VIDEO COPY DETECTION USING A SOFT CASCADE OF MULTIMODAL FEATURES

Menglin Jiang, Yonghong Tian[+], Tiejun Huang

National Engineering Lab for Video Technology, School of EE & CS, Peking University
{mlJiang, yhTian, tjHuang}@pku.edu.cn

***Abstract*—**In the video copy detection task, it is widely recognized that none of any single feature can work well for all transformations. Thus more and more approaches adopt a set of complementary features to cope with complex audio-visual transformations. However, most of them utilize individual features separately and the final result is obtained by fusing results of several basic detectors. Often, this will lead to low detection efficiency. Moreover, there are some thresholds or parameters to be elaborately tuned. To address these problems, we propose a soft cascade approach to integrate multiple features for efficient copy detection. In our approach, basic detectors are organized in a cascaded framework, which processes a query video in sequence until one detector asserts it as a copy. To fully exert the complementarity of these detectors, a learning algorithm is proposed to estimate the optimal decision thresholds in the cascade architecture. Excellent performance on the benchmark dataset of TRECVid 2011 CBCD task demonstrates the effectiveness and efficiency of our approach.

***Keywords*—**Video copy detection, soft cascade architecture, multimodal features

I. INTRODUCTION

The explosive growth of multimedia content on the Internet is revolutionizing the way of content distribution and presenting new challenges to content security and copyright management. As an alternative solution to watermarking, content-based copy detection (CBCD) has drawn more and more attention in recent years. According to TRECVid, CBCD addresses the issue that automatically analyzes a query video's content to determine whether it contains a copy from a given database of reference videos and if so from where the copy comes [1]. Therefore, besides digital right management, it also shows great value in many other video applications such as advertisement tracking, video content filtering, and so on.

However, copy detection is pretty challenging primarily due to the fact that video copies often suffer from severe quality decrease and even change in content, which makes it difficult to extract largely invariant features from a copy and its original reference video. After years of practice, it has become a common view that there is no universal feature that keeps robust to all the audio-visual transformations. Actually, the TRECVid CBCD contest has required all the participants to use both audio and visual features in their copy detection approaches [1]. Most of these approaches first compute several detection results through individual features and then fuse these results into final result. For example, detection results through audio and visual features are fused by picking up the video match with the highest similarity score [2, 3]. Queries asserted as copies by any two of four audio-visual features are accepted as copies, and queries asserted as copies by only one feature will be delivered to further verification [4]. Although such approaches could achieve good detection effectiveness, they also have two drawbacks. One is that the processing time will be at least the sum of time required by each detector. The other is that some thresholds or parameters involved in the late fusion will have to be manually tuned.

To overcome these drawbacks, we introduce a soft cascade approach to combine multimodal features, which is shown in Fig. 1. Three basic detectors based on complementary audio-visual features are organized in a cascaded structure, and they process each query video successively until one detector asserts it as a copy. Since most copies can be correctly detected through the first two efficient detectors, a lot of processing time is saved. Besides, in order to automatically tune the decision thresholds θ_i ($i = 1,2,3$), which are used to determine whether a query is a copy or not, a learning algorithm is proposed. By iteratively adjusting the weights for all the training query videos, posterior detectors are forced to focus on those queries misjudged by anterior detectors, so that these detectors could complement each other as much as possible.

The remainder of this paper is organized as follows. Sec. II describes the cascade architecture and the utilized basic detectors. Sec. III introduces the algorithm for learning the thresholds in the cascade. Sec. IV presents the experimental results and Sec. V concludes this paper.

II. CASCADE ARCHITECTURE FOR CBCD

To improve the efficiency of copy detection approaches which use several features, we introduce the cascade architecture [5] to the copy detection issue. Intuitively, a cascade is constructed by placing a series of detectors in a cascaded order. Efficient but ordinary detectors should stand in the front, while effective but complex detectors should locate in the rear. Generally speaking, a *N*-Stage cascade could be denoted as $D_N = \langle d_1, d_2, \cdots, d_N \rangle$, where d_i ($i = 1,2,\cdots,N$) represents the *i*-th detector. During inquiry process, a query video q is processed by each detector

[+] Dr. Yonghong Tian is the corresponding author.

