Caught Red-Handed: Toward Practical Video-based Subsequences Matching in the Presence of Real-World Transformations

Yi Xu True Price Fabian Monrose Jan-Michael Frahm
University of North Carolina at Chapel Hill
{yix,jtprice,fabian,jmf}@cs.unc.edu

Abstract

Every minute, staggering amounts of user-generated videos are uploaded to on-line social networks. These videos can generate significant advertising revenue, providing strong incentive for unscrupulous individuals that wish to capitalize on this bonanza by pirating short clips from popular content and altering the copied media in ways that might bypass detection. Unfortunately, while the challenges posed by the use of skillful transformations has been known for quite some time, current state-of-the-art methods still suffer from severe limitations. Indeed, most of today's techniques perform poorly in the face of real world copies. To address this, we propose a novel approach that leverages temporal characteristics to identify subsequences of a video that were copied from elsewhere. Our approach takes advantage of a new temporal feature to index a reference library in a manner that is robust to popular spatial and temporal transformations in pirated videos. Our experimental evaluation on 27 hours of video obtained from social networks demonstrates that our technique significantly outperforms the existing state-of-the-art approaches with respect to accuracy, resilience, and efficiency.

1. Introduction

Love it or loathe it, the spread of "pirated" multimedia is here to stay. The dramatic rise of pirated content has been fueled by the staggering amounts of user-generated videos uploaded to online social networks — e.g., over 300 hours of video are uploaded to YouTube every minute [40], the number of video posts per Facebook user has increased by more than 75 percent in one year alone, and nearly 70 million photos and videos are posted on Instagram each day[1]. The fact that many of these uploaded videos generate significant advertising revenue has not gone unnoticed, providing incentive for unscrupulous individuals to capitalize on this bonanza[2]; Internet advertising revenues in the U.S. totaled more than $42.8 billion in 2013. Some studies estimated that in 2007 alone YouTube made as much as $15 million directly from the presence of infringing material on their website, in addition to profits based on the traffic that content lured into their platform [3].

Of course, more traditional forms of content theft still prevail, although large-scale digital piracy is rare [17]. Online piracy websites that host copyrighted content (*e.g.* music and movies) rely on ad networks to generate handsome profits off the work of content creators. According to a 2014 report [8] from the Digital Citizen's alliance, websites that make money exclusively off pirated content earned more than $227 million in ad revenue in 2013. While these file-sharing sites (*e.g.* The Pirate Bay) are routinely held in violation of copyrights for the content they host, several factors can make such a determination less obvious in other cases. For instance, in the United States, whether an uploaded video qualifies as fair use or is in fact copyright infringement depends on factors such as (*i*) the purpose and character of use, (*ii*) the nature of the copyrighted work, (*iii*) the effect of the use upon the potential market and the amount, and (*iv*) substantiality of the portion taken. In infringement cases, the decision on how each of the four factors of fair use applies is left to the discretion of a judge [25]. While the first three factors would pose significant challenges for fully automated approaches, it is possible to use computer-vision approaches to address the last factor and make suggestions regarding infringement as soon as a video is uploaded.

The ability to aid with questions regarding content piracy is becoming ever more important, as copyright infringement is now rampant across social networks. For most of the popular user-generated content (UGC) platforms (*e.g.* YouTube, Facebook, Vimeo), it is the video uploader's responsibility to verify that the uploaded content is indeed in the public domain. Additionally, as part of the Digital Millennium Copyright Act (DMCA) "safe harbor"[3] protection, a UGC platform is obligated to make the most mini-

[1] See `https://instagram.com/press/`.

[2] See *"Copyright and YouTube: Pirate's Playground or Fair Use Forum?"*, K. Hunt, Michigan Tech. Law Review, 2007

mal of efforts to prevent copyright infringement and is required to apply owner-selected policies on any video known to include pirated content for which the content owner has expressed interest in restricting use [29]. However, with the sheer volume of videos uploaded daily, it is difficult for content creators to find potential infringements in a timely manner, especially when the videos may be transformed in non-trivial ways. Consequently, many infringing videos go unnoticed for extended periods, and as a result, the owners of user-generated content platforms have regularly been called upon in court (for what might be considered overzealous applications of copyright law) [24].

