VCDB: A Large-Scale Database for Partial Copy Detection in Videos

Yu-Gang Jiang, Yudong Jiang, and Jiajun Wang

School of Computer Science, Shanghai Key Laboratory of Intelligent
Information Processing, Fudan University, Shanghai, China
ygj@fudan.edu.cn

Abstract. The task of partial copy detection in videos aims at finding if one or more segments of a query video have (transformed) copies in a large dataset. Since collecting and annotating large datasets of real partial copies are extremely time-consuming, previous video copy detection research used either small-scale datasets or large datasets with simulated partial copies by imposing several pre-defined transformations (e.g., photometric or geometric changes). While the simulated datasets were useful for research, it is unknown how well the techniques developed on such data work on real copies, which are often too complex to be simulated. In this paper, we introduce a large-scale video copy database (VCDB) with over 100,000 Web videos, containing more than 9,000 copied segment pairs found through careful manual annotation. We further benchmark a baseline system on VCDB, which has demonstrated state-of-the-art results in recent copy detection research. Our evaluation suggests that existing techniques—which have shown near-perfect results on the simulated benchmarks—are far from satisfactory in detecting complex real copies. We believe that the release of VCDB will largely advance the research around this challenging problem.

Keywords: Video copy detection, benchmark dataset, frame matching, temporal alignment.

1 Introduction

With the popularity of video capture devices and network sharing activities, a huge amount of videos are being transmitted online. This brings increased concerns about copyright issues due to the very low cost of copying a video (or a small fraction in it) and massively distributing it on the Internet. Therefore, video copy detection, which aims at automatically identifying copies in a large dataset, has received significant research attention.

The task of video copy detection is very challenging because of the complex content variations that widely exist among the copied segments, such as scale and lighting changes. Research on copy detection has benefited significantly from the invention of local invariant features like the SIFT [1]. Indexing structures such as the inverted file have also been popularly adopted to enable efficient detection [2]. While great progress has been made, many recent works focused only

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 357–371, 2014.

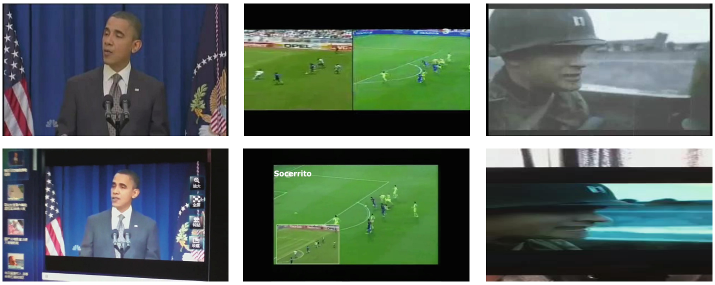

Fig. 1. Three pairs of frames extracted from copied video segments in VCDB. All the copies were found directly from the Internet through careful manual annotation. The complex forms of transformations in VCDB pose new challenges to video copy detection research, as the existing datasets were mostly generated "artificially" by imposing a very few number of pre-defined transformations.

on entire video-level copy detection [3, 4], where a query video and a reference video normally share very long copied segments. Annotations in these datasets were provided only at video-level, i.e., whether or not two videos are copies of each other, preventing research on finer-grained partial copy detection where the copied segments are short and may be even just one single frame. Precise partial copy detection is desired particularly in large datasets so that copyright protection becomes easier.

Because the manual annotation of real partial copies is very difficult and extremely time consuming, recent research on partial copy detection has mostly been done on small scale datasets with *simulated* copies [2], produced by imposing pre-defined transformations like modifications in scale and contrast. While the simulated datasets have been very useful, it is unknown how well the state-of-the-art approaches work on real copies, many of which are too complex to be simulated by just applying a few pre-defined transformations.

This paper introduces a large-scale video copy detection database (VCDB)[1] that aims to address the aforementioned shortcomings of the existing datasets. We construct a dataset of over 100,000 videos downloaded from the Internet, covering a wide range of topics like movies and sports. Through careful manual annotation, approximately 9,200 partial copies were found between around 6,000 pairs of videos. Figure 1 shows a few example frames in the found video segment copies, where the transformations between each pair are very complex. To set up a good baseline and understand the limitations of the existing solutions, we benchmark a popular method that has demonstrated state-of-the-art copy detection results in the literature. We also compare a few popular techniques in this area and provide insightful discussions.