978-0-7695-4711-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ICME.2012.189

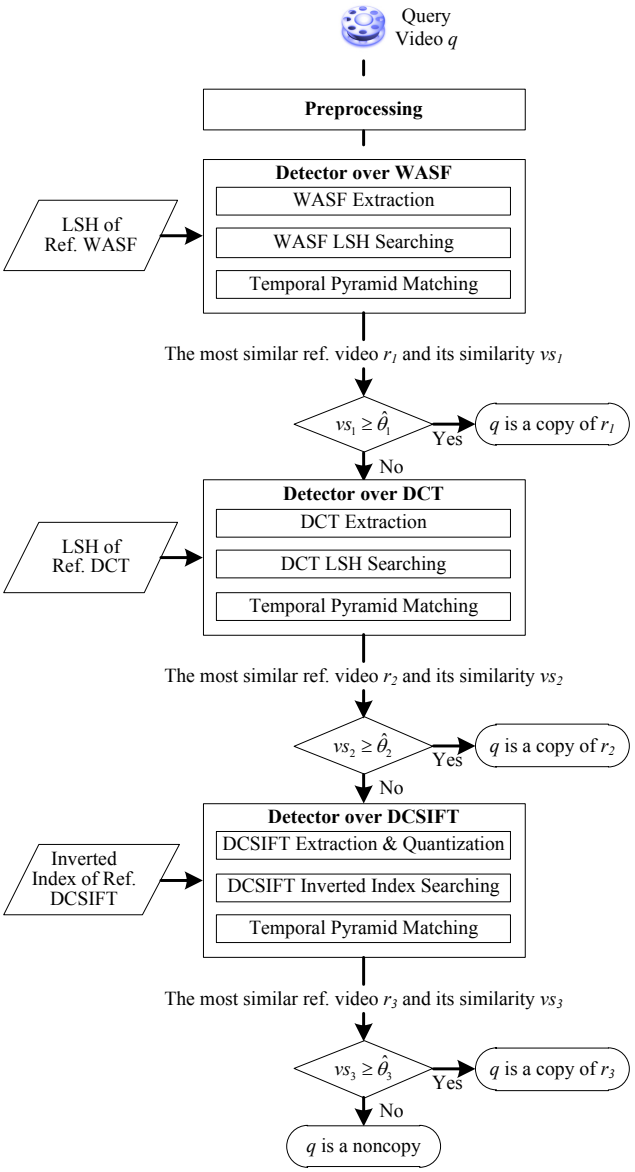

Figure 1. Overview of the proposed approach

successively until one detector asserts it as a copy or all the detectors determine it as a non-copy. To be specific, q is first processed by d_1. A positive detection result, i.e. the returned reference video r_1 has a similarity vs_1 greater than or equal to a predefined threshold θ_1, leads to immediate acceptance of q as a copy. Otherwise, the evaluation of d_2 on q will be triggered. Such process goes on until all the N detectors are executed. Only if q is asserted as a noncopy by all the detectors, will it be accepted as a noncopy. In this architecture, most copies can be detected through the first few detectors, thus saving a major part of processing time. Particularly, if the decision thresholds $\Theta = \{\theta_1, \theta_2, \cdots, \theta_N\}$ are set manually, they could be called "hard thresholds", and the cascade architecture is therefore addressed as "hard cascade".

This section will then describe all the basic detectors with the corresponding preprocessing operations. Given a query video q, the preprocessing module is first utilized to extract audio clips and visual key frames from q. Then in each basic detector, one single feature is used to retrieve these audio clips or visual key frames from a database of reference audio clips or visual key frames. Finally the frame-level retrieval results are passed to temporal pyramid matching (TPM) module, where they would be aggregated into video-level detection result.

A. *Preprocessing*

During preprocessing, first audio frames are obtained by dividing the audio track into segments of 90ms with a 60ms overlap between consecutive frames, and 6-second-long audio clips are constructed by every 198 audio frames with a 5.4-second overlap between adjacent clips. Then visual key frames are uniformly sampled at a rate of 3 frames per second. Finally, additional steps are dedicated to handle picture-in-picture (PiP) and flip transformations. Specifically, Hough transform that detects two pairs of parallel lines is employed to detect and localize the inserted foreground videos. For those queries with PiP, the foreground and original key frames will be processed respectively. Also queries asserted as noncopies will be flipped and matched again to deal with potential flip transformation.

