Content-Based Copy Detection through Multimodal Feature Representation and Temporal Pyramid Matching

LUNTIAN MOU, TIEJUN HUANG, YONGHONG TIAN, MENGLIN JIANG, and WEN GAO,
Peking University

Content-based copy detection (CBCD) is drawing increasing attention as an alternative technology to watermarking for video identification and copyright protection. In this article, we present a comprehensive method to detect copies that are subjected to complicated transformations. A multimodal feature representation scheme is designed to exploit the complementarity of audio features, global and local visual features so that optimal overall robustness to a wide range of complicated modifications can be achieved. Meanwhile, a temporal pyramid matching algorithm is proposed to assemble frame-level similarity search results into sequence-level matching results through similarity evaluation over multiple temporal granularities. Additionally, inverted indexing and locality sensitive hashing (LSH) are also adopted to speed up similarity search. Experimental results over benchmarking datasets of TRECVID 2010 and 2009 demonstrate that the proposed method outperforms other methods for most transformations in terms of copy detection accuracy. The evaluation results also suggest that our method can achieve competitive copy localization preciseness.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering, Search process*; I.4.7 [**Image Processing and Computer Vision**]: Feature Measurement—*Feature representation*

General Terms: Algorithms, Design, Experimentation, Performance, Security, Verification

Additional Key Words and Phrases: Content-based copy detection, feature representation, temporal pyramid matching

ACM Reference Format:

Mou, L., Huang, T., Tian, Y., Jiang, M., and Gao, W. 2013. Content-based copy detection through multimodal feature representation and temporal pyramid matching. ACM Trans. Multimedia Comput. Commun. Appl. 10, 1, Article 5 (December 2013), 20 pages.

DOI: http://dx.doi.org/10.1145/2542205.2542208

1. INTRODUCTION

As a new communication medium characterized by instant dissemination of information, the Internet is expediting the dissemination of visual content while aggravating the proliferation of unauthorized copies. According to YouTube,[1] in May 2012, it handles 4 billion views per day with over 72 hours of video uploaded to the site every single minute. Inevitably, copyright infringement and disputes will

[1]http://www.youtube.com.

This work was partially supported by grants from the Chinese National Natural Science Foundation under contract no. 61035001 and National Basic Research Program of China under contract no. 2009CB320906.

Authors' address: L. Mou, T. Huang, Y. Tian, M. Jiang, and W. Gao, No. 2 Science Building, Peking University, No. 5 Yiheyuan Road, Haidan District, Beijing 1000871, China; Correspondence email: tjhuang@pku.edu.cn.

DOI: http://dx.doi.org/10.1145/2542205.2542208

Fig. 1. Examples of video transformations. T1: camcording; T2: picture in picture; T3: insertion of pattern; T4: re-encoding; T5: change of gamma; T6: decrease in quality; T8: post production; T10: combination of 3 randomly chosen transformations. Note that T7 and T9 have been dropped by TRECVID 2010. Image courtesy Internet Archive [Internet Archive].

occur if no technical measures are taken for checking the ownership of each video clip uploaded to the site. YouTube once was criticized for failing to ensure that uploaded videos comply with the law of copyright. To solve the dilemma, YouTube has introduced Content ID to identify uploaded videos against a database of copyrighted videos by comparing respective ID files extracted from video content.

Traditionally, watermarking embeds a piece of copyright-related information imperceptibly into a video and extracts that information at a later time to claim the ownership. However, its robustness under complicated transformations is questionable. And, despite the imperceptibility of embedded watermarks, the quality of a watermarked video is inevitably degraded to some extent. Last but not least, watermarking is vulnerable to the analog hole: a watermarked video can be recorded and copied through analog means, then re-digitized and distributed as a clear video with no watermark [Huang et al. 2010].

Fortunately, content-based copy detection was proposed recently as an alternative technology to watermarking. As stated in Hampapur and Bolle [2001], the primary idea of content-based copy detection is "the media itself is the watermark". According to TRECVID, content-based copy detection addresses the issue that automatically analyzes a query video's content to determine whether it contains a copy from a given database of reference videos and if so, from where in the database the copy comes [Over et al. 2010]. Here, a copy is a segment of video derived from another video, usually by means of various transformations. Unlike watermarking, content-based copy detection extracts certain intrinsic features from a video and makes them or their dimension-reduced version (i.e., feature vectors) a unique representation of the video content. Such a representation (called mediaprint in Huang et al. [2010]; and video fingerprint or video signature elsewhere) is expected to be robust to a wide range of transformations. Obviously, mediaprint generation does not cause any harm to the quality of a video. Furthermore, content-based copy detection is immune from the analog hole because a mediaprint can be extracted anywhere, any time from any video during its life cycle.

However, content-based copy detection faces several challenges. First, the quality of many copies may have been severely degraded (see Figure 1), and to make things worse, the content of many copies may have been significantly changed. This makes extracting largely-invariant mediaprints from a copy and its original not an easy job. Second, mediaprints should ideally have both competing properties of robustness and discriminability. Third, for frame-based methods without proper temporal fusing mechanism, copies are difficult to be accurately detected and precisely located.

To address these challenges, we propose a comprehensive method for effectively and efficiently detecting copies subjected to complicated transformations. To cope with various audio and video transformations, multimodal feature representation is proposed to exploit the complementary characteristics of audio features and global and local visual features. To resist possible temporal transformations and precisely locate the exact copy segment in a query clip as well as its original in a reference dataset, a frame fusion algorithm, named temporal pyramid matching (TPM) is used.

To evaluate the performance of the proposed method, we have carried out extensive experiments on the benchmarking datasets of TRECVID 2010 and TRECVID 2009 [Over et al. 2010]. Experimental results show that the proposed method outperforms other methods for most transformations in copy detection accuracy at the CBCD competition of TRECVID. Meanwhile, it achieves a competitive performance in copy localization.

Since a part of our work has been published in conference papers [Li et al. 2010; Tian et al. 2011], the main extension and modification to our previous work, are described as follows.

(1) *Improved Multimodal Feature Representation.* Primarily, one feature of dense color SIFT (DC-SIFT) substitutes for two previously used local visual features, that is, SIFT [Lowe 2004] and SURF [Bay et al. 2006]. Other improvements include changing the parameter K of a K-means algorithm used for visual vocabulary calculation in the Bag of Words (BoW) representation from the original 400 to 800, using 16 bins to quantize the orientation of each keypoint instead of the previous 8 bins, and reducing the audio feature named WASF from 126D to 72D.

(2) *Improved Temporal Pyramid Matching.* A pyramid structure is added to intuitively depict the video matching process. And the computational complexity of TPM is analyzed. Moreover, it is evaluated against a state-of-the-art Viterbi-based frame fusion algorithm in Wei et al. [2011].

(3) *Enhanced Experiments.* Besides TRECVID 2010, the performance of the proposed method is also evaluated over TRECVID 2009. The two datasets are significantly different in that TRECVID 2009 comprises professional videos while TRECVID 2010 consists of web videos.

(4) *New Perspective.* A Discussion section is added. After analyzing the drawbacks of current methods, a new perspective on the video copy detection problem is brought up, which includes a new adaboost like architecture to speed up the system response, and an invariant feature analysis model based on typical perceptual models to obtain ideal invariant features.

The contribution of this article mainly lies in two aspects. One is the multimodal feature representation to exploit the complementarity of multimodal features for overall robustness to a wide range of complicated modifications. The other is the temporal pyramid matching algorithm to compute similarity between two video sequences over multiple temporal granularities.

The remainder of this article is organized as follows. In Section 2, the problem of content-based copy detection is formulated. Section 3 presents an overview of related work. In Section 4, the proposed method is described in detail. Then, performance evaluation results are described in Section 5. And a new perspective is provided in Section 6. Finally, Section 7 concludes the paper and outlines future work.

2. PROBLEM FORMULATION

The task of content-based copy detection can be formulated as follows. Given $\mathbb{Q} = \{q_i\}(1 \leq i \leq n)$ and $\mathbb{R} = \{r_j\}(1 \leq j \leq m)$, where q_i and r_j denotes a query video, and a reference video, respectively,

(1) The task is to examine whether the statement of $\exists r_j \in \mathbb{R}$ such that $C(q_i, r_j)$ holds or not, where $C(x, y)$ stands for the binary relationship of x being a copy of y;

Fig. 2. Two pairs of noncopy query clips and corresponding similar reference videos. In each pair, a noncopy query keyframe is shown on the left, with a similar reference keyframe on the right. Image courtesy Internet Archive [Internet Archive].

(2) If this statement holds, then return $[t^B(q_i), t^E(q_i)]$ and $[t^B(r_j), t^E(r_j)]$, which are precise timestamps for the beginning and end of the copy segment in the query q_i, and those of its original in the reference r_j, respectively.