To be compliant with the DMCA's safe harbor protections and stem the tide against such lawsuits, UGC platforms have turned to automated techniques for detecting copyrighted material. The most popular of these systems is YouTube's proprietary Content ID algorithm, which routinely inspects over 400 years worth of video footage every day. The goal of this system is to automatically advise both the uploader and copyright owner about potential infringements. Content owners can select to (1) do nothing and simply track the video, (2) block the video entirely, or (3) redirect all monetization and advertising revenue from the video to the content owner. Today, more than 90% of content owners choose to have the UGC platform redirect all monetization to themselves [1, 27]. As long as accurate assessments are provided, the labor burden for copyright protectors should be lessened. Today, YouTube reports that more than 8,000 partners use Content ID, including major US network broadcasters, movie studios, and record labels, and the service has led to over $2 billion in payouts to creators and rights holders since its launch in 2007.

While the techniques such as Content ID have been successful, they have been widely criticized for false detections [1], which particularly impact independent content creators[3]. Improving the accuracy of such systems remains an important social problem and immense business opportunity. As such, it has garnered much attention from both academia and industry. To that end, the U.S. National Institute of Standards and Technology (NIST) included a separate challenge on video copy detection in their annual TREC Video Retrieval Evaluation (TRECVID) in 2008. In that challenge, synthetic video-based transformations were used to test the performance of different content detection approaches. Four years later, the challenge was prematurely terminated under the claim that near-duplicate video detection was a solved problem.

However, in 2014, Jiang *et al*. [14] showed that the current state of the art — which showed near-perfect results on the simulated benchmarks — are far from satisfactory in detecting complex real-world copies. To highlight the problem, they released the `VCDB` dataset, which contains pirated videos available on YouTube and MetaCafe. Their preliminary evaluations suggested that the transformations observed *in the real world* are very different from the synthetic transformations considered by the academic community to date. For instance, the most widely studied transformation in the TRECVID evaluations, "picture in picture," is rarely seen in real cases, while the more commonly observed transformations in online pirated videos are far more complex. As a result, the techniques that appear to be robust in simulated benchmarks fail miserably in the wild. The work of [14] demonstrated that the assumptions made in the NIST challenges were far too naïve and did not generalize well to what real-world adversaries would do.

In this paper, we go beyond the work of [14] and analyze the difference between real-world and synthetic copies (§3). From this analysis, we introduce a new method for practical content-based copy detection using principles of *temporal consistency*, which are difficult to thwart without significantly degrading the user's viewing experience — contrary to the pirate's goals. We demonstrate that our approach is more robust against spatial transformations and commonly used temporal transformations, and we also highlight cases in which our detection fails. Our method achieves higher accuracy and recall rate on real-world videos from `VCDB` than the existing state-of-the-art approaches.

2. Background and Related Work

There is abundant literature on content-based copy detection (CBCD) approaches, but they all share the same detection structure: feature extraction followed by indexing and searching [22]. The extraction phase identifies fingerprints of both the query video and the reference video. A successful extraction scheme should be robust to transformations. Namely, the fingerprints from near-duplicate video pairs should share a closer distance under a predefined metric than non-copy pairs. The indexing phase efficiently compares the query video against the dataset. Given the large size of real-world reference libraries, efficient data structures such as K-d trees [41] are usually employed.

The feature extraction phase is often the weakest point, targeted for exploitation by adversaries seeking to defeat copy detection algorithms. A devious copyright infringer could, for example, use spatial or temporal transformations to destroy the requisite features without significantly altering the visual experience of the copied portion. To provide context, the remainder of this section discusses several considerations and challenges that impact how well either party can perform their respective tasks.

