TASC: A Transformation-Aware Soft Cascading Approach for Multimodal Video Copy Detection

YONGHONG TIAN, MENGREN QIAN, and TIEJUN HUANG, Peking University

How to precisely and efficiently detect near-duplicate copies with complicated audiovisual transformations from a large-scale video database is a challenging task. To cope with this challenge, this article proposes a transformation-aware soft cascading (TASC) approach for multimodal video copy detection. Basically, our approach divides query videos into some categories and then for each category designs a *transformation-aware* chain to organize several detectors in a cascade structure. In each chain, efficient but simple detectors are placed in the forepart, whereas effective but complex detectors are located in the rear. To judge whether two videos are near-duplicates, a Detection-on-Copy-Units mechanism is introduced in the TASC, which makes the decision of copy detection depending on the similarity between their most similar fractions, called *copy units* (CUs), rather than the video-level similarity. Following this, we propose a CU search algorithm to find a pair of CUs from two videos and a CU-based localization algorithm to find the precise locations of their copy segments that are with the asserted CUs as the center. Moreover, to address the problem that the copies and noncopies are possibly linearly inseparable in the feature space, the TASC also introduces a flexible strategy, called *soft decision boundary*, to replace the single threshold strategy for each detector. Its basic idea is to automatically learn two thresholds for each detector to examine the easy-to-judge copies and noncopies, respectively, and meanwhile to train a nonlinear classifier to further check those hard-to-judge ones. Extensive experiments on three benchmark datasets showed that the TASC can achieve excellent copy detection accuracy and localization precision with a very high processing efficiency.

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, Experimentation, Security

Additional Key Words and Phrases: Video copy detection, multimodal features, transformation-aware soft cascading, copy units, soft decision boundary

ACM Reference Format:
Yonghong Tian, Mengren Qian, and Tiejun Huang. 2015. TASC: A transformation-aware soft cascading approach for multimodal video copy detection. ACM Trans. Inf. Syst. 33, 2, Article 7 (February 2015), 34 pages.
DOI: http://dx.doi.org/10.1145/2699662

1. INTRODUCTION

The past decade saw an explosive growth of video data on the Internet. A recent report by Cisco shows that consumer Internet video traffic accounted for 57% of all consumer Internet traffic in 2012 and is projected to be 69% in 2017. On YouTube, for

This work was supported by the National Basic Research Program of China under contract 2015CB351806 and the National Natural Science Foundation of China under contracts 61035001 and 61390515.

Authors' address: Y. Tian, M. Qian, and T. Huang, Room 2608, Science Building 2, 5 Yiheyuan Road, Haidian, Beijing 100871, China; emails: {yhtian, mrqian, tjhuang}@pku.edu.cn. The authors are also with the Cooperative Medianet Innovation Center, China. T. Huang is the corresponding author.

DOI: http://dx.doi.org/10.1145/2699662

example, a staggering 100 hours of video were uploaded to the site every minute in July 2013. Among the huge volumes of online videos, there exist a noticeable amount of copies or near-duplicate videos. This fact imposes an urgent demand on video copy detection because it is crucial to many video-related applications [Huang et al. 2010; Liu et al. 2013b]. For example, by detecting and then eliminating the semantically and visually identical duplicates, Web-scale video search engines can return results with *semantically coherent* but *visually diversified* content [Wu et al. 2009b]. Similarly, to fend off the severe critics for failing to ensure that the uploaded videos comply with the law of copyright, social media sites such as YouTube and Youku can use copy detection systems to prevent users from uploading copyrighted material.

Basically, the primary idea of video copy detection is to automatically analyze 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 reference video (a.k.a. the origin). Often, copies share the exactly same semantics and the similar scenes with an origin, typically with different visual/audio presentations. A tightly related task is near-duplicate video retrieval (NDVR), which aims to find a ranking list of near-duplicate videos for a given query video from a database [Liu et al. 2013b]. Clearly, video copy detection and NDVR share similar principles, and numerous techniques can be applied to both tasks.

In most cases, video copies are generated from the origins by means of audiovisual transformations. To make things worse, the content of many copies may be significantly changed from their origins. For instance, video content is notably modified after spatial or temporal content-altering operations such as cropping or pattern insertion. Thus, for robust copy detection, invariant features (called *videoprints* [Huang et al. 2010] or *video signatures* [Liu et al. 2013b]) should be extracted from the video as its representation. However, the challenge mainly comes from the fact that there exists no such one-for-all signature that remains robust on all of these transformations. In other words, if we utilize a set of audiovisual features to construct several copy detectors, some of them may be robust against certain types of transformations but vulnerable to other types; other detectors may be the other way around.

Thus, a natural solution is to combine several video signatures, or detectors based on multiple features, to enhance the robustness of a copy detection system. For example, detection results using several audio and visual features separately are fused by selecting video matches with the highest similarity score [Saracoğlu et al. 2009; Liu et al. 2010], or queries asserted as copies by any two of four detectors are accepted as copies [Tian et al. 2012; Mou et al. 2013]. Similar approaches have been successfully used in the TRECVID content-based copy detection (CBCD) task, where most of the participating systems obtain the final result by fusing several detection results [Kraaij and Awad 2011]. In Jiang et al. [2012] and Tian et al. [2013], we proposed a copy detection approach with a cascade of multimodal features. In this approach, detectors based on several complementary audiovisual features are organized in a cascade structure such that efficient but simple detectors are placed in the forepart, whereas effective but complex detectors are located in the rear. To avoid the burdensome manual tuning of thresholds, a soft threshold learning algorithm was further proposed in Tian et al. [2013] to estimate the optimal decision threshold for each detector in the chain.

However, one potential problem of the approach in [Jiang et al. 2012; Tian et al. 2013] is that all query videos, whichever transformations they are subject to, are processed by the same chain of detectors. Clearly, this is only a suboptimal solution since query videos with different transformations may exhibit significantly distinct audiovisual properties. To address this problem, a transformation-aware soft cascading (TASC) approach is proposed. Our basic idea is to divide query videos into some categories and then for each category design a *transformation-aware* chain to organize several detectors in a cascade structure. Given a query video, the TASC first recognizes its category and then hands it over to the corresponding detector chain. We also develop one efficient implementation by utilizing three commonly used multimodal features (i.e., audio fingerprints (AFPs) [Haitsma and Kalke 2012], DCT [Mou et al. 2013], and SIFT Bag-of-Words (BoW) descriptor [Douze et al. 2010]) to construct four different chains. In this four-chain implementation, both the transformation recognition module and different detectors are easy to implement and computationally efficient.

Legally or technologically, only videos in which the length of identical or similar content is more than a predefined minimum value (e.g., 3 to 10 seconds) can be treated as duplicates or near-duplicates. Thus, a Detection-on-Copy-Units mechanism is introduced into the TASC to judge whether two videos are near-duplicates. In other words, given two videos, the decision of copy detection depends on the similarity between their most similar fractions, called *copy units* (CUs), rather than the video-level similarity. Following this, we propose a CU search algorithm to find a pair of CUs from two videos and a CU-based copy localization algorithm to find the precise locations of their copy segments that are with the asserted CUs as the center.

Furthermore, most of existing methods, including our previous works [Tian et al. 2011; Jiang et al. 2012; Tian et al. 2013], are to empirically set or experimentally train a single decision threshold for each detector. In the real-world applications, however, it is difficult to use only one threshold to perfectly divide the copies and noncopies, as they possibly are linearly inseparable in the feature space. To address this problem, the TASC introduces a more flexible strategy, called *soft decision boundary*, by learning two thresholds for each detector and meanwhile training a nonlinear classifier for each chain. Among the two thresholds, the upper one is used to determine whether a query video is a copy, whereas the lower one is used to judge whether it is a noncopy, both with a high degree of confidence. For two videos, if the similarities of their CUs through all detectors in a chain are between the two thresholds, a soft-margin support vector machine (SVM) classifier based on the SIFT keypoint matching between the two CUs is then utilized to further check whether or not they are near-duplicates.

Extensive experiments were conducted on three benchmark datasets, including TRECVID-CBCD [Over et al. 2010], MUSCLE-VCD-2007 [Law-To et al. 2007], and CC_WEB_VIDEO [Wu et al. 2007]. The experimental results showed that compared to several state-of-the-art approaches, the TASC can achieve excellent copy detection accuracy and localization precision with a very high efficiency.

The remainder of this article is organized as follows. After briefly summarizing the related work in Section 2, we formulate the TASC approach and describe its implementation in Section 3. Section 4 then presents the algorithms for CU search, soft boundary learning, and CU-based copy localization. Extensive experiments are presented in Section 5, and we conclude the article in Section 6.

2. RELATED WORK

As an active research problem with great value, video copy detection has attracted a lot of attention in recent years. This section presents a brief review of the related work from the viewpoint of multimodal video copy detection. For more details about NDVR, we refer the readers to a recent survey [Liu et al. 2013b].

2.1. Two Typical Paradigms

In general, the task of copy detection can be formulated as follows. Given a query video $q \in \mathbb{Q}$ and reference videos $\mathbb{R} = \{r_i\}(1 \leq i \leq \mathcal{R})$, the task is to examine whether $\exists r_i \in \mathbb{R}$ such that $A(q, r_i)$ holds, where $A(x, y)$ stands for x being a copy of y by the system; if so,

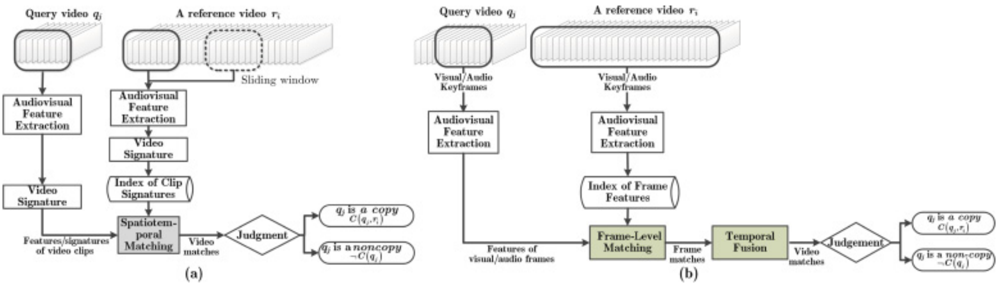


Fig. 1. Sliding window–based (a) and frame-fusion—based (b) paradigms of video copy detection.

then return the locations of the copy segments in q and r_i, namely $[t^{(B)}(q), t^{(E)}(q)]$ and $[t^{(B)}(r_i), t^{(E)}(r_i)]$. Note that slightly differently, the NDVR task is to find all $r_i \in \mathbb{R}$ that meet these conditions.

Often, two typical paradigms can be used for this task. As shown in Figure 1(a), the first one is to extract global features (e.g., spatiotemporal DCT [Coskun et al. 2006]) or an audiovisual signature (e.g., Chiu et al. [2010]) to represent a video clip, and then use a sliding window to scan each $r_i \in \mathbb{R}$ and compute the similarity between q and the windowed clip in r_i. Here an implicit assumption is that the whole video q is or is not a copy. But in practice, probably only one segment of q is a copy of a small segment in r_i. Moreover, this approach is difficult to deal with the task when temporal transformations such as frames inserting or deleting are involved in generating the copy. Instead, a more flexible paradigm is to partition q and r_i into basic indexing units (e.g., audiovisual frames) and then utilize efficient indexing models (e.g., hashing, inverted table) to speed up video matching. As shown in Figure 1(b), it searches a list of similar reference frames for each query frame and then determines video matches by assembling the frame matches with proper temporal fusing strategy. Different from the sliding window–based paradigm that is linearly dependent on the size of $\mathbb{R}$, this paradigm is sublinear and thus can be applicable to large-scale copy detection systems.

Specifically, two main subtasks are involved in the frame fusion–based paradigm: frame-level similarity evaluation and frame fusion. For the first subtask, two key techniques are involved (i.e., multimodal feature representation and fusion), whereas for the second one, the key issue is how to utilize temporal consistency constraints on frame matches to identify several video matches.[1] This article mainly focuses on the first one, and thus we present a brief review of its related work in the following discussion.

2.2. Multimodal Feature Representation

Often, three kinds of features are used for video copy detection in existing work, namely global and local visual features, and audio features. From these features, a video signature can be derived as a compact structure or a higher-level video summarization to represent each video segment or clip on which the copy detection system actually performs.

Based on the statistics of the entire frame or the whole clip, global visual features have the advantages of compactness in size and low computational complexity. Among them, the spatial or temporal ordinal signatures (e.g., Chiu et al. [2008] and Lei et al. [2012]) have been widely used in copy detection. There are also many proposals that

[1] Various methods have been proposed for this subtask, such as 2D Hough transform [Liu et al. 2010], spatiotemporal verification [Douze et al. 2010], Viterbi-based frame fusion [Wei et al. 2011], approximate string matching [Yeh and Cheng 2011], temporal pyramid matching [Tian et al. 2013], and frame matching-result graph [Liu et al. 2013a].

compute video signatures from a specific transform domain, such as DCT [Mou et al. 2013], polar Fourier transform [Swaminathan et al. 2006], and radon transform [Roover et al. 2005], or from special images constructed from the video [Esmaeili et al. 2011].

However, global features cannot effectively deal with more complex transformations, such as postproduction. Instead, local visual features, mostly based on the interest point detection and local descriptor calculation [Joly et al. 2007; Douze et al. 2010; Liu et al. 2013a], are by nature resistant to such content-altering operations since a part of original content always remains in the copy. Among them, scale-invariant feature transform (SIFT) and its derivations, such as SURF [Bay et al. 2006], DC-SIFT [Bosch et al. 2008], and PCA-SIFT [Ke and Sukthankar 2004], could be the most suitable local features for copy detection. To enhance the tolerance of SIFT on some special transformations, such as flip, some new developments, MI-SIFT [Ma et al. 2012] and F-SIFT [Zhao and Ngo 2013], were proposed in recent years. In addition, to improve the compactness of the descriptor and to accelerate feature matching, the BoW technique is used frequently in recent work (e.g., Douze et al. [2010] and Wang et al. [2012]) by building a visual vocabulary for local features and constructing a visual word histogram to represent each frame.

Since a video clip contains both visual and audio components, it is increasingly becoming an important trend to make use of both visual and audio features in video copy detection. Some audio features originally designed for content-based audio retrieval are often used as audio signatures, such as Mel-frequency cepstral coefficients (MFCCs), mean energy, normalized spectral sub-band moments, and audio spectrum flatness (ASF) [Huang et al. 2010]. For more details about audio signatures, we direct the readers to a separate survey [Chandrasekhar et al. 2011].

Several recent studies have started utilizing semantic features (e.g., human faces [Cotsaces et al. 2009], semantic concepts [Min et al. 2012]) for video copy detection. The underlying assumption is that content transformations tend to preserve the semantic information. However, the state-of-the-art performance of semantic concept detection in videos still keeps a very low level (e.g., the best MAP of 20% for 20 concepts on TV08 and 09 benchmark datasets [Tang et al. 2012]). Thus, it is far from generalizing the concept-based copy detection methods to large-scale video databases.

2.3. Fusion of Multimodal Detections

After years of practice, researchers have recognized that no single feature or signature can be both robust and discriminative for copy detection tasks under various transformations [Kraaij and Awad 2011]. Thus, it is beneficial to combine different signatures or several detectors to improve the performance. Typically, two sorts of approaches have been used: feature-level and result-level fusion (or equivalently early and late fusion).

In feature-level fusion approaches, several features are combined into a single representation. For example, in Saracoğlu et al. [2009], two color features, two texture features, and one motion feature are concatenated into a new global visual descriptor for copy detection; in Wu et al. [2009b], color histograms and local SIFT points are combined with the contextual information from time duration and number of views; in Liu et al. [2010], MFCC and RASRA-PLP are combined in a bag-of-audio-words (BoA) representation; and in Song et al. [2011], the authors learn a group of hash functions to map the video keyframes into the Hamming space and generate a series of binary codes to represent the video dataset. Potentially, such a representation can make use of the coherency and correlation across feature spaces. However, none of existing methods has been proven to remarkably boost the robustness of the copy detection system. This is also an open issue at the TRECVID-CBCD task, where few participating approaches can obtain a good detection performance using feature-level fusion [Kraaij and Awad 2011].

Instead, result-level fusion approaches utilize a set of audiovisual features to construct several detectors and then derive the final result by fusing the detection results from these detectors. In Saracoğlu et al. [2009] and Anguera et al. [2011], the fusion rule is to choose the best-matching result in terms of similarity obtained from separate audio and visual matches, whereas in Liu et al. [2010], the fusion is formulated as a reranking problem, which recalculates the similarity scores for all of the individual detection results and then employs four strategies (i.e., average, max, multiply, and logistic) to choose the best match as the final result. In our previous system [Tian et al. 2011], we proposed a verification-based fusion schema: a query video is accepted as a copy only if it is positively asserted by at least two detectors; otherwise, it should be further evaluated using an additional SIFT-based verification module. As such, our system achieved the best overall detection accuracy at the TRECVID-CBCD-2010 task. The system was further extended in Jiang et al. [2012] and Tian et al. [2013] by organizing multiple complementary detectors in a cascade structure, each of which is based on a single audio or visual feature. This can remarkably reduce the processing time for most query videos since the copies can be correctly detected by the first two detectors.