To solve the first problem, invariant features should be extracted from the perceptual content of a video as its representation and correspondingly a distance measurement should be adopted to calculate the similarity between a query video and a reference video. For feature representation, the challenge mainly comes from the fact that the real-world transformations are complicated and there exists no such one-for-all feature that remains robust on all transformations. Transformations may be performed on the components of audio, video, or both of a video file or stream. While audio transformations are composed of relatively few modifications such as compression, companding and mixing with speech, video transformations are much more diverse and complicated. Video transformations can be roughly classified into two categories, namely, spatial modifications and temporal modifications, which can be further divided into the subcategories of content-preserving and content-altering respectively. Spatial content-preserving operations mainly consist of format conversion and quality reduction modifications such as brightness enhancement, noise addition, resolution change, re-encoding and filtering. The perceptual content of a video will be largely maintained after such operations. In contrast, the perceptual content of a video will usually be notably modified after spatial content-altering transformations, for example, cropping, picture-in-picture and pattern insertion. Likewise, video content will be largely preserved after temporal content-preserving transformations including frame rate change and video speed adjustment, which alter the number of frames but keep their order. On the contrary, video content will be significantly changed by temporal content-altering transformations by inserting, deleting, replacing or reordering at any instant along the temporal axis. Besides, discriminability is also demanded so that different video content can be distinguished from each other. This is especially true when non-copy query clips are extremely similar to certain reference videos (see Figure 2).

To solve the second problem, a video matching algorithm should be used to localize the copy in the reference video as well as in the query clip. This implies that the video matching algorithm should be a kind of partial matching algorithm. Obviously, traditional sequence matching algorithms based on the sliding-window mechanism are not applicable due to their high computational complexity. Alternatives could be frame fusion-based methods, which fuse frame-level similarity search results into sequence-level matching results by imposing certain temporal constraints on the fusion process.

3. RELATED WORK

Content-based copy detection mainly consists of two key techniques, namely, feature representation and video matching. This section presents a brief review of related work from these two aspects.

3.1 Feature Representation

According to their intrinsic characteristics, the features used in existing work can be classified into two categories, namely global features and local features.

Global features are generally based on the statistics of the entire frame or the whole clip. Thus, they have the advantages of compactness and low computational complexity. It is observed that global features are largely invariant under content-preserving transformations. Among them, the ordinal measure has been widely used in video copy detection [Hampapur and Bolle 2001; Hua et al. 2004]. Although the ordinal measure has the advantages of compact representation and low computational complexity, it has a drawback: ordinal signatures will significantly change in the case of local luminance variations. Therefore, block-based differential luminance signatures are proposed in Iwamoto et al. [2006] and Lee and Yoo [2006]. However, block-based spatial signatures such as ordinal and differential signatures are susceptible to geometric transformations such as rotation, cropping, and scaling that changes aspect ratios. Meanwhile, many methods have been proposed to compute video signatures from a specific transform domain, such as polar Fourier Transform [Swaminathan et al. 2006], Radon Transform [De Roover et al. 2005] or Singular Value Decomposition (SVD) [Radhakrishnan and Bauer 2008]. It is reported that they have good resilience to affine transformations such as shift and rotation, but are still prone to cropping. Especially, block-based differential signatures computed from DCT domain are widely studied [Lin and Chang 2001; Ahmed et al. 2010] for computationally efficient signatures.

Besides spatial signatures, temporal signatures are also proposed to make use of temporal characteristics of a video such as shot duration [Shivakumar 1999] and locations of keyframes [Cheung and Zakhor 2003]. Similarly, temporal ordinal signature [Chen and Stentiford 2008] and temporal differential signatures [Kim and Vasudev 2005; Oostveen et al. 2002] can be used for copy detection. With respect to audio transformations, audio signatures are reviewed in Cano et al. [2005]. It is reported that the audio signature called WASF in Chen and Huang [2008] outperforms Mel-Frequency Cepstrum Coefficients (MFCC) in Cano et al. [2002] at the distortion of speed acceleration. Multimodal feature representation is increasingly becoming an important trend of feature representation. In fact, the CBCD task of TRECVID already required participants to submit copy detection results achieved by using audio+video queries [Over et al. 2010].

However, global features cannot effectively deal with more complex transformations such as post-production transformations, which usually discard or replace a region of an original frame or a portion of frames in an original clip. Instead, local features are by nature resistant to such content-altering operations since a part of original content always remains in the copy. Local features used in video copy detection are mostly based on the interest point detection and local descriptor calculation [Douze et al. 2010; Joly et al. 2007; Law-To et al. 2006]. According to evaluation results in Mikolajczyk and Schmid [2005], some derivations of SIFT [Lowe 2004] outperform other local descriptors. To accelerate matching of local features, a technique of bag-of-words (BoW) [Sivic and Zisserman 2003] is used frequently in recent literature [Douze et al. 2010].

3.2 Video Matching

Video matching methods can be roughly classified into two categories: sequence matching and frame-fusion-based matching.

The basic idea of sequence matching is direct frame-to-frame matching of two video clips. Due to the fact that a query sequence is usually much shorter than a reference sequence, a sliding window with the same size as the query clip is moved frame by frame along the reference video. Examples of sequence matching methods can be found in the literature [Chen and Stentiford 2008; Hampapur

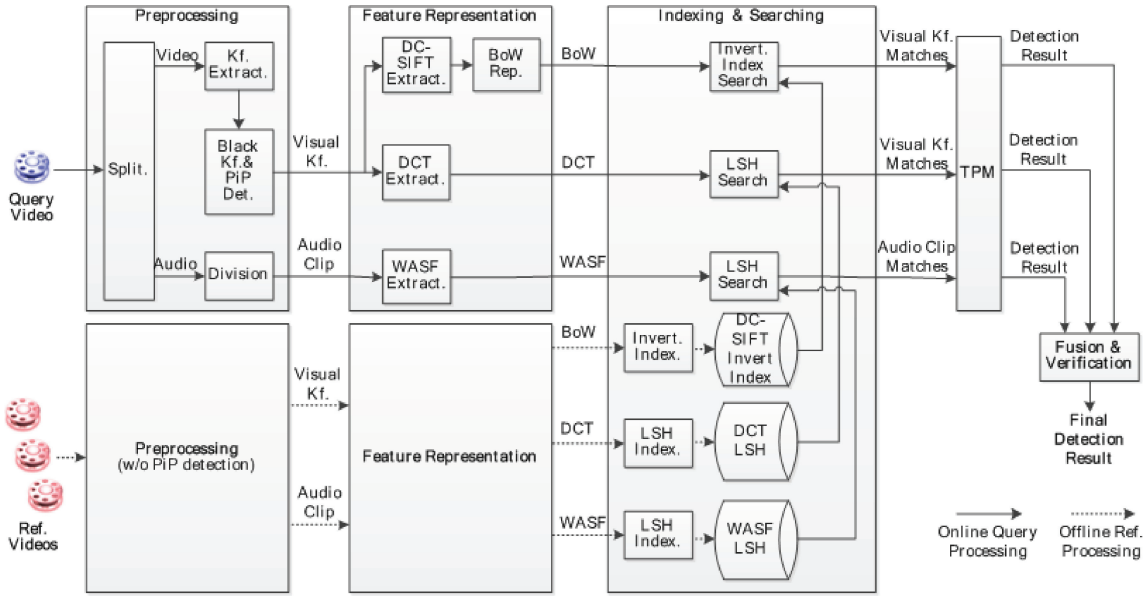

Fig. 3. Framework of the proposed method.

and Bolle 2001; Hua et al. 2004; Kim and Vasudev 2005; Oostveen et al. 2002]. Obviously, an implicit assumption here is that the whole query clip is or is not a copy. But in practice, a copy is probably just a small segment of a query clip. Another major drawback of sequence matching is that matching the query clip with all possible subsequences of a reference video leads to high-computational complexity. Additionally, a copy cannot be effectively detected and precisely localized if post-production transforms such as frame dropping are involved in generating the copy.

A more flexible way for video matching has been recently proposed in Kim et al. [2008] and Wei et al. [2011]. This matching approach tries to search the reference database and obtain a list of similar reference frames for each query frame, and then to determine whether the query is a copy by fusing the obtained reference frames. However, for frame-fusion-based matching without proper temporal fusing mechanism, copies are difficult to be accurately detected and precisely localized. Therefore, a spatio-temporal post-filtering mechanism is presented in Douze et al. [2010] to keep only the frame matches that are consistent with a spatio-temporal model.

4. THE PROPOSED METHOD

In this article, we propose a comprehensive method (see Figure 3) to detect copies that are subjected to complicated transformations. Basically, two core techniques are employed by our method. One is multimodal feature representation, which exploits the complementary characteristics of audio features, global and local visual features to obtain overall robustness to a wide range of transformations. The other is temporal pyramid matching (TPM), which is a frame-to-sequence fusing algorithm used to convert frame-level similarity search results into sequence-level matching results.

4.1 Preprocessing

In our system, some preprocessing operations are performed before feature extraction.