Spatial Features and Transformations Most CBCD approaches use either global or local image descriptors on a per-frame basis. The most well-known global features for

[3] See also P. Tassi, *"The Injustice of the YouTube Content ID Crackdown Reveals Google's Dark Side"* Forbes Magazine, 2013.

copy detection include the ordinal feature [11, 13], spatial correlation descriptor [39], MPEG-7 descriptors [19], and the TIRI descriptor [9]. Popular local features include LBP features [31], Harris corners [15, 20, 21, 28], SIFT/SURF [10, 18, 42], and variations such as the Hessian-based STIP descriptor [36]. Global features are more robust against noise and changes introduced by digitization, but they are often not robust against image shift or the addition of black margins and borders. Local features provide detailed information about every frame in the video, thereby allowing the matching of much shorter copy sequences.

Voting techniques are often used to evaluate the similarity between spatial feature sequences [20]. In some cases, weak temporal order consistency among frames is applied. More advanced schemes use frame fusion [35], bag-of-words models [7], or graph-based matching [6] to improve robustness. These extensions leverage the temporal relation between frames and can handle some temporal transformations (*e.g*. speed adjustment).

Regardless of the similarity measures used, global spatial features are sensitive to bordering and shifting, while local features are sensitive to local spatial transformations. These issues render spatial-information-based CBCD detection systems vulnerable to commonly applied transformations such as cam-recording of the reference video or manual editing. Additionally, although spatial methods may only require short sequences for detection, the complexity of the features make the schemes computationally inefficient. In contrast, for longer sequences (*e.g*. over a minute in length as common in content piracy), our proposed temporal features provide an easier and more efficient pathway.

Temporal Features and Transformations Instead of focusing on individual frames, temporal features characterize changes of pixel values over time. Only a handful of approaches have used temporal features for CBCD. Chenot and Daigneault [5], for example, use a temporal fingerprint based on spectral analysis of intensity changes. Since the fingerprint only relies on average intensity changes, their approach offers some resilience to spatial transformations but remains very sensitive to common temporal adjustments. Since only the first 16 channels of a Fourier transformation are used as a fingerprint, the approach fails to characterize the sudden illumination changes that contain the most useful information for characterizing scene changes.

Chen and Stentiford [4] applied the ordinal measure in the temporal domain to different blocks of a frame. Their application of the ordinal measurement captures temporal characteristics, and by tuning the number of blocks, the algorithm can be used to find a good balance between spatial robustness and detection efficiency. However, their approach fails to characterize the magnitude of temporal changes, and it is sensitive to temporal adjustment such as speed changes and frame dropping. More recently, an video retrieval approach using compromising reflections (*e.g*. as observed through a window) was presented by [38]. They used temporal information to better capture sudden illumination changes in a video and were able to retrieve near duplicate videos in a reference library. Similar to the approach of [4], [38]'s approach is robust to spatial changes, but it also suffers from sensitivity to temporal adjustments.

Figure 1. Copy-reference pairs in the VCDB dataset

Instead of manipulating spatial features (*e.g*. SIFT or SURF), many temporal transformations directly manipulate the video frames via cropping or dropping, or by inserting other video sequences. These transformations impose more stringent requirements on the design of a CBCD algorithm: by design, a certain level of temporal ambiguity must be allowed instead of merely comparing corresponding frames for possible matches.

3. Analysis of Real-World Manipulation

Numerous public datasets exist for evaluating copy detection technologies are publicly available. The most widely used are the MUSCLE-VCD-2007 dataset (with over 100 hours of video) [23] and the IACC dataset (with over 200 hours of video) released by NIST for the TRECVID challenge [30]. However, the transformations used in these datasets are all *synthetic*, without guidance from real-world data. Recently, the more realistic VCDB dataset was published by [14], which contains 528 videos retrieved from two UGC platforms using 28 queries. From this set, the authors manually labelled 9,236 pairs of copied segments. Examples of transformations in the VCDB dataset are shown in Fig. 1.

The videos from the VCDB dataset contain a wide range of content transformations, which is in stark contrast to the pre-defined lab-generated transformations [30]. For instance, 8% of copies were edited to have parts of the original video deleted or extra segments inserted. The distribution of transformations is also different from that in IACC. Among the 9,236 pairs of copies, 36% contain insertion of patterns, 18% are due to cam-recording, 27% have scale changes, and only 2% contain "picture in picture" patterns.