Although these approaches could achieve better detection accuracy than a single detector, a notable drawback is that all query videos, whichever transformations to they are subject, are processed by the same set of detectors. In other words, they ignore the fact that query videos with different transformations may exhibit significantly distinct audiovisual properties. To address this problem, some recent studies were devoted to investigate video copy detection methods for some specific transformations. For example, Liu et al. [2013a] proposed a twin-threshold segmentation and a graph-based sequence matching method for detecting copies with picture-in-picture (PiP); Kim et al. [2014b] proposed a video copy detection approach against rotation and flipping by extracting two complementary region binary patterns from keyframes and deriving a new video fingerprint. Wu et al. [2009a] proposed an approach based on transformation recognition. In their approach, seven visual features were used to recognize the transformation types and accordingly to construct several detectors, each for one transformation type. Note that their recognition method was built on 10 single transformations, with average accuracy of 78.7%. Therefore, it is difficult to extend it to more complex transformations, even the combination of several transformations. Thus, we need to design a more reasonable strategy for transformation recognition. More importantly, an architecture is needed to organize different detectors in a principled way such that the good detection effective and high processing efficiency can be achieved.

3. TASC: TRANSFORMATION-AWARE SOFT CASCADING

In this section, we first summarize the multimodal detector cascading approach proposed in our previous work [Jiang et al. 2012; Tian et al. 2013] and then describe the TASC and its implementation.

3.1. Cascading of Multimodal Detectors

In Jiang et al. [2012] and Tian et al. [2013], multiple detectors based on complementary audiovisual features are organized in a cascade structure. Formally, the N-stage cascade can be expressed as $\mathbb{D}_N = \langle d_1, d_2, \ldots, d_N \rangle$ with a set of decision thresholds $\Theta_N = \{\theta_1, \theta_2, \ldots, \theta_N\}$, where d_n ($1 \leq n \leq N$) denotes the n^{th} detector and θ_n is its decision threshold. In this system, any query video q is first processed by d_1 where a positive detection result (i.e., the returned reference video r_1 with a similarity s_1 where $s_1 \geq \theta_1$) will lead to the acceptance of q as a copy; otherwise, the evaluation of d_2 on q will be triggered. Only if q is asserted as a noncopy by all detectors will it be accepted as a noncopy. In practice, most copies can be detected through the first several detectors.

Generally, three basic principles should be taken into account when designing such a cascade:

(1) *Complementarity*: The used detectors should be complement each other. One of them may be robust against certain transformations but vulnerable to the others; other detectors may be the other way around.
(2) *Simple-to-complex*: A series of detectors should be organized in a simple-to-complex order. Namely, efficient but simple detectors should be placed in the forepart, whereas effective but complex detectors should be located in the rear.
(3) *Terminated-by-PA*: To determine whether $\exists r \in \mathbb{R}$ for a query q such that $A(q, r)$ holds, q should be sequentially processed until one detector asserts it as a copy (i.e., a *Positive Assertion*) or all determine it as a noncopy.[2]

Note that different from the Terminated-by-NA policy that is employed in the boosting algorithm, the Terminated-by-PA strategy is used here mainly because each detector, if it is specially designed for some transformations, has a high confidence to make a positive assertion for a query video with these transformations; otherwise, it cannot determine if the query video is a noncopy, as the video is possibly subject to some other transformations. In this case, a noncopy query should be processed by all detectors in the cascade.

3.2. The TASC

This article further extends the multimodal detector cascading framework to a more general approach—TASC. Basically, three additional mechanisms are taken into account in the TASC:

(1) *Transformation-Awareness*: Several sets of detectors are organized in multiple cascade chains, each of which is devoted to process query videos of a specific category.
(2) *Detection-on-Copy-Units*: To judge whether two videos are near-duplicates, the decision should depend on the similarity between their most similar fractions CUs rather than their video-level similarity.
(3) *Soft-Decision-Boundary*: Instead of using a single decision threshold for each detector, a more flexible soft decision boundary strategy is used, by learning two thresholds for each detector and meanwhile training a nonlinear classifier for each chain.

In the following discussion, we will explain these mechanisms in detail and then present the TASC.

Considering the fact that query videos with different transformations may exhibit distinct audiovisual properties, it is obviously not an optimal solution to use the same set of detectors to process all query videos. Figure 2 shows two examples in which both are judged by the DCT-based detector as duplicates. However, the two frames in (a) are indeed duplicates, but those in (b) are not. Thus, two cascade chains should be constructed by respectively including or excluding the DCT-based detector. More generally, let $\mathcal{G}=\{g_1, \ldots, g_M\}$ denote the categories and $\mathbb{D}_m=\langle d_{m,1}, \ldots, d_{m,N_m}\rangle$ denote the detector chain for the m^{th} category where N_m is the number of detectors. Then the TASC

[2] For the NDVR task whose goal is to find $\forall r \in \mathbb{R}$ with $A(q, r)$, q should be sequentially processed throughout the cascade so that all such r are found out or all detectors determine it as a noncopy.

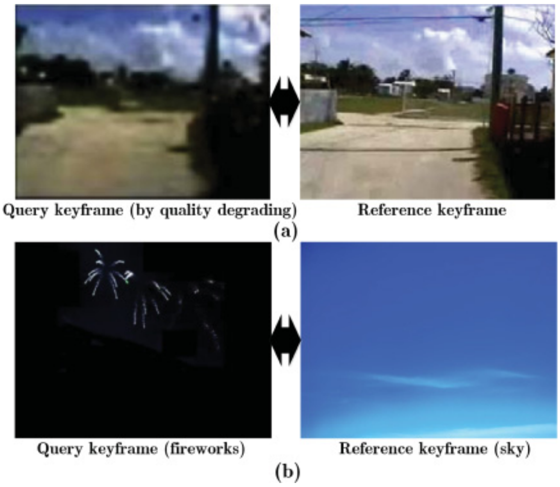
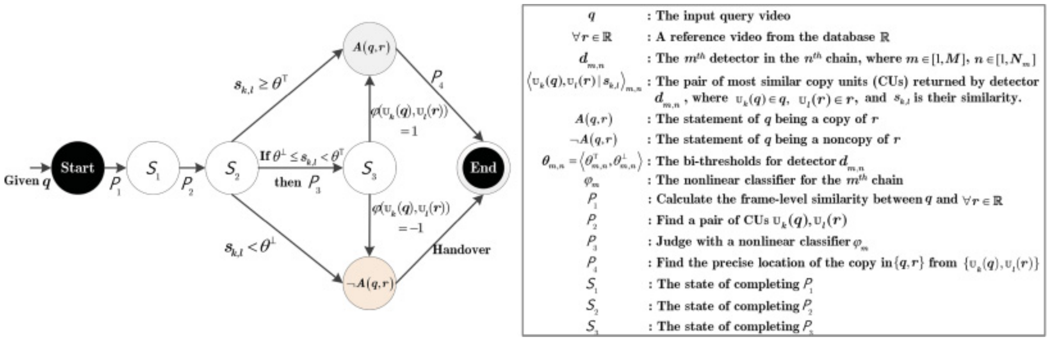

Query keyframe (by quality degrading)
Reference keyframe
(a)
Query keyframe (fireworks)
Reference keyframe (sky)
(b)

Fig. 2. Two examples that both are judged by the DCT-based detector as duplicates. However, the two frames in (a) are indeed duplicates, but those in (b) are not.


Given q
Start
P_1
S_1
P_2
S_2
$s_{k,l} \geq \theta^{\top}$
$A(q,r)$
P_4
If $\theta^{\perp} \leq s_{k,l} < \theta^{\top}$
then P_3
S_3
$\varphi(\upsilon_k(q),\upsilon_l(r)) = 1$
$\varphi(\upsilon_k(q),\upsilon_l(r)) = -1$
End
$s_{k,l} < \theta^{\perp}$
$\neg A(q,r)$
Handover

q	: The input query video
$\forall r \in \mathbb{R}$	: A reference video from the database $\mathbb{R}$
$d_{m,n}$	: The m^{th} detector in the n^{th} chain, where $m \in [1, M]$, $n \in [1, N_m]$
$\langle \upsilon_k(q), \upsilon_l(r) \mid s_{k,l} \rangle_{m,n}$	: The pair of most similar copy units (CUs) returned by detector $d_{m,n}$, where $\upsilon_k(q) \in q$, $\upsilon_l(r) \in r$, and $s_{k,l}$ is their similarity.
$A(q,r)$	: The statement of q being a copy of r
$\neg A(q,r)$	: The statement of q being a noncopy of r
$\theta_{m,n} = \langle \theta^{\top}_{m,n}, \theta^{\perp}_{m,n} \rangle$	: The bi-thresholds for detector $d_{m,n}$
φ_m	: The nonlinear classifier for the m^{th} chain
P_1	: Calculate the frame-level similarity between q and $\forall r \in \mathbb{R}$
P_2	: Find a pair of CUs $\upsilon_k(q), \upsilon_l(r)$
P_3	: Judge with a nonlinear classifier φ_m
P_4	: Find the precise location of the copy in $\{q, r\}$ from $\{\upsilon_k(q), \upsilon_l(r)\}$
S_1	: The state of completing P_1
S_2	: The state of completing P_2
S_3	: The state of completing P_3

Fig. 3. The state machine of a detector (with the nonlinear classifier). The main notations used in this section are also shown in the right side of the figure.

can be expressed as a set of transformation-aware chains of cascading detectors

$$\mathbb{D} = \left\{ \begin{array}{c} \mathbb{D}_1 \\ \mathbb{D}_2 \\ \vdots \\ \mathbb{D}_M \end{array} \right\} = \left\{ \begin{array}{cccc} d_{1,1} & d_{1,2} & \cdots & d_{1,N_1} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,N_2} \\ \vdots & \vdots & \ddots & \vdots \\ d_{M,1} & d_{M,2} & \cdots & d_{M,N_M} \end{array} \right\}, \tag{1}$$

where M is the number of chains (also the number of categories) and $d_{m,n}$ denotes the n^{th} detector in the m^{th} chain (with its state machine shown in Figure 3). Basically, the TASC consists of a preprocessing module, a transformation recognition module, and the detector chains $\mathbb{D}$. Given a query video q, the system first performs some preprocessing operations (e.g., frame extraction) and recognizes its category (denoted by g_m). Then q is processed successively by each detector in the m^{th} chain until one asserts it as a *copy* or all determine it as a *noncopy* of any video in the reference database $\mathbb{R}$.

The other two mechanisms are basically related to the design of detectors in the TASC. Legally or technologically, only videos in which the length of identical or similar content is more than a predefined value (e.g., 3 to 10 seconds) can be treated as duplicates or near-duplicates. Such a predefined minimum length defines the basic

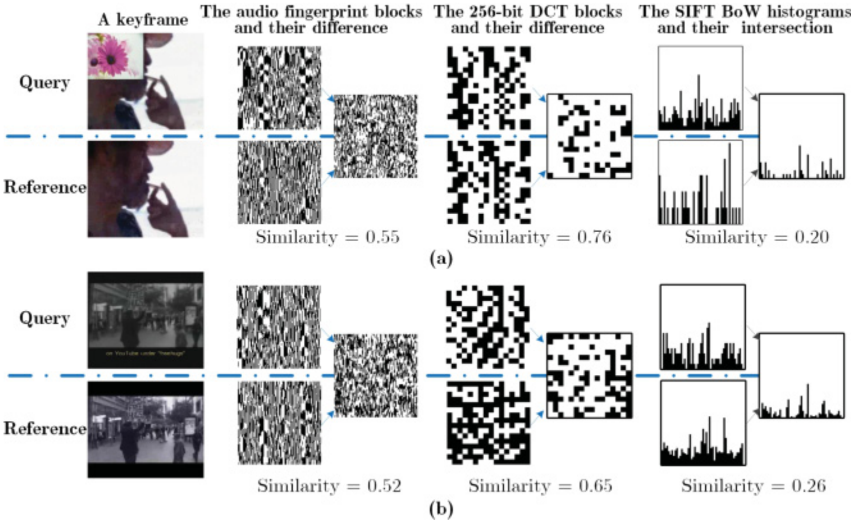


Fig. 4. Two examples that the three detectors all judge the two videos as noncopies by using the learned thresholds (0.65 for AFPs [Haitsma and Kalke 2012], 0.85 for DCT features [Mou et al. 2013], and 0.3 for SIFT BoW features [Douze et al. 2010]). In this figure, each AFP (and DCT) feature is depicted as a block diagram by using a black/white block to denote the bit 1/0, whereas the difference is also represented as a diagram where a black/white block denotes that their corresponding bits are different/the same.

unit for copy detection. Between two videos, we call the *most similar fractions with the predefined length* as a pair of CUs. Thus, to judge whether $A(q, r)$ holds for q and r, the decision should be based on the *segment-level* similarity between their CUs rather than the *video-level* similarity between q and r. In other words, let $\langle \mathbb{U}_k(q), \mathbb{U}_l(r)|s_{k,l}\rangle$ denote the CUs between q and r where k and l are their beginning locations, then

$$A(\mathbb{U}_k(q), \mathbb{U}_l(r)) \Rightarrow A(q, r). \tag{2}$$

Obviously, it is easier to judge whether two videos are copies or not by using the segment-level similarity of their CUs than by using the whole video-level similarity, as in practice many videos only share one or more near-duplicate segments with each other. Moreover, this mechanism can remarkably reduce the computational complexity of the following processes in copy detection.

Often, most of existing methods are to empirically set or experimentally train a single decision threshold for each detector. In the real-world applications, however, it is difficult to use only one threshold to perfectly divide the copies and noncopies. Two such examples are shown in Figure 4, where the three detectors all determine the query video as a noncopy of the reference video by using their optimal thresholds. In other words, the copies and noncopies are possibly linearly inseparable in the feature space. To address this problem, the TASC introduces a more flexible strategy—soft decision boundary—to replace the single threshold strategy. Its basic idea is to automatically learn two decision thresholds for each detector to examine the easy-to-judge copies and noncopies respectively, and meanwhile to train a nonlinear classifier for each chain to further check those hard-to-judge ones.

Formally, each detector $d_{m,n}$ can be expressed as a base hypothesis $\hbar_{m,n}$: $\mathbb{Q}_m \times \mathbb{R} \rightarrow \{-1, +1\}$, where $\mathbb{Q}_m$ is the m^{th} category of query videos, and for $q \in \mathbb{Q}_m$ and $r \in \mathbb{R}$, $y = \pm 1$ denote $A(q, r)$ and $\neg A(q, r)$, respectively. Then the final output of each chain $\mathbb{D}_m = \langle d_{m,1}, \ldots, d_{m,N_m}\rangle$ is always a convex combination of base hypotheses

$f_{\mathbf{w}_m}(q,r)=\sum_{n=1}^{N_m} w_{m,n}\hbar_{m,n}(q,r)$, where $w_{m,n}$ is the weight of $\hbar_{m,n}$ satisfying $w_{m,n}\geq 0$ and $\sum_{n=1}^{N_m} w_{m,n}=1$. Therefore, the (hard) margin of (q,r) can be defined as

$$\varrho(q,r,\mathbf{w}_m) = y(q,r)f_{\mathbf{w}_m}(q,r) = y(q,r)\sum_{n=1}^{N_m} w_{m,n}\hbar_{m,n}(q,r). \tag{3}$$

For the chain $\mathbb{D}_m$, the hard margin $\wp(\mathbf{w}_m)$ is thus defined as the smallest margin over the set $\mathbb{Q}_m$. In the TASC, the base hypothesis $\hbar_{m,n}$ is specified by two thresholds (i.e., the *upper threshold* $\theta_{m,n}^{\top}$ and the *lower threshold* $\theta_{m,n}^{\perp}$), where $\theta_{m,n}^{\top}$ is used to definitely determine whether $A(q,r)$ holds, whereas $\theta_{m,n}^{\perp}$ is used to definitely judge whether $\neg A(q,r)$ holds. This copy detection process is therefore called a *hard decision*.

For those video pairs that are not separable by convex combinations of the base hypotheses, the TASC also introduces an additional classifier, φ_m, for $\mathbb{D}_m$. Different from $d_{m,n}\in\mathbb{D}_m$, φ_m is a strong classifier based on the features with high discriminative ability. It can define a nonlinear boundary between copies and noncopies. In contrast, we call copy detection with the classifier φ_m a *soft decision*.

By assembling the two sorts of decision margins, the soft boundary of $\mathbb{D}$ can be expressed as

$$\Pi = \begin{bmatrix} \Theta_1|\varphi_1 \\ \Theta_2|\varphi_2 \\ \vdots \\ \Theta_M|\varphi_M \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_{1,1} & \cdots & \boldsymbol{\theta}_{1,N_1} & | & \varphi_1 \\ \boldsymbol{\theta}_{2,1} & \cdots & \boldsymbol{\theta}_{2,N_2} & | & \varphi_2 \\ \vdots & \ddots & \vdots & | & \vdots \\ \boldsymbol{\theta}_{M,1} & \cdots & \boldsymbol{\theta}_{M,N_M} & | & \varphi_M \end{bmatrix}, \tag{4}$$

where $\boldsymbol{\theta}_{m,n}=\langle\theta_{m,n}^{\top}, \theta_{m,n}^{\perp}\rangle$ are the bi-thresholds for $d_{m,n}$.