The main contribution of this work is the construction of a large dataset with realistic partial video copies, which requires significant efforts in both design and annotation. Through benchmarking state-of-the-art techniques on the new

[1] Available at: `http://www.yugangjiang.info/research/VCDB/`

comprehensive dataset, we observe that the systems that have produced near-perfect results on the simulated datasets like TRECVID [5, 6] are still far from satisfactory. This opens up new opportunities to continue further research around this problem to make copy detection algorithms practically more effective.

The rest of the paper is organized as follows. We review related works in Section 2. Section 3 describes the construction and annotation of VCDB. Section 4 briefly introduces the baseline system and Section 5 discusses the evaluation results. Finally, Section 6 concludes this paper.

2 Related Work

We first discuss related datasets for video copy detection, and then review a few representative approaches.

Video Copy Detection Datasets: Although the problem of video copy detection has been investigated for decades, very few benchmark datasets have been constructed. Many researchers constructed their own datasets and did not release them for cross-site comparison. For instance, Indyk et al. [7] downloaded 2,000 clips of news, music videos and movie trailers. The duration of these clips is between 2 and 5 minutes. Copies were generated by the authors using pre-defined transformations including inserting TV logos and using various camcordings, frame rates, etc. Joly et al. [8] collected 1,040 hours of TV video data stored in MPEG1 format, containing contents in various categories like commercials, news, sports and TV shows. Copies were also created by imposing some transformations.

Perhaps the first well-known public benchmark is the Muscle-VCD, created by Law-To et al. [9], which contains around 100 hours of videos collected from the Internet, TV archives and movies. Videos are in different resolutions and formats. There are two kinds of queries representing two practical situations: (1) ST1: entire video copy (normally between 5 minutes and 1 hour), where the videos may be slightly recoded and/or noised. (2) ST2: partial video copy, where two videos only share one or more short segments. This scenario was also simulated by using video-editing softwares to impose a few transformations. The "transformed" segments were later used as queries to search their original versions in the dataset. The duration of a segment normally ranges from 1 second to 1 minute.

The importance of video copy detection was also recognized by the U.S. National Institute of Standards and Technology, whose annual TRECVID evaluation [10] included a separate task on copy detection in 2008. Each year a benchmark dataset was generated and released only to the registered participants of the task. The TRECVID datasets were constructed in a very similar way to the Muscle-VCD. The 2008 edition, used in several recent works like [6, 2], contains 200 hours of TV programs and around 2,000 query clips. Each query was generated using a software to randomly extract a segment from the dataset and impose a few pre-defined transformations. The copy detection task

Table 1. Comparison of video copy detection datasets, sorted by construction year. VCDB is the only one containing real partial copies.

	Reference	Year	Partial Copy	Type of Copies
Indyk et al.	[7]	1999	N	Real
Joly et al.	[8]	2003	Y	Simulated
Muscle-VCD	[9]	2007	Y	Simulated
CC_Web	[3]	2007	N	Real
TRECVID 2008	[10]	2008	Y	Simulated
UQ_Video	[4]	2011	N	Real
VCDB	—	2014	Y	Real

of TRECVID was terminated in 2011 because near-perfect results were reported. However, as will be shown later in this paper, the existing approaches cannot detect many real partial copies.

Different from the datasets mentioned earlier, which are all simulated based on pre-defined transformations, a few datasets have real video copies directly obtained from the Internet. The CC_Web dataset constructed by Wu et al. [3] has been popularly used, which consists of 12,790 videos collected from the video search results of Google, YouTube and Yahoo!. Another recent dataset, called UQ_Video [4], was constructed by extending the CC_Web with more background distraction videos. Both datasets were created for near-duplicate video detection, which by definition is different from the copy detection problem. For instance, two videos containing the same scenes but originally captured from two different cameras could be near-duplicates but not copies. Many copies in the two datasets are easy to be detected as the transformations among them are very limited. In addition, the labels in the datasets are only available on video-level, indicating whether or not two videos are copies of each other without the timestamps of the copied segments. Therefore they are not suitable for evaluating the techniques of partial copy detection. We summarize these datasets and compare them with VCDB in Table 1.

Video Copy Detection Approaches: Several noteworthy copy detection systems have been proposed in the past decade. We briefly describe a few representative ones. Works on entire video-level copy detection relied on the use of global features like color histogram and local features like the LBP [11, 3, 4]. Reasonably good results were obtained as the samples used in the experiments were mostly simple with limited content variations.

