Xu, *Senior Member, IEEE,* and Zhenfeng Zhu

***Abstract*—Content-based video copy detection is very important for copyright protection in view of the growing popularity of video sharing websites, which deals with not only whether a copy occurs in a query video stream but also where the copy is located and where the copy is originated from. While a lot of work has addressed the problem with good performance, less effort has been made to consider the copy detection problem in the case of a continuous query stream, for which precise temporal localization and some complex video transformations like frame insertion and video editing need to be handled. We attempt to attack the problem by presenting a frame fusion based copy detection approach, which converts video copy detection to frame similarity search and frame fusion under a temporal consistency assumption. Our work focuses mainly on the frame fusion stage due to its critical role in copy detection performance. The proposed frame fusion scheme is based on a Viterbi-like algorithm, comprising an online back-tracking strategy with three relaxed constraints. The experimental results show that the proposed approach achieves high localization accuracy in both the query stream and the reference database even when a query video stream undergoes some complex transformations, while achieving comparable performance compared with state-of-the-art copy detection methods.**

***Index Terms*—Frame fusion, HMM, video copy detection, viterbi algorithm.**

I. Introduction

CONTENT-BASED video copy detection (CBCD), which offers an alternative to the watermarking technique, plays an important role in digital copyright protection, media tracking, law enforcement investigations, and so on. Normally, the watermarking technique performs copyright detection by retrieving the secret information embedded in a target video. It means that some secret information must be embedded before the video archive is distributed. In practice, it may be difficult to fulfill the requirement since huge amounts of video data have earlier been distributed without such processing. Compared to watermarking technology, the CBCD does not pose any additional requirements [16], which detects a copy by matching a query video with a reference database [6], [12], [17].

Generally speaking, CBCD refers to judging whether a query video contains any content originated from copyright protected video via some feature extraction and matching techniques [35]. The key challenge in CBCD is how to precisely localize the pair of a copy and its original clip in both the query video stream and the reference database despite various video transformations on the copy. This challenge becomes more difficult and complicated as the size of reference database increases. To this end, a lot of work has been done in recent years. In the earlier work, the main effort focuses on frame feature extraction and video matching based on the aligned frames. For example, those reported in [2], [8], [16], [18], and [24] treat a whole query video as a detection unit and attempt to match it with all possible subsequences of equal length within a long reference video, where a threshold is set to determine if there is a copy or not. However, those schemes fail if only a small segment in the query video is a copy in many practical applications [17]. An example is a broadcast stream in which only some clips are potential copies. Therefore, more flexible detection methods need to be designed to address this issue. Recently, frame fusion based methods provide a possibility to detect copied segments [4], [6], [7], [15]. These methods first search the reference database and return a list of similar reference frames for each query frame. Then the copies can be determined by fusing these returned reference frames according to a temporal consistency assumption. However, those methods generally process the query video in batch, that is, the query frames in one batch need to be parsed beforehand. This may limit or compromise the application or performance of those schemes for detecting copies in a continuous query video stream, such as the broadcast video stream.

To address the above problem, we consider a frame fusion based copy detection approach, which detects copies by similar frame search and frame fusion under a temporal consistency assumption. In this paper, our work focuses mainly on the critical frame fusion stage that performs copy determination and temporal localization. The proposed frame fusion scheme employs a Viterbi-like dynamic programming algorithm which comprises an online back-tracking strategy with three relaxed

Manuscript received December 6, 2009; revised May 16, 2010; accepted September 1, 2010. Date of publication January 13, 2011; date of current version February 24, 2011. This work was supported in part by the National Natural Science Foundation of China (61025013, 60970092), Sino-Singapore JRP (2010DFA11010), 973 Program (2011CB302204), the Open Foundation of National Laboratory of Pattern Recognition (2009JBZ006-3), and the State Scholarship for Study Aboard (2008709012). This paper was recommended by Associate Editor T. Fujii.

S. Wei is with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China, and also with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (e-mail: shkwei@google.com).

Y. Zhao and Z. Zhu are with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China (e-mail: yzhao@bjtu.edu.cn; zhfzhu@bjtu.edu.cn).

C. Zhu is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (e-mail: eczhu@ntu.edu.sg).

C. Xu is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100029, China (e-mail: csxu@nlpt.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2105554

constraints, namely, emission constraint, transition constraint, and gap constraint. In particular, when a new query frame is read and a list of similar reference frames is retrieved for it, the emission constraint and transition constraint are then used to build transition relationship between reference frames in the current list and reference frames in previous lists. Finally the gap constraint is employed to determine the starting and ending positions of complete paths. Using the online back-tracking, we can get a few complete paths at current time instant, which correspond to the original video clips. Note that the starting and ending positions indicate the boundaries of potential copies in the query video stream.

Compared with most existing frame fusion methods, the proposed frame fusion approach differs in three major aspects as follows.

1) An online back-tracking strategy to deal with the copy detection problem in a continuous query video stream. Instead of processing the query video stream in batch as done in *INRIA-LEAR* [4], [5], the back-tracking strategy detects copies "online" from a continuous video stream. When a new query frame comes in, the algorithm back-tracks all the partial best paths at the time instant and determines whether there is a copy sequence ending at this time instant. Due to the online processing feature, the proposed scheme can easily handle the copy detection problem in a continuous query video stream.
2) Some mechanisms to handle complex transformations and to tolerate matching offset and misalignment. The key constraint of frame fusion is the temporal consistency which assumes that similar video clips should be continuously similar in their aligned frames. However, it is usually difficult to meet this strict temporal consistency in practice due to some mismatches or offsets in key frame selection, feature representation, and frame matching. Instead of using the strict temporal consistency, we replace it with three more relaxed constraints, namely, emission constraint, transition constraint, and gap constraint. For example, the transition constraint can tolerate the matching offset and misalignment by relaxing the transition relationship among adjacent frames to that among frames in the same shot or adjacent shots.
3) A flexible copy determination manner. In addition to tolerating some video transformations like frame insertion by searching more candidate reference frames, the introduced gap constraint is also delegated to distinguish copy clips from non-copy video stream. Unlike previous work such as *INRIA-LEAR* [4], [5] which makes a decision with a threshold, the gap constraint determines a copy by automatically and accurately localizing its boundaries in the query video stream without requiring an explicit threshold. By limiting the transition scope of returned reference frames, the gap constraint can determine the starting and ending time instants of potential copies where no explicit threshold is involved in the decision making, thus avoiding the difficulty of threshold selection.

To facilitate the following discussions, we clarify some terms used in this paper. A copy refers to a video clip originated from a copyright protected video. "Original video" and "reference video" are interchangeable throughout this paper, which mean the copyright protected video. The remainder of this paper is organized as follows. We first review the related work for video copy detection in Section II. Section III presents an overall framework of the proposed CBCD system. In Section IV, we formulate the frame fusion problem and present a feasible solution. In Section V, we further refine the solution by modifying the conventional Viterbi algorithm and introducing an additional gap constraint. Section VI describes the experimental setup and evaluation criteria in detail. The experimental results and analysis are presented in Section VII. Finally, we conclude this paper in Section VIII.

II. Related Work

Content-based video copy detection involves two key techniques: feature extraction and video matching. We will review the existing work from these two aspects.

A. Feature Extraction

A copy is usually a transformed version of the reference clip. That is, the video signal of a copy is distorted from its original version. Therefore, the features used for copy detection should not only be distinctive enough for identification but also be robust enough to tolerate signal distortions. According to the feature nature, we can classify them into global features and local features.

Many research studies [2], [3], [8], [16], [24], especially in the earlier work, have paid much attention to the extraction of global features so as to deal with a variety of simple signal distortions. For example, Oostveen *et al.* [24] proposed a global binary feature for tolerating the changes in resolution and contrast. Ordinal measure based features [3], [8], [16] are employed for dealing with color degradation and change of display format. On the whole, however, global features are normally based on the statistics of the entire frame or the whole clip. Although those features are usually more compact and can be extracted more efficiently, they can only deal with some simple transformations. For some post-production transformations such as picture in picture, they are not workable since partial matching is needed in these cases. Compared with global features, the local features are automatically resistant to the transformations caused by some post-production operations [11] since a part of original content always remains in the copy. Most of local features used in CBCD are based on the interest points. Normally, all the interest points in a frame are detected first, and then a local descriptor is computed around each interest point. A lot of methods exist for both interest point detection and local descriptor calculation. Different combinations construct different extraction schemes [4]–[6], [11], [12], [29], [33], [34], [36], [37]. In [1], [6], and [9], for example, a fast Hessian detector is used to detect the interest points, and a SIFT local descriptor [20] is computed around each interest point. To improve the matching efficiency of local features, some efforts have been made on dimensionality reduction [14], [17] and local feature selection [11]. Another alternative scheme for

compacting features is the bag-of-features used frequently in the latest literature [1], [4], [5]. The key idea is to represent each frame as an orderless collection of local descriptors [10]. Usually, a visual word vocabulary is generated first by clustering a large training set of local features, and each cluster center is treated as a visual word. Afterward, the local features in a frame are mapped to those visual words, and then a visual word histogram is built for representing the frame. A main advantage lies in that it can generate a more compact representation as well as keep the partial matching feature. Moreover, the bag-of-features scheme can facilitate building index construction for speeding up search process. In our video copy detection system, we adopt the bag-of-features scheme in [23].