B. *Frame-level retrieval*

Three independent basic detectors are constructed over one local visual feature, one global visual feature and one audio feature respectively. Each feature is briefly described as follows.

1) Local visual feature: A dense color version of SIFT [6] (DCSIFT) is adopted mainly to cope with content-altering visual transformations such as camcording, PiP and postproduction. The differences between DCSFIT and traditional SIFT reside in "dense" and "color". "Dense" means the module of interest point detection is discarded and multi-scale dense sampling is used instead. "Color" implies the descriptor is calculated not from a grayscale image but from a color image. Concretely speaking, sub-descriptors are computed from each LAB component and then concatenated to form the final descriptor.

Furthermore, the bag-of-words (BoW) framework [7] is employed and boosted in our approach. Since vector quantization might degrade the descriptors' discriminability, information of position, scale and orientation is also taken into account so that only features mapped to the same descriptor cluster and with similar position, scale and orientation will be regarded as matches. In particular, *K*-means algorithm is conducted on a random subset (10M) of reference DCSIFT descriptors to obtain 800 clusters, thus each descriptor could be quantized into a cluster ID. Also position, scale and orientation are quantized into 4, 2 and 16

bins respectively. Consequently, the optimized visual vocabulary contains $800\times4\times2\times16=102,400$ visual words. To accelerate frame retrieval, all the reference DCSIFT features are quantized as visual words and stored in an inverted index during offline process. During online process, DCSIFT BoW is obtained from each query key frame through the same feature extraction and quantization method. By searching the inverted index, reference key frames that have similar appearance and spatial layout can be found efficiently.

2) Global visual feature: A global feature denoted as "DCT" [4] is employed. Based on the relationship between low-frequency DCT coefficients of adjacent image blocks, DCT feature can effectively resist content-preserving visual transformations such as re-encoding, change of gamma and decrease in quality. Moreover, DCT feature is computationally efficient and compact (256 bits per image). Hamming distance is used as the distance metric. To speed up feature matching, all the reference DCT features are indexed by locality sensitive hashing (LSH) [8].

3) Audio feature: Weighted audio spectrum flatness (WASF) [9] is used to address audio transformations such as MP3 compression and multiband companding. WASF extends the MPEG-7 descriptor - audio spectrum flatness (ASF) [10] by introducing human auditory system (HAS) functions to weight audio spectrum, making the resulted feature more consistent with the outer ear and middle ear models of HAS. In brief, a 72-D WASF feature is extracted from each 6-second-long audio clip. Euclidean distance is adopted to measure the dissimilarity between two WASF features, and all the reference WASF features are stored in LSH for efficient feature matching.

4) Results of frame-level retrieval: Given a query video q, a detector picks up the top $K_1=20$ similar reference key frames (audio clips) for each query key frame (audio clip), obtaining a collection FM which contains a series of frame-level matches fm:

$$fm=\langle q,t(q),r,t(r),fs\rangle \tag{1}$$

which means the key frame (audio clip) of reference video r at timestamp $t(r)$ is a match to the key frame (audio clip) of query video q at timestamp $t(q)$ with a similarity fs.

C. Temporal Pyramid Matching

Given the frame matches FM of a query video, temporal pyramid matching (TPM) [4] is employed to pick up the most similar reference video clip. Note that the sequential pyramid matching (SPM) in [4] is renamed TPM here to avoid confusion with spatial pyramid matching [11].