To accurately locate partial copies, particularly those under severe content transformations, more advanced techniques are needed. In [5], local features are extracted and quantized using the bag-of-visual-words (BoV) representations, which are then indexed by an inverted file structure for efficient retrieval. In [2], local descriptors were also used, but were quantized into an aggregated representation similar to the Fisher Vectors [12, 13]. The aggregated features were then encoded using an indexing structure for efficient frame retrieval or matching.

Finally, a modified Hough voting scheme was used to fuse the frame matching results and produce segment level copy predictions. Similarly, another system introduced by Tan et al. [14] used standard bag-of-words representations of the local descriptors, which were indexed in an inverted file structure. The matched local descriptors across two frames were further filtered by a geometric consistency verification method, which is able to reject outlier wrong matches that are geometrically not consistent to a majority of matches. After that, a temporal network model was constructed and the partial video copies can be found by solving a network flow optimization problem.

As can be summarized from the above works, most copy detection systems start from the extraction of local features, which are then used for frame-level matching. Finally, the frame matching results are sent into a temporal alignment method to identify the copied segments. The main differences of these systems lie in the choices of the efficient descriptor matching method (e.g., using the product quantization [15] or its extended version [16]), the geometric verification scheme (e.g., using the Weak Geometric Consistency [5] or its variant [17]), or the final copy segment identification algorithm. The first two steps, i.e., the local descriptor matching and geometric verification, are technically very similar to the approaches for image-based object retrieval, which has been extensively studied in the vision community [18, 5, 19–22].

3 Creating VCDB

3.1 Database Collection

All the videos in VCDB were downloaded from video-sharing websites YouTube and MetaCafe. In order to collect representative partial copies, we started from 28 carefully selected queries, covering a wide range of topics such as commercials, movies, music videos, public speeches, sports, etc. We downloaded the top returned search results of the queries from the two websites, and manually picked on average around 20 videos per query. These videos are all relevant to the query and many of them share partial copies. In total we have 528 videos (approximately 27 hours) in the core dataset, forming around 6,000 candidate pairs ($\binom{20}{2}\times 28$) requiring manual annotation.

To make the task of copy detection in VCDB close to the realistic application scenario, we further downloaded 100,000 videos from YouTube as background distraction videos. We skimmed over these distraction videos to reduce the chance of having copies of videos in the core dataset. The final VCDB consists of both the core dataset and the distraction videos.

3.2 Annotation

Annotating 6,000 pairs of videos on frame level is an extremely difficult task, particularly when many copies in the core dataset are short segments. Figure 2 gives an example of multiple partial copies between two videos. Manually identifying the boundaries of the segments is very time-consuming. Different from

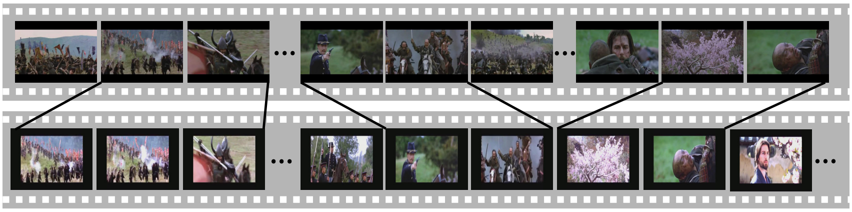

Fig. 2. An example of a video pair containing multiple partial copies. Similar cases are frequently seen in VCDB.

simple image or video annotation tasks that can be performed on crowdsourcing websites like the Amazon MTurk, the task of annotating partial copies is sophisticated as it requires more inputs with precise operations. This makes it very difficult to design a good interface on the MTurk for the novice workers. We therefore employed seven part-time annotators, who were well trained before performing the task.

An annotation tool was developed with careful design to finish the task efficiently. Each time two videos were shown to an annotator, who can view them separately or in parallel with different start times to compare them. The annotator can then input the timestamps of all the found copied segments. To speed up the effort, the transitivity property of the video copies was utilized in the annotation tool. Specifically, if two segments are copies of the same segment in another video, they are very likely to be copies of each other. Notice that the transitivity property does not always hold, since the bridging segment may contain two different scenes (a.k.a. picture-in-picture) that are copies of the two segments respectively (see an example of picture-in-picture in the middle of Figure 1). Therefore, the tool will automatically recommend these *candidate* segment pairs from transitivity propagation to the annotator for confirmation. This function can largely reduce the annotation time, because the annotator rarely needs to manually specify the boundary frames of these segments. The entire annotation process finished in about one month (around 700 man-hours).