B. Video Matching

As the other key aspect, video matching plays an important role in the content-based video copy detection. A lot of matching methods have been proposed in recent years. According to the difference of matching manner, these matching methods can roughly be classified into two groups: sequence matching and frame fusion based matching.

The key idea of sequence matching lies in that two video clips are matched directly by frame-to-frame matching. Given a short query sequence $Q = (q_{1,}q_{2,} \cdots, q_N)$ and a long reference sequence $R = (r_1, r_2, \cdots, r_M)$, those methods slide the Q in the matching with R and result in a series of scores between Q and total $(M - N + 1)$ subsequences of R. Then a judgment mechanism involving a threshold is employed to determine whether the query Q is originated from a subsequence of R. Examples include the matching methods proposed in [2], [8], [16], [18], and [24]. However, those matching methods cannot efficiently deal with the scenario where a copy is only a small segment of the query video. With the sequence matching, we need to match all the possible subsequences in both the query and reference videos, which results in high computational complexity. In addition, it is also difficult for sequence matching to cope with some post-production transformations, such as frame dropping, fast/slow motion, although some attempts [3] have been made to alleviate it. Moreover, it is usually difficult to find an appropriate threshold beforehand due to various copy types.

Recently, frame fusion based matching methods [4], [6], [7], and [15], which provide a more flexible manner for copy detection, attract increasing attention. Unlike the sequence matching, the frame fusion based matching avoids directly matching query video with all equal-length reference clips. Instead, it first searches the reference database and returns a list of similar reference frames for each query frame. Then the copies can be determined by fusing the reference frames in the returned lists. Previous methods are based mainly on the statistics of the whole returned reference matches. That is, the whole query video needs to be processed beforehand. For example, a 2-D Hough histogram is employed for frame fusion by accumulating the votes on video identifier and time shift [4]. However, since the matches of all the query frames need to be obtained beforehand, this kind of scheme is not suitable for coping with a continuous query streams. In addition, while previous schemes pay more attention to detection precision and similarity search efficiency, they normally skip the localization precision and frame fusion efficiency.

The proposed frame fusion can smartly overcome these limitations by combining a back-tracking strategy and three relaxed constraints into a Viterbi-like algorithm. By dynamically determining the starting and ending time instants of potential copies, the proposed approach can detect video copies in an online manner. In fact, similar strategy is also used for detecting sequential gesture patterns in [31], which employs a logical DP matching for efficiently detecting similar subsequences.

C. Copy Detection

A complete detection process normally involves both feature extraction and video matching. Different combinations form different copy detection schemes. For example, an initial step for the sequence matching method is to calculate the distances between frames in two aligned video clips. Different feature schemes result in different similarity measurements. Likewise, similarity matching between query frames and reference frames is also required in the frame fusion based matching method.

However, for frame fusion based matching methods, there is another key difference among different schemes, that is, the frame fusion strategy. The performance of different schemes remains uncertain due to various fusion strategies. Our main effort just focuses on the critical problem of frame fusion.

III. Overall Framework

Although our work focuses mainly on the frame fusion phase, we also need a complete video copy detection system to validate our scheme. The system architecture is illustrated in Fig. 1. Each component will be detailed in the following subsections.

A. Keyframe Extraction

It is redundant and time-consuming to process all the frames within a video, thus the keyframe selection is a necessary step for improving the efficiency of copy detection. In our scheme, we combine the shot-based sampling scheme and the uniform sampling scheme [4] into a unified framework. First, we partition the reference video into shots using the method proposed in [25]. Then we uniformly sample each shot at a fixed sampling rate. In our experiment, we sample three frames/s. The main feature of this sampling strategy is that each uniformly sampled frame is associated with a shot boundary. This boundary information is very important for tolerating the matching offset caused by some signal distortions or the imperfection of low-level features. In addition, the shot boundary information is also useful for localizing where the copy is derived from, which will be explained later. Note that for the query video, we just uniformly sample three frames per second, while shot detection step is not performed in order to speed up the query process.

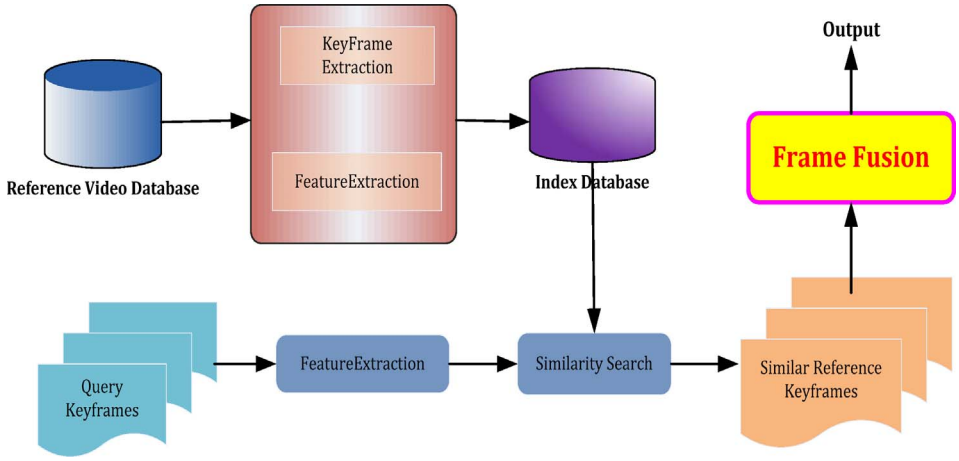

Fig. 1. Framework of proposed video copy detection system.

B. Feature Extraction

As mentioned in Section II, local features are inherently resistant to the transformations caused by some post-production operations, such as cropping and shifting. Therefore, our copy detection system is also based on the local features. In particular, Hessian-Affine region extractor [22] is employed to extract the affine-invariant key points for each frame, and then the SIFT descriptor [20] is used to represent each key point by a 128-dimensional vector. In our experiments, the software of [21] is used for both detecting Hessian-Affine regions and generating SIFT descriptors with default parameter settings. After that, the bag-of-features approach in [23] is further employed for compacting the feature representation. The key idea of this approach is to perform hierarchical k-means clustering on a training descriptor set to construct a visual vocabulary where each cluster centriod is treated as a visual word. Given a new local descriptor extracted from a frame, it is mapped to a visual word by searching the nearest centriod in the visual vocabulary. We implemented this procedure by using the VLFeat package available in [32]. In our experiments, a visual vocabulary with four levels and 100 000 leaf nodes is used for evaluating the proposed method, and another smaller vocabulary with four levels and 10 000 leaf nodes is employed for validating the effect of description on the overall detection performance.

To facilitate similarity search and indexing, we further map each visual word in the vocabulary into a unique pseudo-word. This means that each visual word is explicitly represented with a unique text string, and a frame containing lots of descriptors is transferred to a text document with pseudo-words. In this way, we can directly index and search visual contents using the existing tools in the text information retrieval field.

C. Similarity Search and Indexing

As discussed above, pseudo-word text documents are separately generated for frames in both the query and reference video. Therefore, some existing similarity matching models in the text retrieval area can be employed directly. In our scheme, we adopt the Okapi BM25 scoring function [28], [27], which represents state-of-the-art scoring function in the text information retrieval area. Given a query q containing pseudo-words $\{w_1, w_2, \ldots, w_m\}$, the Okapi BM25 score ranking a reference document d is

$$s(q,d) = \sum_{i=1}^{m} RSJ(w_i) \cdot \frac{f(w_i,d)\cdot(k+1)}{f(w_i,d)+k\cdot\left(1-b+b\cdot\frac{|d|}{avgdl}\right)} \quad (1)$$

where $f(w_i, d)$ is the term frequency of w_i in the document d, $|d|$ is the length of the document d in pseudo-words, $avgdl$ is the average length of the documents in the test database. k and b are free parameters, which are set to 2 and 0.75, respectively. $RSJ(w_i)$ is the Robertson-Sparck Jones weight [15] of the query term w_i, which is computed as follows:

$$RSJ(w_i) = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \quad (2)$$

where N is the total number of documents in the reference database, n is the number of documents containing w_i, R and r are two parameters related to relevance feedback. Since no relevance feedback is used in our scheme, we set both R and r to zero.

Likewise, we also improve the scoring efficiency by building an inverted table, which stores a mapping from the pseudo-words to the reference frames. The inverted table is equivalent to the index database in Fig. 1. In our experiment, we use the implementation in the Lemur toolkits [19] for both similarity search and text document indexing.

D. Frame Fusion

For each query frame, a list of similar reference frames with their scores is returned according to the scoring function (1). Our purpose is to determine whether the query video contains a copy derived from the reference video by fusing the returned reference frames according to the temporal relationships among them. In this paper, we target at this critical frame fusion phase, which will be discussed in more detail in the next two sections.

IV. Formulation of Frame Fusion

A. Problem Definition

After similarity search, each query frame is associated with a list of similar reference frames, which is denoted as follows:

$$Q = (q_1, q_2, \cdots, q_t, \cdots q_T) \tag{3}$$

$$L = (L_1, L_2, \cdots, L_t, \cdots L_T) \tag{4}$$

where q_t is the key frame at time instant t of the query sequence Q, and L_t is the list of similar reference frames returned for q_t. T is the length of the query stream, which may be a very large number even up to infinite. We assume that the top most similar reference frames are returned for each query frame, hence the length of each list is fixed to M.