(1) *Video/Audio Splitting*. A reference or query video clip is first split into components of a video and its accompanying audio.

(2) *Sampling*. Visual keyframes are obtained by uniformly sampling at a rate of 3 frames per second. Meanwhile, audio frames are obtained by dividing the audio signal into segments of 60ms with a 40ms overlap between consecutive frames. Thus, a 4-second-long audio clip is constructed by every 198 audio frames, with a 3.8 seconds overlap between adjacent clips.

(3) *Special Effects Detection*. Some special effects should be detected so that counter-measures can be taken for more robust copy detection. Currently, two kinds of special effects detection are employed, namely, black frame detection and picture in picture (PiP) detection. Black frames are detected and then discarded to avoid their disturbance on copy detection. For PiP detection, Hough transform [Ballard 1981] is employed in detecting and localizing the inserted foreground videos. Once PiP is detected, features will be extracted from the foreground and the whole original query keyframes respectively.

4.2 Multimodal Feature Representation

To keep robustness under diverse complicated transformations, we exploit complementarities among multimodal features. The multimodal features used in the proposed method are a local visual feature of dense color SIFT (DC-SIFT) [Lowe 2004], a global visual feature based on DCT and an audio feature named WASF [Chen and Huang 2008]. The complementary characteristics of visual features and audio features are obvious in that they represent different types of perceptual information. And the complementarities between local and global visual features lie in that the former can effectively handle spatial content-altering transformations while the latter is capable of resisting spatial content-preserving but quality-degrading operations. The local feature of DC-SIFT is chosen because it can better represent scenes as well as objects [Bosch et al. 2008]. The details of proposed multimodal feature representation are depicted as follows.

4.2.1 *Local Visual Feature.* DC-SIFT [Bosch et al. 2008] is employed to cope with spatial content-altering transformations such as camcording, picture in picture, pattern insertion and post-production operations. The DC-SIFT descriptor differs from the SIFT descriptor in that there is no keypoint detection and localization. Instead, regular grids with overlapping (i.e., dense sampling) are used for descriptor construction, and grids with single color values are discarded. Then, SIFT descriptors are computed at points on a regular grid with spacing M pixels, here, $M = 21$, 33, and 45. At each grid point, SIFT descriptors are computed over circular support patches with radii $r = 10.5$, 16.5, and 22.5 pixels. A 3×3 descriptor array for 8 orientation histogram bins is constructed for the three Lab components of each point. Consequently, each point is represented by a $9 \times 8 \times 3 = 216$ dimension SIFT descriptor. Furthermore, the Bag of Words (BoW) is applied in converting each feature vector into a visual word. For this purpose, a K-means algorithm ($K = 800$) is implemented on a random subset (10M) of the keypoint descriptors to calculate a visual vocabulary. However, BoW representation might lead to loss of discriminability for the descriptors. Therefore, the information of position, orientation and scale for each keypoint is taken into account so that only keypoints mapped to the same visual word and with roughly the same position, orientation and scale will be regarded as matches. In particular, the spatial region of a keyframe is divided into 1×1, 2×2 and 4×4 multi-granularity cells, leading to three integers (0-20) indexing the position of each keypoint. Similarly, the orientation and the scale of each keypoint are quantized into 16 and 2 bins, respectively.

4.2.2 *Global Visual Feature.* A new DCT feature is designed to resist spatial content-preserving operations such as strong re-encoding, change of gamma and decrease in quality. Like the DCT feature proposed by Lin and Chang [2001], our DCT feature is also based on the invariance retained by DCT coefficients at same positions of different blocks. But it differs from the original DCT feature mainly in that sub-band energy is introduced as an alternative to individual DCT coefficient (see Figure 4).

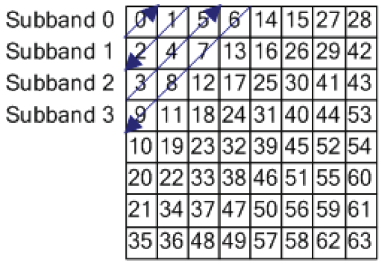

Fig. 4. Illustration of DCT sub-band indexing.

Specifically, the luminance component Y is first extracted from a decoded image and resized to the 64×64 resolution. Then, the resized image is divided into 64 8×8 blocks and a 2-D DCT is applied to each block to obtain the corresponding DCT coefficient matrix. After that, the first four sub-band energies of each block are calculated by summing up the absolute values of the DCT coefficients of each sub-band, resulting in a 256-D DCT feature vector:

$$\langle f_{0,0}, \ldots, f_{0,63}, \ldots, f_{3,0}, \ldots, f_{3,63} \rangle (0 \le i \le 3, 0 \le j \le 63), \tag{1}$$

where $f_{i,j}$ denotes the energy level of the sub-band i of block j. The magnitudes of each two neighboring components are compared to obtain one bit of 0 or 1 according to the following rule:

$$d_{i,j} = \begin{cases} 1, & \text{if } f_{i,j} \ge f_{i,j+1(\bmod 64)} \\ 0, & \text{otherwise} \end{cases} (0 \le i \le 3, 0 \le j \le 63). \tag{2}$$

Consequently, a 256-D descriptor is constructed by concatenating each obtained bit:

$$\langle d_{0,0}, \ldots, d_{0,63}, \ldots, d_{3,0}, \ldots, d_{3,63} \rangle (0 \le i \le 3, 0 \le j \le 63). \tag{1}$$

4.2.3 *Audio Feature.* Weighted audio spectrum flatness (WASF) [Chen and Huang 2008] extends the MPEG-7 descriptor - Audio Spectrum Flatness (ASF) in MPEG [2002] by introducing Human Auditory System (HAS) functions to weight audio data. In brief, a 14-D single WASF feature is first extracted from each 60ms audio frame. Then, for a 4-second-long audio clip with 198 audio frames, single WASF features of different frames are assembled and reduced to a 72-D WASF feature vector using Audio Signature specified in MPEG [2002].

4.3 Feature Indexing

Efficient indexing techniques should be exploited to speed up feature searching so that excessive computation demanded by exhaustive search can be avoided. DC-SIFT BoWs are stored into an inverted index with word IDs and quantized information of position, orientation and scale (see Figure 5). LSH [Gionis et al. 1999] is an algorithm for solving the approximate near neighbor search in high-dimensional spaces in sublinear time. In our system, an LSH is implemented to index DCT and WASF features.

4.4 Frame-Level Matching

Once the feature indexing for the reference database is ready, frame-level matching can be performed for each keyframe (audio clip) of the query video based on the multimodal feature representation.

4.4.1 *Similarity Measurement.* For DC-SIFT, the similarity between a query frame and a reference frame is defined as the average percentage of identical words to the larger number of words owned by

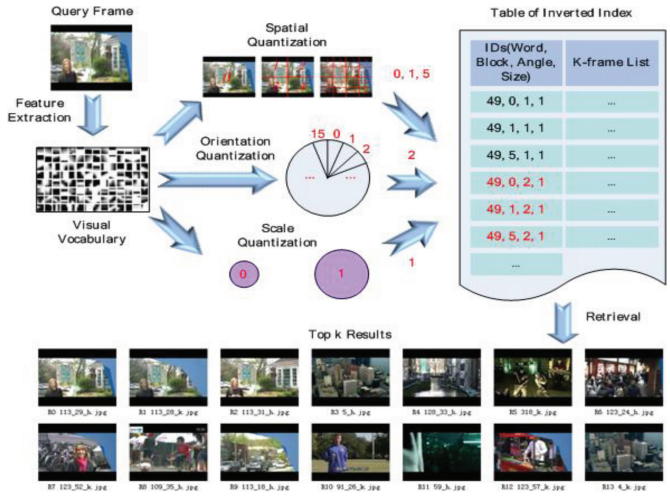

Fig. 5. Similarity search of keyframes using the inverted index of BoWs. Image courtesy Internet Archive [Internet Archive].

either the query frame or the reference frame at a specific spatial position (cell) i:

$$s_f^{DC-SIFT} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{\left|\langle w_i\rangle \cap \left\langle w_i'\right\rangle\right|}{Max\left(\left|\langle w_i\rangle\right|, \left|\left\langle w_i'\right\rangle\right|\right)}, \tag{2}$$

where w_i and w_i' denote words that appear at location i for the query and the reference video frames respectively, and N refers to the number of spatial positions, experimentally chosen as 21. With respect to DCT descriptors in binary representation and WASF descriptors in nonbinary representation, Hamming Distance and Euclidean Distance are adopted for similarity measurement respectively.