Figure 3 depicts the state machine of each detector $d_{m,n}$ in the TASC, together with the classifier φ_m. Given a query video q, $d_{m,n}$ should first perform two processes sequentially:

—The similarity evaluation process P_1, to calculate the frame-level similarity between q and $\forall r\in\mathbb{R}$;

—The CU search process P_2, to find a pair of CUs $\langle\mathbb{u}_k(q), \mathbb{u}_l(r)|s_{k,l}\rangle$ between q and $\forall r\in\mathbb{R}$.

Then the TASC can take action among three options:

(1) if $s_{k,l} \geq \theta_{m,n}^{\top}$, then $d_{m,n}$ asserts $A(q,r)$ holds;
(2) if $s_{k,l} < \theta_{m,n}^{\perp}$, then $d_{m,n}$ asserts $\neg A(q,r)$ holds; and
(3) otherwise, if $n < N_m$, then q will be handed over to the next detector $d_{m,n+1}$, whereas if $n = N_m$, $\langle\mathbb{u}_k(q), \mathbb{u}_l(r)\rangle$ should be further checked by the process P_3, where

—The soft decision process P_3, to utilize a nonlinear classifier φ_m to check $\langle\mathbb{u}_k(q), \mathbb{u}_l(r)\rangle$ to determine whether $A(q,r)$ holds or not.

Note that if the output of $d_{m,n}$ is that $A(\mathbb{u}_k(q), \mathbb{u}_l(r))$ holds (accordingly, $A(q,r)$ holds), then it will further perform the process P_4, where

—The CU-based localization process P_4, to find the precise locations of the copy segments in q and r that are with $\langle\mathbb{u}_k(q), \mathbb{u}_l(r)\rangle$ as the center.

In summary, the TASC provides a general framework to organize detectors in a cascade and transformation-sensitive way, which is expected to achieve high detection accuracy while minimizing the processing time. The detectors in each chain can be elaborately designed by utilizing the audiovisual characteristics of the query videos in that category. However, here the TASC does not specify how each detector works (e.g., utilizing which audio or visual features to perform the similarity evaluation process P_1), as long as it outputs a CU for a given query video. Thus, we will present an efficient implementation of the TASC in Section 3.3 and then discuss how to design the algorithms for the processes P_2 through P_4 in Section 4, which are basically independent of the variant implementations of the TASC.

3.3. A Four-Chain Implementation of the TASC

In practice, the implementation of the TASC is mainly related to three issues: which audiovisual features are utilized, how to design a reasonable classification strategy, and how to organize the detectors into several chains. In addition, the preprocessing operations should also be considered.

3.3.1. Preprocessing. In the implementation, some preprocessing operations should be performed first. Visual keyframes are obtained by uniformly sampling the visual component at a rate of 3 frames per second, whereas an audio frame with a length of 0.37 seconds is extracted from the audio signal for every interval of 11.6 milliseconds. As in Haitsma and Kalke [2012], the overlap factor of two consecutive audio frames is set to 31/32. Thus, for a 3-second-long video segment, a total of 10 visual keyframes and about 256 audio frames can be extracted. Often, this segment length should be consistent with the predefined length of CUs.

PiP detection is also performed in the preprocessing step. Instead of using a simple Hough transform–based method, we use a recently proposed PiP detection method [Qian et al. 2014] that introduces the spatiotemporal slicing to establish the probability measurement of the corresponding edge surface and then uses an optimization algorithm to refine vertical and horizontal edge lines. For queries with PiP, the foreground and nonforeground keyframes will be processed respectively to check whether the corresponding videos are copies. In addition, queries asserted as noncopies will be flipped and matched again to deal with flip transformation.

3.3.2. Multimodal Features. To keep robustness to diverse transformations, the system should extract several complementary features from audio and video frames [Tian et al. 2013]. Generally, three kinds of features should be used: audio, global, and local visual features. The complementarity of visual and audio features is obvious; that between global and local visual features lies in that the former is capable of resisting quality-degrading operations (e.g., blur, noise), whereas the latter can cope well with a wide range of other transformations [Mou et al. 2013]. In this implementation, the details about the used features are described as follows:

(1) *Audio feature*: Our previous systems [Tian et al. 2012, 2013; Mou et al. 2013] used the weighted audio spectrum flatness (WASF) [Chen and Huang 2008] as the audio feature. Despite having good performance for some audio transformations, such as MP3 compression, its computational complexity is high. Instead, this implementation uses a robust AFP proposed in [Haitsma and Kalke 2012]. AFP extracts a 32-bit subfingerprint for each audio frame by calculating energy differences along the frequency and time axes. As in Haitsma and Kalke [2012], the bit errors are used to measure the similarity between two AFPs, and all AFPs for reference videos are organized in a hash lookup table for quick search. Our experimental results in Section 6 show that AFP only takes 1/13 the processing time of WASF on the same dataset.
(2) *Global visual feature*: As a kind of compact and computationally efficient feature, the 256-D DCT [Mou et al. 2013] is still used as the global visual feature in this implementation. In the DCT-based detector, Hamming distance is used as the distance metric. For each frame in a reference video, the first 64 bits of its DCT feature

are indexed by locality sensitive hashing (LSH). Then a two-stage strategy can be adopted for fast DCT feature matching, which first searches in the LSH table for all candidates whose difference with the first 64 bits of the query frame is no more than 3 bits and then conducts the exact comparison of the whole DCT features between these candidates and the query frame. Note that here the comparison between two DCT features is computationally very efficient (only four CPU clock cycles when using the "popcnt" instruction).

(3) *Local visual feature*: In our previous systems [Tian et al. 2013; Mou et al. 2013], DC-SIFT [Bosch et al. 2008] was used as as the local feature. It can obtain excellent detection accuracy at the cost of a very long processing time. Thus, in this implementation, we use SIFT to replace DC-SIFT and also apply the BoW technique to convert each SIFT descriptor into a visual word (1,000 words generated from the Flickr 1M dataset). Meanwhile, the position in the 2×2 partition of the image, scale (large vs. small), orientation (quantized into 12 bins), and Laplacian response (positive vs. negative) are also taken into account. Only two features mapped to the same word and with similar position, scale, orientation, and Laplacian response can be regarded as a match. The similarity between two frames is defined as the average percentage of the matches. For all reference videos, SIFT BoWs are stored into an inverted index table for quick matching. Note that the original SIFT descriptors of each CU are also used in the soft decision process P_3 to further examine those hard-to-judge copies and noncopies.

3.3.3. Transformation Recognition. Generally, different transformations may produce different effects on audiovisual content. For example, some audio transformations, such as removal of audio signal or mixture with speech, will remarkably change the acoustic content of a video, whereas some others (e.g., mp3 compression and companding) will not. Similarly, video content is largely preserved after spatial or temporal content-preserving operations, such as format conversion, degrade in quality (e.g., noise addition, resolution change, and re-encoding). In contrast, video is notably modified after spatial or temporal content-altering operations such as cropping, PiP, and pattern insertion [Tian et al. 2013]. Therefore, ideally, we can group the query videos according to whether the audiovisual content is notably modified or not. Following this idea, we have tried a strategy that identifies three transformation categories: (1) content-preserving audio transformations, no matter which visual transformations exist; (2) content-altering audio transformations with content-preserving visual transformations; and (3) content-altering audio and visual transformations. However, the task of correctly recognizing these transformation categories for query videos is challenging and even more difficult than the copy detection tasks themselves. Our preliminary experiments show that by training two SVM classifiers for audiovisual content-altering detection, the average recognition accuracy can only reach 78.9%. In this case, the recognition errors will heavily influence the copy detection performance of the whole system.

Therefore, a more reasonable strategy should not only facilitate the utilization of different features but also should be easy to recognize the categories. Toward this end, this implementation adopts a strategy that classifies the query videos according to the *perceptual distinguishability* between them and their near-duplicates. We have shown two contrasting examples in Figure 2: the query keyframe in (a), despite being heavily transformed, can be easily determined as a copy of the reference keyframe; on the contrary, it is often hard to distinguish the two frames in (b) if their color information is ignored. This is mainly because we cannot extract discriminative features (e.g., DCT features) from those *low-contrast* images. Similar observation can also be found in the audio signal when inserting mute audio as an editing effect. Following these observations, four categories are identified in this implementation if a query video consists of the following types of frames:

—***G1***: Mute audio and low-contrast visual frames
—***G2***: Normal visual frames with mute audio
—***G3***: Normal audio frames and low-contrast visual frames
—***G4***: Normal audiovisual frames.

As such, the transformation recognition can be formulated as two simple detection tasks: mute audio detection and low-contrast frame detection. For mute audio detection, if the average energy of all 2,048 samples in an audio frame is lower than a very low value (e.g., 3), we can treat it as a mute frame. For low-contrast frame detection, each original frame is first transformed into the gray image and then quantized into 32 bins. After that, if the sum of any three adjacent bins takes a very high percentage (e.g., 97%), this frame can be treated as a low-contrast one. In practice, both detection tasks can obtain very high recognition rates. Moreover, the simple computation also makes the whole transformation recognition computationally efficient.

3.3.4. Design of Transformation-Aware Detector Chains. Given the preceding multimodal features and transformation recognition, the third issue is to organize the detectors into several transformation-aware chains. Obviously, for categories **G1** and **G2** where no (or very weak) audio signal exists in the query video, the AFP-based detector should not be used; similarly, for categories **G1** and **G3** where low-contrast visual frames exist in the video, the DCT-based detector should be excluded. Therefore, the detector chains can be expressed as

$$\mathbb{D} = \left\{ \begin{array}{c} \mathbb{D}_1 \\ \mathbb{D}_2 \\ \mathbb{D}_3 \\ \mathbb{D}_4 \end{array} \right\} = \left\{ \begin{array}{ccc} d_{\mathrm{SIFT_{BoW}}} & & \\ d_{\mathrm{DCT}} & d_{\mathrm{SIFT_{BoW}}} & \\ d_{\mathrm{AFP}} & d_{\mathrm{SIFT_{BoW}}} & \\ d_{\mathrm{AFP}} & d_{\mathrm{DCT}} & d_{\mathrm{SIFT_{BoW}}} \end{array} \right\}, \tag{5}$$

where $d_{\mathrm{SIFT_{BoW}}}$ (or d_{AFP}, d_{DCT}) denotes the detector that is based on the SIFT BoW feature (or the AFP and DCT features, respectively). Accordingly, the soft boundary for $\mathbb{D}$ in Equation (4) can be rewritten as

$$\Pi = \begin{bmatrix} \Theta_1|\varphi_1 \\ \Theta_2|\varphi_2 \\ \Theta_3|\varphi_3 \\ \Theta_4|\varphi_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_{1,1} & & & |\varphi_1 \\ \boldsymbol{\theta}_{2,1} & \boldsymbol{\theta}_{2,2} & & |\varphi_2 \\ \boldsymbol{\theta}_{3,1} & \boldsymbol{\theta}_{3,2} & & |\varphi_3 \\ \boldsymbol{\theta}_{4,1} & \boldsymbol{\theta}_{4,2} & \boldsymbol{\theta}_{4,3} & |\varphi_4 \end{bmatrix}. \tag{6}$$

Note that although the detector $d_{\mathrm{SIFT_{BoW}}}$ is used in all chains, the corresponding decision thresholds (i.e., $\boldsymbol{\theta}_{1,1}$, $\boldsymbol{\theta}_{2,2}$, $\boldsymbol{\theta}_{3,2}$, and $\boldsymbol{\theta}_{4,3}$) are different in most cases, as they are learned from different training data. The analogous observations also hold for d_{AFP} and d_{DCT}.

4. THE ALGORITHMS

This section will describe three algorithms used in the TASC, including the CU search algorithm for the process P_2, the soft boundary learning algorithm for the process P_3, and the CU-based localization algorithm for the process P_4. Note that these algorithms are basically independent of the implementation details of the TASC and thus can be applied to its different implementations. For more readability, Table I shows some main notations used in this section.

4.1. The CU Search Algorithm

Given the frame-level similarity results between a query video q_j and a reference video $\forall r \in \mathbb{R}$ provided by the process P_1, the CU search process P_2 to find a CU pair, denoted by $\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r)|s_{k,l}\rangle$ where k and l are their beginning locations, which is then used by

Table I. Some Main Notations Used in This Section

Notation	Meaning
$\mathbb{D}_m = \langle d_{m,1}, \ldots, d_{m,N_m} \rangle$	The m^{th} detector chain, in which $d_{m,n}$ is the n^{th} detector
$\mathbb{Q}_m = \{q_1, \ldots, q_{J_m}\}$	The training query videos for the m^{th} chain
$q_j = \{Q_1, \ldots, Q_{L_q}\} \in \mathbb{Q}_m$	A query video with L_q keyframes
$r = \{R_1, \ldots, R_{L_r}\} \in \mathbb{R}$	A reference video with L_r keyframes
$\mathbf{S} = [Sim(Q_{i'}, R_{j'})]_{L_q, L_r}$	The frame-level similarity matrix
$\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r) \mid s_{k,l} \rangle$	A CU pair in $\langle q_j, r \rangle$ with its similarity $s_{k,l}$
$P(k, l, \ell)$	Sum of the frame-level similarities in $\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r) \rangle$
$\mathbb{C}_{m,n}(q_j, r)$	All candidate CUs in $\langle q_j, r \rangle$ found by $d_{m,n}$
$\mathbb{U}^{\top}_{m,n}$ or $\mathbb{U}^{\perp}_{m,n}$	The CU set when training $\theta^{\top}_{m,n}$ or $\theta^{\perp}_{m,n}$
$\mathbb{U}^*_m$	The CU set when training φ_m for $\mathbb{D}_m$
$[t^{(\mathrm{B})}(x), t^{(\mathrm{E})}(x)]$	The precise location of a segment in x
$\Theta^*_m = [\boldsymbol{\theta}^*_{m,1}, \ldots, \boldsymbol{\theta}^*_{m,N_m}]$	The optimal thresholds for $\mathbb{D}_m$
$\Upsilon(q_j, r, \theta_{m,n})$	The decision cost of (q_j, r) by $\theta_{m,n}$
$\ddot{A}\ (q_j, r)$	The statement that q_j is a copy of r in the ground truth
$A(q_j, r)$	The statement that q_j is asserted as a copy of r by the system
ℓ	The predefined CU length
$\bar{\theta}_m$	The localization termination threshold

detector $d_{m,n}$ to decide whether $A(q_j, r)$, holds. Let ℓ denote the predefined length of a CU. We suppose that there are totally L_q keyframes in q_j (denoted by $Q_1, \ldots, Q_{L_q}$) and L_r keyframes in r (denoted by $R_1, \ldots, R_{L_r}$). Note that if $L_q < \ell$ or $L_r < \ell$, according to the Detection-on-Copy-Units mechanism, $d_{m,n}$ can directly determine that q_j is not a copy of r. Then the CU search algorithm is to find k^* and l^*, $1 \le k^* \le L_q$, $1 \le l^* \le L_r$, that maximize the following objective function:

$$P(k, l, \ell) = \sum_{i=0}^{\ell-1} Sim(Q_{k+i}, R_{l+i}), \tag{7}$$

where $Sim(X, Y)$ denotes the similarity between two frames X and Y. For the two videos q_j and r, there are at most $(L_q - \ell+1) \times (L_r - \ell + 1)$ possible CU pairs. Among them, only the ones whose $P()$ are larger than a predefined threshold can be viewed as the candidate CUs (denoted by $\mathbb{C}_{m,n}(q_j, r)$). In our experiments, the threshold is set to $\ell \times \theta^{\perp*}_{m,n}$, where $\theta^{\perp*}_{m,n}$ is the optimal lower threshold for $d_{m,n}$ that is learned using Algorithm 2.

Figure 5 visualizes two examples for the CU search, where each block diagram represents a frame-level similarity matrix. In Figure 5(a), there is only one copy segment between the two videos. In this case, each candidate CU pair (marked by the dotted blue lines) corresponds to a slant that has the length of ℓ. As such, we can find a slant (marked by the solid red line) that maximizes the sum of the similarity values between all frame pairs in this line. Figure 5(b) shows another case that there are more than one copy segments between the two videos. In this case, we need to find all candidate CUs and sort them in the descending order by similarity; then for each candidate CU, repeat the copy detection P_3 process and the localization process P_4 until no copy segment pairs can be found again.