The frame fusion is to detect possible copies from Q by fusing the reference frames in the returned lists. Since a copy is usually a small part of the query sequence, we further define subsequences for both the query sequence and the sequence of lists as follows:

$$Q_{sub}(i, j) = \{(q_i, q_{i+1}, \cdots, q_t, \cdots q_j) | 1 \le i \le T, i \le t \le j \le T\} \tag{5}$$

$$L_{sub}(i, j) = \{(L_i, L_{i+1}, \cdots, L_t, \cdots L_j) | 1 \le i \le T, i \le t \le j \le T\} \tag{6}$$

where $Q_{sub}(i, j)$ is a temporally successive subsequence from time instant i to j in the query sequence Q; $L_{sub}(i, j)$ is the corresponding list subsequence of $Q_{sub}(i, j)$.

Therefore, the problem is changed to determining whether $Q_{sub}(i, j)$ is a copy by fusing the reference frames in $L_{sub}(i, j)$ A key step of frame fusion is to reconstruct reference frame sequences from $L_{sub}(i, j)$ according to the temporal consistency information. In fact, $L_{sub}(i, j)$ can be treated as a frame sequence hypothesis space $H_{sub}(i, j)$ containing total $M^{(j-i+1)}$ reference frame sequences with length $(j - i + 1)$, which can be denoted as

$$H_{sub}(i, j) = \{(h_i, h_{i+1} \cdots, h_t, \cdots, h_j) | 1 \le i \le T, \\ i \le t \le j \le T, h_t \in L_t\} \tag{7}$$

where $H_{sub}(i, j)$ is constructed by concatenating the reference frames selected from the aligned lists of similar reference frames; h_t is a frame selected from L_t at time instant t.

Let h be a reference frame sequence in the hypothesis space $H_{sub}(i, j)$. Now, judging whether $Q_{sub}(i, j)$ is a copy is equivalent to checking whether there exists a reference frame sequence h that meets a certain temporal consistency assumption.

B. Problem Solution

In fact, the frame fusion problem can be further transformed to the decoding problem of hidden Markov model (HMM) [26]. The following shows their high similarity in definition.

1) *HMM decoding problem:* Given a particular emission sequence $E_{seq} = (e_1, e_2, \cdots, e_t, \cdots e_T)$ and a model $\lambda = \{T_r, E_m\}$, from state set $S = (s_1, s_2, \cdots, s_n, \cdots s_N)$, how we can find a state sequence $h = (h_1, h_2, \cdots, h_t, \cdots, h_T)$ that is most likely to have generated the emission sequence E_{seq}. Here, *Tr* and *Em* are the state transition and emission probability matrices, respectively; h_t is a state selected from S at time instant t of the state sequence h.
2) *Frame fusion problem:* Given a query subsequence $Q_{sub}(i, j)$ and a list subsequence $L_{sub}(i, j)$ under some constraints, how we can find a reference subsequence $h = (h_i, h_{i+1}, \cdots, h_t, \cdots, h_j)$ that is most likely to have generated the query subsequence $Q_{sub}(i, j)$. Here, h_t is a frame selected from L_t at time instant t.

We can easily convert the frame fusion problem into HMM decoding problem. In particular, the query subsequence can be directly treated as the emission sequence E_{seq}, and the reference frame in $L_{sub}(i, j)$ consitue the state set S after *Unique* operation. Here, the *Unique* symbol denotes the duplicate-removal operation on $L_{sub}(i, j)$, by which the corresponding state set S can be constituted with the remained frames from $L_{sub}(i, j)$. The conversion model can be formulated as follows:

$$E_{seq} = (q_i, q_{i+1}, \cdots, q_t, \cdots, q_j) \Leftarrow (e_i, e_{i+1}, \cdots, e_t, \cdots, e_j) \tag{8}$$

$$S = \{s_1, s_2 \cdots, s_n, \cdots, s_N\} \Leftarrow Unique\{L_i, L_{i+1}, \cdots, L_t, \cdots, L_j\} \tag{9}$$

$$H(i, j = \{(h_i, h_{i+1}, \cdots, h_t, \cdots, h_j,) | 1 \le i \le T, \\ i \le t \le j \le T, h_t \in S\}) \tag{10}$$

$$\begin{aligned} h^* &= \underset{h \in H(i,j)}{argmax}\ P(E_{seq}, h) \\ &= \underset{h \in H(i,j)}{argmax}\ P(h) \cdot P(E_{seq}|h) \\ &= argmax\{P((h_i, h_{i+1}, \cdots, h_t, \cdots, h_j)) \cdot \\ &\quad P((e_i, e_{i+1}, \cdots, e_t, \cdots, e_j)|(h_i, h_{i+1}, \cdots, h_t, \cdots, h_j))\} \end{aligned} \tag{11}$$

where $H(i, j)$ is a superset of $H_{sub}(i, j)$, and h_t is a frame selected from S.

In our context, $P((h_i, h_{i+1}, \cdots, h_t, \cdots, h_j))$ reflects the transition relationship among the returned reference frames, whereas $P((e_i, e_{i+1}, \cdots, e_t, \cdots, e_j)|(h_i, h_{i+1}, \cdots, h_t, \cdots, h_j))$ implies the similarity measurement between the query sequence $(q_i, q_{i+1}, \cdots, q_t, \cdots, q_j)$ and a reference frame sequence $(h_i, h_{i+1}, \cdots, h_t, \cdots, h_j)$. We employ the first-order Markov chain to model the transition relationship, which assumes that the present state is only dependent on the previous state. That is, $P(h_t|h_{t-1}, \cdots, h_i = P(h_t|h_{t-1}), t = i + 1, \cdots, j$. For similarity measurement, since we perform an independent similarity search for each query frame, $P(e_t|h_t)$, $t = i, \cdots, j$, are independent of each other. Therefore, we can rewrite the objective function (11) as

$$\begin{aligned} h^* = \underset{h \in H(i,j)}{argmax}\{&P(h_i) \cdot P(e_i|h_i) \\ &\cdot \Pi_{t=i+1}^{j}\ P(h_t|h_{t-1})\ P(e_t|h_t)\}. \end{aligned} \tag{12}$$

In order to calculate the above objective function, we need to estimate both $P(h_t|h_{t-1})$ and $P(e_t|h_t)$, i.e., the state transition probability $Tr = (P(s_y|s_x)|x\ and\ y \in \{1, 2, \cdots, N\})$, and the

emission probability $Em = (P(e_t|s_x)|x \in \{1, 2, \cdots, N\}, t \in \{i, i+1, \cdots, j\})$, where N is the total number of states. To this end, two relaxed constraints are given in the following. Note that we assume $P(s_x)$ follows the uniform distribution. Hence, $P(s_x)$ is set to $\frac{1}{N}$, that is, $P(h_i)$ is set to $\frac{1}{N}$.

1) *Transition Constraint:* According to the strict temporal consistency assumption, if a query frame subsequence $(q_i, q_{i+1}, \cdots, q_t, \cdots, q_j)$ is a copy of a reconstructed reference frame subsequence $(h_i, h_{i+1}, \cdots, h_t, \cdots, h_j)$, then the reconstructed reference frame sequence should be a temporally successive frame sequence in a reference video. That is, for any two frames h_{t-1} and $h_t, t = i+1, \cdots, j$, h_{t-1} can transfer to h_t if and only if h_t is the next frame of h_{t-1} in the same reference clip. However, there may be some drawbacks with this assumption. First, a copy is usually a transformed version of its original video clip, thus it is difficult to get perfect matches along all the aligned key frames in two clips due to the limitation of feature representation. We even cannot align them if the copy is obtained by dropping some frames from or inserting some frames into its original video clip. In addition, adjacent frames in a video clip are usually perceptually similar with each other due to the high redundancy of video content. Therefore, even if all the frames in the sequence $(h_i, h_{i+1}, \cdots, h_t, \cdots, h_j)$ belong to the same reference clip and the clip indeed generates the query sequence $(q_i, q_{i+1}, \cdots, q_t, \cdots, q_j)$, it is also less possible to guarantee the reference sequence to be aligned with the query sequence in the completely correct temporal order. Hence, we constrain the transition relationship with a more relaxed assumption. Assume that for any two reference frames (states) s_x and s_y, s_x can transfer to s_y if and only if s_x and s_y are in the same shot or in two adjacent shots. In this way, we can tolerate a certain matching offset. This constraint is described as follows:

$$P(s_y|s_x) = \begin{cases} \partial_1, & s_x \in V_t \text{ and } s_y \in V_t \\ \partial_2, & s_x \in V_t \text{ and } s_y \in V_{t+1} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where V_t and V_{t+1} are two temporally successive shots in the same reference video; ∂_1 and ∂_2 are two constants representing transition probabilities, which are set to 1 and 0.8 in our experiment, respectively.

2) *Emission Constraint:* As discussed above, the state set S consists of all the unrepeated similar reference frames in the returned lists. For a specific emission e_t (i.e., the query frame q_t), since its corresponding list L_t containing similar reference frames is only a subset of state set S, not all the states will produce the emission. Therefore, we have

$$P(e_t|s_x) = P(q_t|s_x) = \begin{cases} score(q_t, s_x), & \text{if } s_x \in L_t \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where *score* (*, *) is the scoring function for similarity search. All similarity scores are normalized beforehand.