To dig deeper into the data, we built a graphical user interface for frame-level labelling and manually labelled a

subset of 210 videos that were all over 30 seconds long. Among those videos, we found 4402 copy-reference segment pairs, all of which were manually verified. These data enable us to run a thorough analysis at frame-level precision. We focused on the temporal characteristic of the transformed videos. Two types of temporal transformations were immediately apparent: speed adjustment and segment editing (*i.e*. video cropping and temporal reversing).

Speed Adjustments To characterize the observed video speed transformations, we computed the distribution of video speed. We observed that 44.2% of the pirated videos have no temporal scaling, and 2.2% are scaled by more than 20%. To analyze the consistency of the speed adjustments, we computed the standard deviation of the video speed within each pirated content. The result (not shown) revealed that as much as 97.7% of the pirated videos do not have different speeds within the video.

Cropping, Editing and Reversing The data collected from the UGC platforms indicated that roughly 40% of the videos contained more than 80% content copied from elsewhere. These results are consistent with the observations of [14]. Over 8% of the samples contain edited videos wherein segments were either deleted or inserted. 2% of the videos were temporally reversed. The low percentage of temporal reversion makes sense because few videos are still watchable once reversed.

The aforementioned analysis of real-world transforms revealed that *although the magnitude of the intensity signal in the copy video may vary significantly due to spatial transformations, the temporal positions of sudden intensity changes remain relatively constant*. Our conjecture is that this phenomenon occurs because harsh temporal adjustments appear to degrade the viewing experience more than similarly strong spatial adjustments. In our approach, we take advantage of this observation to design a robust technique for content-based subsequence matching.

4. Our Approach

A high-level depiction of our approach is shown in Fig. 2. Our approach consists of two main components: a knowledge extraction stage and a video sub-sequence matching (or "copy detection") stage. In first stage, we preprocess the reference library by extracting derivative features for all its videos (i.e., computing their "fingerprints" in stage ❶). These derivative features encode the temporal position of illumination changes in the videos, which can be leveraged to find linkages among a collection of videos.

To build an indexing structure that is robust against temporal transformations, local temporal features that cover a short time span around a peak in the derivative feature are extracted from derivative features of each video (stage ❷).[4] They encode the gradient signal profile of significant illumination changes in a manner that is robust to temporal transformations. A K-d tree structure is built upon the local features of all reference videos.

The detection stage is used to determine if a video contains infringing material. In this phase, the uploaded video's derivative and local features are also computed. The local features are then used as a first approximation (stage ❸) to retrieve reference videos with similar local intensity profiles. Each retrieved local feature provides a set of potential candidates for matching reference videos. The collection of possible matches is then considered for further scrutiny using full video information (stage ❹). The comparison to the full video, or large segments thereof, requires a known temporal transformation. However, it is important to note that our local features do not provide an immediate estimate of the required frame rate adjustment and starting/ending position of the segment: We must iteratively estimate these parameters between each candidate and the uploaded video.

To accomplish this, we train a SVM to recognize near-duplicate video sequence pairs. Using this classifier and the position of the local features in the reference video, we are able to estimate the matching segment and then derive the scaling factor based on this alignment. Afterwards, we refine the matching segment based on the current estimate of the scale. If the SVM score of the resulting sub-sequence pair is lower than a predefined threshold, the uploaded video is deemed as non-infringing.

4.1. Derivative Feature Extraction

Our derivative features are computed as the temporal gradient of the video frames' average intensity signal s_t. The temporal gradient is calculated as $ds(t) = s(t+1) - s(t)$. Based on the observation of Xu *et al*. [38] that brightness changes uniquely characterize a video, we convert the temporal gradient $ds(t)$ into the derivative feature f by only preserving its significant ($|ds(t)| > 1$) local extrema ("peaks") given by

$$f(t) = \begin{cases} ds(t), & \text{if } |ds(t)| > 1 \wedge |ds(t)| > |ds(t-1)| \\ & \quad \wedge |ds(t)| > |ds(t+1)| \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

We extend the single feature per frame of Xu *et al*. [38] to a hierarchical representation also computing the temporal feature on a grid of $n \times n$ image tiles, which span the entire set of images. In all our experiments, $n = 3$ is used. This results in a 10-dimensional feature vector for each frame.