To obtain the optimal solution $\{k^*, l^*\}$, we need to calculate $P()$ for all $(L_q - \ell+1) \times (L_r - \ell + 1)$ slants, using totally $(L_q - \ell+1) \times (L_r - \ell+1) \times (\ell-1)$ add operations. Namely, the computation complexity is approximately $\mathcal{O}(\ell L_q L_r)$. To reduce the computation, we observe that for a long slant starting from $(Q_{i'}, R_{j'})$ $(1 \le i' \le L_q - \ell,\ 1 \le j' \le L_r - \ell)$, with the length of K $(\ell \le K \le min(L_q - \ell,\ L_r - \ell))$, there are totally K calculations of $P()$ using Equation (7) (i.e., $P(i', j', \ell), P(i'+1, j'+1, \ell), \ldots, P(i'+K, j'+K, \ell)$). Here for any

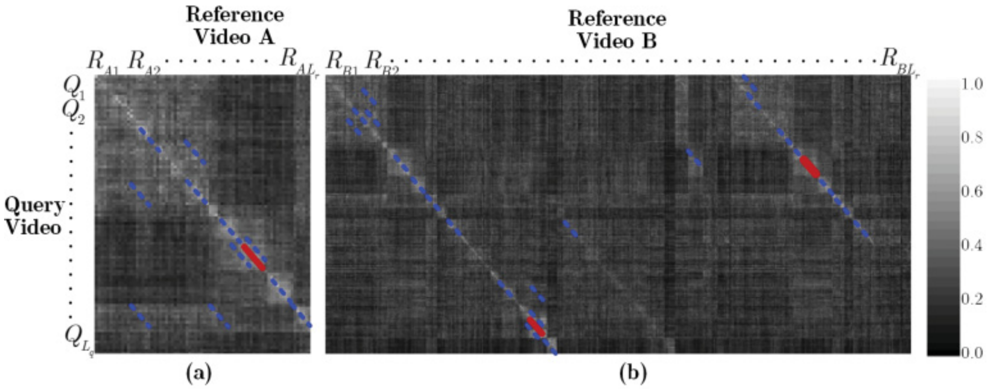


Fig. 5. Two examples for the CU search, where only one copy segment (a) and more than one copy segments (b) exist between between a query video and a reference video. Here each block diagram represents a similarity matrix between two videos, in which the dotted blue lines denote the candidate CUs, whereas the the solid red lines represent the desired CUs.

two adjacent terms $P(k, l, \ell)$ and $P(k+1, l+1, \ell)$ where $i' \leq k < i' + K$ and $j' \leq l < j' + K$, $P(k+1, l+1, \ell-1)$ is calculated twice. Then we introduce the matrix $\mathbf{E}$ by

$$\mathbf{E}(k, l) = \begin{cases} Sim(Q_k, R_l), & k = 1 \text{ } or \text{ } l = 1; \\ \mathbf{E}(k-1, l-1) + Sim(Q_k, R_l), & otherwise, \end{cases} \tag{8}$$

where $k \in [1, \ldots, L_q]$ and $l \in [1, \ldots, L_r]$. Then $P(k, l, \ell)$ can be rewritten as

$$P(k, l, \ell) = \mathbf{E}(k + \ell - 1, l + \ell - 1) - \mathbf{E}(k-1, l-1). \tag{9}$$

In other words, given the precalculated accumulative matrix $\mathbf{E}$, we only need one add operation for the calculation of $P()$, instead of $(\ell - 1)$ add operations directly using Equation (7). Since the calculation of $\mathbf{E}$ is about $\mathcal{O}(L_q L_r)$, the total computation complexity can thus be reduced to approximately $\mathcal{O}(L_q L_r + (L_q - \ell + 1) \times (L_r - \ell + 1)) \approx \mathcal{O}(2 L_q L_r)$. Algorithm 1 presents the pseudocode of the CU search process P_2.

Note that this algorithm needs the frame-level similarity matrix $\mathbf{S}$ from the similarity evaluation process P_1 as its input. Let $\delta_{m,n}$ denote the similarity evaluation time between two frames in detector $d_{m,n}$ (in terms of basic operations), then the computation complexity of P_1 is approximately $\mathcal{O}(\delta_{m,n} L_q L_r)$, which is much larger than $\mathcal{O}(2 L_q L_r)$ of the CU search process P_2. Actually, the computation of P_1 can also be accelerated by using the indexing structure of the used features (e.g., the hash lookup table for AFP, the LSH for DCT, and the inverted table for SIFT BoW). Given a query video q_j, each detector $d_{m,n}$ picks up top $\mathcal{K}$ similar reference keyframes (audio frames) for each query keyframe (audio frame) from the index structure, obtaining a collection $\{\mathbb{m}_{m,n}(Q_k, R_l)\}_{Q_k \in q_j, R_l \in r}$ of frame matches ($\mathcal{K} = 20$ in this work). After that, we can only calculate the similarity for all frame pairs in the slants that contain $\{\mathbb{m}_{m,n}(Q_k, R_l)\}$ and perform the process P_2 in these slants. In practice, the number of such slants is limited. Thus, the computation complexity of P_1 can be remarkably reduced.

4.2. The Soft Boundary Learning Algorithm

In the TASC, each detector $d_{m,n}$ takes two thresholds (i.e., $\theta^{\top}_{m,n}$ and $\theta^{\perp}_{m,n}$) to examine the easy-to-judge copies and noncopies (called *hard decision*), whereas each chain $\mathbb{D}_m$ also trains one nonlinear classifier to check those hard-to-judge ones (called *soft decision*). Thus, for $\mathbb{D}_m$, the soft boundary learning problem can be divided into two subtasks:

ALGORITHM 1: The CU Search Algorithm

```
Input: A query video $q_j = \{Q_1, \ldots, Q_{L_q}\}$, a reference video $r = \{R_1, \ldots, R_{L_r}\}$, the similarity
    matrix $\mathbf{S} = \left[Sim(Q_{i'}, R_{j'})\right]_{L_q, L_r}$, the $d_{m,n}$'s optimal lower threshold $\theta^{\perp *}_{m,n}$, and the
    predefined CU length $\ell$.
Output: A CU pair $\langle \Cup_k(q_j), \Cup_l(r) | s_{k,l} \rangle$ and the candidate CU set $\mathbb{C}_{m,n}(q_j, r)$.
1. Calculate the accumulative matrix E.
$\mathbf{E} \leftarrow \left[0.0\right]_{0 \sim L_q, 0 \sim L_r}, \mathbb{C}_{m,n}(q_j, r) \leftarrow \emptyset$;
for $i' = 1, \ldots, L_q; j' = 1, \ldots, L_r$ do
    $\mathbf{E}(i', j') \leftarrow \mathbf{S}(i', j') + \mathbf{E}(i' - 1, j' - 1)$;
end
2. Calculate the most similar segments.
$E_{Max} \leftarrow 0.0$;
for $i'=0, \ldots, L_q - \ell; j'=0, \ldots, min(L_r - \ell, L_q - \ell - i')$ do
    $E_{Cur} \leftarrow \mathbf{E}(i'+\ell, j'+\ell) - \mathbf{E}(i', j')$;
    if $E_{Cur} > E_{Max}$ then
        $E_{Max} \leftarrow E_{Cur}, k \leftarrow i' + 1, l \leftarrow j + 1$;
    end
    if $E_{Cur} > \ell \times \theta^{\perp *}_{m,n}$ then
        $\mathbb{C}_{m,n}(q_j, r) = \mathbb{C}_{m,n}(q_j, r) \cup \{i', j', E_{Cur}/\ell\}$;
    end
end
for $j'=1, \ldots, L_r - \ell; i'=0, \ldots, min(L_q - \ell, L_r - \ell - j')$ do
    $E_{Cur} \leftarrow \mathbf{E}(i'+\ell, j'+\ell) - \mathbf{E}(i', j')$;
    if $E_{Cur} > E_{Max}$ then
        $E_{Max} \leftarrow E_{Cur}, k \leftarrow i' + 1, l \leftarrow j' + 1$;
    end
    if $E_{Cur} > \ell \times \theta^{\perp *}_{m,n}$ then
        $\mathbb{C}_{m,n}(q_j, r) = \mathbb{C}_{m,n}(q_j, r) \cup \{i', j', E_{Cur}/\ell\}$;
    end
end
3. Return the CU.
$\Cup_k(q) \leftarrow \{Q_k, \ldots, Q_{k+\ell-1}\}$; $\Cup_l(r) \leftarrow \{R_l, \ldots, R_{l+\ell-1}\}$;
$s_{k,l} \leftarrow E_{Max}/\ell$.
```

learning bi-thresholds $\Theta^*_m = [\boldsymbol{\theta}^*_{m,1}, \ldots, \boldsymbol{\theta}^*_{m,N_m}]$ where $\boldsymbol{\theta}^*_{m,n} = \langle \theta^{\top *}_{m,n}, \theta^{\perp *}_{m,n} \rangle$ are the optimal bi-thresholds for $d_{m,n}$, and training a nonlinear classifier φ_m.

4.2.1. Learning Bi-Thresholds for Hard Decision. In Jiang et al. [2012] and Tian et al. [2013], a soft threshold learning algorithm was proposed to automatically determine the optimal thresholds. However, it learns a single decision threshold for each detector. Moreover, that algorithm is built on the video-level similarity between two videos, whereas the TASC utilizes the segment-level similarity of their CUs to decide whether or not they are near-duplicates. Therefore, we will describe how to extend that algorithm to learn bi-thresholds for hard decision in the TASC.

Ideally, for a detector $d_{m,n} \in \mathbb{D}_m$, its upper threshold $\theta^{\top}_{m,n}$ is used to definitely determine whether a query video q_j is a copy of r, whereas the lower one $\theta^{\perp}_{m,n}$ is used to definitely judge whether q_j is a noncopy. In other words, when $d_{m,n}$ uses $\theta^{\top}_{m,n}$ to judge whether $A(q_j, r)$ holds, the results should be without any false alarm (i.e., false positive). This is analogous to the NOFA profile in the TRECVID-CBCD task [Over et al. 2010] where a very high cost is set to an individual false alarm. Such an idea is also illustrated in Figure 6(a) through (c), which visualizes the bi-thresholds respectively for the AFP-, DCT-, and SIFT BoW-based detectors using the TRECVID-CBCD-2010 data.

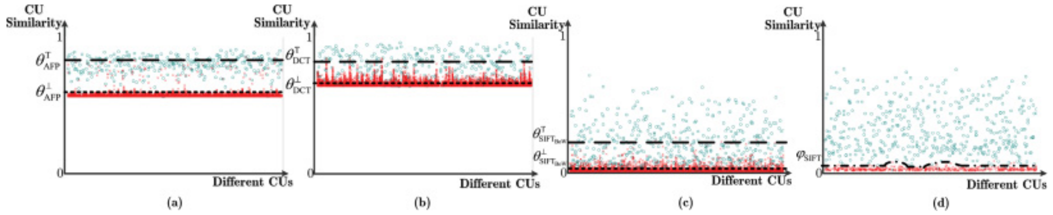


Fig. 6. Visualize the bi-thresholds for AFP (a), DCT (b), and SIFT BoW-based (c) detectors, as well as the soft boundary of the classifier φ_{SIFT}, using the TRECVID-CBCD-2010 data. In each subfigure, a cycle mark (o) denotes that the two CUs are duplicates in the ground truth, whereas a cross mark (x) denotes that they are nonduplicates; the dashed line annotated by $\theta^{\top}_{X}$ represents the upper threshold for the corresponding detector, whereas the dotted line annotated by $\theta^{\perp}_{X}$ represents the lower one.

Let $\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r) | s_{k,l} \rangle$ denote the CU pair of a query video $q_j \in \mathbb{Q}$ and a reference video $r \in \mathbb{R}$, where $s_{k,l}$ is the segment-level similarity between $\mathbb{u}_k(q_j)$ and $\mathbb{u}_l(r)$ (note that $s_{k,l}$ should be the maximal value among the similarity values of all candidate CUs $\mathbb{C}_{m,n}(q_j, r)$). We also let $\ddot{A}(q_j, r)$ denote the statement that q_j is indeed a copy of r in the ground truth, whereas $A(q_j, r)$ denote that q_j is asserted as a copy of r by the system. Thus, the decision cost of $d_{m,n}$ with $\theta^{\top}_{m,n}$ on $A(q_j, r)$, denoted by $\Upsilon(q_j, r, \theta^{\top}_{m,n})$, is evaluated in the following four cases:

(1) If $\ddot{A}(q_j, r)$ and $s_{k,l} \geq \theta^{\top}_{m,n}$, then it is a *true positive* (TP) and $\Upsilon(q_j, r, \theta^{\top}_{m,n})$ is set to zero.
(2) If $\neg\ddot{A}(q_j, r)$ but $s_{k,l} \geq \theta^{\top}_{m,n}$, then it is a *false positive* (FP) and $\Upsilon(q_j, r, \theta^{\top}_{m,n})$ is set to a large value c_L (e.g., $\mathrm{c}_L = 100$).
(3) If $\neg\ddot{A}(q_j, r)$ and meanwhile $s_{k,l} < \theta^{\top}_{m,n}$, then it is a *true negative* (TN) and $\Upsilon(q_j, r, \theta^{\top}_{m,n})$ is set to zero.
(4) Otherwise (namely, $\ddot{A}(q_j, r)$ and $s_{k,l} < \theta^{\top}_{m,n}$), $d_{m,n}$ is not able to utilize $\theta^{\top}_{m,n}$ to judge whether $\mathbb{u}_k(q_j)$ is or is not a copy of $\mathbb{u}_l(r)$, and thus $\Upsilon(q_j, r, \theta^{\top}_{m,n})$ is set to a small value c_S (e.g., $\mathrm{c}_S = 1$).

By summarizing the four cases, we can get

$$\Upsilon\left(q_j, r, \theta^{\top}_{m,n}\right) = \begin{cases} 0, & \left(\ddot{A}(q_j, r) \cap \left(s_{k,l} \geq \theta^{\top}_{m,n}\right)\right) \cup \left(\neg\ddot{A}(q_j, r) \cap \left(s_{k,l} < \theta^{\top}_{m,n}\right)\right); \\ \mathrm{c}_L, & \left(\neg\ddot{A}(q_j, r) \cap \left(s_{k,l} \geq \theta^{\top}_{m,n}\right)\right); \\ \mathrm{c}_S, & otherwise. \end{cases} \tag{10}$$

Note that if there are more than one copy segments between q_j and r (as shown in Figure 5(b)), the CUs from each copy segment should be used to calculate $\Upsilon(q_j, r, \theta^{\top}_{m,n})$.

Similarly, we can use $\theta^{\perp}_{m,n}$ as the decision threshold of $d_{m,n}$ to definitely reject noncopies and then derive the decision cost $\Upsilon(q, r, \theta^{\perp}_{m,n})$ as follows:

$$\Upsilon\left(q_j, r, \theta^{\perp}_{m,n}\right) = \begin{cases} 0, & \left(\neg\ddot{A}(q_j, r) \cap \left(s_{k,l} < \theta^{\perp}_{m,n}\right)\right) \cup \left(\ddot{A}(q_j, r) \cap \left(s_{k,l} \geq \theta^{\perp}_{m,n}\right)\right); \\ \mathrm{c}_L, & \left(\ddot{A}(q_j, r) \cap \left(s_{k,l} < \theta^{\perp}_{m,n}\right)\right); \\ \mathrm{c}_S, & otherwise. \end{cases} \tag{11}$$

For each detector $d_{m,n}$, the two thresholds $\theta^{\top}_{m,n}$ and $\theta^{\perp}_{m,n}$ work independently, thus they can be learned in two separate but similar training tasks. Given a training set $\mathbb{Q}$, we first divide it into M subsets, each of which corresponds to one category (denoted by $\mathbb{Q}_m = \{q_1, \ldots, q_{J_m}\}$ for the m^{th} category). Then for $\forall m \in [1, \ldots, M]$, we need to perform the similarity evaluation process P_1 to calculate the frame-level similarity between $\forall q_j \in \mathbb{Q}_m$ and $\forall r \in \mathbb{R}$. After that, we perform the CU search process P_2 to collect all CUs between

$\mathbb{Q}_m$ and $\mathbb{R}$, and then construct two subsets:

$$\begin{aligned} \mathbb{U}^{\top}_{m,n} &= \{\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r)\rangle | \forall q_j, \forall r, \ddot{A}(q_j, r), \omega^{\top}_{m,j} > 0\}, \\ \mathbb{U}^{\perp}_{m,n} &= \{\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r)\rangle | \forall q_j, \forall r, \neg\ddot{A}(q_j, r), \omega^{\perp}_{m,j} > 0\}, \end{aligned} \tag{12}$$

where $\omega^{\top}_{m,j} \in \mathbf{W}^{\top}_m$(or $\omega^{\perp}_{m,j} \in \mathbf{W}^{\perp}_m$) denotes the weight of q_j when training $\theta^{\top}_{m,n}$ (or $\theta^{\perp}_{m,n}$). Since the learning procedures for $\theta^{\top}_{m,n}$ or $\theta^{\perp}_{m,n}$ are almost the same, we use $\theta_{m,n}$ to denote either $\theta^{\top}_{m,n}$ or $\theta^{\perp}_{m,n}$ in the following discussion, with its corresponding CU set $\mathbb{U}_{m,n} \in \{\mathbb{U}^{\top}_{m,n}, \mathbb{U}^{\perp}_{m,n}\}$ and weight vector $\mathbf{W}_m \in \{\mathbf{W}^{\top}_m, \mathbf{W}^{\perp}_m\}$. Note that here the weight vector $\mathbf{W}_m$ is introduced such that detectors in each chain are enforced to only focus on the queries that are incorrectly detected by their antecessors. Thus,

$$\omega_{m,j} = \begin{cases} 1, & n = 1; \\ 0, & n > 1, \displaystyle\sum_{\forall r \in \mathbb{R}} \Upsilon\big(q_j, r, \theta^*_{m,n-1}\big) = 0; \\ \alpha_j \omega_{m,j}, & otherwise, \end{cases} \tag{13}$$

where $\theta^*_{m,n-1}$ is the learned upper or lower threshold for $d_{m,n-1}$ and $\alpha_j = \frac{\sum_{\forall r} \Upsilon(q_j, r, \theta^*_{m,n-1})}{\sum_{\forall q_j} \sum_{\forall r} \Upsilon(q_j, r, \theta^*_{m,n-1})/J_m}$ is a regularization factor .