So far, given a query subsequence, we need to find the most likely state sequence that might have generated the query subsequence according to objective function (12). If we exhaustively evaluate all possible state sequences, it is computationally prohibitive. Therefore, the Viterbi algorithm, as a kind of dynamic programming algorithm, is employed. At any time instant, the Viterbi algorithm avoids tracking all possible paths by keeping only the most likely path (the partial best path) for each state. After arriving at the end of the query sequence, a whole best path can be obtained by a path backtracking process.

V. Frame Fusion Using Viterbi-Like Algorithm

In Section IV, we formulate the problem of frame fusion and provide a feasible solution. However, it raises two problems if we directly follow it. First, high computational cost is inevitable. In order to detect copied clips, we need to separately check all possible and temporally successive query subsequences. If the query length is T, then we need to check a total of $\frac{T\cdot(T+1)}{2}$ query subsequences. Given the average length T_{sub} of subsequences and the length M of each list, a computational complexity of $o(\frac{T\cdot(T+1)}{2} \cdot M^{T_{sub}})$ is required using the exhaustive search method. Even if we can reduce the complexity to $o(\frac{T*(T+1)}{2} \cdot T_{sub} \cdot M^2)$ using the Viterbi algorithm [26], it is still expensive. More importantly, this method is not convenient for dealing with unbounded query stream since it is impossible to obtain all possible query subsequences beforehand in this case.

To address these problems, we refine the scheme by modifying the conventional Viterbi algorithm and introducing an additional gap constraint.

A. Viterbi-Like Algorithm

Instead of separately checking all possible query subsequences, we attempt to detect copies from a continuous query video stream in an online manner. The core problem is how to precisely localize boundaries of copies in the continuous query stream. To this end, we introduce an additional gap constraint and redefine the partial best path. The gap constraint provides a mechanism for distinguishing copies from non-copy video clips as well as tolerating some video transformations like frame insertion operation. The redefined partial best path records a reference frame sequence that is most likely to generate its corresponding query subsequence. The starting and ending nodes of each path corresponding to a reconstructed reference video clip are determined by the gap constraint. The following describes them in details.

1) *Gap Constraint:* Given the query subsequence $(q_{i-\Delta t}, \cdots, q_{i-1}, q_i, q_{i+1}, \cdots, q_t, \cdots q_j, q_{j+1}, \cdots, q_{j+\Delta t})$ and a reference frame sequence $(h_{i-\Delta t}, \cdots, h_{i-1}, h_i, h_{i+1}, \cdots, h_t, \cdots, h_j, h_{j+1}, \cdots, h_{j+\Delta t})$, if there is not any transition from h_i to the previous Δt reference frames, the time instant i can serve as a possible starting point of a copy. The constraint means that we can determine the starting instant of a copy based on the transition relationship among similar reference frames at different time instants. Likewise, we can determine the ending instant in the same way. In addition, the gap constraint can also deal with video transformations caused by inserting some non-copy frames. The length of Δt limits the maximum number of non-copy keyframes allowed to be inserted in the copy clip. We describe this constraint as follows.

The query subsequence $(q_i, q_{i+1}, \cdots, q_t, \cdots, q_j)$ is possibly a copy of the reference sequence $(h_i, h_{i+1}, \cdots, h_t, \cdots, h_j)$ if and only if

$$\begin{cases} P(h_i|h_{i-1}) = 0 \\ P(h_i|h_{i-2}) = 0 \\ P(h_i|h_{i-\Delta t}) = 0 \end{cases} \tag{15}$$

$$\text{and} \begin{cases} P(h_j|h_{j+1}) = 0 \\ P(h_j|h_{j+2}) = 0 \\ P(h_j|h_{j+\Delta t}) = 0. \end{cases} \tag{16}$$

Equations (15) and (16) can determine the starting instant i and the ending instant j, respectively.

2) *Partial Best Path:* Let $\delta(t, x)$ be the best score of all possible state sequences starting at any states at any previous time instants and ending at state s_x at time instant t. The partial best path $Path(t, x)$ is the state sequence (or reference frame sequence) which achieves the best score. Note that the partial best path ending at state s_x at time t may start at any previous time instants, instead of starting at time $t = 1$ as Viterbi algorithm does.

At time $t = 1$, $\delta(t, x)$ is calculated as follows:

$$\delta(1, x) = Em(1, x) \tag{17}$$

where $Em(1, x)$ is the emission probability from state s_x to emission e_1, which can be calculated according to the emission constraint (14). At time $2 \le t \le \infty$, $\delta(t, x)$ is calculated as follows:

$$\delta(t, x) = \begin{cases} \mu\left(\tilde{t}^*, n^*\right) + Em(t, x), & \text{if } Em(t, x) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

$$\mu\left(\tilde{t}^*, n^*\right) = \delta\left(\tilde{t}^* *, n^*\right) \cdot Tr(n^*, x) \tag{19}$$

$$\left(\tilde{t}^*, n^*\right) = \begin{cases} \underset{1 \le \tilde{t} < t, 1 \le n, \le N_t}{argmax} \left(\delta\left(\tilde{t}, n\right) .Tr(n, x)\right), t \le \Delta t \\ \underset{t - \Delta t + 1 \le \tilde{t} < t, 1 \le n, \le N_t}{argmax} \left(\delta\left(\tilde{t}, n\right) .Tr(n, x)\right), t > \Delta t \end{cases} \tag{20}$$

where N_t is the size of state set at time instant t; $(\tilde{t}^*, n^*)$ indicates the state S_{n*} at time instant, $\tilde{t}^*$ from which $\delta(t, x)$ achieves the best score, $\mu\left(\tilde{t}^*, n^*\right)$ indicates the maximum transition score from states at previous different time instants to state s_x at time instant t; $Tr(n, x)$ is the transition probability from state s_n to s_x, which can be calculated according to the transition constraint (13). For (18), the right side is $\mu\left(\tilde{t}^*, n^*\right) + Em(t, x)$ which is different from the original form $\mu\left(\tilde{t}^*, n^*\right) \cdot Em(t, x)$ in the conventional Viterbi algorithm. By this way, when the state s_x at time instant t has no any transition with states at previous different time instants (i.e., $\mu\left(\tilde{t}^*, n^*\right) = 0$), it can serve as a starting point of new partial best paths if $Em(t, x > 0)$,. In addition, using this form can also void the problem of data overflow due to the product of a large number of values which are far lower than one.

Note that (19) implies the gap constraint (15). If $\mu\left(\tilde{t}^*, n^*\right) = 0$, then $\delta(t, x) = Em(t, x)$, which means that no transition exists between the state s_x at time t and any states in the past. That is, the state s_x at time instant t can serve as a starting node of new partial best paths. The back-tracking strategy is formulated in (20). It means that at any time instant t, we search backward

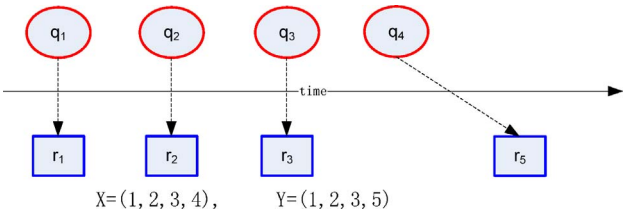

Fig. 2. Example illustrating how to get two time stamp sequences for a pair of query sequence and partial best path. Note that the subscripts of query frames and reference frames indicate the time instants relative to the first frame in query sequence and partial best path, respectively.

the best partial paths obtained at several time instants in the past. In this way, we can iteratively use the previous detection information and speed up the copy detection.

Using the Viterbi-Like algorithm, the problem of frame fusion becomes the back-tracking of the partial paths. Once certain partial best paths meet the gap constraint, we can then localize both the copies in the query video stream and their original clips in the reference video. Table I gives the detailed description of the Viterbi-like algorithm. Note that both emission and transition matrices are updated dynamically since new emissions and states are generated when new query frames come. Still, the construction of the transition and emission matrices is based on the proposed transition and emission constraints (13), (14), respectively. Because the dynamic programming method used in the conventional Viterbi algorithm is remained, the efficiency of frame fusion is high.

The main difference from the conventional Viterbi algorithm is that Viterbi-like algorithm back-tracks the partial best paths at each time instant, instead of back-tracking once after traveling forward all the time instants. It means that the back-tracking process will never stop until arriving at the end of the query stream. In this way, it can deal with a continuous query stream by making an online decision. Another difference lies in that the Viterbi-like algorithm builds the partial best path for the current time instant by looking back several time instants in the past, instead of checking only the previous one time instant as the conventional Viterbi algorithm does. Equation (20) implies this strategy. By this way, we can enlarge the match scope and then tolerate more complex transformations.