Briefly speaking, a copy is detected through the following four steps. First, 2-D Hough transform, with one dimension representing reference video ID and the other representing δt, is conducted on FM to vote in $K_2=10$ hypotheses $\langle r,\delta t\rangle$, where $\delta t=t(q)-t(r)$ specifies the temporal offset between a query frame and a reference frame. Second, for each hypothesis, the extent of copy in query video and reference video, denoted as $\left[t^B(q),t^E(q)\right]$ and $\left[t^B(r),t^E(r)\right]$, are identified by picking up the first and the last matches fm in FM that accord with this hypothesis. Third, $\left[t^B(q),t^E(q)\right]$ and $\left[t^B(r),t^E(r)\right]$ are partitioned into increasingly finer segments and video similarities are computed at multiple granularities. In each resolution, only frames within aligned segments can be matched across two sequences. The video similarity vs is calculated by accumulating the weighted similarities from multiple resolutions. And a candidate video-level match can be expressed as follows:

$$vm(q)=\langle q,t^B(q),t^E(q),r,t^B(r),t^E(r),vs\rangle \tag{2}$$

which means the sequence $\left[t^B(q),t^E(q)\right]$ of query q is a potential copy derived from the sequence $\left[t^B(r),t^E(r)\right]$ of reference r with a similarity score vs. Finally, the video match with the highest similarity vs among all the K_2 candidate matches is retained. Only if vs is greater than or equal to a predefined threshold θ, will this detector asserts q as a copy.

III. LEARNING SOFT THRESHOLDS

It is obvious that hard cascade has several drawbacks: artificial adjustment of thresholds $\Theta=\{\theta_1,\theta_2,\cdots,\theta_N\}$ can hardly reach optimal performance; it lacks in generalization ability and is very burdensome. To solve these issues, a learning algorithm is designed to automatically select the optimal thresholds $\hat{\Theta}=\{\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_N\}$, which can be named "soft thresholds". Accordingly, the cascade involving soft thresholds can be described as "soft cascade" and denoted as $\hat{D}_N=\langle\hat{d}_1,\hat{d}_2,\cdots,\hat{d}_N\rangle$. For example, Fig. 1 is expressed as $\hat{D}_3=\langle\hat{d}_{WASF},\hat{d}_{DCT},\hat{d}_{DCSIFT}\rangle$. Next we'll first define the error rate of a copy detection approach, and then present the threshold learning algorithm which aims at minimizing the error rate of the soft cascade.

A. Optimization problem

First of all, following the normalized detection cost rate (NDCR) measure used in TRECVid CBCD task [12], we define the cost of a single query video. For a given query q, supposing detector d with threshold θ returns a video match $vm(q)$ like (2), the cost of q w.r.t. θ, denoted as $c(q,\theta)$, is calculated according to the following rules: *1)* When q is actually a copy and also asserted as a copy by d, i.e. $vs\ge\theta$, *a)* if $vm(q)$ indicates the correct reference video clip, i.e. the reference video ID r is right and the copy extent

$\left[t^B(r), t^E(r)\right]$ overlaps with the ground truth, then it is a true positive (TP) and $\mathrm{c}(q,\theta)$ is set to zero; *b)* if $vm(q)$ identifies a wrong reference clip, then it causes a false positive (FP) and a false negative (FN) simultaneously, and $\mathrm{c}(q,\theta)$ is set to sum of c_{FP} and c_{FN}, which represent the penalty for a FP and a FN respectively. *2)* If q is a copy but asserted as a noncopy by d, i.e. $vs < \theta$, then $vm(q)$ is a FN and $\mathrm{c}(q,\theta)$ is set to c_{FN}. *3)* If q is in fact a noncopy but asserted as a copy, then $vm(q)$ is a FP and $\mathrm{c}(q,\theta)$ is set to c_{FP}. *4)* If q is a noncopy and also asserted as a noncopy, then $vm(q)$ is a true negative (TN) and $\mathrm{c}(q,\theta)$ is set to zero. The above rules could be summarized as follows:

$$\mathrm{c}(q,\theta) = \begin{cases} 0, & \begin{array}{c} if\left(C(q) \wedge vs \geq \theta \wedge T\left(vm(q)\right)\right) \\ \vee\left(\neg C(q) \wedge vs < \theta\right) \end{array} \\ c_{FP} + c_{FN}, & if\left(C(q) \wedge vs \geq \theta \wedge \neg T\left(vm(q)\right)\right) \\ c_{FN}, & if\left(C(q) \wedge vs < \theta\right) \\ c_{FP}, & if\left(\neg C(q) \wedge vs \geq \theta\right) \end{cases} \quad (3)$$

where $C(q)$ means q is indeed a copy, $T\left(vm(q)\right)$ denotes that $vm(q)$ recognizes the correct reference video clip. $\langle c_{FP}, c_{FN} \rangle$ is determined based on the application profile. In this paper it is set to $\langle 2, 0.2 \rangle$, owing to our belief that a FP is much worse than a FN in applications such as copyright protection.