3.3 Statistics

As a benchmark dataset, it is important that the copies in VCDB are representative and diverse. In total, 9,236 pairs of partial copies were found. Figure 3 gives one example copy from videos downloaded by each of the 28 queries. As can be clearly seen, there are a wide range of content transformations among the partial copies in VCDB, which cannot be fully covered by the very few predefined transformations used in generating the existing datasets with simulated copies.

We manually went through all the 9,236 pairs to count the number of copies according to a few major transformations popularly used in generating the simulated datasets. We found that around 36% of them contain “insertion of

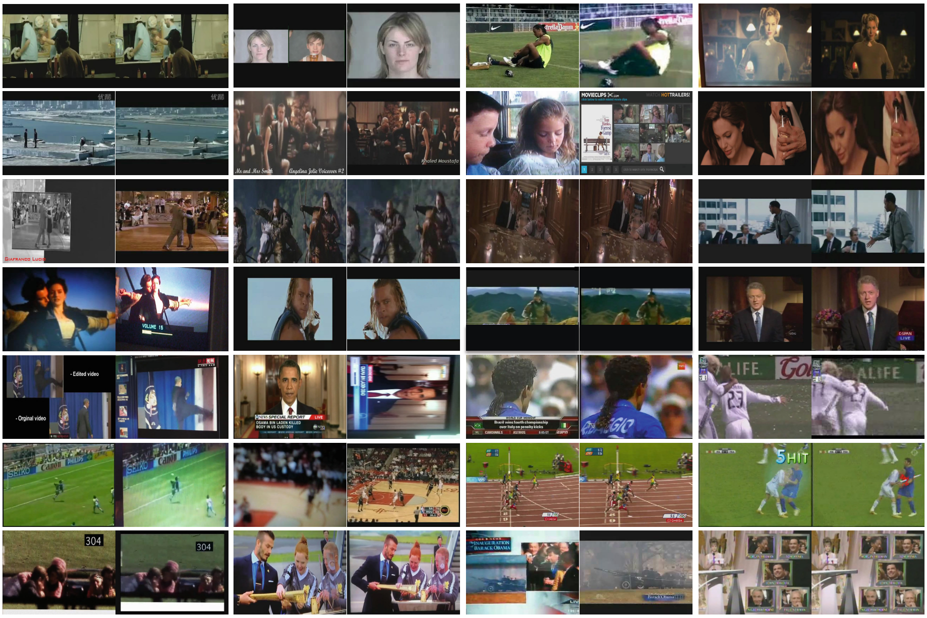

Fig. 3. Example frame copies from the videos downloaded by the 28 queries, respectively. Ordered from left to right and top to bottom, the corresponding queries are topics about commercials (3), movies (11), music video (1), public speeches (3), sports (6), surveillance event (1), and others (3).

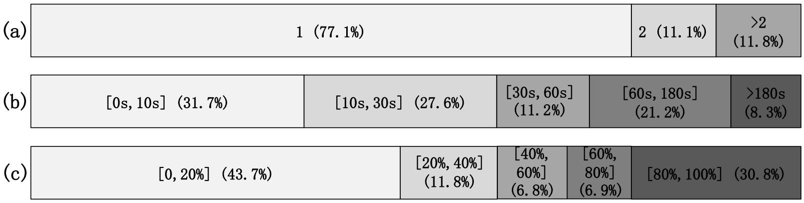

Fig. 4. Statistics of VCDB: (a) the number of partial copies per video pair, among those having at least one copy; (b) the duration of the partial copies; and (c) the percentage of the duration of the copy segments in the corresponding parent videos. See texts for more explanations.

patterns", 18% are from "camcording", 27% have scale changes, and 2% contain "picture in picture" patterns. These percentages are quite different from that in the simulated datasets. Many "insertion of patterns" copies exist in the practical scenario because of the logos of different TV channels, and the "picture in picture" patterns frequently seen in the simulated copies do not seem to be popular in real cases.