B. Pruning and Localization

Each partial best path is a reference frame sequence, which corresponds to a query subsequence. Because we make a decision when each new query frame comes, we need to determine whether the partial best paths (reference sequences) at current time instant indeed generate their corresponding query sequences. Intuitively, we still need a hard threshold to solve this problem. Here, the hard threshold means a fixed score value that is used for all queries to determine whether copies occur in queries. However, according to gap constraint, only the partial best paths which meet the equation groups (15), (16) are retained. The corresponding query subsequences of the retained paths are treated as copies. Note that we need to check the following Δt query frames so as to determine whether the partial best paths at current time instant are ended. In order

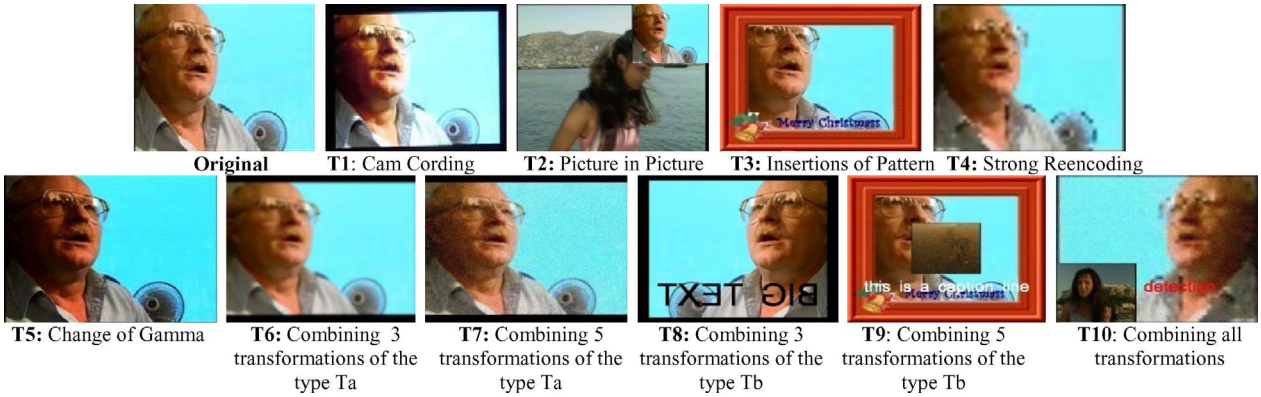

Fig. 3. Example frames from the transformed query videos.

to further filter out false copies, we compute the Pearson's correlation coefficient between the aligned time stamps of corresponding query sequence and partial best path. Fig. 2 shows how to get two time stamp sequences for a pair of query sequence and partial best path with four frames. Given two time stamp sequences $X = (x_1, x_2, \cdots, x_n)$ and $y = (y_1, y_2, \cdots, y_n)$, the Pearson's correlation coefficient is given by

$$\rho(X,Y) = \frac{n \cdot \left(\sum_{i=1}^{n} x_i \cdot y_i\right) - \left(\sum_{i=1}^{n} x_i\right) \cdot \left(\sum_{i=1}^{n} y_i\right)}{\sqrt{n \cdot \left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2} \cdot \sqrt{n \cdot \left(\sum_{i=1}^{n} y_i^2\right) - \left(\sum_{i=1}^{n} y_i\right)^2}}. \tag{21}$$

The correlation coefficient ranges from -1 to 1, and it is greater than 0 if two sequences are positive correlation. In our context, only the pairs with positive correlation are reserved. The resulting partial best paths are further filtered out by pruning those short paths. Here, we consider a path as short path if the number of its nodes is less than six.

After we have obtained a series of partial best paths and query subsequences, we need to consider how to localize them in the reference database and in the query stream, respectively. In fact, since the frames in the query sequence are ordered temporally, the starting and ending time instants of a path localize a possible copy in the query video stream. To find the location of a partial best path in the reference database, the boundary information of shots is used. According to the transition constraint, all the reference frames in a path may be out of order temporally, but they must be in the same shot or adjacent shots. Therefore, we can localize the reference clip in the database by searching lower and upper boundary of frames in the corresponding shots.

Since our localization of reference clips is based on the boundaries of shots, it is very likely to generate the same reference clip for different paths. In our scheme, for all the paths corresponding to the same reference clip, we only remain the one with the largest length.

VI. Evaluation Setup

A. Reference Dataset

Our video copy detection system is evaluated based on the Sound & Vision dataset used in TRECVID 2008 search

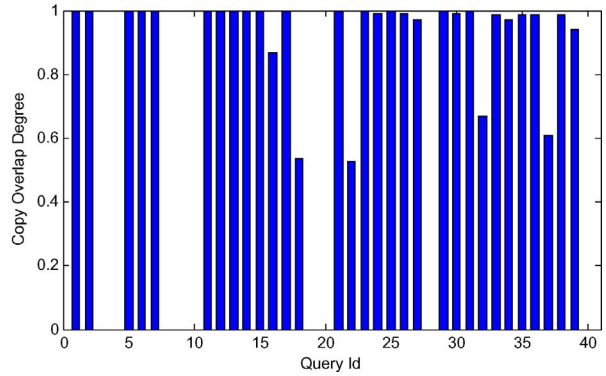

Fig. 4. Copy overlap degree evaluation on separate queries.

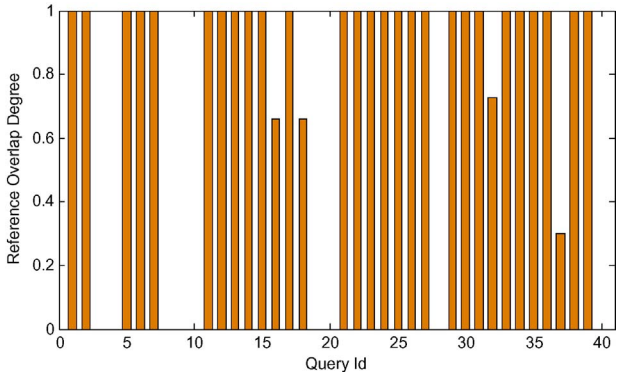

Fig. 5. Reference overlap degree evaluation on separate queries.

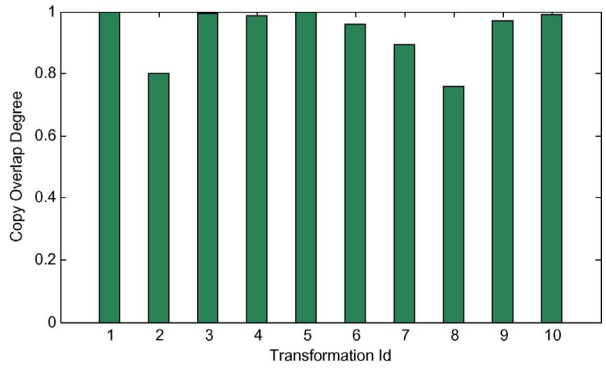

Fig. 6. Copy overlap degree evaluation on varied transformations.

TABLE I

FRAME FUSION USING VITERBI-LIKE ALGORITHM

Parameters

t : the time instant in query stream
x : the index of the x^{th} state in the current state set
$E_{seq,t}$: the subsequence of query at time instant t
S_t : the state set at time instant t
L_t : the list of similar reference frames returned for q_t
N_t : the total number of states at time instant t
$s_{t,x}$: the x^{th} state in the state set S_t
$Path(t,x)$: the partial best path arriving at the state $s_{t,x}$

Initialization, $t = 1$ and $1 \le x \le N_1$

1) Construct emission sequence and state set for time instant t:
   $E_{seq,1} = (e_1) \Leftarrow (q_1)$
   $S_1 = \{s_{1,1}, s_{1,2}, \cdots, s_{1,N_1}\} \Leftarrow Unique\{L_1\}$
2) Calculate probability matrices Tr and Em
3) Calculate best score for any state $s_{1,x}$:
   $\delta(1,x) = Em(1,x)$
4) Backtrack the partial best path for any state $s_{1,x}$:
   If $\delta(1,x) > 0$
      $Path(1,x) = (s_{1,x})$
   Else
      $Path(1,x) = ( )$
   End

Reduction, $2 \le t \le T$ and $1 \le x \le N_t$

1) Construct emission sequence and state set for time instant t:
   If $2 \le t \le \Delta t$
      $E_{seq,t} = (e_1, e_2, \cdots, e_t) \Leftarrow (q_1, q_2, \cdots, q_t)$
      $S_t = \{s_{t,1}, s_{t,2}, \cdots, s_{t,N_t}\} \Leftarrow Unique\{L_1, L_2, \cdots, L_t\}$
   Else
      $E_{seq,t} = (q_{t-\Delta t+1}, q_{t-\Delta t+2}, \cdots, q_t) \Leftarrow (e_{t-\Delta t+1}, e_{t-\Delta t+2}, \cdots, e_t)$
      $S_t = \{s_{t,1}, s_{t,2}, \cdots, s_{t,N_t}\} \Leftarrow Unique\{L_{t-\Delta t+1}, L_{t-\Delta t+2}, \cdots, L_t\}$
   End
2) Update probability matrices Tr and Em
3) Search the best state in previous time instants, which leads to the best score at state $s_{t,x}$:
$$(\tilde{t}^*, n^*) = \begin{cases} \underset{1 \le \tilde{t} < t, 1 \le n \le N_t}{argmax} \ (\delta(\tilde{t}, n) \cdot Tr(n,x)), t \le \Delta t \\ \underset{t-\Delta t+1 \le \tilde{t} < t, 1 \le n \le N_t}{argmax} \ (\delta(\tilde{t}, n) \cdot Tr(n,x)), t > \Delta t \end{cases}$$
4) Calculate the maximum transition score from states at previous time instants to $s_{t,x}$:
   $\mu(\tilde{t}^*, n^*) = \delta(\tilde{t}^*, n^*) \cdot Tr(n^*, x)$
5) Calculate the best score for any state $s_{t,x}$:
   If $Em(t,x) > 0$
      $\delta(t,x) = \mu(\tilde{t}^*, n^*) + Em(t,x)$
   Else
      $\delta(t,x) = 0$
   End
6) Backtrack the best path for any state $s_{t,x}$:
   If $Em(t,x) > 0$
      If $\mu(\tilde{t}^*, n^*) > 0$
         $Path(t,x) = (Path(\tilde{t}^*, n^*), s_{t,x})$
      Else
         $Path(t,x) = (s_{t,x})$
      End
   Else
      $Path(t,x) = ( )$
   End