4.4.2 *Determination of Frame Matches.* Given a query video, top K_1 ($K_1 = 20$) reference keyframes (audio clips) are chosen as candidate matches for each query keyframes (audio clips) by each kind of feature (i.e., DC-SIFT, DCT, and WASF), resulting in a collection $\mathbb{M}_f$ of frame-level candidate matches m_f: $m_f = \left\langle v_q, t(q), v_r, t(r), s_f\right\rangle$, which means the keyframe (audio clip) of reference video v_r at timestamp $t(r)$ is a candidate match to the keyframe (audio clip) of query video v_q at timestamp $t(q)$ with a similarity of s_f. And histogram equalization is applied to make s_f values computed through different features more evenly distributed and comparable. Given the range of similarity score [0, 1] is divided into 1000 bins and the bin number Bin is calculated as $\lfloor s_f \times 1000 \rfloor$, and p_i is the learned frequency of the ith bin, then s_f can be normalized as:

$$s_f = \min\left\{1.0, \sum_{i=0}^{Bin} p_i\right\}. \tag{3}$$

4.5 Temporal Pyramid Matching

Inspired by spatial pyramid matching [Lazebnik et al. 2006] which conducts pyramid match kernel [Grauman and Darrell 2005] in 2-D image space, we adapt the kernel to 1-D temporal space, leading to the concept of temporal pyramid matching (TPM) (see Figure 6).

Although the frames of two matched video sequences should have consistent timestamps, due to temporal transformations, the timestamps of two matched frames of a copy and its original are allowed

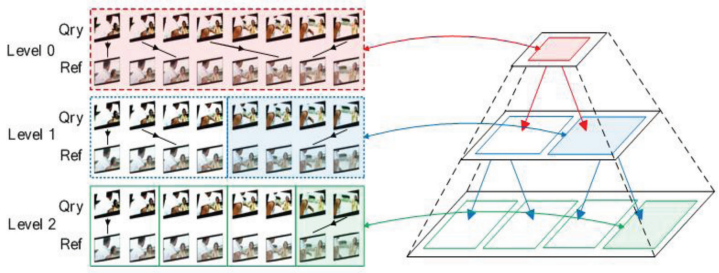

Fig. 6. A toy example for TPM (L = 2).

to have a moderate deviation. To address this problem, TPM partitions videos into increasingly finer temporal segments and compute video similarities over each granularity.

Given the candidate frame matches $\mathbb{M}_f$, a 2-D Hough transform is first conducted on $\mathbb{M}_f$ to vote in K_2 ($K_2 = 10$) hypotheses $<v_r, \delta t>$, where $\delta t = t(q) - t(r)$ specifies the temporal offset between a query video and a reference video. Then, for each hypothesis, the beginning and the end of a potential video copy are identified by picking up the first and the last matches m_f in $\mathbb{M}_f$ that accord with the hypothesis. After that, the two subsequences of $[t^B(q), t^E(q)]$ and $[t^B(r), t^E(r)]$ at level ℓ are uniformly divided into $C = 2^\ell$ segments respectively, namely $ts_{q,0}, \ldots, ts_{q,C-1}$ and $ts_{r,0}, \ldots, ts_{r,C-1}$. Similarity scores of frame matches across the two subsequences are then accumulated to form the similarity of each two corresponding segments. The similarity of the two subsequences at level $\ell (\ell = 0, 1, \ldots, L)$ (in practice, $L = 3$) is obtained by averaging the pairwise segment similarity values.

$$s_{v,i}^{\ell} = \sum \left\{ s_f | \langle v_q, t(q), v_r, t(r), s_f \rangle \in M_f, t(q) \in ts_{q,i}, t(r) \in ts_{r,i} \right\}, \tag{4}$$

$$s_v^{\ell} = \frac{1}{n_f} \sum_{i=0}^{C-1} s_{v,i}^{\ell}, \tag{5}$$

where n_f is the number of keyframes in $[t^B(q), t^E(q)]$. The weight of level ℓ is set to 2^{-L} for $\ell = 0$, and $2^{\ell - L - 1}$ for $\ell = 1, \ldots, L$ to penalize matches in coarser levels. Finally, the video similarity score s_v is calculated by accumulating the weighted similarities from multiple levels:

$$s_v = 2^{-L} s_v^0 + \sum_{\ell=1}^{L} 2^{\ell - L - 1} s_v^{\ell}. \tag{6}$$

Only if s_v is greater than or equal to a threshold T_1, will m_v be accepted as a video match. That is, a video-level match can be expressed as: $m_v = \langle v_q, t^B(q), t^E(q), v_r, t^B(r), t^E(r), s_v \rangle (s_v \geq T_1)$, which means the subsequence $[t^B(q), t^E(q)]$ of a query video v_q is a copy originated from the subsequence $[t^B(r), t^E(r)]$ of a reference video v_r with a similarity score of s_v. In the case that several candidate video matches may meet this constraint, only the one with the highest similarity score is retained.

Since TPM only needs a set of frame-level matches as its input, it is suitable for various frame-based visual/audio features. Moreover, it is computationally efficient (see Table IV in 5.4). Let N_f denote the number of keyframes (audio clips) in v_q, N_{ref} be the number of reference videos and $N_{\delta bin}$ be the number of bins for $\delta t = t_q - t_r$ in the 2-D Hough Transform respectively, then the amount of computational operations (basically, comparison and assignment) required by TPM for one query video can be approximated as:

$$N_f * K_1 * K_2 * (L + 1) + N_{ref} * N_{\delta bin} * \log K_2. \tag{7}$$

4.6 Fusion and Verification

A result-level fusion mechanism is utilized to fuse the copy detection results obtained by different features. Specifically, if a query video is simultaneously asserted by at least two features as a copy of the same reference video and the two assertions overlap temporally, then the query video is accepted as a copy of the reference video. Formally, let $\bar{m}_v = \langle v_q, \bar{t}^B(q), \bar{t}^E(q), v_r, \bar{t}^B(r), \bar{t}^E(r), \bar{s}_v \rangle$ and $\hat{m}_v = \langle v_q, \hat{t}^B(q), \hat{t}^E(q), v_r, \hat{t}^B(r), \hat{t}^E(r), \hat{s}_v \rangle$ be two video-level matches detected by two features, if (10) is satisfied, then (11) will be accepted as a final detection result:

$$[\bar{t}^B(q), \bar{t}^E(q)] \cap [\hat{t}^B(q), \hat{t}^E(q)] \neq \phi, \ \ [\bar{t}^B(r), \bar{t}^E(r)] \cap [\hat{t}^B(r), \hat{t}^E(r)] \neq \phi, \tag{8}$$

$$\langle v_q, \max(\bar{t}^B(q), \hat{t}^B(q)), \min(\bar{t}^E(q), \hat{t}^E(q)), v_r, \max(\bar{t}^B(r), \hat{t}^B(r)), \min(\bar{t}^E(r), \hat{t}^E(r)), \max(\bar{s}_v, \hat{s}_v) \rangle. \tag{9}$$

And if a query video is reported as a copy only by one feature, it should be further verified using only original DC-SIFT descriptors (see formula (4)). The average frame similarity will be taken as the similarity value of the reported video match. Only if this recalculated similarity value is above a threshold T_2, will it be accepted as a real copy. This verification is due to the consideration that the BoW representation inevitably causes a decrease in the discriminability of DC-SIFT.

Note that the specific values of thresholds T_1 and T_2 are learned on a training dataset. The values are chosen so that specific requirements of false positive rate and false negative rate can be met.

5. EXPERIMENTS

The performance of the proposed method is primarily evaluated using the benchmarking CBCD (also abbreviated as CCD) task of TRECVID 2010.[2] In this section, we first describe the evaluation proxy (i.e., the dataset and evaluation methodology) and then present the experimental results based on this proxy. To further evaluate the effectiveness of the proposed method, experiments are also carried out over the dataset of TRECVID 2009.[3]

5.1 Datasets

In TRECVID 2010, the reference database has 425-hour videos. It is composed of 11,524 videos collected from the Internet, thereby diverse in content, style and format, and varied in quality. A query dataset of 10,976 videos,[4] each being averagely 70 seconds long, is constructed using tools developed by IMEDIA.[5] The construction process applies a combination of eight video transformations and seven audio transformations (see Table I) to three types of videos: reference video only, reference video embedded into a nonreference video and non-reference video only, resulting in 56 compound transformations. Consequently, 32-hour video queries and 28-hour audio queries are derived from 425-hour reference videos. With respect to TRECVID 2009, its reference database contains 394-hour videos, with both video and audio queries being 32 hours. In contrast to TRECVID 2010, the video transformation of V1 is excluded from TRECVID 2009. Therefore, the total numbers of copy detection tasks are 56 and 49 for TRECVID 2010 and TRECVID 2009, respectively.

5.2 Evaluation Metrics

A detection result is considered correct if it at least overlaps with the fragment from which the query is derived. To evaluate the performance of a copy detection system, TRECVID adopts three metrics, namely, NDCR, Mean F1 and Mean Processing Time.

[2]http://www-nlpir.nist.gov/projects/tv2010/tv2010.html.
[3]http://www-nlpir.nist.gov/projects/tv2009/tv2009.html.
[4]40 queries are dropped by TRECVID 2010 due to synchronization issues between audio and video.
[5]http://www-rocq.inria.fr/imedia/.