Figure 4 further shows some statistics of VCDB. We see that, among the video pairs that have at least one partial copy, nearly 80% of them contain just

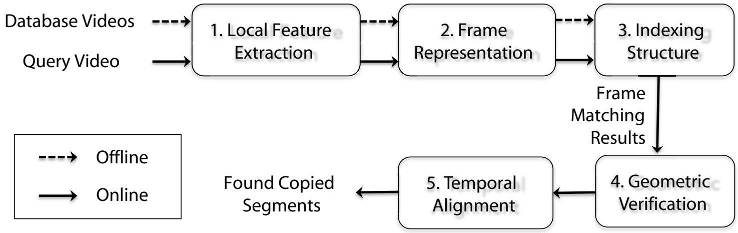

Fig. 5. The general framework of a video copy detection system.

one copied segment and as high as 20% contain two or more partial copies. In addition, 32% of the found copied segments are less than 10 seconds and another 28% are between 10 and 30 seconds, which are very short. More importantly, according to Figure 4(c), we see that 44% of the copies are shorter than 1/5 of their parent videos and only 31% of them occupy over 80% of the parent videos. This confirms the fact that most copies in VCDB are partial video segments.

4 The Baseline System

To evaluate the capability of current copy detection techniques, and also to understand the difficulty of VCDB, we benchmark a system that has produced strong performances on various datasets. This also sets up a good baseline for future systems to compare against. Most state-of-the-art video copy detection systems follow a basic pipeline as shown in Figure 5. The first several core components of the baseline system (modules 1–4 of Figure 5) are based on the work of Herve et al. [5]. For temporal alignment, we adopt and compare two options [14, 2]. In the following we briefly describe the techniques used in all the modules.

Feature Extraction and Frame Representation: First, frames are uniformly sampled from the videos and local SIFT descriptors are computed on each of the frames. The popular BoV representation is then used to quantize the SIFT features from each frame. The codebook used in generating the BoV representations is constructed by hierarchical k-means, which segments the SIFT feature space into many Voronoi cells.

Indexing and Hamming Embedding: The inverted file structure is adopted to index the frames for efficient online frame matching. Hamming embedding is used [5] to alleviate the effect of quantization errors in the traditional BoV representation. Specifically, the key idea of Hamming embedding is to partition each Voronoi cell into a few subspaces. Each subspace is represented by a very short binary code, so that the feature similarity within the cell can be measured by the Hamming distance that can be efficiently computed. With Hamming embedding, two SIFT feature matches only when they fall in the same Voronoi cell and their Hamming distance within the cell is smaller than a threshold. This is better than directly using more Voronoi cells (i.e., more clusters) because using more cells will incur significant quantization error [23].

Geometric Verification: The SIFT matches found by the inverted file and Hamming embedding are not always correct. One important reason is that the BoV representation and the indexing structure do not capture any geometric information such as the orientations of the local image patches. The matching accuracy can be improved by geometric verification as a post-processing step to exclude "wrong" matches that are not consistent with a majority of matches geometrically. For this, a weak geometric consistency (WGC) method [5] is adopted. WGC is based on the angle and scale parameters of the SIFT descriptors, which are used to adjust the matching scores of video frames. The underlying assumption is that the matching score of a frame pair should be enhanced if the matched SIFT features are transformed by consistent angles and scales. Similarly the score should be reduced if the matched features are transformed inconsistently. As both the angle and the scale parameters are embedded in the SIFT descriptors, WGC can be very efficiently computed. For more details of the Hamming embedding and the WGC, readers are referred to [5].

Alignment by Temporal Network: Two frames are considered to be a copy pair if they have a sufficient number of matched SIFT features over a threshold. The next step is to align the matched frames and identify the copied video segments, by considering both the visual similarity and the temporal information. Note that this alignment process also has the capability of further filtering the wrong frame matches by checking temporal (in)consistency. We adopt two methods to achieve this goal. The first one was proposed by Tan et al. [14], who formulated the problem by network flow optimization. Given a query video Q and a database video R, a temporal network is constructed by querying the top-k similar frames from R using Q. After that directed edges are established across the frames in the top-k lists by chronologically linking the frames according to their timestamps. The value (edge weight) of the link (edge) is the similarity value between the corresponding frames. Finally, optimization is performed to identify the longest path (segment) by considering three constraints: the maximum difference between the timestamps of two successively aligned frames, the minimum length of a copied segment, and the minimum similarity value between the matched frames.

Alignment by Temporal Hough Voting: The second temporal alignment method adopted in this work is called temporal Hough transform proposed in [2]. Denote $s(t_q, t_d) > 0$ as the matching score between a query frame at time t_q and a reference database frame at time t_d. A histogram $h(\delta)$ is computed to accumulate the frame matching scores for the matched pair within a window of δ frames: $h(\delta) = \sum_{t_q \in Y} s(t_q, t_q + \delta)$, where Y is the set of timestamps of the query and $s(t_q, t_q + \delta) = 0$ if the timestamp $t_q + \delta$ does not exist in the database video. Peaks are then searched in the histogram, and the matched segments are identified around the peaks. In addition, because consecutive frames in videos can be visually very similar, we often see bursts of matches which bias the scores returned by the Hough histogram. To alleviate this issue, a re-weighting scheme is adopted to normalize the matching scores. The normalized scores are used as input to compute the histogram $h(\delta)$. This alignment method was used in a system [2] that produced competitive results on the TRECVID dataset of simulated partial copies.