By assembling the individual costs of $d_{m,n}$ on all CUs in $\mathbb{U}_{m,n}$, the overall decision cost of $d_{m,n}$ with respect to $\theta_{m,n}$ can be expressed as follows:

$$\varepsilon(\mathbb{U}_{m,n}, \theta_{m,n}) = \sum_{\forall \langle \mathbb{u}_k(q_j), \mathbb{u}_l(r)\rangle \in \mathbb{U}_{m,n}} \omega_{m,j} \times \Upsilon(q_j, r, \theta_{m,n}). \tag{14}$$

Therefore, the optimal threshold $\theta^*_{m,n}$ can be obtained by solving the following minimization problem:

$$\theta^*_{m,n} = \mathop{\arg\min}_{\theta_{m,n} \in [\check{s}_{m,n}, \hat{s}_{m,n}]} \varepsilon(\mathbb{U}_{m,n}, \theta_{m,n}), \tag{15}$$

where $\check{s}_{m,n}$ and $\hat{s}_{m,n}$ denote the minimal/maximal segment-level similarity values among CUs in $\mathbb{U}_{m,n}$.

As a result, the learning procedure for the bi-thresholds $\{\Theta_m\}_{1 \le m \le M}$ for all detectors in $\mathbb{D}$ is summarized in Algorithm 2.

4.2.2. Learning Nonlinear Classifiers for Soft Decision. As mentioned earlier, for a CU $\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r)\rangle$ whose similarity $s_{k,l}$ is between $\theta^{\top}_{m,n}$ and $\theta^{\perp}_{m,n}$, the detector $d_{m,n}$ is not able to utilize the bi-thresholds to determine whether or not they are duplicates. If the similar assertions are obtained throughout all detectors in $\mathbb{D}_m$, it should be further checked by a nonlinear classifier φ_m. Note that here φ_m requires that the input feature should characterize the similarity or difference between $\mathbb{u}_k(q)$ and $\mathbb{u}_l(r)$ in a CU.

In the field of copy detection, the keypoint matching based on the SIFT descriptor (not the SIFT BoW feature) is well recognized for its good stability and discriminative ability, despite that it is computationally expensive for large number of points and the high dimension [Liu et al. 2013a]. In this study, we choose the SIFT descriptor to describe visual characteristics of each 3-second-long video segment in a CU. This is computationally feasible since such a segment only contains 10 visual keyframes (i.e., $\ell = 10$), and the calculations in SIFT feature extraction can be reused with the SIFT BoW-based detector. Thus, given a CU pair $\langle \mathbb{u}_k(q), \mathbb{u}_l(r)\rangle$ where $\mathbb{u}_k(q) = \{Q_k, Q_{k+1}, \ldots, Q_{k+\ell-1}\}$ and $\mathbb{u}_l(r) = \{R_l, R_{l+1}, \ldots, R_{l+\ell-1}\}$, the frame-level similarity $Sim(Q_{k+i}, R_{l+i})$ is calculated as the average percentage of the SIFT keypoint matches between two frames $\forall Q_{k+i} \in \mathbb{u}_k(q)$

ALGORITHM 2: The Bi-Thresholds Learning Algorithm

```
Input: The $M$ chains $\mathbb{D}=\{\mathbb{D}_1, \dots, \mathbb{D}_M\}$ where $\mathbb{D}_m = \langle d_{m,1}, \dots, d_{m,N_m} \rangle$, a training set $\mathbb{Q}=\{\mathbb{Q}_1, \dots,$
       $\mathbb{Q}_M\}$ where $\mathbb{Q}_m=\{q_1, \dots, q_{J_m}\}$ for the $m^{th}$ category, and a reference database $\mathbb{R}$.
Output: $\Theta^*_m=[\boldsymbol{\theta}^*_{m,1}, \dots, \boldsymbol{\theta}^*_{m,N_m}]$ for $1 \leq m \leq M$, where $\boldsymbol{\theta}^*_{m,n}=\langle \theta^{\top *}_{m,n}, \theta^{\perp *}_{m,n} \rangle$ are the optimal bi-thresholds
        for $d_{m,n}$.
for $m = 1, \dots, M$ do
    1. Initialize weights $\{\omega^{\top}_{m,j}, \omega^{\perp}_{m,j}\} \leftarrow 1$ for $1 \leq j \leq J_m$;
    2. for $n = 1, \dots, N_m$ do
        2.1. Evaluate the frame-level similarity using the process $P_1$, for $\forall q_j \in \mathbb{Q}_m$ and $\forall r \in \mathbb{R}$;
        2.2. Perform the process $P_2$ to collect all CUs and then construct two CU sets:
        $\mathbb{U}^{\top}_{m,n}=\{\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r) \rangle | \forall q_j, \forall r, \ddot{A}(q_j, r), \omega^{\top}_{m,j}>0\}$,
        $\mathbb{U}^{\perp}_{m,n}=\{\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r) \rangle | \forall q_j, \forall r, \neg\ddot{A}(q_j, r), \omega^{\perp}_{m,j}>0\}$;
        for $\theta_{m,n} \in \{\theta^{\top}_{m,n}, \theta^{\perp}_{m,n}\}$, with its corresponding $\mathbb{U}_{m,n} \in \{\mathbb{U}^{\top}_{m,n}, \mathbb{U}^{\perp}_{m,n}\}$ and $\mathbf{W}_m \in \{\mathbf{W}^{\top}_m, \mathbf{W}^{\perp}_m\}$ do
            2.3. Evaluation the cost $\Upsilon(q_j, r, \theta_{m,n})$ for $\forall \langle \mathbb{u}_k(q_j), \mathbb{u}_l(r) \rangle \in \mathbb{U}_{m,n}$;
            2.4. Calculate the max/min similarities:
            $\hat{s}_{m,n}=\max\{s_{k,l} | \langle \mathbb{u}_k(q_j), \mathbb{u}_l(r) \rangle \in \mathbb{U}_{m,n}\}$,
            $\check{s}_{m,n}=\min\{s_{k,l} | \langle \mathbb{u}_k(q_j), \mathbb{u}_l(r) \rangle \in \mathbb{U}_{m,n}\}$;
            2.5. Find the optimal threshold $\theta^*_{m,n}$ for $d_{m,n}$ by Equation (15);
            2.6. Update $\omega_{m,j}$ using Equation (13), for $1 \leq j \leq J_m$;
        end
    end
    3. Return $\Theta^*_m = [\boldsymbol{\theta}^*_{m,1}, \dots, \boldsymbol{\theta}^*_{m,N_m}]$.
end
```

and $\forall R_{l+i} \in \mathbb{u}_l(r)$ where $0 \leq i < \ell$. Consequently, a ℓ-D vector is constructed as the input of φ_m by concatenating these frame-level similarities:

$$\mathbf{v}(\mathbb{u}_k(q), \mathbb{u}_l(r)) = [Sim(Q_k, R_l), \dots, Sim(Q_{k+\ell-1}, R_{l+\ell-1})]^{\mathrm{T}}. \tag{16}$$

In this study, we use SVM with soft margin (e.g., LibSVM [Chang and Lin 2011]) as the classifier, as it utilizes the kernel trick to map the original feature space into a high-dimensional space where a maximum soft margin hyperplane can be constructed for classification. For each chain $\mathbb{D}_m$, a nonlinear classifier φ_m can be trained over the training set $\mathbb{U}^*_m$ that contains the CUs between $\forall q_j \in \mathbb{Q}_m$ and $\forall r \in \mathbb{R}$ whose decision cost by the last detector d_{m,N_m} is larger than zero—that is,

$$\mathbb{U}^*_m = \{\langle \mathbb{u}_k(q_j), \mathbb{u}_l(r) \rangle | \Upsilon(q_j, r, \theta_{m,N_m}) > 0\}. \tag{17}$$

Figure 6(d) illustrates an example of the soft boundary trained on the TRECVID-CBCD-2010 dataset. We can see that compared to all other detectors shown in Figure 6(a) through (c), the CU similarity distribution when using the SIFT keypoint matching is linearly more separable. Thus, based on this, the soft-margin SVM can further improve the detection performance. It should also be noted that since only the SIFT descriptor is used in the nonlinear classifiers, the four-chain implementation of the TASC just needs to train two soft-margin SVMs: one for categories **G1** and **G3** and the other for categories **G2** and **G4**.

4.3. The Localization Algorithm

If a query q_j is asserted as a copy of $r \in \mathbb{R}$, then the remaining task is to locate the precise timestamps of the copy segments in q_j and r, namely $[t^{(\mathrm{B})}(q_j), t^{(\mathrm{E})}(q_j)]$ and $[t^{(\mathrm{B})}(r), t^{(\mathrm{E})}(r)]$. In our previous work [Jiang et al. 2012; Tian et al. 2013], a multiscale sequence matching method—temporal pyramid matching (TPM)—was proposed for the copy localization task. However, it cannot be applied here because the TASC adopts the

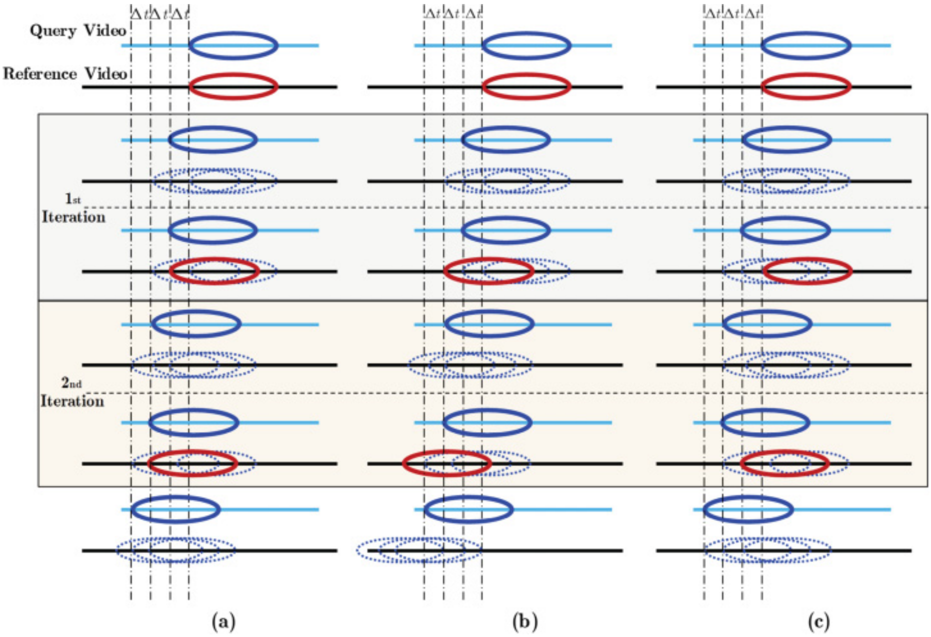


Fig. 7. Schematic illustration of the CU-based localization algorithm for the normal-speed (a), the frame rate–increasing (b), and frame-dropping (c) cases, respectively.

CUs as the sole basis for copy detection. In other words, the goal of the copy localization process P_4 is to find copy segments in q_j and r that are with the asserted CU as their center. Toward this end, we propose a CU-based localization algorithm.

Figure 7(a) illustrates the conceptual scheme of our CU-based localization algorithm. Given a query video $q_j = \{Q_1, \ldots, Q_{L_q}\}$ and a reference video $r = \{R_1, \ldots, R_{L_r}\}$, their CU pair $\langle \mathbb{u}_k(q_j) \mathbb{u}_l(r) \rangle$ is asserted as copies by the detector chain $\mathbb{D}_m$. Starting from this CU pair, a backward scanning and a forward scanning are used to determine the beginning and ending timestamps of the copy segments, respectively. Take the backward scanning as the example. Two sliding windows of the length ℓ are used to iteratively backward scan q_j and r. The sliding window in q_j starts from the starting frame of $\mathbb{u}_k(q_j)$ (i.e., the k^{th} frame) and moves backward with the step length of Δt at each iteration, whereas that in r starts from the starting frame of $\mathbb{u}_l(r)$ (i.e., the l^{th} frame) and moves backward with three step lengths (i.e., 0, Δt, $2\Delta t$) at each iteration. Note that here, Δt may be one to several frames (we set Δt=1 in the experiments). As such, we can obtain three possible matches between the windowed segments in q_j and r at each iteration (marked by the dotted ellipses in the figure). After that, the segment-level similarity is calculated for each match. The match with maximal similarity (marked by the bold solid ellipses) will be selected as the starting points at the next iteration. This iteration process will be terminated when the similarity of the selected match is lower than a threshold $\bar{\theta}_m$, where $\bar{\theta}_m$ can be calculated on the training set $\mathbb{Q}_m$ as the minimal segment-level similarity in all ground-truth copy segments.

Note that due to utilization of the variable step lengths for the sliding window on the reference video, this CU-based localization algorithm can cope well with the temporal content-altering operations such as frame-rate change and frame dropping. Figure 7(b) and (c) illustrate the two cases. Moreover, if more than one copy segments exist between

between q_j and r, and several CUs are asserted as copies by $\mathbb{D}_m$, we can repeat the CU-based localization algorithm for each CU pair.

Algorithm 3 summarizes the CU-based localization algorithm. Here, $P(i', j', \ell)$ is defined in Equation (7). In particular, if $i' \notin [1, L_q - \ell + 1]$ or $j' \notin [1, L_r - \ell + 1]$, then $P(i', j', \ell) = -\infty$.

```
ALGORITHM 3: The CU-Based Localization Algorithm
Input: A query video $q_j = \{Q_1, \dots, Q_{L_q}\}$, a reference video $r = \{R_1, \dots, R_{L_r}\}$, the similarity
       matrix $\mathbf{S} = \left[Sim(Q_{i'}, R_{j'})\right]_{L_q, L_r}$, the asserted CU $\langle \mathbb{u}_k(q), \mathbb{u}_l(r) | s_{k,l} \rangle$, the predefined CU
       length $\ell$, and the termination threshold $\bar{\theta}_m$.
Output: The copy locations $[t^{(B)}(q_j), t^{(E)}(q_j)]$ in $q_j$ and $[t^{(B)}(r), t^{(E)}(r)]$ in $r$.
1. The backward scanning.
$t^{(B)}(r) \leftarrow l$;
for $t^{(B)}(q_j) = k, \dots, 1$ do
    Calculate $P(t^{(B)}(q_j), t^{(B)}(r) - i, \ell)$ for $i \in [0, 2]$;
    $\{S_{max}, i^*\} = \max_{i \in [0,2]} P(t^{(B)}(q_j), t^{(B)}(r) - i, \ell)$;
    if $S_{max} < \ell \times \bar{\theta}_m$ then break;
    $t^{(B)}(r) \leftarrow t^{(B)}(r) - i^*$;
end
2. The forward scanning.
$t^{(E)}(r) \leftarrow l$;
for $t^{(E)}(q_j) = k, \dots, L_q - \ell + 1$ do
    Calculate $P(t^{(E)}(q_j), t^{(E)}(r) + i, \ell)$ for $i \in [0, 2]$;
    $\{S_{max}, i^*\} = \max_{i \in [0,2]} P(t^{(E)}(q_j), t^{(E)}(r) + i, \ell)$;
    if $S_{max} < \ell \times \bar{\theta}_m$ then break;
    $t^{(E)}(r) \leftarrow t^{(E)}(r) + i^*$;
end
$t^{(E)}(q_j) \leftarrow t^{(E)}(q_j) + \ell - 1; t^{(E)}(r) \leftarrow t^{(E)}(r) + \ell - 1$;
3. Return $[t^{(B)}(q_j), t^{(E)}(q_j)]$ and $[t^{(B)}(r), t^{(E)}(r)]$.
```

For a given query q_j and a reference video $r \in \mathbb{R}$, if the similarity matrix $\mathbf{S} = \left[Sim(Q_{i'}, R_{j'})\right]_{L_q, L_r}$ is precalculated, the computation of each scanning process is mainly ascribed to the computation of $P()$. Let L_c denote the frame number of the copy segment in q_j ($L_c \ll L_q$ in most cases), then the overall computation complexity of this algorithm is approximately $\mathcal{O}(3\ell L_c)$ and much lower than that of P_1 or P_2. Therefore, the process P_4 is computationally very efficient.