Output

If $Path(t,x)$ meets the condition (16)
   Output the $Path(t,x)$
Else
   Continue the frame fusion process
End

and high level feature extraction tasks. This dataset contains approximately 200 h of videos including news magazine, science news, educational programming, and archival video. Using the frame sampling strategy proposed in Section III,

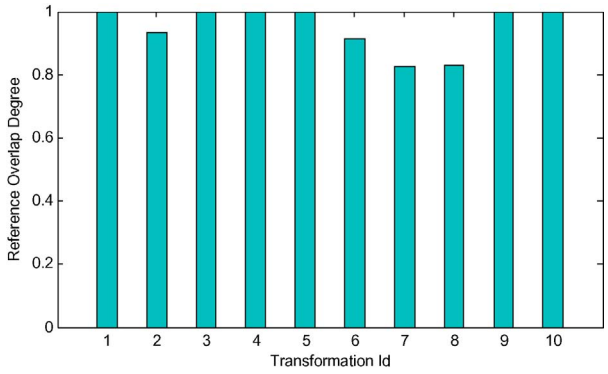

Fig. 7. Reference overlap degree evaluation on varied transformations.

TABLE II

SINGLE TRANSFORMATION LIST

Type	Decrease in Quality (Ta)	Post-Production (Tb)
Single Transformations	Blur	Crop
	Gamma	Shift
	Frame dropping	Contrast
	Contrast	Caption (text insertion)
	Compression	Flip (vertical mirroring)
	Ratio	Insertion of pattern
	Noise	Picture in picture

over 2 million keyframes are sampled from the video dataset. All our experiments are carried out on this reference dataset.

B. Query Construction

In our experiment, four video clips are selected from the reference dataset, and one video clip is selected from non-reference dataset. Each clip is inserted into a non-reference video stream. After that, all the combined video streams are transformed separately by applying ten complex transformations. The total 50 query video streams are generated for testing our proposed method. Note that only 40 of these queries indeed have copies. In our experiment, ten complex transformations are constructed by combining one or more single video transformations. Table II lists all the single video transformations of two types, and Fig. 3 illustrates these ten complex transformations with some example frames.

C. Evaluation Criteria

The miss rate and the false alarm rate are two criteria generally used for evaluating the detection accuracy. They are defined as follows:

$$R_{Miss} = \frac{N_{rel} - N_{ret\_rel}}{N_{rel}} \tag{22}$$

$$R_{FA} = \frac{N_{ret} - N_{ret\_rel}}{N_{ret}} \tag{23}$$

where N_{ret} is the total number of the returned results, N_{ret_rel} is the number of true positives in the returned results, and N_{rel} is the total number of true positives.

Although the above evaluation criteria can give a good measurement on the overall detection performance, they do not take into account the localization precision. Thus two more criteria for evaluating the localization precision are defined.

1) *Copy Overlap Degree:* This criterion measures the overlap degree in time duration between the detected copy and its ground truth

$$CopyOverlap(i) = \frac{Overlap(Q_i, Q_{gi})}{Length(Q_{gi})} \tag{24}$$

where $Overlap(Q_i, Q_{gi})$ is the time span of overlap between the detected copy clip Q_i and its ground truth clip Q_{gi}, $Length(Q_{gi})$ is the total time span of the ground truth clip Q_{gi}.

2) *Reference Overlap Degree:* This criterion measures the overlap degree in time duration between the asserted reference clip of a copy and its ground truth

$$RefOverlap(i) = \frac{Overlap(R_i, R_{gi})}{Length(R_{gi})} \tag{25}$$

where $Overlap(R_i, R_{gi})$ is the time span of overlap between the asserted reference clip R_i and its ground truth R_{gi}, $Length(R_{gi})$ is the total time span of the ground truth clip R_{gi}. Note that NIST TRECVID 2008 provides all the ground truth data for evaluation.

VII. Experimental Results

A. Average Localization Precision

As stated in Section V, the gap constraint plays a key role in detecting the boundaries of copies. In addition, since the length of list returned for each query frame determines the tolerance degree to the matching errors, it also has an important effect on the localization precision. In this section, we have performed experiments to evaluate the effects of the gap constraint and list length. To evaluate the localization precision in the query stream, the average copy overlap degree is computed over the queries containing copies. A total of 12 runs are performed with various combinations of the gap and list lengths. The results are displayed in Table III. The columns show the localization performance when varying the gap lengths, and the rows display the performance when varying the list lengths. The high overlap degree means high localization precision. There is a clear tendency that the localization precision is improved with increasing either the gap length or the list length. As discussed above, the gap length determines the gap between the first frame of a possible copy and previous non-relevant frames. A big gap length means a higher possibility that the candidate clip starting from this frame can be a copy. In addition, the gap constraint also provides a mechanism for tolerating the frame dropping transformation. The bigger the gap length is, the stronger the tolerance degree to frame dropping is. Similarly, a list with large length gives more chances to identify a complete copy. Similar tendency can also be obtained for the localization precision in the reference database. Table IV shows the experimental results.

B. Localization Precision on Separate Queries

In addition to the average overlap degree measurement, we also evaluate the effectiveness of the proposed method on the separate queries. The histograms of both the query and reference overlap degrees are given in Figs. 4 and 5, respectively. For most of queries, our algorithm achieves high localization precision in both the query and reference videos, though it fails in localizing a few of queries, such as query 8. In two figures, some empty bins mean that our copy detection system fails when detecting the copies in the corresponding queries. We will discuss it in more detail when comparing the proposed method with state-of-the-art systems. Note that only the run with the gap length 3 and list length 200 is displayed. In fact, similar conclusion can also be obtained for other runs.

TABLE III
Evaluation on Average Copy Overlap Degree

Δt \ M	1	10	100	200
1	0.623897	0.729671	0.772739	0.802450
2	0.730087	0.802712	0.837362	0.913016
3	0.754871	0.816329	0.882743	0.935390

TABLE IV
Evaluation on Average Reference Overlap Degree

Δt \ M	1	10	100	200
1	0.835474	0.855282	0.870448	0.906880
2	0.859533	0.893803	0.894382	0.946784
3	0.871408	0.905232	0.924617	0.950131

C. Localization Precision on Varied Transformations

A copy is usually a distorted version of its original video clip. Different transformations will distort the original video signal in different manners. A good copy detection system should not only tolerate various distortions but also distinguish copies from non-copy clips. In this section, we carry out experiments to evaluate localization performance on different transformations. Ten transformations used in TRECVID 2008 CBCD task are employed in our experiments. For each transformation, we have five queries, and four of them indeed insert copies in their video streams. Likewise, we use copy overlap degree and reference overlap degree to evaluate the localization precision. The difference is that the average overlap degree for each transformation is obtained by averaging over the queries undergoing the transformation.

The histograms of query and reference overlap degrees are plotted in Figs. 6 and 7, respectively. Our algorithm achieves high localization precision in both the query and reference videos for most of transformation types. This means that the proposed algorithm indeed provides a robust mechanism for tolerating various transformations. Still, only the run with the gap length 3 and list length 200 is displayed, and similar conclusion can also be obtained for other runs.

D. Comparison with State-of-the-Art Methods

In the above subsections, we validate the effectiveness of the proposed algorithm by tuning different impact factors. In this subsection, we will compare the proposed algorithm with the state-of-the-art techniques. In our experiment, three leading copy detection systems [4], [13], which achieve the

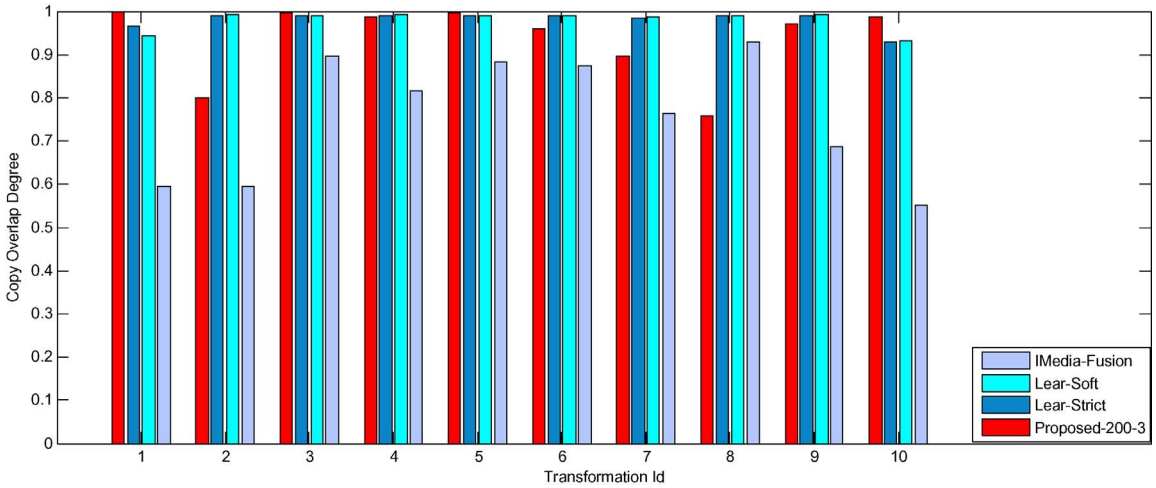

Fig. 8. Comparison with state-of-the-art copy detection systems on copy overlap degree with varied transformations.