Discussions: Here we briefly discuss the rationale of selecting the baseline techniques. One important guideline is that the selected techniques should be representative and have shown consistently good results on multiple datasets. We underline that, although the methods of [5] and [2] were proposed a few years ago, to our knowledge they still represent a state-of-the-art solution, and systems developed on top of them have demonstrated outstanding performance in competitions such as the TRECVID [10]. Very few new copy detection methods have been developed recently. This is probably because the near-perfect results on the traditional simulated databases have delivered a wrong signal that the video copy detection problem might already be successfully solved. Perhaps the most related approach proposed recently is by Revaud et al. [16], who used a different frame representation called VLAD [13] and an extended version of the product quantization [15] for event retrieval in large video databases. We also implemented this pipeline on VCDB but observed slightly worse results than the adopted baseline. This is probably because the approach was designed for similar video event retrieval, which emphasizes more on similar semantics, not necessarily the same visual patterns, and therefore did not enforce strong geometric consistency of the matched local feature points.

5 Experiments

In this section we discuss experimental results. While our main purpose is to analyze the results of the aforementioned techniques on VCDB, we also conduct experiments on a small and popular benchmark, the Muscle-VCD dataset [9], in order to ensure that all the methods are correctly implemented and to examine the power of the baseline system.

5.1 Muscle-VCD

For this dataset, we focus on the ST2 of partial copies as described in Section 2. In total there are 21 query segments, and performance is evaluated by QF=$1 - |missed\ frames|/|groundtruth\ frames|$ and QS=$(|correct| - |false\ alarm|)/|returned\ segments|$, following [9]. Throughout all the experiments in this work, we use uniform frame sampling to extract two frames every second. Note that using more frames may lead to slightly better results, but evaluating this factor is beyond the focus of this paper.

Using the baseline system with the first temporal alignment method, i.e., the temporal network, we achieve 0.81 for QS and 0.70 for QF. To our knowledge

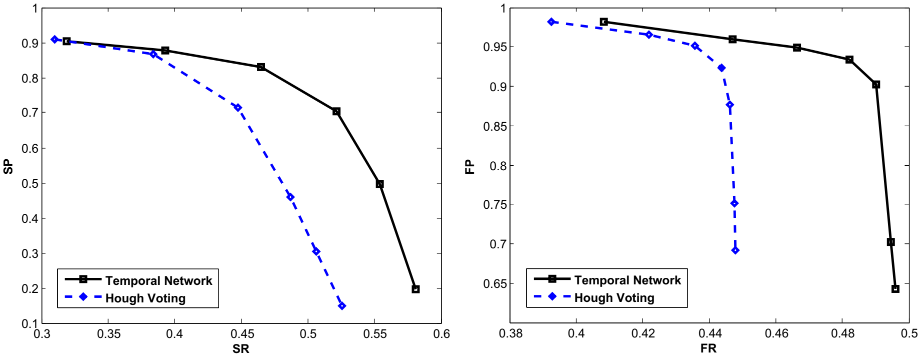

Fig. 6. Precision-recall curves of the baseline system on the core dataset of VCDB, using the two temporal alignment methods respectively. **Left:** segment-level results. **Right:** frame-level results. Overall, the performance is much worse than that reported on the existing datasets with simulated copies, indicating that partial copy detection in realistic videos remains a challenging problem that deserves future research.

the best results achieved on this dataset are 0.86 and 0.76 respectively for the two criteria, reported in [14] where a similar method to the baseline system was used. The small performance gap is mainly due to the use of different geometric verification methods. An improved version of the WGC was used in [14], while our baseline uses the standard WGC.