We propose a classifier to evaluate whether two aligned sequences are near-duplicates. We adopt the similarity metric of [38] to measure the element-wise similarity of the derivative feature vector pairs. This yields a 10-dimensional

[4] As this paper does not use spatial features, we henceforth use the terms "local temporal feature" and "local feature" interchangeably.

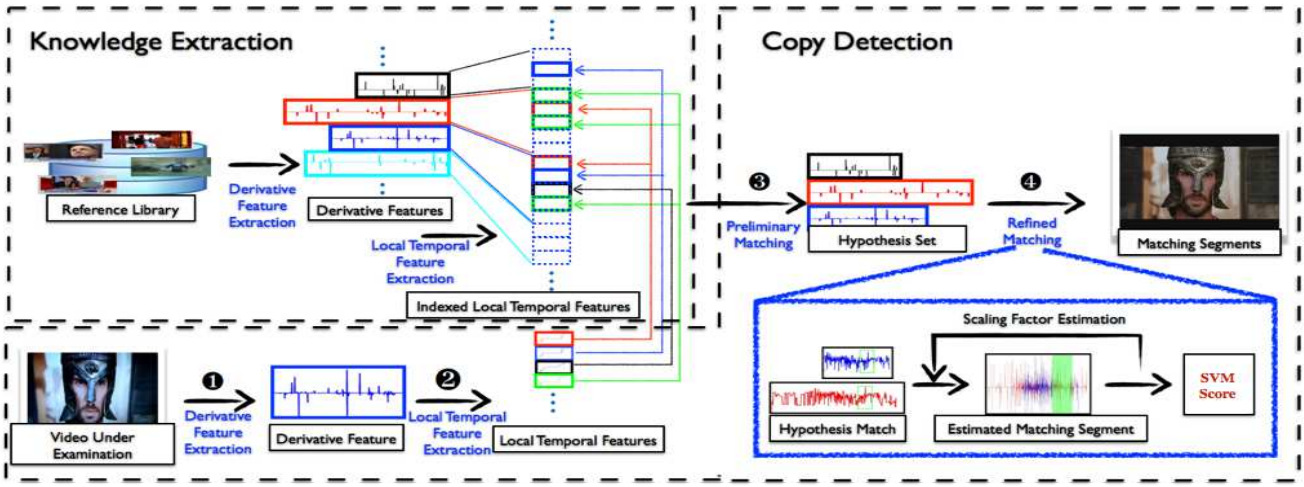

Figure 2. Overview of proposed approach

similarity score vector. The score vector mainly encodes the difference between temporal positions of changes in illumination between two videos (the potential original and its potential copy). To consider the intrinsic properties of the uploaded video's segment, we concatenate the similarity score vector with the length of the uploaded video and the peak rate to form a 12-dimensional vector. We train a SVM on such descriptors, using a training set of 20% of our manually labeled sequence-pairs as positive samples and randomly selected negative samples. The classifier outputs a SVM score that measures how likely a particular sequence pair is a duplicate. If the score is higher than a given threshold, we declare the sequence pair to be near-duplicates.

4.2. Local Feature Extraction

For fast retrieval, local features are computed from the derivative feature values using a sliding window of $t = 5$ seconds at a 10 Hz video sampling rate. These values were empirically chosen. We focus on the sliding window position where there is a peak at the center of the window. Also, we only consider the sliding windows where the magnitude of the center peak is larger than at least 60% of other peaks in the window. This choice is guided by the intuition that high peaks — *representing strong changes in illumination* — are less likely to be noise and are therefore more likely to be preserved during the creation of a pirated version of the original video. We return to this choice later in §5.

To achieve robustness to temporal transformations, each peak in the window of a reference video casts votes for its position in its temporal neighborhood in the new video. The weights of the votes for different temporal positions follow a Gaussian function. Intuitively, the resulting vector of votes for the sliding window defines the local feature after it is normalized to have an L_2-norm of one.