The error rate of detector d on training set $Q = \{q_1, q_2, \cdots, q_M\}$ w.r.t. threshold θ is defined as the weighted sum of the cost of each query video:

$$\epsilon(Q,\theta) = \sum_{j=1}^{M} w_j \bullet c\left(q_j, \theta\right) \quad (4)$$

where $W = \{w_1, w_2, \cdots, w_M\}$ is the set of weights for each query video. The goal of threshold optimization is to minimize the error rate ϵ of a soft cascade $\hat{D}_N$.

B. Threshold learning algorithm

The threshold learning is grounded on two core ideas. One is that the optimal threshold should bring about a good tradeoff between FPs and FNs, and lead to the minimum error rate $\hat{\epsilon}$. For this purpose, $\epsilon(Q,\theta)$ should be calculated at a range of thresholds, sweeping from the minimum video similarity returned by d to the maximum similarity, so that the similarity score associated with the minimum error rate $\hat{\epsilon}$ is chosen as the optimal threshold $\hat{\theta}$. The other insight is that posterior detectors should focus on the queries which are incorrectly detected by anterior detectors, so that the overall system reaches its peak performance. To this end, weights of those misjudged queries should be augmented for posterior detectors. Such policy is inspired by the classifier learning approach [5] in the domain of object recognition. The learning procedure is summarized in Fig. 2.

Input: A hard cascade $D_N = \langle d_1, d_2, \cdots, d_N \rangle$ and a training set $Q = \{q_1, q_2, \cdots, q_M\}$

Output: Optimal thresholds $\hat{\Theta} = \{\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_N\}$

Solution:

1. Initialize weights $w_{1,j} = \frac{1}{M}$ for $j = 1, 2, \cdots, M$.

2. For $i = 1, 2, \cdots, N$

2.1. Normalize the weights:

$sum = \sum_{j=1}^{M} w_{i,j}$, $w_{i,j} = \frac{w_{i,j}}{sum}$ for $j = 1, 2, \cdots, M$

so that W_i is a probability distribution.

2.2. Evaluate d_i on Q, record the detection results:

$VM_i = \{vm_i(q_j) \mid j = 1, 2, \cdots, M\}$

And collect the video similarities:

$VS_i = \{vs \mid \langle q, t^B(q), t^E(q), r, t^B(r), t^E(r), vs \rangle \in VM_i\}$

2.3. Find the optimal threshold for d_i:

$\hat{\theta}_i = \arg\min_{\theta_i \in VS_i} \epsilon_i(Q, \theta_i)$

And record the minimum error rate:

$\hat{\epsilon}_i = \epsilon_i\left(Q, \hat{\theta}_i\right)$

2.4. Update the weights for d_{i+1}:

$$w_{i+1,j} = \begin{cases} w_{i,j} \bullet \frac{\hat{\epsilon}_i}{1-\hat{\epsilon}_i}, & if\ c_i\left(q_j, \hat{\theta}_i\right) = 0 \\ w_{i,j}, & otherwise \end{cases}$$

3. Return $\hat{\Theta} = \{\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_N\}$

Figure 2. Algorithm for learning soft thresholds

IV. EXPERIMENTS

Experiments are conducted over the TRECVid 2011 CBCD task [1, 12 & 13]. In this section, we'll first describe the CBCD task and then present the experimental results.

A. Data set & evaluation metrics

CBCD adopts a 425-hour-long reference database composed of poor-quality web videos. According to [13], 201 raw queries are first derived from the reference database and another non-reference database, among which a third is pure reference clip, a third is reference clip embedded in non-reference clip, and a third is non-reference clip. Then these queries are attacked by $8 \times 7 = 56$ transformations (c.f. TABLE I), creating 11,256 final queries, which are averagely 73 seconds long.