5.2 VCDB

This subsection presents results on VCDB. We first report performance on the core dataset, and then discuss the results of large scale experiments by incrementally adding the background distraction videos. Each segment of the 9,236 pairs is used as a query. Performance is measured by the standard precision and recall, which are widely adopted and can nicely reflect the power of a copy detection system. A detected pair of copied segments is considered correct if both segments have intersection frames with a ground-truth pair. We do not set a minimum percentage of the overlapped time window because hitting a ground-truth pair with one single frame will be adequate in practical applications such as copyright protection. More formally, the segment-level precision (SP) and recall (SR) are defined as: SP=$|correctly\ retrieved\ segments|/|all\ retrieved\ segments|$ and SR=$|correctly\ retrieved\ segments|/|groundtruth\ copy\ segments|$. In addition, we also measure frame-level precision and recall on the core dataset as auxiliary criteria to understand how accurate the baseline system is, which are defined as: FP=$|correctly\ retrieved\ frames|/|all\ retrieved\ frames|$ and FR=$|correctly\ retrieved\ frames|/|groundtruth\ copy\ frames|$.

Results on the core dataset are shown in Figure 6. To plot the precision-recall curves, we adjust the thresholds of the frame matching scores and the minimum numbers of matched frames needed for temporal alignment to achieve different

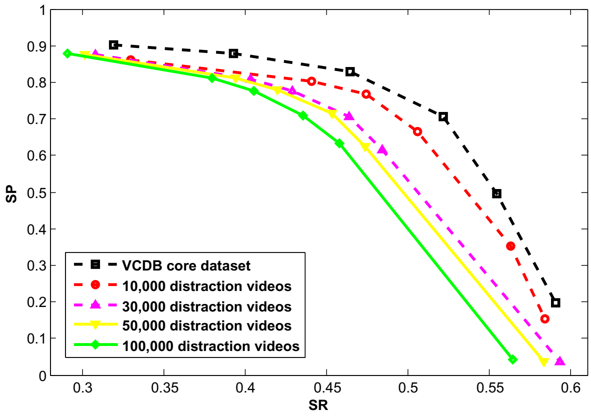

Fig. 7. Precision-recall curves of large-scale copy detection on VCDB, using the temporal network method with different numbers of background distraction videos

levels of detection precisions and recall rates. We compare the two temporal alignment methods on this core dataset. The only technical difference behind the two curves is the use of different alignment methods. As can be observed from the figure, the temporal network method produces better results in most cases, which shows the effectiveness of explicitly enforcing the several constraints in an optimization framework. This method is slightly slower than the Hough voting based method but is practically acceptable as the number of matched frames is limited after thresholding. In addition, we see that the frame-level recall tends to be saturated at 0.5 for the temporal network method and at 0.45 for the Hough voting. This indicates that around half of the copied frames are difficult to be identified by the baseline system.

Overall, the results on this core dataset are far from satisfactory. The baseline system with the temporal network alignment method achieves very impressive results on the Muscle-VCD dataset, but can only attain a segment-level recall of around 0.48 at a similar precision of 0.80. The frame-level recall is similar at the same level of precision. This clearly verifies our argument that partial copy detection under realistic scenario is much more challenging.

Next we move on to the large scale copy detection experiments by gradually adding the background distraction videos. We use four distraction set sizes with 10,000, 30,000, 50,000, and 100,000 (the entire VCDB) videos respectively. Results are visualized in Figure 7. As expected, the performance drops with an increasing number of the added distraction videos. However, the degradation is quite insignificant considering the large number of distraction videos included in the experiments, particularly when recall is smaller than 0.4. This indicates that the baseline system is not very sensitive to background noises, which is quite appealing as robustness is very important in large scale real applications. In addition, similar to the trends shown from the small-scale experiment on the VCDB core dataset, one can observe from Figure 7 that the existing

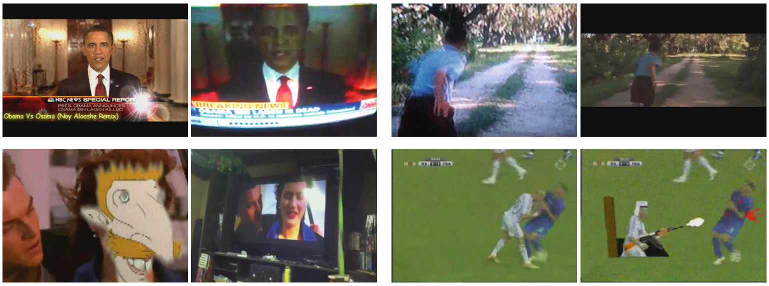

Fig. 8. Four frame pairs that are difficult to be detected, which contain very severe and complex content variations

techniques face difficulties in locating nearly 50% of the partial copies. These copies are valuable resources as they pose new challenges for future research. Figure 8 shows examples of a few failure cases.