5. EXPERIMENTS

In this section, we discuss several experiments that were conducted to prove the effectiveness and efficiency of our TASC approach. The main objectives were twofold: (1) to explore how different components of the TASC works and (2) to evaluate whether the TASC can effectively and efficiently detect and locate video copies. Toward these ends, we adopted three most widely used benchmark datasets in the experiments, including TRECVID-CBCD [Kraaij and Awad 2011], MUSCLE-VCD-2007 [Law-To et al. 2007], and CC_WEB_VIDEO [Wu et al. 2007].

TRECVID-CBCD. The TRECVID-CBCD datasets [Kraaij and Awad 2011] are widely recognized as the largest and most challenging benchmarks for video copy detection. The 425-hour reference database contains 11,503 videos collected from the Internet, thereby diverse in content, style, format, and quality. Meanwhile, two query sets were constructed for the TRECVID-CBCD 2010 and 2011 tasks, respectively, by *randomly* applying a combination of 8 visual and 7 audio transformations (a total of 56 transformations) to three types of video: reference video only, reference video embedded into a nonreference video, and nonreference video only. Among them, the TRECVID-CBCD-2010 dataset contains a total of 10,936 query videos, whereas the TRECVID-CBCD-2011 dataset contains 11,256 query videos. In our experiments, the former was used to train the TASC and evaluate the performance of its different components, whereas the latter was used to evaluate its overall performance.

In the datasets, the detection results are often evaluated for each transformation in terms of normalized detection cost rate (NDCR), Mean F1, and mean processing time (MP-Time) [Kraaij and Awad 2011]. NDCR is the primary metric to evaluate the detection effectiveness as follows [Over et al. 2010]:

$$NDCR = P_{Miss} + \beta * R_{FA}, \tag{18}$$

where P_{Miss} is defined as the percentage of misses (i.e., false negatives) in all queries containing a copy, R_{FA} is calculated as the percentage of false alarms (i.e., false positives) measured on the full reference dataset, and β is a weighting factor.[3] The second measure, Mean F1, is used to assess the localization accuracy once a copy has been correctly detected. It is defined as the harmonic mean of precision and recall, where precision is the length percentage of the asserted copy that is indeed an actual copy, whereas recall is the length percentage of the actual copy that is subsumed in the asserted copy. The third measure, MP-Time, is the average processing time (in seconds) for a query. Obviously, the less NDCR, higher Mean F1, and shorter MP-Time, the better.

MUSCLE-VCD-2007 [Law-To et al. 2007]. MUSCLE-VCD-2007 consists of 101 videos with a total length of 80 hours, collected from different sources such as Web videos, TV archives, and movies with different bitrates, resolutions, and formats. Two sets of query videos were constructed by applying transformations to some reference and nonreference videos: ST1 with 15 videos and ST2 with 3 videos. For ST1, the task is to determine whether a query is a copy of a reference video. Thus, the evaluation metric, quality (**Q**), is calculated as the percentage of correct answers. For ST2, the task is to find the copy segments with the boundaries. Two evaluation metrics, QualitySegment (**QS**) and QualityFrame (**QF**), are used to measure the detection effectiveness and localization precision, respectively:

$$\mathbf{QS} = \frac{TP_{Seg} - FA_{Seg}}{N_{Seg}}, \tag{19}$$

$$\mathbf{QF} = 1 - \frac{F_{Miss}}{F_{Total}}, \tag{20}$$

where TP_{Seg} (or FA_{Seg}) is the number of correctly matched (or mismatched) video segments, N_{Seg} is the total number of segments in all queries, F_{Miss} denotes the number of mismatched frames, and F_{Total} is the total number of frames in all queries.

CC_WEB_VIDEO [Wu et al. 2007]. CC_WEB_VIDEO contains 24 sets of video clips (a total of 12,790 videos) collected from YouTube, Yahoo! Video, and Google Video. For each set of videos, the most popular video is used as the query video, whereas the other videos are labeled as “redundant” or “irrelevant” in the ground truth. These redundant videos are (approximately) identical to the query video but different in lengths,

[3] Two profiles are evaluated at the TRECVID-CBCD task: NOFA, which aims to reduce the false alarm rate to 0. and BALANCED, which sets an equal cost for false alarms and misses. For simplicity, this article only reports the results on the BALANCED profile.

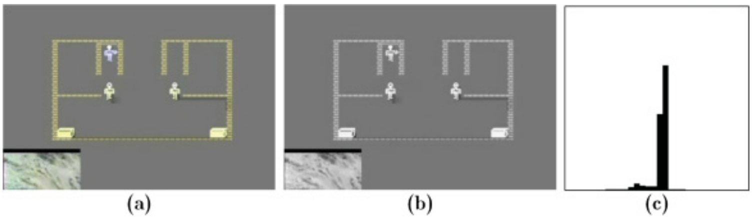


Fig. 8. An example of the misrecognized low-contrast frames: the original frame (a), its gray image (b), and its histogram (c).

formats, encoding parameters, photometric variations (color, lighting changes), editing operations (caption, logo, and border insertion), and certain modifications (frames add/remove). The performance is evaluated by mean average precision (MAP) as follows:

$$AP = \frac{1}{J}\sum_{i=1}^{J}\frac{i}{r_i}, \tag{21}$$

where J is the number of relevant videos and r_i is the rank of the i^{th} relevant video. Meanwhile, the *precision-recall curve* is also used to assess the performance of near-duplicate detection.

Note that in the experiments, we used the TRECVID-CBCD-2010 dataset as the training set for the TASC. Then the learned parameters and classifiers were applied to the other three datasets. This is reasonable, because our criteria for copy detection should remain the same, whichever sources from these datasets were constructed. Of course, this also presents a greater challenge for the TASC. All experiments were carried out on a Windows Server 2008 with 32-core 2.0GHz CPUs and 32GB memory.

5.1. How It Works

In the first set of experiments, the main objective was to explore how different components of the TASC work, including the transformation recognition, the individual detectors, the soft decision boundary with bi-thresholds and nonlinear classifiers, and the CU-based localization. In these experiments, we randomly divided the TRECVID-CBCD-2010 dataset into four folds of equal size: one subset was used for training, whereas the other three subsets were used for validation.

5.1.1. Transformation Recognition Results. This experiment evaluated the effectiveness of the transformation recognition in the TASC. To do so, we manually labeled the category IDs for all query videos in the TRECVID-CBCD-2010 dataset. For low-contrast frames, we labeled the ground truth through subjective evaluation, whereas for mute audio clips, the ground truth could be obtained using the automatic detection tool with manual verification. In the TASC, the transformation recognition is formulated as two simple detection tasks: mute audio detection and low-contrast frame detection. In the experiment, the recognition rates of mute audio and low-contrast frames could reach 100% and 96%, respectively. Figure 8 shows a failure example of the low-contrast frame detection, where the given high-contrast frame is misrecognized as a low-contrast one because its gray histogram has two dominant neighboring bins. Note that this is also the reason why we need to manually label the ground truth for low-contrast frames. Given the detection results of mute audio and low-contrast frames in a CU, a decision rule was used to determine the category of that CU according to the definition of transformation categories. Table II shows the corresponding confusion matrix. On average, the average

Table II. The Confusion Matrix of Transformation Recognition on the TRECVID-CBCD-2010 Dataset

		Asserted Category			
		G1	***G2***	***G3***	**G4**
Actual Category	***G1***	**0.99**	0.01	0	0
	G2	0.05	**0.95**	0	0
	G3	0	0	**0.99**	0.01
	G4	0	0	0.04	**0.96**

transformation recognition accuracy is about 97.3% in the four categories. The results show that the transformation recognition in the TASC is very effective.

5.1.2. Performance of Individual Detectors. The second experiment evaluated the performance of individual detectors under various audiovisual transformations. Five detectors were involved in this experiment, including SIFT BoW, DCT, and AFP used in the TASC and DC-SIFT BoW and WASF used in our previous work [Tian et al. 2013; Mou et al. 2013]. Note that here all of these detectors utilize the CUs as the sole basis for copy detection and localization.

Table III(a) shows the detection performance of individual detectors in terms of NDCR and MP-Time. Here, for each detector, we used the soft threshold learning algorithm [Tian et al. 2013] to train a single optimal threshold. Overall, the detection results of all detectors are much worse than those in Tian et al. [2013] and Mou et al. [2013]. This is mainly because the features extracted from each CU pair are much less that those from the whole video segments. Among the three visual detectors, the performance of the DC-SIFT–based detector is remarkably better than those of SIFT-based and DCT-based detectors. However, its MP-Time is nearly 17 times that of the SIFT-based detector, making it practically infeasible to many real-world copy detection tasks. Comparatively, the DCT-based detector performs better on some content-preserving transformations such as V4, whereas the performance of the SIFT-based detector is superior on several content-altering transformations, namely V3, V5, and V8. As for the efficiency, their MP-Times are varied for different transformations. For example, MP-Times for V2, V8, and V10 are much longer than those for the other transformations. This is because an additional processing will be paid for PiP and flip effects in all V2 videos and parts of V8 and V10 videos. One surprising finding is that both of the two detectors seemed totally incapable of resisting V1, V2, V6, and V10 (i.e., with NDCRs of larger than 0.6). Through an in-depth analysis, we found that this might be caused by the “single decision threshold” strategy, since at least one of them could correctly detect most of copies for query videos that were subject to these transformations if a small number (e.g., 10) of false alarms were tolerated. This indicates that the used detectors may be effective for CU-based copy detection, although what we really need is probably to introduce a more flexible decision strategy for each detector.

To further validate this conjecture, we conducted a supplementary experiment by learning the optimal bi-thresholds using Algorithm 2 and implementing a simplest nonlinear classifier for each detector. This nonlinear classifier, called AvgSim, was simply based on the average SIFT-based similarity among all frames in a CU pair. Given all CUs that were identified by a specific detector, it thus could be used to judge the ones whose similarities were between the two thresholds. In this case, the more accurately the detector identified the CUs, the higher the detection performance that could be achieved. Table III(b) shows the detection performance of individual detectors with the learned bi-thresholds and the simple AvgSim classifier. We can see that the detection performance of the three visual detectors improves significantly. In particular, the average NDCR of the SIFT-based detector improved from 0.701 to 0.255, whereas

Table III. Detection Performance of Individual Detectors on the TRECVID-CBCD-2010 Dataset[1]

(a) Each detector with a single threshold learned by the soft threshold learning algorithm.

Metric	Detector	V1	V2	V3	V4	V5	V6	V8	V10	**AVG**
		(A1)	(A2)	(A3)	(A14)	(A5)	(A6)	(A7)		
NDCR	DC-SIFT BoW	0.636	0.569	0.208	0.292	0.115	0.446	0.415	0.492	**0.397**
	SIFT BoW	1.000	0.869	0.246	1.000	0.284	0.746	0.661	0.799	**0.701**
	DCT	0.908	0.738	0.777	0.331	0.514	0.744	0.885	0.892	**0.724**
	WASF	0.806	0.776	1.000	1.000	0.910	0.933	1.000	—	**0.918**
	AFP	0.238	0.262	0.269	0.262	0.285	0.308	0.300	—	**0.275**
MP-Time[2]	DC-SIFT BoW	172.493	513.615	243.896	260.327	189.152	234.372	406.465	379.177	**299.937**
	SIFT BoW	12.384	27.400	14.267	17.788	11.632	14.455	22.621	21.210	**17.720**
	DCT	4.663	9.991	6.452	5.728	4.710	5.236	9.387	8.642	**6.851**
	WASF	8.979	8.988	8.986	8.979	8.979	8.980	8.979	—	**8.981**
	AFP	0.777	0.740	0.556	0.640	0.778	0.582	0.672	—	**0.678**

(b) Each detector with the learned bi-thresholds and a simple implementation of the nonlinear classifier that is based on the average SIFT-based similarity among all frames in a CU pair.

Metric	Detector	V1	V2	V3	V4	V5	V6	V8	V10	**AVG**
		(A1)	(A2)	(A3)	(A14)	(A5)	(A6)	(A7)		
NDCR	DC-SIFT BoW	0.177	0.215	0.092	0.092	0.085	0.154	0.138	0.262	**0.152**
	SIFT BoW	0.223	0.208	0.092	0.437	0.100	0.400	0.169	0.414	**0.255**
	DCT	0.615	0.623	0.362	0.169	0.200	0.338	0.715	0.554	**0.447**
	WASF	0.470	0.470	0.575	0.940	0.597	0.746	0.948	—	**0.678**
	AFP	0.238	0.262	0.262	0.262	0.277	0.285	0.277	—	**0.266**
MP-Time[2]	DC-SIFT BoW	173.289	515.711	244.831	261.302	190.248	235.294	407.829	380.590	**301.137**
	SIFT BoW	12.717	28.238	14.612	18.307	12.230	15.109	23.126	21.914	**18.282**
	DCT	4.807	10.639	7.109	6.142	5.831	6.143	10.031	9.238	**7.493**
	WASF	16.948	16.925	16.735	16.820	16.865	16.688	16.763	—	**16.820**
	AFP	7.181	7.146	6.962	7.044	7.181	6.987	7.077	—	**7.083**

[1] On the TRECVID-CBCD datasets, transformations are V1. Cam-cording; V2. PiP; V3. Insertions of pattern; V4. Re-encoding; V5. Change of gamma; V6. Decrease in quality; V8. Postproduction; V10. Combination of three randomly chosen transformations; 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 compression; A7. Bandwidth filter, mix with speech, and compression.
[2] Here the MP-Times are excluded the processing times (in seconds) for CU-based localization.

that of the DCT-based detector improved from 0.724 to 0.447, only at the additional cost of about 1-second MP-Time for AvgSim. This shows that the soft decision boundary strategy can indeed improve the detection performance of each detector remarkably. Moreover, the DCT-based detector shows good robustness (lower NDCR values) to V2 and V4, whereas the SIFT-based detector can achieve better performance on other visual transformations. In other words, they demonstrate a strong complementarity in dealing with different visual transformations.

For audio, Table III(a) shows that the NDCR values of the WASF-based detector are all close to one. On the contrary, the AFP-based detector shows very good detection performance on nearly all audio transformations. More importantly, the AFP-based detector is computationally very efficient since it only takes about $^1/_{13}$ processing time that of the WASF-based detector on this dataset. Table III(b) also shows the detection performance of the two audio detectors with the learned bi-thresholds and the simple AvgSim classifier. We can see that the NDCRs of the AFP-based detector are slightly improved (from 0.275 to 0.266). This means that most of the CUs identified by the AFP-based detector can be effectively recognized as copies or noncopies using either a single decision threshold or bi-thresholds. We also noted that AvgSim took about

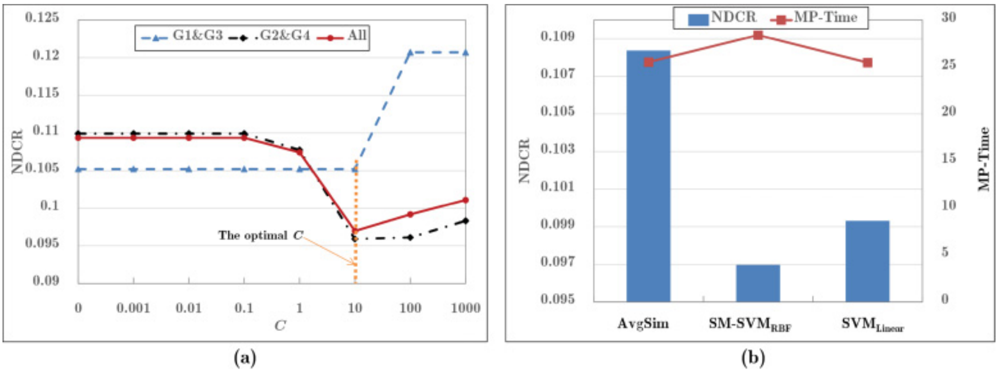


Fig. 9. (a) The NDCR curves when different C values are adopted in SM-SVM$_{RBF}$. (b) The detection performance of the TASC with three implementations of the nonlinear classifiers: AvgSim, SM-SVM$_{RBF}$ with $C = 10$ and SVM$_{Linear}$.

6 to 8 seconds on average for each audio detector. This time was mainly paid for SIFT feature extraction and similarity evaluation for both the original and flipped visual frames.

Overall, the experimental results can support our conjecture about the complementarity of SIFT BoW, DCT, and AFP features. In other words, none of them can resist all of the transformations, whereas a good overall performance may be achieved by appropriately combining them together. Moreover, the results also confirm the necessity of utilization of the soft decision boundary strategy for copy detection.

5.1.3. Performance of Soft Decision Boundary. After experimentally verifying its necessity, this experiment evaluated the performance of the soft decision boundary strategy in the TASC. As mentioned in Section 4.2, the TASC utilizes the soft-margin SVMs as the nonlinear classifiers. In our experiments, the soft-margin SVMs were equipped with the RBF kernel (denoted by SM-SVM$_{RBF}$) and implemented on the LibSVM [Chang and Lin 2011]. Two comparison methods were used as well, including the simplest AvgSim, and SVM$_{Linear}$ (i.e., the SVM classifier using the linear kernel). It should be noted that the four-chain implementation of the TASC just needs to train two nonlinear classifiers: one for categories **G1** and **G3** and the other for categories **G2** and **G4**.