Table I. Transformations Used in the TRECVID CBCD Task

CATEGORY	LABEL	TYPE
VIDEO TRANS.	V1	Simulated camcording
	V2	Picture in picture Type 1 (The original video is inserted in front of a background video)
	V3	Insertion of pattern
	V4	Strong re-encoding
	V5	Change of gamma
	V6	Decrease in quality
	V8	Post Production
	V10	Combination of 3 randomly chosen transformations out of V1-V8
AUDIO TRANS.	A1	Do nothing
	A2	Mp3 compression
	A3	Mp3 compression and multiband companding
	A4	Bandwidth limit and single-band companding
	A5	Mix with speech
	A6	Mix with speech and multiband compress
	A7	Bandpass filter, mix with speech and compress
COMB. TRANS.	Mn	Vx+Ay => M[(x-1)*7+y]

The first measure is normalized detection cost rate (NDCR), which is used to measure the detection effectiveness to each transformation. The probability of a miss error (P_{Miss}) and the false alarm rate (R_{FA}) are combined into a single detection cost rate, which is normalized as follows:[6]

$$NDCR = P_{Miss} + \beta * R_{FA} = \frac{\mathrm{FN}}{\mathrm{N_{Target}}} + \left(\frac{\mathrm{C}_{FA}}{(\mathrm{C_{Miss}} * \mathrm{R_{Target}})}\right) * \left(\frac{\mathrm{FP}}{(\mathrm{T_{Ref}} * \mathrm{T_{Qry}})}\right), \tag{10}$$

where FN and FP stands for false negative (i.e., miss) and false positive (i.e., false alarm) respectively; C_{Miss} and C_{FA} are the costs of an individual Miss and an individual False Alarm, respectively, with C_{Miss} being 1, and C_{FA} being 1 for BALANCED and 1000 for NOFA (no false alarm); $\mathrm{N_{Target}}$ is the total number of copies, and R_{Target} is the a priori target rate for the application of interest; T_{Ref} and T_{Qry} are the total length (in hours) of the entire reference dataset and that of the queries for a transformation, respectively.

To measure the accuracy of finding the exact extent of the copy in the reference video, the second measure of Mean F1 is defined as the harmonic mean of a special version of precision and recall. Specifically, precision is the percentage of the asserted copy that is indeed an actual copy and recall is the percentage of the actual copy that is subsumed in the asserted copy:

$$precision = \frac{t''^{E}(r) - t''^{B}(r)}{t^{E}(r) - t^{B}(r)}; recall = \frac{t''^{E}(r) - t''^{B}(r)}{t'^{E}(r) - t'^{B}(r)}. \tag{11}$$

The third measure of Mean Processing Time (MPT) is mean time (in seconds) needed to process a query. Processing time is defined as the full time required to process queries from mpg files to yield the result file.

In alignment with the problem formulation given in Section 2, NDCR is to evaluate how well problem (1) is resolved, while Mean F1 is to measure how precise problem (2) is answered. Of course, MPT is for measuring the efficiency of solving both problem (1) and (2).

[6]http://www-nlpir.nist.gov/projects/tv2010/Evaluation-cbcd-v1.3.htm#eval.

Table II. Performance of Individual Features

METRICS	FEATURES	TRECVID DATASETS	V1 (A1)	V2 (A2)	V3 (A3)	V4 (A4)	V5 (A5)	V6 (A6)	V8 (A7)	V10	AVG
NDCR	DC-SIFT	2010	0.285	0.154	0.054	0.146	**0.038**	0.223	0.292	0.200	0.174
		2009		0.112	0.030	0.090	**0.024**	0.142	0.201	0.149	0.107
	DCT	2010	*1.000*	0.377	0.246	0.200	**0.146**	0.323	0.585	0.415	*0.412*
		2009		0.224	0.164	0.119	**0.104**	0.231	0.410	0.306	0.223
	WASF	2010	**0.108**	0.131	0.131	0.146	0.269	0.254	0.269		0.187
		2009	**0.090**	**0.090**	**0.090**	**0.090**	0.194	0.172	0.187		0.130
	Combined	2010	0.054	0.032	0.022	0.041	**0.013**	0.044	0.061	0.097	0.046
		2009		0.005	**0.001**	**0.001**	0.002	0.038	0.070	0.068	0.026
MEAN F1	DC-SIFT	2010	0.890	**0.945**	0.928	0.923	0.934	0.891	0.901	0.918	0.916
		2009		0.937	0.934	0.939	**0.947**	0.904	0.896	0.923	0.926
	DCT	2010	*0.000*	0.946	0.969	0.970	0.970	0.964	**0.978**	0.943	*0.843*
		2009		0.942	0.971	0.967	**0.974**	0.944	0.942	0.938	0.954
	WASF	2010	0.927	0.923	0.923	**0.931**	0.921	0.903	0.921		0.921
		2009	**0.932**	**0.932**	**0.932**	**0.932**	0.909	0.917	0.911		0.926
	Combined	2010	0.922	0.949	**0.952**	0.940	0.937	0.936	0.938	0.932	0.938
		2009		0.938	**0.944**	0.932	0.930	0.921	0.925	0.928	0.931
MEAN PROC. TIME	DC-SIFT	2010	223	208	**132**	149	141	206	219	234	189
		2009		141	83	92	**79**	137	152	157	120.143
	DCT	2010	7.26	10.03	5.54	5.46	**5.22**	5.76	10.64	7.01	7.115
		2009		7.32	3.96	3.71	**3.53**	4.57	7.46	5.09	5.091
	WASF	2010	**70.1**	72.3	72.6	71.8	73.4	73.2	72.7		72.3
		2009	53.2	54.4	54.3	54.9	55.6	**53.1**	55.3		54.25
	Combined	2010	269.36	287.13	**191.64**	208.56	200.32	265.86	283.74	295.11	250.22
		2009		142.85	149.83	**130.64**	135.16	134.21	136.84	131.65	137.31

5.3 Performance of Individual Features

The experimental results are shown in Table II (in each row, best values are marked in bold). It can be seen that DC-SIFT shows good robustness (less NDCR values) to content-altering transformations such as V1, V2, V8, and V10. Although DCT performs well on content-preserving transformations such as V3, V4, V5, and V6, it is vulnerable to content-altering transformations such as V1, V2, V8, and V10. In particular, it is totally incapable of resisting V1 (with NDCR being 1 and Mean F1 being 0). Examples for the complementarity between DC-SIFT and DCT are shown in Figure 7. The reason why Figure 7(a) is detected only by DC-SIFT is that DCT is sensitive to content-altering operations while a certain amount of feature points can still be extracted by DC-SIFT from the modified content. On the contrary, Figure 7(b) is due to insensitivity of low frequency DCT coefficients to the change of visual detail, whereas few feature points can be extracted when the quality of video is severely degraded. Moreover, compared with DC-SIFT and WASF, DCT has the advantages of excellent Mean F1 (except V1) and short Mean Processing Time. In fact, DC-SIFT is computationally inefficient. With

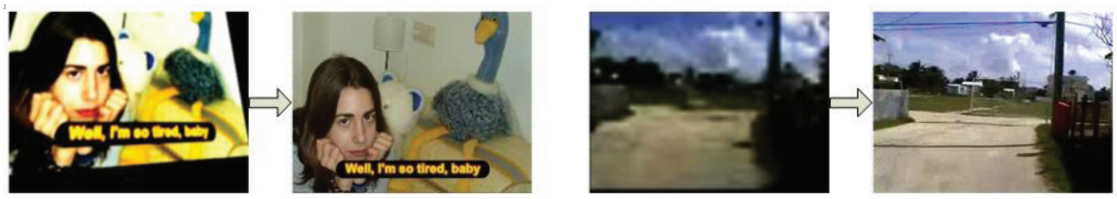

Fig. 7. Examples for the complementarity. (a) Detected only by DC-SIFT; (b) Detected only by DCT. In each pair, a query keyframe is shown on the left, with its reference on the right. Image courtesy Internet Archive [Internet Archive].

Table III. Performance Comparison between Single-level Matching and Multi-level Matching

	TRECVID 10		TRECVID 09	
ℓ	SINGLE LEVEL	TPM	SINGLE LEVEL	TPM
0 (1 ts)	0.273		0.219	
1 (2 ts)	0.247	0.223	0.192	0.179
2 (4 ts)	0.226	0.195	0.177	0.132
3 (8 ts)	0.202	**0.174**	0.173	**0.107**
4 (16 ts)	0.214	0.181	0.185	0.110

respect to WASF, it handles well simple audio transformations such as A1, A2, A3, and A4, while being sensitive to more complex transformations.