6 Conclusions

We have introduced a new dataset called VCDB for partial copy detection in videos, which is—to our knowledge—the only large scale dataset containing realistic partial video copies. Most previous video copy detection research used simulated datasets, on which near-perfect results have been frequently reported. Because of this, copy detection is sometimes considered as a solved problem. This has largely limited the needed progress of copy detection research. With over 9,000 carefully annotated partial copies and over 100,000 videos, VCDB goes far beyond the existing benchmarks and poses new challenges to the research around this problem.

We evaluated a baseline system on VCDB, which is built upon techniques that have produced state-of-the-art performances on related tasks. The performance of the system is far from satisfactory, indicating that VCDB is arguably a good benchmark for future investigations. We also compared two temporal alignment methods on VCDB and observed that the temporal network method with optimization using explicit constraints tends to be a better solution. The best recall rate on VCDB is just close to 0.60 when the precision significantly drops to 0.20. This suggests that future research on copy detection should pay particular attention on the frame matching stage to overcome the difficulties caused by the severe content variations.

Acknowledgments. This work was supported in part by a National 863 Program (#2014AA015101), a Key Technologies Research and Development Program (#2013BAH09F01), a grant from the National Natural Science Foundation of China (#61201387), four grants from the Science and Technology Commission of Shanghai Municipality (#13PJ1400400, #13511504503, #12511501602 and #14511106900), and a grant from Ministry of Education (#20120071120026), China.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
2. Douze, M., Jégou, H., Schmid, C., Pérez, P.: Compact video description for copy detection with precise temporal alignment. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 522–535. Springer, Heidelberg (2010)
3. Wu, X., Hauptmann, A.G., Ngo, C.W.: Practical elimination of near-duplicates from web video search. In: ACM MM (2007)
4. Song, J., Yang, Y., Huang, Z., Shen, H.T., Hong, R.: Multiple feature hashing for real-time large scale near-duplicate video retrieval. In: ACM MM (2011)
5. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
6. Douze, M., Jegou, H., Schmid, C.: An image-based approach to video copy detection with spatio-temporal post-ltering. IEEE TMM 12(4), 257–266 (2010)
7. Indyk, P., Iyengar, G., Shivakumar, N.: Finding pirated video sequences on the internet. Technical Report, Stanford University (1999)
8. Joly, A., Frelicot, C., Buisson, O.: Robust content-based video copy identification in a large reference database. In: Bakker, E.M., Lew, M., Huang, T.S., Sebe, N., Zhou, X.S. (eds.) CIVR 2003. LNCS, vol. 2728, pp. 414–424. Springer, Heidelberg (2003)
9. Law-To, J., Joly, A., Boujemaa, N.: Muscle-VCD-2007: a live benchmark for video copy detection (2007), `http://www-rocq.inria.fr/imedia/civr-bench/`
10. U.S. National Institute of Standards and Technology: TREC video retrieval evaluation, `http://trecvid.nist.gov/`
11. Law-To, J., Chen, L., Joly, A., Laptev, I., Buisson, O., Gouet-Brunet, V., Boujemaa, N., Stentiford, F.: Video copy detection: a comparative study. In: CIVR (2007)
12. Perronnin, F., Dance, C.R.: Fisher kernels on visual vocabularies for image categorization. In: CVPR (2007)
13. Jegou, H., Douze, M., Schmid, C., Perez, P.: Aggregating local descriptors into a compact image representation. In: CVPR (2007)
14. Tan, H.K., Ngo, C.W., Hong, R., Chua, T.S.: Scalable detection of partial near-duplicate videos by visual-temporal consistency. In: ACM MM (2009)
15. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE TPAMI 33(1), 117–128 (2011)
16. Revaud, J., Douze, M., Schmid, C., Jegou, H.: Event retrieval in large video collections with circulant temporal encoding. In: CVPR (2013)
17. Zhao, W.L., Ngo, C.W.: Flip-invariant sift for copy and object detection. IEEE TIP 22(3), 980–991 (2013)
18. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)

19. Perronnin, F., Liu, Y., Sanchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: CVPR (2010)
20. Arandjelovi, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR (2012)
21. Avrithis, Y., Tolias, G.: Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval. IJCV (2013)
22. Zhang, S., Yang, M., Wang, X., Lin, Y., Tian, Q.: Semantic-aware co-indexing for image retrieval. In: ICCV (2013)
23. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR (2008)