Figure 9(a) shows the NDCR curves when different regularization parameter C values (from 0 to 1,000) are adopted in SM-SVM$_{RBF}$. Clearly, when $C = 0$, SM-SVM$_{RBF}$ becomes the hard-margin SVM. We can see that the best performance of SM-SVM$_{RBF}$ could be achieved when $C = 10$, whether for a single chain or for all chains. Thus, in the rest of experiments, we used SM-SVM$_{RBF}$ with $C = 10$.

Figure 9(b) compares both NDCRs and MP-times of the TASC for three implementations of the nonlinear classifiers. Clearly, the detection effectiveness of SM-SVM$_{RBF}$ is remarkably better than both AvgSim and SVM$_{Linear}$ despite that its MP-Time is slightly longer (more than 3 seconds on average).

5.1.4. Performance of CU-Based Localization. The fourth experiment evaluated the effectiveness and efficiency of the CU-based localization algorithm. To do so, we evaluated the localization precision (in terms of Mean F1) for different detectors and the whole TASC. Table IV shows the results. We can see that for individual detectors, the SIFT-based detector has the best Mean F1 of 0.945, slightly better than that of the DCT- and AFP-based ones (0.935 and 0.913, respectively). Overall, the Mean F1 of the TASC achieves 0.938. Considering the fact that the visual keyframes are extracted at a rate of three frames per second, this localization precision is pretty good. Meanwhile, the

Table IV. The Localization Performance of CU-Based Localization with Different Detectors and the Whole TASC on the TRECVID-CBCD-2010 Dataset

Metric	Detector	V1 (A1)	V2 (A2)	V3 (A3)	V4 (A14)	V5 (A5)	V6 (A6)	V8 (A7)	V10	**AVG**
Mean F1	SIFT BoW	0.926	0.949	0.952	0.936	0.951	0.938	0.956	0.956	**0.945**
	DCT	0.921	0.944	0.954	0.914	0.958	0.943	0.935	0.915	**0.935**
	AFP	0.941	0.937	0.925	0.926	0.888	0.886	0.886	—	**0.913**
	TASC	0.929 (0.942)	0.944 (0.941)	0.944 (0.939)	0.914 (0.938)	0.943 (0.934)	0.949 (0.935)	0.951 (0.935)	0.929 —	**0.938**
MP-Time*	SIFT BoW	0.057	0.056	0.057	0.057	0.057	0.057	0.057	0.057	**0.057**
	DCT	0.027	0.026	0.026	0.026	0.026	0.026	0.026	0.026	**0.026**
	AFP	0.010	0.010	0.010	0.010	0.010	0.010	0.010	—	**0.010**
	TASC	0.034 (0.031)	0.035 (0.031)	0.030 (0.031)	0.025 (0.031)	0.027 (0.031)	0.027 (0.031)	0.040 (0.031)	0.031 —	**0.031**

*Here the MP-Times are the mean processing times (in seconds) only for CU-based localization while excluding times for all other operations.

average MP-Time of 0.031 seconds also validates the high efficiency of the CU-based localization algorithm in the TASC.

5.2. Whether It Works

In the second set of experiments, the main objective was to see whether our TASC approach could really work. Toward this end, we compared it with several state-of-the-art methods on the TRECVID-CBCD-2011, MUSCLE-VCD-2007, and CC_WEB_VIDEO datasets by using the parameters and classifiers trained on the TRECVID-CBCD-2010 dataset. On each dataset, we strictly followed the same evaluation proxy that was originally proposed by the dataset designer to make the experimental results comparable to other methods.

5.2.1. Results on the TRECVID-CBCD-2011 Dataset. This experiment compared the TASC with several cutting-edge methods on the TRECVID-CBCD-2011 dataset. These methods included (1) two variants of our previous system, the single soft threshold method in Jiang et al. [2012] and Tian et al. [2013] (i.e., $\mathbb{D}_3^{(\mathrm{S})} = \langle d_{\mathrm{WASF}}^{(\mathrm{S})}, d_{\mathrm{DCT}}^{(\mathrm{S})}, d_{\mathrm{DCSIFT}}^{(\mathrm{S})} \rangle$, denoted by "SoftD3"), and the hard threshold version ($\mathbb{D}_3^{(\mathrm{H})} = \langle d_{\mathrm{WASF}}^{(\mathrm{H})}, d_{\mathrm{DCT}}^{(\mathrm{H})}, d_{\mathrm{DCSIFT}}^{(\mathrm{H})} \rangle$, denoted by "HardD3") [Tian et al. 2012; Mou et al. 2013] that achieved the *best* overall performance at the TRECVID-CBCD-2011 task; (2) two best methods from the other 21 participants at the TRECVID-CBCD-2011 task, CRIM-VISI [Gupta et al. 2011] and INRIA-LEAR [Ayari et al. 2011], and the median performances on each transformation among all approaches at this task [Kraaij and Awad 2011], denoted by "Median"; (3) three state-of-the-art methods, including the subspace learning–based video fingerprinting (SLFP) method [Cirakman et al. 2012], the nearest-neighbor mapping (NNM) method [Gupta et al. 2012], and the randomly projected binary features (RPBF) method (RPBF) [Wu et al. 2012]. Note that among the last three methods, only the NNM method [Gupta et al. 2012] presents the detection results on different transformations on this dataset.

Table V shows the comparison results. We can see that the TASC obtained the best detection performance (with average NDCR of 0.047) and comparable localization precision (with average Mean F1 of 0.947). SoftD3 also achieved very good NDCR and Mean F1. Among the other three state-of-the-art methods, RPBF got the best Mean F1 (i.e., 0.952 on average) but had very poor detection performance, whereas NNM could obtain a comparable average NDCR.

Official evaluation results at the TRECVID-CBCD-2011 task showed that our HardD3 method achieved excellent NDCR performance (e.g., 34 best "Actual NDCR"

Table V. Comparison between the TASC and Several State-of-the-Art Results on the TRECVID-CBCD-2011 Dataset

Method		Avg. NDCR	Avg. Mean F1	Avg. MP-Time
TASC		**0.047**	0.947	**26.823**
State-of-the-Art Methods	SoftD3	0.054	0.951	163.184
	NNM	0.102	0.833	NA[2]
	RPBF	0.545	**0.952**	NA
	SLFP	0.900	0.800	NA
TRECVID-CBCD-2011 Evaluation[1]	HardD3	0.055	0.950	172.291
	CRIM-VISI	0.159	0.715	2,792.014
	INRIA-LEAR	0.217	0.943	2,079.294
	Median	1.050	0.889	191.535

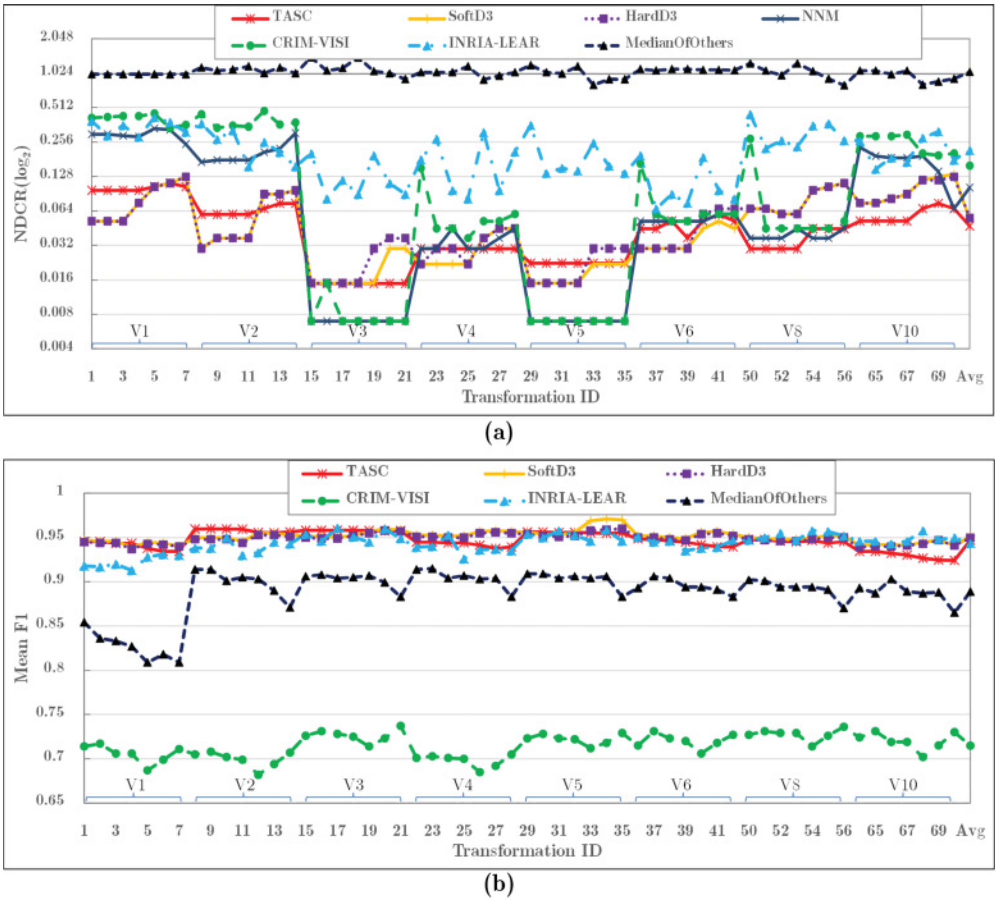
[1] Their MP-Times are only used as references since they were executed on different platforms.
[2] NA means that the paper did not provide the corresponding results (similarly hereinafter).

for the BALANCED profile) and very good Mean F1 performance (i.e., average F1 of 0.95 on all transformations) [Kraaij and Awad 2011]. Comparatively, CRIM-VISI won 20 best NDCR (particularly on V3 and V5) but had very poor localization precision with average Mean F1 of 0.715, whereas INRIA-LEAR showed good Mean F1 (0.944, on average). As an extension of HardD3 and its soft threshold variant SoftD3, the TASC can further improve the detection effectiveness. Figure 10 depicts the performance curves of these methods on the 56 transformations. We can see that the TASC even exhibits excellent detection accuracy on some most complex transformations, such as V8 (i.e., postproduction) and V10 (i.e., combinational transformations). More importantly, the MP-Time of the TASC is only about $^{1}/_{6}$ of SoftD3 and HardD3, demonstrating very high computational efficiency.

5.2.2. Results on the MUSCLE-VCD-2007 Dataset. This experiment evaluated the performance of the TASC on the MUSCLE-VCD-2007 dataset by using the parameters and classifiers trained on the TRECVID-CBCD-2010 dataset. Our objective was to test the robustness and generalization of the TASC across different datasets. By comparison, eight state-of-the-art results on this dataset were also cited directly from the literature, including Anguera et al. [2009], Tan et al. [2009], Cui et al. [2010], Yeh and Cheng [2011], Zheng et al. [2011], Ren et al. [2012], Kim et al. [2014a], and Wu and Aizawa [2014]. For simplicity, they are denoted by Anguera2009, Tan2009, Cui2010, Yeh2011, Zheng2011, Ren2012, Kim2014, and Wu2014, respectively.

Note that two tasks are involved in this dataset: for a given query video, only the most similar duplicate video should be returned in the ST1 task, whereas all duplicate or near-duplicate video segments with the boundaries should be returned in the ST2 task. For performance evaluation, only the detection accuracy of the top-1 result should be evaluated in ST1 (in terms of $\mathbf{Q}$), whereas both the detection accuracy and localization precision should be evaluated in ST2 (in terms of $\mathbf{QS}$ and $\mathbf{QF}$). Thus, in ST2, the TASC should find all candidate CUs for a given query video and then repeat the copy detection process P_3 and the localization process P_4 until no copy segment pairs can be found again.

Table VI shows the experimental results. In ST1, many methods, including our TASC, achieved excellent detection performance with $\mathbf{Q}$ of 1.0. This means that they could correctly detect all copies on the ST1 database. In ST2, the TASC also showed significant advantage, with $\mathbf{QS}$ of 1.0 and $\mathbf{QF}$ of 0.977. These results are even better than the most recent results in Wu2014, which obtained $\mathbf{QS}$ of 0.95 and $\mathbf{QF}$ of 0.9. One possible reason is that all query videos in the MUSCLE-VCD-2007 dataset are not



(a)

(b)

Fig. 10. Comparison between the TASC approach and several state-of-the-art results on the TRECVID-CBCD-2011 dataset over 56 transformations. (a) NDCR (y-axis in log_2 coordinate). (b) Mean F1.

Table VI. Comparison between the Proposed TASC and the State-of-the-Art Methods on the MUSCLE-VCD-2007 Dataset

Metric		TASC	Anguera2009	Tan2009	Cui2010	Yeh2011	Zheng2011	Ren2012	Kim2014	Wu2014
ST1	**Q**	**1.00**	**1.00**	**1.00**	**1.00**	0.93	**1.00**	0.93	0.93	**1.00**
ST2	**QS**	**1.00**	0.88	0.9	0.86	0.86	0.9	0.93	0.86	0.95
	QF	**0.977**	NA	0.82	NA	NA	0.85	NA	NA	0.9

subject to any audio transformation, making them very easy to be recognized as copies or noncopies by the TASC. In addition, the TASC is very computationally efficient on this dataset. Its MP-Time is averagely 34.663 seconds for 749.3-second-long query videos in ST1 and 45.733 seconds for 896.9-second-long query videos in ST2. Overall speaking, the experimental results show that the TASC exhibits excellent detection performance, good robustness, and high efficiency on the MUSCLE-VCD-2007 dataset.

5.2.3. Results on the CC_WEB_VIDEO Dataset. The last experiment evaluated the performance of the TASC on the CC_WEB_VIDEO dataset. Given a query video, the TASC should retrieve all duplicate and near-duplicate videos. On this dataset, the performance should be evaluated in terms of MAP and the precision-recall curve. In this

Table VII. Comparison between the Proposed TASC and the State-of-the-Art Methods on the CC_WEB_VIDEO Dataset

Metric	TASC	Wu2007	Shang2010	Cai2012	Zhou2012	Song2013	Wu2014
MAP	**0.986**	0.952	0.953	0.918	0.956	0.958	0.922

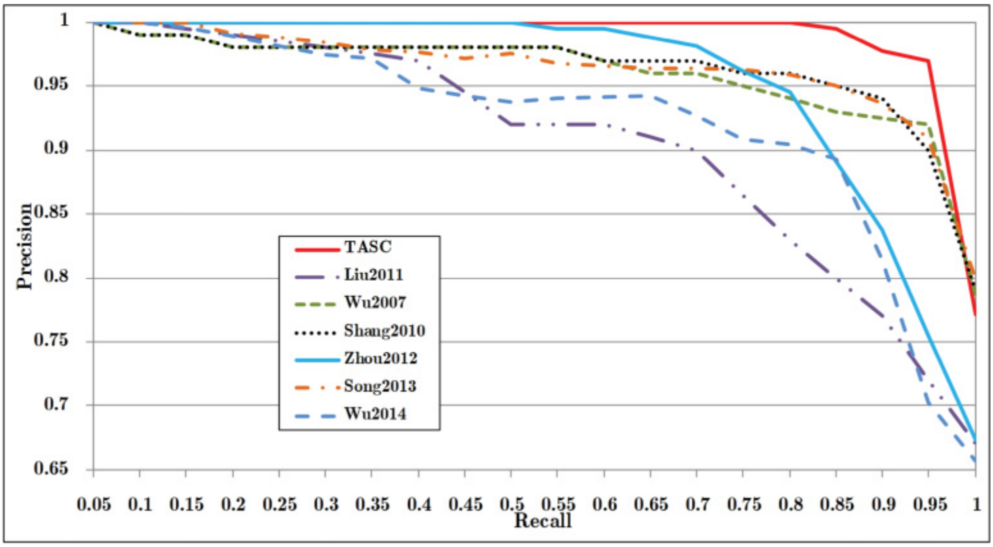


Fig. 11. The average precision-recall curves of different methods on the CC_WEB_VIDEO dataset.

case, the localization process P_4 should not be performed. For comparison, seven state-of-the-art results were also collected from the literature, including Wu et al. [2007], Shang et al. [2010], Liu et al. [2011], Cai et al. [2012], Zhou et al. [2012], Song et al. [2013], and Wu and Aizawa [2014]. Similarly, they are denoted by Wu2007, Shang2010, Liu2011, Cai2012, Zhou2012, Song2013, and Wu2014, respectively.

Note that Liu 2011 did not provide its MAP result, whereas Cai2012 did not provide its precision-recall data. Thus, they were excluded in the corresponding comparisons.

Table VII shows the comparison of the MAP results. Among all of these methods, the TASC obtained the best MAP (i.e., 0.986), with about 3% improvement over the best result Song2013 in the literature. Figure 11 depicts the precision-recall curves of different methods. We can see that the TASC can obtain the precision of 100% even when the recall reaches 80%. This is consistent with the Soft-Decision-Boundary mechanism, which enables the TASC to preferably find more results under the prerequisite of keeping as high detection accuracy as possible.