The experimental results on individual features show that none of the individual features can resist all the transformations; however, the individual features may complement each other so that an ideal overall performance can be achieved. For convenience of comparison, the performance of the combined features is also added here in Table II (marked in red). Obviously, the NDCR performance of the combined features is much better than that of any individual feature. This is because the total number of copies detected by the combination is approximately the sum of the copies detected by each individual feature. Consequently, the Mean F1 of the combined features is about the average of those of individual features. As the Mean Processing Time is concerned, the result of the combination is about the sum of time needed by each individual feature. The minor difference between the summation and the actual time can be attributed to the fusion strategy where same copies reported by any two features will be accepted directly without further verification. This also contributes to the better MPT on TRECVID 2009 than on TRECVID 2010. Another factor for less MPT is that the most time-consuming transformation of V1 is excluded from TRECVID 2009.

5.4 Performance of TPM

Taking frame-level similarity search results as input, TPM produces sequence-level matching results through similarity evaluation over temporal multigranularities. Intuitively, such a multigranularity similarity fusion framework can make the copy detection system robust to temporal transformations such as sampling. This is experimentally supported by the NDCR performances achieved by DC-SIFT using single-level and multilevel matching mechanisms respectively (see Table III). In our method, TPM adopts a structure of four levels ($\ell = 0, 1, 2, 3$) to strike a balance between coarse matching and fine matching. The results at $\ell = 0$ are unsatisfactory due to the malposed frame matches included in the video similarity calculation, resulting in many false positives. On the contrary, video matching at $\ell = 4$ misses some short copies for inadequacy in strictly aligned frame matches. With respect to $\ell \geq 5$,

Table IV. Performance Comparison between TPM and VFF

Metrics	Fusion Methods	TRECVID Datasets	V1	V2	V3	V4	V5	V6	V8	V10	AVG
NDCR	TPM	2010	0.285	0.154	0.054	0.146	0.038	0.223	0.292	0.200	**0.174**
		2009		0.112	0.030	0.090	0.024	0.142	0.201	0.149	**0.107**
	VFF	2010	0.346	0.207	0.131	0.200	0.116	0.285	0.354	0.269	0.239
		2009		0.164	0.090	0.142	0.090	0.194	0.245	0.187	0.159
Mean F1	TPM	2010	0.890	0.945	0.928	0.923	0.934	0.891	0.901	0.918	**0.916**
		2009		0.937	0.934	0.939	0.947	0.904	0.896	0.923	**0.926**
	VFF	2010	0.901	0.918	0.909	0.913	0.912	0.907	0.916	0.910	0.911
		2009		0.916	0.921	0.917	0.920	0.914	0.913	0.919	0.917
Frame Fusion Time (s)	TPM	2010	0.004	0.004	0.004	0.004	0.004	0.004	0.004	0.004	**0.004**
		2009		0.004	0.004	0.004	0.004	0.004	0.004	0.004	**0.004**
	VFF	2010	0.103	0.102	0.103	0.103	0.103	0.103	0.103	0.103	0.103
		2009		0.102	0.101	0.101	0.102	0.102	0.103	0.101	0.102

it is inapplicable because the number of keyframes sampled from the shortest copies may be less than 32 (i.e., 2^5).

Further, the performance of TPM is compared with the matching method proposed in Wei et al. [2011], which formulates the frame fusion problem as the decoding problem of a hidden Markov model. The method used by Wei is a Viterbi-based frame fusion algorithm (shorted as VFF). Again, DC-SIFT is used here. It can be easily figured out from Table IV that TPM outperforms VFF in terms of all three evaluation metrics. Particularly, TPM is much faster than VFF.

5.5 Performance of the Proposed Method

The overall performance of the proposed method (labeled as Improved) is compared against those of its previous versions (Perseus and Kraken), the best performances of all other methods (BestExceptUs) as well as the median (Median) over each task at TRECVID 2010. For Improved, parameter optimization has been carried out to the feature descriptors as well as TPM. Note that Kraken differs from Perseus only by substituting a higher similarity threshold T_1 for the verification module to prevent false positives. For our absence from the CBCD competition of TRECVID 2009, the performance of the proposed method is only evaluated against the best performances of all other methods (Best09) and Median for TRECVID 2009. And due to the limitation on length of this article, listed here are only the results for BALANCED profile. It must be noted that, the BestExceptUs or Best09 does not refer to any single method; instead, it refers to a virtual method which represents the best performances of all methods except ours over each copy detection task.

5.5.1 *NDCR.* Perseus and Kraken achieved excellent NDCR performances at the CBCD competition: among 56 tasks, they together achieved 51 best (lowest) "Optimal NDCR" (see Figure 8 (a)). The reason why Perseus outperforms Kraken over most transformations could be that the verification adopted by Perseus is more effective in filtering false positives than the higher threshold used by Kraken. With respect to Improved, its NDCR performance is slightly worse than that of Perseus. This is mainly attributed to the fact that Perseus uses two local visual features of SIFT and SURF, while Improved uses only DC-SIFT.

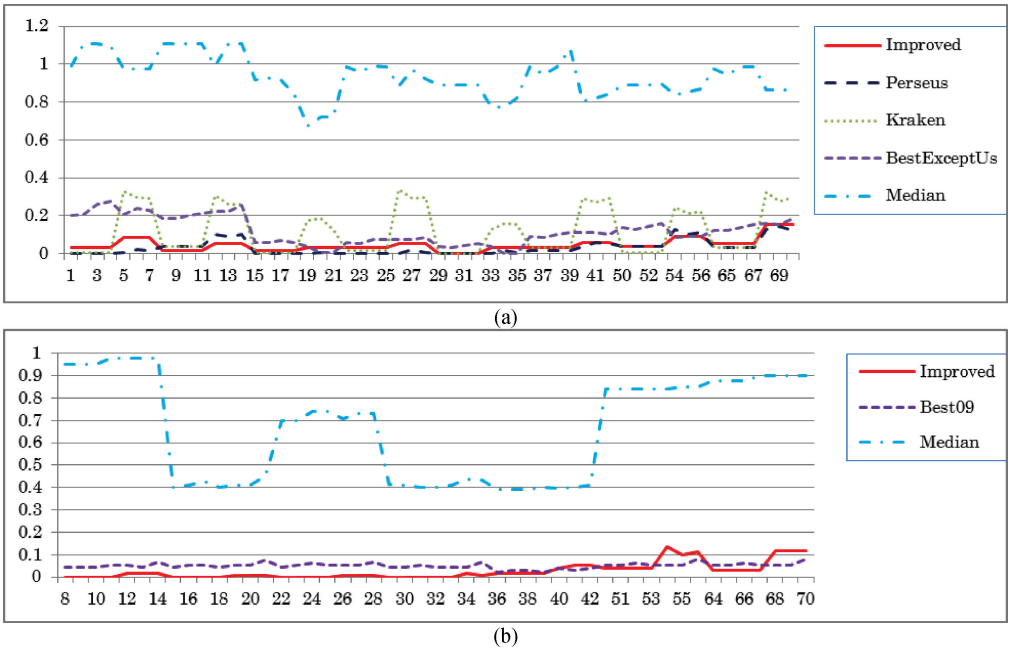

Fig. 8. NDCR performances. (a) TRECVID 2010; (b) TRECVID 2009. The horizontal axis is for sequence numbers of copy detection tasks (totally 56 for TRECVID 10, and 49 for TRECVID 09), while the vertical axis is for NDCR values.

5.5.2 *Mean F1.* Perseus and Kraken achieved competitive Mean F1 values of around 0.9 with minor deviations (see Figure 9). Since only true positives are taken into account at Mean F1 evaluation, Perseus and Kraken achieved almost the same performances. The performance difference between our method and BestExceptUs may be attributed to the "overcautious" strategy for copy localization in (11). Due to parameter optimization, the Mean F1 of Improved has been improved to about 0.95.

5.5.3 *Mean Processing Time.* Since two computation-intensive local visual features were involved, the original MPT performances of Perseus and Kraken were much worse than the Median. Therefore, multithreading and multicore programming techniques have been adopted to tremendously decrease Mean Processing Time [Li et al. 2010]. Furthermore, the substitution of DC-SIFT for SIFT and SURF by Improved has resulted in a further decrease in Mean Processing Time (see Figure 10).

5.5.4 *Summary.* These experimental results show that our method, either Improved or Persus/Kraken, achieves the best copy detection accuracies for the overwhelming majority of copy detection tasks, over either dataset of TRECVID 2010 or TRECVID 2009. Compared with Persus/Kraken, Improved performs much better in terms of copy localization preciseness and processing efficiency and it outperforms Kraken significantly at copy detection accuracy.

6. DISCUSSION

It can be seen from the experimental results that the proposed method has greatly improved the performance of a video copy detection system, and can be viewed as a big step towards solving the problem of video copy detection. After CBCD competition at TRECVID 2011, the CBCD competition is terminated because of limited space for performance improvement. However, has the problem of video copy detection completely been solved? The answer is Not Yet.