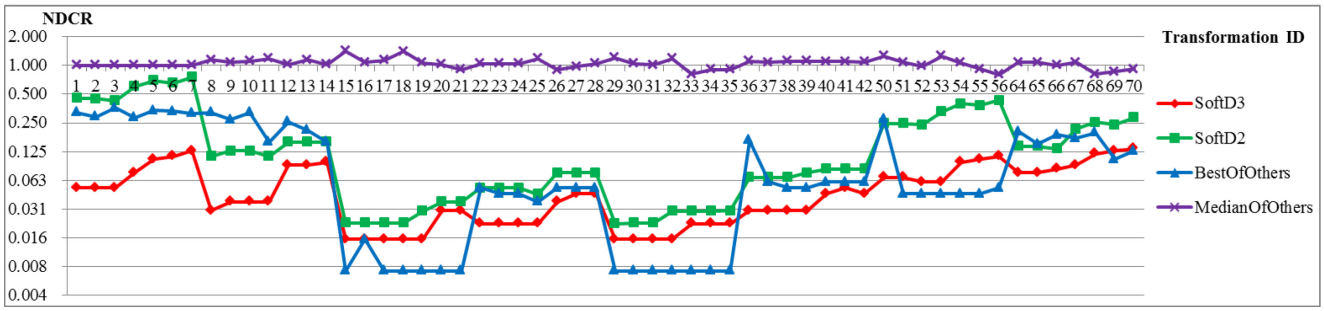

Figure 3. Actual NDCR for BALANCED profile

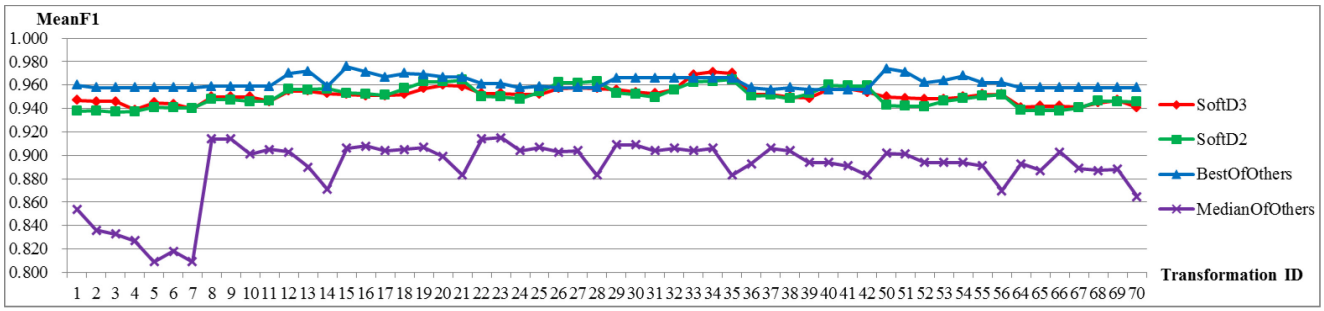

Figure 4. Actual MeanF1 for BALANCED profile

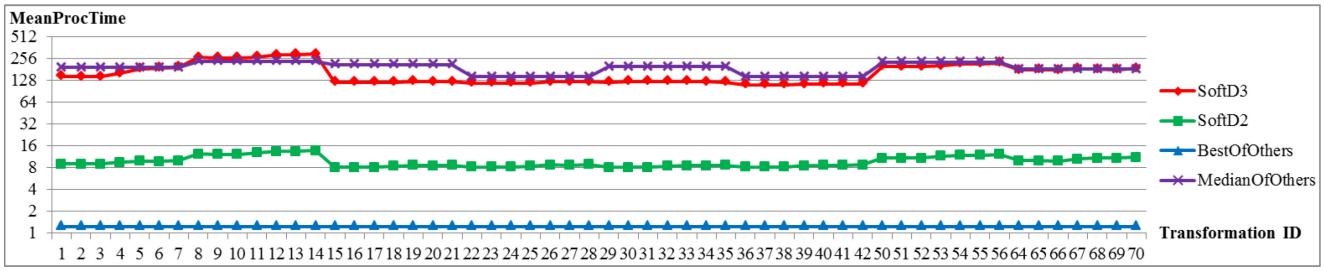

Figure 5. MeanProcTime for BALANCED profile

The evaluation metrics consist of three measures and are calculated separately for each transformation [12]. The primary measure is normalized detection cost rate (*NDCR*), which evaluates a system's detection effectiveness. It is similar to the error rate defined by (3) and (4), except that the weights for different queries vary in (4) but are the same in NDCR. The second diagnostic measure is *MeanF1*, which is the average F1 for all the TPs. F1 evaluates a system's accuracy in copy localization, that is to say, how much the asserted extent $\left[t^B(r), t^E(r)\right]$ overlaps with that in the ground truth. The last measure is *MeanProcTime*, i.e. the average time for a system to process a query. Since all the participants test their approaches under different environments, MeanProcTime is more useful to comparison between one participant's different approaches.

B. *Performance of the proposed approach*

Seven approaches are evaluated, including three detectors discussed in Sec. II and another detector over traditional SIFT feature, a hard cascade $D_3 = \langle d_{WASF}, d_{DCT}, d_{DCSIFT} \rangle$ denoted as "HardD3", two soft cascades of $\hat{D}_3 = \langle \hat{d}_{WASF}, \hat{d}_{DCT}, \hat{d}_{DCSIFT} \rangle$ and $\hat{D}_2 = \langle \hat{d}_{WASF}, \hat{d}_{DCT} \rangle$ denoted as "SoftD3" and "SoftD2". Our testing environment is: OS-Windows Server 2008; CPU-Intel X7550 2.00 GHz, 32 Core; Memory-32 GB. The best and median performances are picked up to reflect 40 approaches from 22 other participants, denoted as "BestOfOthers" and "MedianOfOthers" respectively. Note that BestOfOthers along with MedianOfOthers does not refer to one single approach, but is computed over each transformation separately. The average performances over 56 transformations are summarized in TABLE II. Detailed performances of SoftD3, SoftD2, BestOfOthers and MedianOfOthers are shown in Fig. 3, Fig. 4 and Fig. 5.