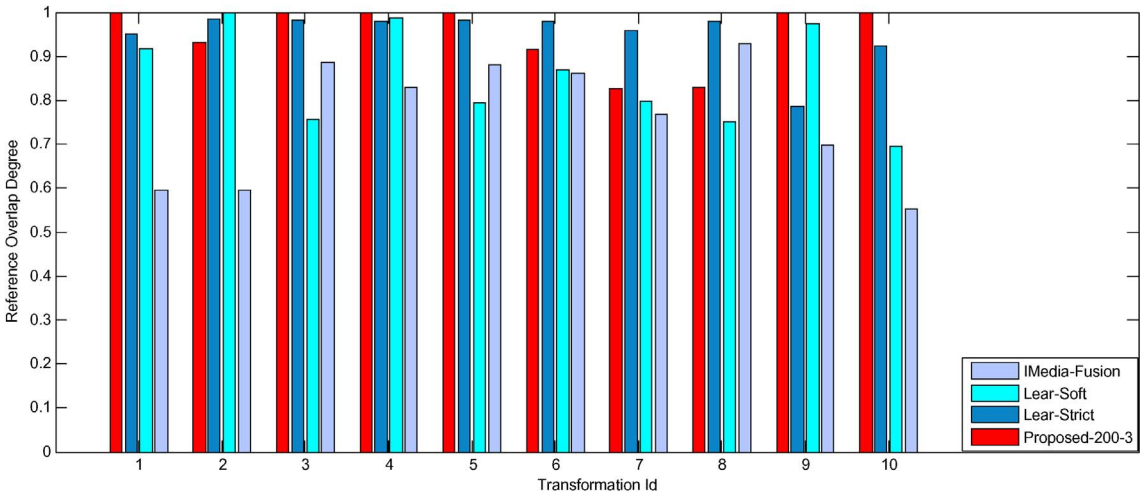

Fig. 9. Comparison with state-of-the-art copy detection systems on reference overlap degree with varied transformations.

best detection performance in NIST TRECVID 2008 CBCD task, are used for comparison. These three systems are named as *Lear-Strict*, *Lear-Soft*, and *IMedia-Fusion*, respectively. Note that both *Lear-Strict* and *Lear-Soft* come from the same scheme *INRIA-LEAR* with different parameter settings in [4] and [36]. The main difference between *Lear-Strict* and *Lear-Soft* is that *Lear-Strict* only keeps the top-ranked copies.

First, we compare four copy detection systems on the average precision in both the copy and reference localizations. Table V lists all the evaluation results. For the copy localization, the proposed method obtains quite good performance though it is a little weaker than the top two systems. For the reference localization, the proposed method gets almost the best performance. That is, the proposed method indeed works well for localizing the copy in both the query and reference videos.

Second, we compare these systems on the localization performance for different transformations. The histograms of both query and reference overlap degrees are plotted in Figs. 8 and 9, respectively. For the copy localization, the proposed method obtains the best performance for four transformations, i.e., T1, T3, T5, and T10. The performances for other transformations except T8 are also good. After analyzing the similarity search results, we find that for the copies undergoing T8 transformation, even no true relevant reference frames are returned for the copy frames near the boundaries. This means that only a part of copy is detected, which leads to low query overlap degree. The main reason lies in to some extent that the used SIFT descriptor is not robust to flip transformation and we do not handle this operation further. In our future work, we will take the mirroring operation into account in designing feature extraction scheme.

For the reference localization, the proposed method achieves the best performance for six of ten transformations. The performance for other transformations is also comparable with

TABLE V
COMPARISON ON AVERAGE COPY AND REFERENCE OVERLAP DEGREE

System	Average Copy Overlap Degree	Average Reference Overlap Degree
Lear-Strict	0.981406	0.950439
Lear-Soft	0.979899	0.843860
IMedia-Fusion	0.770349	0.770694
Proposed	0.935390	0.950131

TABLE VI
COMPARISON ON MISS RATE AND FALSE ALARM RATE

System	R_{miss}	R_{FA}
Lear-Strict	0.000000	0.166667
Lear-Soft	0.075000	0.663636
IMedia-Fusion	0.200000	0.360000
Proposed	0.225000	0.288900

TABLE VII
EFFECT OF VOCABULARY SIZE ON THE OVERALL PERFORMANCE

Vocabulary Size	Average Copy Overlap Degree	Average Reference Overlap Degree	R_{miss}	R_{FA}
10 000	0.896870	0.914383	0.325000	0.372100
100 000	0.935390	0.950131	0.225000	0.288900

the best ones. This means that our proposed copy detection method is indeed robust to various video distortions. Encouragingly, the proposed method achieves the best localization performance for the most complex transformation T10 in both the query and reference videos. That is, our method can tolerate severe signal distortions.

Finally, we compare all systems on the overall detection performance, i.e., the miss rate and the false rate. Table VI lists all the evaluation results. Although the performance of the proposed algorithm is not as good as other systems in the miss rate, it still achieves a comparable performance with the system *IMedia-Fusion*, especially in the false alarm rate. As discussed in [6], a complete copy detection system comprises a few key components including the sampling rate of key frames, feature extraction, similarity search as well as frame fusion results. The overall performance of such a system depends on the aggregated result of all the constituents. In our scheme, we focus mainly on the frame fusion stage. Although the proposed frame fusion method achieves high localization precision, it indeed slightly reduces the detection precision. In fact, this issue can be solved by enhancing the other components. For example, the detection performance can be notably improved by simply changing the size of visual vocabulary since there is a tradeoff between the robustness and discriminability of bag-of-features [23]. A larger size of visual vocabulary means better discriminability capability. The experimental results are illustrated in Table VII. When we change the vocabulary size from 10 000 to 100 000, the detection performance has a remarkable improvement.

E. Evaluation on Frame Fusion Efficiency

In the proposed approach, the frame fusion problem can be dealt with in real-time due to the high efficiency of the Viterbi-like algorithm. However, real-time frame fusion is a vital but not the only step toward real-time copy detection. The detection system can still be slow if the similar reference frame cannot be retrieved in real-time. Since our work focuses mainly on frame fusion, real-time similar search is out of scope of this paper. Because we cannot get the copy detection systems in [4] due to the limitation of computer power and copyright, we directly adopt their output results available in Trecvid'08 [30]. However, since the output results do not contain the time cost information in the frame fusion stage, we cannot make an exact comparison on frame fusion efficiency. Fortunately, the authors of *INRIA-LEAR* give detailed information of time cost about every stage of copy detection in their new paper [5] to be appeared. While that information is achieved via different database and computer configures, we manage to present a rough comparison of frame fusion efficiency. Since the frame rate of video is 25 frames per second (F/S), the system must process a query frame within $1/25 = 0.04$ s (i.e., 0.04S/F) if it wants to achieve a real-time frame fusion. As shown in Table VIII, although *INRIA-LEAR* can perform a real-time frame fusion (0.0021S/F $<$ 0.04S/F) when frame sampling rate is very low (i.e., 1F/S), it loses the real-time property (0.0438S/F $>$ 0.04S/F) when frame sampling rate is high (i.e., 12.5F/S which is its default setting). In contrast, the proposed frame fusion method with a compromised sampling rate (i.e., 3F/S) can fuse candidate reference frames in real-time even if the main parameter setting varies greatly, see 0.0374S/F for Frame list = 200 and 0.0014S/F for Frame list = 50, respectively. Note that when Frame list = 50, the frame fusion efficiency of the proposed scheme outperforms *INRIA-LEAR* even if its sampling rate is higher than one of the sampling rates of *INRIA-LEAR*, compared 0.0014S/F with 0.0021S/F and 0.0438S/F. Here, the average frame fusion time is calculated over the total number of query frames (not only the query keyframes). In conclusion, our proposed system indeed achieves real-time frame fusion even with less powerful computer than *INRIA-LEAR*.

TABLE VIII
COMPARISON ON RUNTIME OF THE PROCESSING STAGES (HOURS: MINUTES: SECONDS)

System	Stages	Frame Sampling Rate		
		1 F/S	3F/S	12.5F/S
INRIA-LEAR	Frame fusion	0H:12M	N/A	4:16
	Average frame fusion	0.0021S/F	N/A	0.0438S/F
Proposed system (frame list = 200)	Frame fusion	N/A	0H:55M	N/A
	Average frame fusion	N/A	0.0374S/F	N/A
Proposed system (frame list = 50)	Frame fusion	N/A	10M:28S	N/A
	Average frame fusion	N/A	0.0014S/F	N/A

For *INRIA-LEAR* in [4] and [36], the database contains 21 h 11 min and the total length of queries is 3 h 54 min, whereas lengths of database and queries are about 200 h and 59 min for our system, respectively.

VIII. CONCLUSION

In this paper, we proposed a frame fusion based copy detection approach, which involves similar frame search and frame fusion. Our work focused mainly on the critical frame fusion stage. The proposed frame fusion scheme employs a Viterbi-like dynamic programming algorithm that comprises an online back-tracking strategy and three relaxed constraints. The major advantages of this scheme lie in the following three novel aspects: 1) propose a real-time frame fusion, which can apply to the copy detection problem in a continuous query video stream; 2) relax strict temporal consistency constraints

to handle complex transformations and tolerate matching offset and misalignment; and 3) avoid the difficult problem of threshold selection and provide precisely temporal localization. In addition, since the proposed frame fusion procedure is under a dynamic programming framework, the fusion efficiency is very high. The experimental results show that the proposed approach achieves high localization accuracy in both the query stream and the reference database and provides good tolerance to some difficult video transformations.