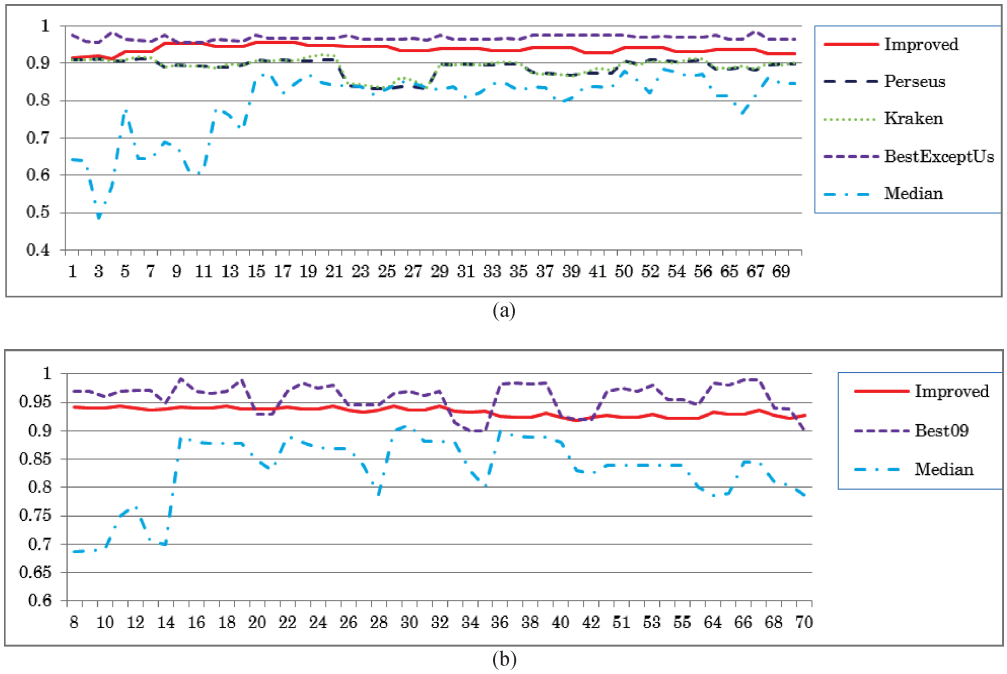

Fig. 9. Mean F1 Performances. (a) TRECVID 2010; (b) TRECVID 2009. The horizontal axis is for sequence numbers of copy detection tasks (totally 56 for TRECVID 10, and 49 for TRECVID 09), while the vertical axis is for Mean F1 values.

For one thing, the effective methods are not efficient enough. Take the proposed method for example. The mean processing time is linearly dependent on the duration of the query video and also on the total time length of the dataset of reference videos. Once the dataset of reference videos gets to extra-large, the response time of the system will be unacceptably long. Nevertheless, video sharing on the Internet calls for instant response. Therefore, a real-time video copy detection system is needed. Thus, the decision fusion by fusing results reported by multiparallel detectors is no longer suitable. Adopting the adaboost-like structure, we can design a scalable cascading architecture in the way that each query is sequentially processed by individual detectors until it is asserted as a copy. By arranging simple and fast detectors before complicated and slow detectors, the mean processing time can be significantly reduced. To exploit the correlation of multimodal features to improve fusion performance, a boosting approach called MultiFusion [Wang and Kankanhalli 2010] could be utilized. Additionally, the multimodal feature representation could be converted into a complete binary mediaprint representation to reach the goal of real-time video copy detection.

For another, the multimodal features are not ideal invariant features. Ideal invariant features should be not only discriminative to distinguish different video content, but also robust to allow common video processing transformations. Despite the fact that existing invariant features bear both discrimination and robustness to some extent, there is a lack of theories and models for these invariant features to be used in video copy detection. Therefore, research efforts are needed on establishing an invariant feature analysis model. Such a model should consist of at least three stages (see Figure 11), namely, transformations analysis, perceptual model analysis and optimization, and invariant feature evaluation. First, all kinds of audio and video transformations are analyzed and a matrix containing all single and combined transformations will be output to the next stage as robustness constraint. Second, a

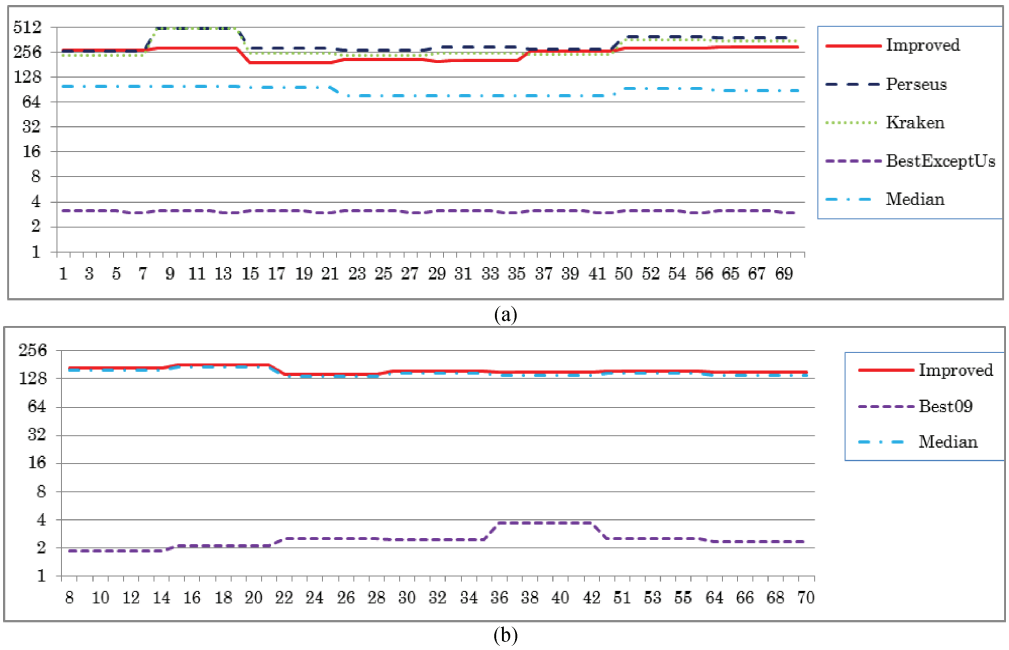

Fig. 10. Mean Processing Time performance. (a) TRECVID 2010; (b) TRECVID 2009. The horizontal axis is for sequence numbers of copy detection tasks (totally 56 for TRECVID 10, and 49 for TRECVID 09), while the vertical axis is for mean processing time values. Note that the vertical axis adopts a logarithmic scale.

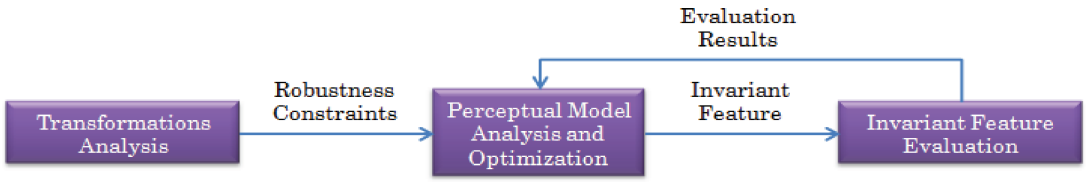

Fig. 11. An invariant feature analysis model.

typical perceptual model is analyzed and optimized to partly or wholly meet the robustness constraint, leading to representation of certain kind of invariant feature. Third, performance evaluation is carried out to assess the discrimination and robustness of the invariant feature, and the evaluation results will be fed back to the second stage to further optimize the perceptual model and the representation of the invariant feature as well. Such iteration can be repeated many times until the ideal invariant feature is achieved. Some exploratory work in analyzing well-known perceptual models for invariant features can be found in Mou et al. [2011, 2012], which use Sparse Coding and Standard Model in the field of neural science.

7. CONCLUSION

In this article, we propose a method combing multimodal feature representation and temporal pyramid matching (TPM) to address the challenges posed by detecting video copies subjected to complicated transformations. Multimodal feature representation exploits the complementary characteristics of multimodal features to achieve robustness to various transformations. A frame-to-sequence

fusion algorithm called TPM is proposed to convert frame-level similarity search results into sequence-level video matching results based on similarity evaluation over temporal multigranularities. Performance evaluation over the benchmarking datasets of TRECVID CBCD demonstrates that the proposed method is effective in copy detection and localization. Future work will be devoted to designing a cascade architecture to achieve more efficient copy detection, and to exploring new invariant features following the invariant feature analysis model.

REFERENCES

AHMED, F., SIYAL, M. Y., AND ABBAS, U. V. 2010. A secure and robust hash-based scheme for image authentication. *Signal Process. 90*, 5, 1456–1470. DOI: http//dx.doi.org/10.1016/j.sigpro.2009.05.024.

BALLARD, D. H. 1981. Generalizing the Hough transform to detect arbitrary shapes. *Patt. Recog. 13*, 2, 111–122. DOI: http//dx.doi.org/10.1016/0031-3203(81)90009-1.