1) Performance of basic detectors: Experiments show that the selected three detectors are indeed effective and complementary. The complementarity lies in two aspects. One is that these detectors strike different balance between effectiveness and efficiency. DCSIFT obtains lowest NDCR at the cost of longest MeanProcTime (note that DCSIFT surpasses SIFT w.r.t. NDCR), while DCT and WASF gain higher NDCR within much shorter time. The other aspect is

TABLE I. TRANSFORMATIONS IN TRECVID 2011 CBCD

Video	V1-Simulated camcording
	V2-Picture-in-Picture (PiP)
	V3-Insertion of pattern
	V4-Strong re-encoding
	V5-Change of gamma
	V6-Decrease in quality
	V8-Post production
	V10-Rondom combination of 3 trans. out of V1-V8
Audio	A1-Do nothing
	A2-Mp3 compression
	A3-Mp3 compression and multiband companding
	A4-Bandwidth limit and single-band companding
	A5-Mix with speech
	A6-Mix with speech and multiband compress
	A7- Mix with speech, bandpass filter, and compress
Mixed	M[(i-1)×7+j] is composed of Vi and Aj

TABLE II. PERFORMANCE SUMMARY

Approach		Avg. NDCR	Avg. MeanF1	Avg. MeanP.T.
Basic Detector	DCSIFT	0.117	0.955	249.636
	SIFT	0.210	0.953	138.550
	DCT	0.344	0.953	6.381
	WASF	0.206	0.949	5.486
Cascade	HardD3	0.060	0.951	172.291
	SoftD3	**0.054**	0.951	163.184
	SoftD2	0.178	0.950	9.752
TRECVid Evaluation	BestOfOthers	0.117	**0.962**	**1.250** [a]
	MedianOfOthers	1.050	0.889	191.535

a. One of other participants' approaches could process a query video within 1.30 seconds, but it suffers from high NDCR ($Avg.NDCR = 6.408$) and low MeanF1 ($Avg.MeanF1 = 0.001$).

that these detectors are robust to different transformations. The complementarity between visual features and WASF is obvious. As for two visual features, detailed results prove that DCT is more robust than DCSIFT to severe blur and noise, while the overall robustness of DCSIFT is better than that of DCT.