REFERENCES

[1] T. Can and P. Duygulu, "Searching for repeated video sequences," in *Proc. ACM Int. Workshop Multimedia Information Retrieval*, 2007, pp. 207–216.

[2] L. Chen and F. W. M. Stentiford, "Video sequence matching based on temporal ordinal measurement," *Pattern Recognit. Lett.*, vol. 29, no. 13, pp. 1824–1831, 2008.

[3] C.-Y. Chiu, C.-S. Chen, and L.-F. Chien, "A framework for handling spatiotemporal variations in video copy detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 412–417, Mar. 2008.

[4] M. Douze, A. Gaidon, H. Jegou, M. Marszatek, and C. Schmid, "Inrialear's video copy detection system," in *Proc. TRECVID*, 2008.

[5] 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, Jun. 2010.

[6] N. Gengembre and S.-A. Berrani, "A probabilistic framework for fusing frame-based searches within a video copy detection system," in *Proc. ACM Int. Conf. Content-Based Image Video Retrieval*, 2008, pp. 211–220.

[7] A. Hampapur and R. M. Bolle, "VideoGREP: Video copy detection using inverted file indices," IBM Research Division. Thomas. J. Watson Research Center, Tech. Rep., 2002 [Online]. Available: http://www.research.ibm.com/ecvg/pubs/arun-vgrep.html

[8] X.-S. Hua, X. Chen, and H.-J. Zhang, "Robust video signature based on ordinal measure," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1. Oct. 2004, pp. 685–688.

[9] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vision*, 2008, pp. 304–317.

[10] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Toward optimal bag-of-features for object categorization and semantic video retrieval," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2007, pp. 494–501.

[11] A. Joly and O. Buisson, "Discriminant local features selection using efficient density estimation in a large database," in *Proc. ACM SIGMM Int. Workshop Multimedia Inform. Retrieval*, 2005, pp. 201–208.

[12] 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.

[13] A. Joly, J. Law-To, and N. Boujemaa, "INRIA-IMEDIA TRECVID 2008: Video copy detection," in *Proc. TRECVID*, 2008.

[14] Y. Ke, R. Sukthankar, and L. Huston, "An efficient parts-based near-duplicate and sub-image retrieval system," in *Proc. ACM Int. Conf. Multimedia*, 2004, pp. 869–876.

[15] H.-S. Kim, J. Lee, H. Liu, and D. Lee, "Video linkage: Group based copied video detection," in *Proc. ACM Int. Conf. Content-Based Image Video Retrieval*, 2008, pp. 397–406.

[16] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 127–132, Jan. 2005.

[17] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa, "Robust voting algorithm based on labels of behavior for video copy detection," in *Proc. ACM Int. Conf. Multimedia*, 2006, pp. 835–844.

[18] S. Lee and C. D. Yoo, "Robust video fingerprinting for content-based video identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 983–988, Jul. 2008.

[19] Lemur. *The Lemur Toolkit for Language Modeling and Information Retrieval* [Online]. Available: http://www.lemurproject.org

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] K. Mikolajczyk, *Binaries for Affine Covariant Region Descriptors* [Online]. Available: http://www.robots.ox.ac.uk/_vgg/research/affine

[22] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. Comput. Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[23] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Oct. 2006, pp. 2161–2168.

[24] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Visual Lecture Notes in Computer Science*, vol. 2. Springer, 2002, pp. 117–128.

[25] C. Petersohn, "TRECVID 2004: Shot boundary detection system," in *Proc. TRECVID*, 2004.

[26] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[27] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," *J. Am. Soc. Inform. Sci.*, vol. 27, no. 3, pp. 129–146, 1976.

[28] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3," in *Proc. TREC-3*, 1995, pp. 109–126.

[29] S. I. Satoh, M. Takimoto, and J. Adachi, "Scene duplicate detection from videos based on trajectories of feature points," in *Proc. ACM Int. Workshop Multimedia Inform. Retrieval*, 2007, pp. 237–244.

[30] TRECVID. (2008). *TREC Video Retrieval Evaluation* [Online]. Available: http://www-nlpir.nist.gov/projects/trecvid

[31] S. Uchida, A. Mori, R. Kurazume, R.-I. Taniguchi, and T. Hasegawa, "Logical DP matching for detecting similar subsequence," in *Proc. 8th ACCV*, vol. 4843. Nov. 2007, pp. 628–637.

[32] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms* [Online]. Available: http://www.vlfeat.org

[33] K. Vaiapury, P. K. Atrey, M. S. Kankanhalli, and K. Ramakrishnan, "Non-identical duplicate video detection using the SIFT method," in *Proc. IET Int. Conf. Visual Inform. Eng.*, 2006, pp. 537–542.

[34] X. Wu, C.-H. Ngo, A. G. 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.

[35] X. F. Yang, Q. B. Sun, and Q. Tian, "Content-based video identification: A survey," in *Proc. Int. Conf. Inform. Technol. Res. Educ.*, 2003, pp. 50–54.

[36] X. M. Zhou, X. F. 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, May 2009.

[37] J. K. Zhu, S. C. H. Hoi, M. R. Lyu, and S. C. Yan, "Near-duplicate keyframe retrieval by nonrigid image matching," in *Proc. ACM Int. Conf. Multimedia*, 2008, pp. 41–50.

Shikui Wei received the B.E. degree in electrical engineering from Hebei University, Hebei, China, in 2003, the M.E. degree in signal and information processing from Beijing Jiaotong University (BJTU), Beijing, China, in 2005, and the Ph.D. degree from the Institute of Information Science, BJTU, in 2010.

Currently, he is a Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include computer vision, image/video analysis and retrieval, and copy detection.

Yao Zhao (M'05) received the B.E. degree from Fuzhou University, Fuzhou, China, in 1989, and the M.E. degree from Southeast University, Nanjing, China, in 1992, both from the Radio Engineering Department, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1996.

He was an Associate Professor with BJTU in 1998 and a Professor in 2001. From 2001 to 2002, he was a Senior Research Fellow with the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. He is currently the Director of the Institute of Information Science, BJTU. Currently, he is leading several national research projects from the 973 Program, the 863 Program, the National Science Foundation of China, and the Fok Ying Tong Education Foundation. His current research interests include image/video coding, fractals, digital watermarking, and content-based image retrieval.

Ce Zhu (M'03–SM'04) received the B.S. degree from Sichuan University, Chengdu, China, and the M.Eng. and Ph.D. degrees from Southeast University, Nanjing, China, in 1989, 1992, and 1994, respectively, all in electronic and information engineering.

He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He has authored or co-authored over 80 papers and holds two granted patents. He has edited one book and contributed two book chapters. His current research interests include image/video coding, streaming and processing, joint source-channel coding, 3-D video, multimedia systems, and applications.

Dr. Zhu currently serves as an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING and IEEE SIGNAL PROCESSING LETTERS. He has served on technical/program committees, organizing committees, and as a track/session chair for over 40 international conferences.

Changsheng Xu (M'97–SM'99) is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is the Executive Director of the China-Singapore Institute of Digital Media, Singapore. His current research interests include multimedia content analysis, image processing, pattern recognition, and computer vision. He has published over 200 refereed book chapters, journal, and conference papers in these areas.

He is an Associate Editor of the *ACM/Springer Multimedia Systems Journal* and received the 2008 Best Editorial Member Award. He is on the Editorial Board of the *Journal of Multimedia* and the *International Journal of Multimedia Intelligence and Security*. He serves as a guest editor of special issues on the IEEE TRANSACTIONS ON MULTIMEDIA, *ACM Transactions on Multimedia Computing, Communication and Applications*, *International Journal of Multimedia Tools and Applications*, and *Pattern Recognition Letter*. He served as the Program Co-Chair of ACM Multimedia in 2009, the Program Co-Chair of the 2009 International Conference on Internet Multimedia Computing and Services, the General Co-Chair of the Pacific-Rim Conference on Multimedia in 2008, a Short Paper Co-Chair of ACM Multimedia in 2008, the General Co-Chair of the 2007 Asia-Pacific Workshop on Visual Information Processing, the Program Co-Chair of the 2006 Asia-Pacific Workshop on Visual Information Processing, Industry Track Chair and Area Chair of the 2007 International Conference on Multimedia Modeling. He is on organizing committees and program committees in many prestigious multimedia conferences, including ACM Multimedia, ICME, PCM, CIVR, MMM, among others. He is the Director of Programs of ACM SIG Multimedia Beijing Chapter. He is a member of ACM.

Zhenfeng Zhu received the B.E. and M.E. degrees from the Wuhan University of Science and Engineering, Wuhan, China, and the Harbin Institute of Technology, Harbin, China, in 1996 and 2001, respectively, both in electromechanical engineering, and the Ph.D. degree in pattern recognition and intelligence system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2005.

He joined the Institute of Information Science, Beijing Jiaotong University, Beijing, in 2005, and is currently an Associate Professor. His current research interests include image and video understanding, pattern recognition, and computer vision.