BAY, H., TUYTELAARS, T., AND GOOL, L. V. 2006. SURF: Speeded Up Robust Features. In *Proceedings of the 9th European Conference on Computer Vision (ECCV'06)*, (Graz, Austria). 404–417. DOI: http://dx.doi.org/10.1007/11744023_32.

BOSCH, A., ZISSERMAN, A., AND MUÑOZ, X. 2008. Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 4, 712–727. DOI: http://dx.doi.org/10.1109/TPAMI.2007.70716.

CANO, P., BATLLE, E., KALKER, T., AND HAITSMA, J. 2005. A review of audio fingerprinting. *J. VLSI Signal Process. 41*, 3, 271–284. DOI: http://dx.doi.org/10.1007/s11265-005-4151-3.

CANO, P., BATLLE, E., MAYER, H., AND NEUSCHMIED, H. 2002. Robust sound modeling for song detection in broadcast audio. In *Proceedings of AES 112th International Convention* (Germany).

CHEN, J. AND HUANG, T. 2008. A robust feature extraction algorithm for audio fingerprinting. In *Proceedings of the 9th Pacific Rim Conference on Multimedia (PCM'08)*, 887–890. DOI: http://dx.doi.org/10.1007/978-3-540-89796-5_106.

CHEN, L. AND STENTIFORD, F. W. M. 2008. Video sequence matching based on temporal ordinal measurement. *Patt. Recog. Lett. 29*, 13, 1824–1831. DOI: http://dx.doi.org/10.1016/j.patrec.2008.05.015.

CHEUNG, S. S. AND ZAKHOR, A. 2003. Efficient video similarity measurement with video signature. *IEEE Trans. Circuits Syst. Video Technol. 13*, 1, 59–74. DOI: http://dx.doi.org/10.1109/TCSVT.2002.808080.

DE ROOVER, C., DE VLEESCHOUWER, C., LEFÈBVRE, F., AND MACQ, B. 2005. Robust video hashing based on radial projections of key frames. *IEEE Trans. Signal Proc. 53*, 10, 4020–4037. DOI: http://dx.doi.org/10.1109/TSP.2005.855414.

DOUZE, M., JÉGOU, H., AND SCHMID, C. 2010. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Trans. Multimedia 12*, 4, 257–266. DOI: http://dx.doi.org/10.1109/TMM.2010.2046265.

GIONIS, A., INDYK, P., AND MOTWANI, R. 1999. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*. 518–529.

GRAUMAN, K. AND DARRELL, T. 2005. The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV'05)*. 1458–1465. DOI: http://dx.doi.org/10.1109/ICCV.2005.239.

HAMPAPUR, A. AND BOLLE, R. M. 2001. Comparison of distance measures for video copy detection. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'01)*. 737–740. DOI: http://dx.doi.org/10.1109/ICME.2001.1237827.

HUA, X.-S., CHEN, X., AND ZHANG, H.-J. 2004. Robust video signature based on ordinal measure. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*. 685–688. DOI: http://dx.doi.org/10.1109/ICIP.2004.1418847.

HUANG, T., TIAN, Y., GAO, W., AND LU, J. 2010. Mediaprinting: Identifying multimedia content for digital rights management. *Computer. 43*, 12, 28–35. DOI: http://dx.doi.org/10.1109/MC.2010.356.

IWAMOTO, K., KASUTANI, E., AND YAMADA, A. 2006. Image signature robust to caption superimposition for video sequence identification. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'06)*. 3185–3188. DOI: http://dx.doi.org/10.1109/ICIP.2006.313046.

INTERNET ARCHIVE. www.archive.org.

JOLY, A., BUISSON, O., AND FRÉLICOT, C. 2007. Content-based copy retrieval using distortion-based probabilistic similarity search. *IEEE Trans. Multimedia 9*, 2, 293–306. DOI: http://dx.doi.org/10.1109/TMM.2006.886278.

KIM, C. AND VASUDEV, B. 2005. Spatiotemporal sequence matching for efficient video copy detection. *IEEE Trans. Circuits Syst. Video Technol. 15*, 1, 127–132. DOI: http://dx.doi.org/10.1109/TCSVT.2004.836751.

KIM, H., LEE, J., LIU, H., AND LEE, D. 2008. Video linkage: Group based copied video detection. In *Proceedings of the ACM International Conference on Content-Based Image Video Retrieval (CIVR'08)*. 397–406. DOI: http://dx.doi.org/10.1145/1386352.1386404.

LAW-TO, J., BUISSON, O., GOUET-BRUNET, V., AND BOUJEMAA, N. 2006. Robust voting algorithms based on labels of behavior for video copy detection. In *Proceedings of the ACM International Conference on Multimedia (MM)*. (Santa Barbara, CA). 835–844. DOI: http://dx.doi.org/10.1145/1180639.1180826.

LAZEBNIK, S., SCHMID, C., AND PONCE, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 19th IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2169–2178. DOI: http://dx.doi.org/10.1109/CVPR.2006.68.

LEE, S. AND YOO, C. D. 2006. Video fingerprinting based on centroids of gradient orientations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)*. 401–404. DOI: http://dx.doi.org/10.1109/ICASSP.2006.1660364.

LI, Y., MOU, L., SU, C., FANG, X., QIAN, M., JIANG, M., WANG, Y., TIAN, Y., HUANG, T., AND GAO, W. 2010. PKU@TRECVID2010: Copy detection with visual-audio feature fusion and sequential pyramid matching. In *Online Proceedings of TRECVID 2010 Workshop*.

LIN, C.-Y. AND CHANG, S.-F. 2001. A robust image authentication method distinguishing jpeg compression from malicious manipulation. *IEEE Trans. Circuits Syst. Video Technol. 11*, 2, 153–168. DOI: http://dx.doi.org/10.1109/76.905982.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis. 60*, 2, 91–110. DOI: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

MIKOLAJCZYK, K. AND SCHMID, C. 2005. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell. 27*, 10, 1615–1630. DOI: http://dx.doi.org/10.1109/TPAMI.2005.188.

MOU, L., HUANG, T., TIAN, Y., LIAN, S., AND CHEN, X. 2011. Robust and discriminative image authentication based on sparse coding. In *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC'11)*. 323–326. DOI: http://dx.doi.org/10.1109/CCNC.2011.5766482.

MOU, L., CHEN, X., TIAN, Y., AND HUANG, T. 2012. Robust and discriminative image authentication based on standard model feature. In *Proceedings of IEEE International Symposium on Circuits & Systems (ISCAS'12)*. 1131–1134. DOI: http://dx.doi.org/10.1109/ISCAS.2012.6271431.

MPEG. 2002. ISO/IEC 15938-4:2002 Information technology – Multimedia content description interface – Part 4: Audio.

OOSTVEEN, J., KALKER, T., AND HAITSMA, J. 2002. Feature extraction and a database strategy for video & fingerprinting. *Vis. Lect. Notes Comput. Sci. 2*, 117–128. DOI: http://dx.doi.org/10.1007/3-540-45925-1_11.

OVER, P., AWAD, G. M., FISCUS, J., ANTONISHEK, B., MICHEL, M., SMEATON, A. F., KRAAIJ, W., AND QUÉNOT, G. 2010. TRECVID 2010 – An overview of the goals, tasks, data, evaluation mechanisms, and metrics. In *Proceedings of TRECVid*.

RADHAKRISHNAN, R. AND BAUER, C. 2008. Robust video fingerprints based on subspace embedding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'08)*. 2245–2248. DOI: http://dx.doi.org/10.1109/ICASSP.2008.4518092.

SHIVAKUMAR, N. N. 1999. Detecting digital copyright violations on the Internet. Ph.D. Dissertation, Stanford University.

SIVIC, J. AND ZISSERMAN, A. 2003. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV'03)*. 1470–1477. DOI: http://dx.doi.org/10.1109/ICCV.2003.1238663.

SWAMINATHAN, A., MAO, Y., AND WU, M. 2006. Robust and secure image hashing. *IEEE Trans. Inf. Forensics Security 1*, 2, 215–230. DOI: http://dx.doi.org/10.1109/TIFS.2006.873601.

TIAN, Y., JIANG, M., MOU, L., FANG, X., AND HUANG, T. 2011. A multimodal video copy detection approach with sequential pyramid matching. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'11)*. 3629–3632. DOI: http://dx.doi.org/10.1109/ICIP.2011.6116504.

WANG, X. AND KANKANHALLI, M. 2010. MultiFusion: A boosting approach for multimedia fusion. *ACM Trans. Multimedia Comput. Commun. Appl. 6*, 4, Article 25, DOI: http://dx.doi.org/10.1145/1865106.1865109.

WEI, S., ZHAO, Y., ZHU, C., XU, C., AND ZHU, Z. 2011. Frame fusion for video copy detection. *IEEE Trans. Circuits Syst. Video Technol. 21*, 1, 15–28. DOI: http://dx.doi.org/10.1109/TCSVT.2011.2105554.

Received Novmember 2011; revised November 2012 and March 2013; accepted April 2013