2) Performance of soft cascade: First, performance of HardD3 shows that by integrating complementary detectors, cascade architecture could dramatically improve detection effectiveness as well as efficiency (note that its average MeanProcTime is shorter than that of DCSIFT). Second, performance of SoftD3 indicates that the threshold learning algorithm could further enhance effectiveness and efficiency. Among all the approaches, SoftD3 achieves compelling performance: it wins 35 best NDCR, 3 best MeanF1 (others are close to the best ones), and its MeanProcTime are better than the median ones on most transformations. Finally, result of SoftD2 demonstrates the flexibility of soft cascade architecture. By using only two efficient detectors, SoftD2 achieves competitive NDCR, excellent MeanF1 within pretty short MeanProcTime.

V. CONCLUSION

We have proposed a soft cascade video copy detection approach, which integrates multimodal detectors and learns optimal decision thresholds automatically. Extensive experiments over benchmark data set prove that our approach is both effective and efficient. Further endeavors will be devoted to introducing a transformation recognition model and learning $\hat{\Theta}$ for each transformation.

ACKNOWLEDGMENT

This work is partially supported by grants from the Chinese National Natural Science Foundation under contract No. 60973055, a grant from National Key Technologies R&D Program of China under contract No. 2009BAH51B01, and the CADAL project.

REFERENCES

[1] NIST, "Guidelines for TRECVid 2011," http://www-nlpir.nist.gov/projects/tv2011/tv2011.html, Available in Nov. 2011.
[2] A. Saracoglu, E. Esen, T. K. Ates, B. O. Acar, U. Zubari, E. C. Ozan, E. Ozalp, A. A. Alatan, and T. Ciloglu, "Content Based Copy Detection with Coarse Audio-Visual Fingerprints," *CBMI'09*, Chania, pp. 213-218, Jun. 3-5, 2009.
[3] Y. Liu, W. Zhao, C. Ngo, C. Xu, and H. Lu, "Coherent Bag-of Audio Words Model For Efficient Large-Scale Video Copy Detection," *ACM CIVR'10*, Xi'an, China, pp. 89-96, Jul. 5-7, 2010.
[4] Y. H. Tian, M. L. Jiang, L. T. Mou, X. Y. Fang, and T. J. Huang, "A Multimodal Video Copy Detection Approach with Sequential Pyramid Matching," *IEEE ICIP'11*, Brussels, Belgium, pp. 3629-3632, Sep. 11-14, 2011.
[5] P. Viola, and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *IEEE CVPR'01*, Vol. 1, Kauai, HI, USA, pp. I-511-I-518, Dec. 8-14, 2001.
[6] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, Vol. 60, No. 2, pp. 91-110, 2004.
[7] J. Sivic, and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *IEEE ICCV'03*, Nice, France, pp. 1470-1477, Oct. 13-16, 2003.
[8] A. Gionis, P. Indyk, and R. Motwani, "Similarity Search in High Dimensions via Hashing," *VLDB'99*, Edinburgh, Scotland, pp. 518-529, 1999.
[9] J. P. Chen, and T. J. Huang, "A Robust Feature Extraction Algorithm for Audio Fingerprinting," *PCM'08*, Tainan, Taiwan, pp. 887-890, Dec. 9-13, 2008.
[10] Mpeg, "ISO/IEC 15938-4:2002 Information Technology - Multimedia Content Description Interface - Part 4: Audio," 2002.
[11] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *IEEE CVPR'06*, Vol. 2, New York, USA, pp. 2169-2178, Jun. 17-22, 2006.
[12] NIST, "CBCD Evaluation Plan TRECVid 2010 (v3)," http://www-nlpir.nist.gov/projects/tv2011/Evaluation-cbcd-v1.3.htm, Available in Nov. 2011.
[13] NIST, "Building Video Queries for TRECVid 2008 Copy Detection Task," http://www-nlpir.nist.gov/projects/tv2011/Trec-Vid2008CopyQueries.pdf, Available in Nov. 2011.