5.2.4. Summary. Extensive experiments on three benchmark datasets showed that the TASC can achieve excellent copy detection accuracy and localization precision with a very high processing efficiency. However, it should be admitted that video copy detection is a very challenging task due to the complex audiovisual transformations. Figure 12 shows some examples of failure cases on the TRECVID-CBCD-2011 dataset and the CC_WEB_VIDEO dataset. In Figure 12(a), the miss detection is mainly caused by the addition of black borders and heavy degradation on the query video; in Figure 12(b) and (c), the two miss detections are mainly due to the insertion of a heavily degraded clip into the query video as PiP; and in Figure 12(d), the false alarm is caused by the insertion of a short clip (less than the predefined CU length) into the query video. Clearly, we need to further improve the performance of the TASC when dealing with

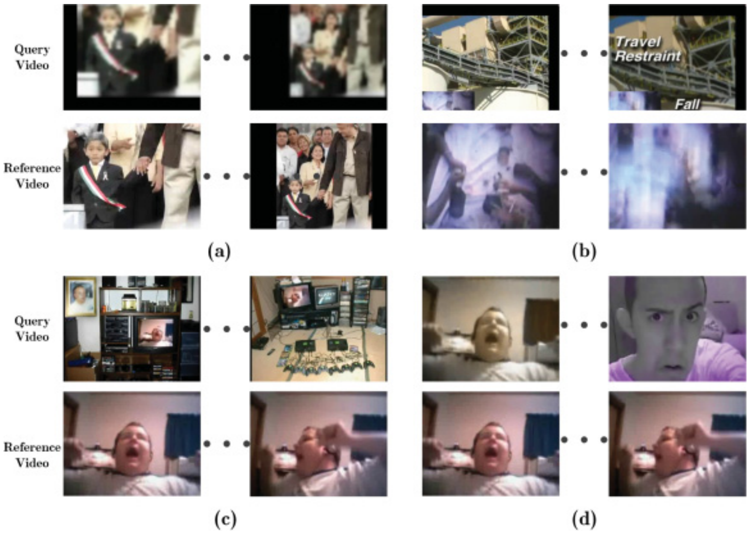


Fig. 12. Some examples of failure cases: two miss detections (a, b) on the TRECVID-CBCD-2011 dataset; one miss detection (c) and one false alarm (d) on the CC_WEB_VIDEO dataset. For simplicity, each video is only shown with two frames.

more complex transformations (e.g., PiP with heavy degradation, insertion of black borders).

6. CONCLUSION

This article proposes a TASC approach for multimodal video copy detection. Our main contributions are summarized as follows:

—We propose the TASC to organize multiple multimodal detectors in a cascading and transformation-aware way, which is expected to achieve high detection accuracy while minimizing the processing time. One efficient implementation is also developed by utilizing three commonly used multimodal features (i.e., AFP, DCT, and SIFT BoW) to construct four different chains.

—A Detection-on-Copy-Units mechanism is introduced in the TASC, which makes the decision of copy detection depending on the similarity between their most similar CUs rather than the video-level similarity. To do so, we also propose a CU search algorithm to find a pair of CUs from two videos and a CU-based localization algorithm to find the precise locations of their copy segments that are with the asserted CUs as the center.

—To address the problem that the copies and noncopies are possibly linearly inseparable in the feature space, we introduce a flexible soft decision boundary strategy in the TASC and then propose a bi-threshold learning algorithm for hard decision and utilize a soft-margin SVM classifier based on the SIFT keypoint matching for soft decision.

Due to its excellent processing performance, the TASC is capable of satisfying various requirements in practical copy detection applications. For example, the TASC-based system will be used by the Chinese government to discover pirated videos on the Internet, whereas Baidu (the Chinese search engine giant) also tries to use this technology in her video search engine to eliminate the semantically and visually identical duplicates from video search results. In the future work, we intend to further optimize the performance and scalability of the TASC on these practical applications.

REFERENCES

Xavier Anguera, Juan Manuel Barrios, Tomasz Adamek, and Nuria Oliver. 2011. Multimodal fusion for video copy detection. In *Proceedings of the 19th ACM International Conference on Multimedia (MM'11)*. 1221–1224.

Xavier Anguera, Pere Obrador, and Nuria Oliver. 2009. Multimodal video copy detection applied to social media. In *Proceedings of the 1st SIGMM Workshop on Social Media (WSM'09)*. 57–64.

Mohamed Ayari, Jonathan Delhumeau, Matthijs Douze, Hervé Jégoun, Danila Potapov, Jérôme Revaud, Cordelia Schmid, and Jiangbo Yuan. 2011. INRIA@TRECVID'2011: Copy detection and multimedia event detection. In *Proceedings of the Workshop on Emerging Multimedia Systems and Applications at the 2014 IEEE International Conference on Multimedia and Expo*.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. SURF: Speeded up robust features. In *Computer Vision—ECCV 2006.* Lecture Notes in Computer Science, Vol. 3951. Springer, 404–417.

Anna Bosch, Andrew Zisserman, and Xavier Muñoz. 2008. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 4, 712–727.

Yang Cai, Wei Tong, Linjun Yang, and Alexander G. Hauptmann. 2012. Constrained keypoint quantization: Towards better bag-of-words model for large-scale multimedia retrieval. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR'12)*. Article No. 16.

Vijay Chandrasekhar, Matt Sharifi, and David A. Ross. 2011. Survey and evaluation of audio fingerprinting schemes for mobile query-by-example applications. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. 801–806.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 3, Article No. 27.

Jianping Chen and TieJun Huang. 2008. A robust feature extraction algorithm for audio fingerprinting. In *Proceedings of the 9th Pacific Rim Conference on Multimedia*. 887–890.

Chih-Yi Chiu, Chu-Song Chen, and Lee-Feng Chien. 2008. A framework for handling spatiotemporal variations in video copy detection. *IEEE Transactions on Circuits and Systems for Video Technology* 18, 3, 412–417.

Chih-Yi Chiu, Hsin-Min Wang, and Chu-Song Chen. 2010. Fast min-hashing indexing and robust spatio-temporal matching for detecting video copies. *ACM Transactions on Multimedia Computing, Communications, and Applications* 6, 2, Article No. 10.

Cisco Visual Networking Index: Forecast and Methodology, 2012–2017, White Paper, May 29, 2013. http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html.

Ozgun Cirakman, Bilge Gunsel, Neslihan Serap Sengor, and Sezer Kutluk. 2012. Content-based copy detection by a subspace learning based video fingerprinting scheme. *Multimedia Tools and Applications* 71, 1381–1409. DOI:10.1007/s11042-012-1269-8

Baris Coskun, Bulent Sankur, and Nasir Memon. 2006. Spatio-temporal transform based video hashing. *IEEE Transactions on Multimedia* 8, 6, 1190–1208.

Costas Cotsaces, Nikos Nikolaidis, and Ioannis Pitas. 2009. Semantic video fingerprinting and retrieval using face information. *Image Communication* 24, 7, 598–613.

Peng Cui, Zhipeng Wu, Shuqiang Jiang, and Qingming Huang. 2010. Fast copy detection based on Slice Entropy Scattergraph. In *Proceedings of the 2010 IEEE International Conference on Multimedia and Expo (ICME'10)*. 1236–1241.

Matthijs Douze, Hervé Jégou, and Cordelia Schmid. 2010. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Transactions on Multimedia* 12, 4, 257–266.

Mani Malek Esmaeili, Mehrdad Fatourechi, and Rabab Kreidieh Ward. 2011. A robust and fast video copy detection system using content-based fingerprinting. *IEEE Transactions on Information Forensics and Security* 6, 1, 213–226.

Vishwa Gupta, Parisa Darvish Zadeh Varcheie, Langis Gagnon, and Gilles Boulianne. 2011. CRIM at TRECVID-2011: Content-based copy detection using nearest-neighbor mapping. In *Online Proceedings of TRECVID 2011*.

Vishwa Gupta, Parisa Darvish Zadeh Varcheie, Langis Gagnon, and Gilles Boulianne. 2012. Content-based video copy detection using nearest-neighbor mapping. In *Proceedings of the 11th International Conference on Information Science, Signal Processing, and Their Applications.* 918–923.

Jaap Haitsma and Ton Kalke. 2012. A highly robust audio fingerprinting system. In *Proceedings of the International Symposium on Music Information Retrieval*. 107–115.

Tiejun Huang, Yonghong Tian, Wen Gao, and Jan Lu. 2010. MediAprinting: Identifying multimedia content for digital rights management. *Computer* 43, 12, 28–35.

Menglin Jiang, Yonghong Tian, and TieJun Huang. 2012. Video copy detection using a soft cascade of multimodal features. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'12)*. 374–379.

Alexis Joly, Olivier Buisson, and Carl Frélicot. 2007. Content-based copy retrieval using distortion-based probabilistic similarity search. *IEEE Transactions on Multimedia* 9, 2, 293–306.

Yan Ke and Rahul Sukthankar. 2004. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 506–513.

Semin Kim, Jae Young Choi, Seungwan Han, and Yong Man Ro. 2014a. Adaptive weighted fusion with new spatial and temporal fingerprints for improved video copy detection. *Image Communication* 29, 7, 788–806.

Semin Kim, Seung Ho Lee, and Yong Man Ro. 2014b. Rotation and flipping robust region binary patterns for video copy detection. *Journal of Visual Communication and Image Representation* 25, 373–383.

Wessel Kraaij and George Awad. 2011. TRECVID 2011 content-based copy detection: Task overview. In *Online Proceedings of TRECVid 2010*. Available at http://www-nlpir.nist.gov/projects/tvpubs/tv11.slides/tv11.ccd.slide s.pdf.

Julien Law-To, Alexis Joly, and Nozha Boujemaa. 2007. MUSCLE-VCD-2007: A live benchmark for video copy detection. Retrieved January 12, 2015, from http://www.rocq.inria.fr/imedia/civr-bench/.

Yanqiang Lei, Weiqi Luo, Yuangen Wang, and Jiwu Huang. 2012. Video sequence matching based on the invariance of color correlation. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 9, 1332–1343.

Hong Liu, Hong Lu, and Xiangyang Xue. 2013a. A segmentation and graph-based video sequence matching method for video copy detection. *IEEE Transactions on Knowledge and Data Engineering* 25, 8, 1706–1718.

Jiajun Liu, Zi Huang, Hongyun Cai, Heng Tao Shen, Chong Wah Ngo, and Wei Wang. 2013b. Near-duplicate video retrieval: Current research and future trends. *ACM Computing Surveys* 45, 4, Article No. 44.

Jiajun Liu, Zi Huang, Heng Tao Shen, and Bin Cui. 2011. Correlation-based retrieval for heavily changed near-duplicate videos. *ACM Transactions on Information Systems* 29, 4, Article No. 21.

Yang Liu, Wan-Lei Zhao, Chong-Wah Ngo, Chang-Sheng Xu, and Han-Qing Lu. 2010. Coherent bag-of audio words model for efficient large-scale video copy detection. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'10)*. 89–96.

Rui Ma, Jian Chen, and Zhong Su. 2012. MI-SIFT: Mirror and inversion invariant generalization for SIFT descriptor. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'12)*. 228–236.

Hyun Seok Min, Jae Young Choi, Wesley De Neve, and Yong Man Ro. 2012. Near-duplicate video clip detection using model-free semantic concept detection and adaptive semantic distance measurement. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 8, 1174–1187.

Luntian Mou, Tiejun Huang, Yonghong Tian, Menglin Jiang, and Wen Gao. 2013. Content-based copy detection through multimodal feature representation and temporal pyramid matching. *ACM Transactions on Multimedia Computing, Communications, and Applications* 10, 1, Article No. 5.

Paul Over, Martial Michel, George Awad, Jon Fiscus, Brian Antonishek, Alan F. Smeaton, Wessel Kraaij, and Georges Quénot. 2010. TRECVID 2010: An overview of the goals, tasks, data, evaluation mechanisms, and metrics. In *Online Proceedings of TRECVid 2010*.

Mengren Qian, Luntian Mou, Jia Li, and Yonghong Tian. 2014. Video picture-in-picture detection using spatio-temporal slicing. In *Proceedings of the ICME 2014 Workshop on Emerging Multimedia Systems and Applications*.

Jennifer Ren, Fangzhe Chang, Thomas Wood, and John R. Zhang. 2012. Efficient video copy detection via aligning video signature time series. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval (ICMR'12)*. ACM, New York, NY, Article No. 14.

Cédric De Roover, Christophe De Vleeschouwer, Frédéric Lefèbvre, and Benoiît M. Macq. 2005. Robust video hashing based on radial projections of key frames. *IEEE Transactions on Signal Processing* 53, 10, 4020–4037.

Ahmet Saracoğlu, Ersin Esen, Tuğrul K. Ateş, Banu Oskay, Ünal Zubari, Ezgi C. Ozan, Egemen Özalp, A. Aydin Alatan, and Tolga Çiloğlu. 2009. Content based copy detection with coarse audio-visual fingerprints.

In *Proceedings of the 7th International Workshop on Content-Based Multimedia Indexing (CBMI'09)*. 213–218.

Lifeng Shang, Linjun Yang, Fei Wang, Kwok-Ping Chan, and Xian-Sheng Hua. 2010. Real-time large scale near-duplicate Web video retrieval. In *Proceedings of the International Conference on Multimedia (MM'10)*. ACM New York, NY, 531–540.

Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Richang Hong. 2011. In *Proceedings of the International Conference on Multimedia (MM'11)*. 423–432.

Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Richang Hong. 2013. Multiple feature hashing for large scale near-duplicate video retrieval. *IEEE Transactions on Multimedia* 15, 8, 1997–2008.

Ashwin Swaminathan, Yinian Mao, and Min Wu. 2006. Robust and secure image hashing. *IEEE Transactions on Information Forensics and Security* 1, 2, 215–230.

Hung-Khoon Tan, Chong-Wah Ng, Richard Hong, and Tat-Seng Chua. 2009. Scalable detection of partial near-duplicate videos by visual-temporal consistency. In *Proceedings of the 17th ACM International Conference on Multimedia (MM'09)*. ACM, New York, NY, 145–154.

Sheng Tang, Yan-Tao Zheng, Yu Wang, and Tat-Seng Chua. 2012. Sparse ensemble learning for concept detection. *IEEE Transactions on Multimedia* 14, 1, 43–54.

Yonghong Tian, TieJun Huang, and Wen Gao. 2012. Multimodal video copy detection using multi-detectors fusion. *IEEE ComSoc MMTC E-Letter* 7, 7, 6–9.

Yonghong Tian, TieJun Huang, Menglin Jiang, and Wen Gao. 2013. Video copy-detection and localization with a scalable cascading framework. *IEEE Multimedia* 20, 3, 72–86.

Yonghong Tian, Menglin Jiang, Luntian Mou, Xiaoyu Fang, and Tie-Jun Huang. 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.

Lei Wang, Dawei Song, and Eyad Elyan. 2012. Improving bag-of-visual-words model with spatial-temporal correlation for video retrieval. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. 1303–1312.

Shikui Wei, Yao Zhao, Ce Zhu, Changsheng Xu, and Zhenfeng Zhu. 2011. Frame fusion for video copy detection. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 1, 15–28.

Chenxia Wu, Jianke Zhu, and Jiemi Zhang. 2012. A content-based video copy detection method with randomly projected binary features. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 21–26.

Xiao Wu, Alexander G. Hauptmann, and Chong-Wah Ngo. 2007. Practical elimination of near-duplicates from Web video search. In *Proceedings of the ACM Conference on Multimedia*. 218–227.

Xiao Wu, Chong-Wah Ngo, Alexander G. Hauptmann, and Hung-Khoon Tan. 2009b. Real-time near-duplicate elimination for Web video search with content and context. *IEEE Transactions on Multimedia* 11, 2, 196–207.

Zhipeng Wu and Kiyoharu Aizawa. 2014. Self-similarity-based partial near-duplicate video retrieval and alignment. *International Journal of Multimedia Information Retrieval* 3, 1–14.

Zhipeng Wu, Shuqiang Jiang, and Qingming Huang. 2009a. Near-duplicate video matching with transformation recognition. In *Proceedings of the 17th ACM International Conference on Multimedia*. 549–552.

Mei-Chen Yeh and Kwang-Ting Cheng. 2011. Fast visual retrieval using accelerated sequence matching. *IEEE Transactions on Multimedia* 13, 2, 320–329.

Wan-Lei Zhao and Chong-Wah Ngo. 2013. Flip-invariant SIFT for copy and object detection. *IEEE Transactions on Image Processing* 22, 3, 980–991.

Ligang Zheng, Guoping Qiu, Jiwu Huang, and Hao Fu. 2011. Salient covariance for near-duplicate image and video detection. In *Proceedings of the 2011 18th IEEE International Conference on Image Processing (ICIP'11)*. 2537–2540.

Xiangmin Zhou, Lei Chen, and Xiaofang Zhou. 2012. Structure tensor series-based large scale near-duplicate video retrieval. *IEEE Transactions on Multimedia* 14, 4, 1220–1232.

Received June 2014; revised September 2014; accepted October 2014