4.3. Indexing and Retrieval

We use a K-d tree [2] to efficiently index the local features. We note that the VCDB dataset contains over 27 hours of video, wherein we identified 17,438 local features. Given a new video, we extract local features, after which the N strongest features of every five-minutes-long sub-sequence are used to locate similar local features among all reference videos using the K-d tree structure.

For each of these N features, we retrieve the nearest α fraction of local features from K-d tree. Fig. 3 shows an example of a sample video's local feature and its closest neighbor. This process produces a list of potentially infringing sub-sequences in the new video and their correspondence in the reference set. Unfortunately, given the limited temporal context of the local features, these correspondences contain false positives, which have to be eliminated.

4.4. Detecting Copied sub-sequences

To minimize false positives, we use the frame correspondences as seeds for a more thorough duplicate *sub-sequence* detection step. In this step, our goal is to determine the sub-sequence that is most likely to be copied. In order to correctly identify these copied sub-sequences, we need to know both their temporal *positions* and the temporal *scales* that might have been applied by the pirate.

Sub-sequence estimation We begin our sub-sequence estimation by assuming that the scaling factor is one (*i.e.* there is no speed adjusted to the copied segments). We align the reference and uploaded videos by matching the suggested frame correspondence. We exhaustively check all sub-sequences containing the frame correspondence with length less than $\Theta = 60$ seconds, which is already enough for the trained classifier to determine whether the sub-sequence pair is a copy. The estimated copy sub-sequence is the one with the largest SVM score.

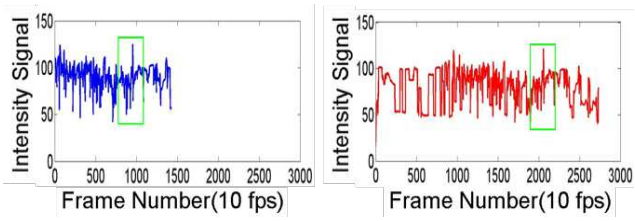

Figure 3. Local feature in uploaded video and its closest neighbor in the reference set.

Temporal scale estimation Once a sub-sequence is identified as containing potentially infringing content, we refine the scale estimate for that sub-sequence. Given that, in practice, the observed range of temporal scales is limited (per the analysis in §3), we explore scales from 0.8 to 1.2 quantized into 100 steps. We then select the scale (*i.e.* the slow-down or speed-up factor) that yields the highest similarity between the sub-sequence from the uploaded and the reference video. To normalize temporal uncertainties of the peaks, we change the sub-sequence's temporal domain to its logarithmically transformed time axis with the center peak at the origin. In this logarithmic domain, the uncertainty caused by the scale estimation error becomes uniform across the sub-sequence, and the impact of quantization error is mitigated when estimating the scaling factor. We use this modified scoring scheme to determine the best scale estimation for the identified overlapping sub-sequence.

Once we have an estimate for the scaling factor, the longest overlapping sub-sequence is recalculated given the newly estimated temporal scale. An example of a correctly scaled and matched sub-sequence for a pirated video is shown in Fig. 4. In practice, we found this process to converge within three iterations. As a final optimization, we only consider sub-sequences flagged as potential copies if their SVM score is higher than Φ. The threshold Φ can be tuned to balance precision and recall rate.

5. Evaluation

As we address the piracy of longer content such as movies, TV series, sports events, and podcasts that are prime subjects for monetization, we evaluated the performance of our approach on videos in the VCDB dataset that are longer than 30 seconds in length. To provide a comparison with the state of the art, we use the same evaluation methodology used in the VCDB benchmark, whereby performance is measured using the standardized metrics of precision and recall. Specifically, the segment-level precision (*SP*) and recall (*SR*) in these benchmarks are defined as

$$SP = \frac{|correctly\ retrieved\ segments|}{|all\ retrieved\ segments|} \quad (2)$$

$$[SR = \frac{|correctly\ retrieved\ segments|}{|groundtruth\ copy\ segments|} \quad (3)$$

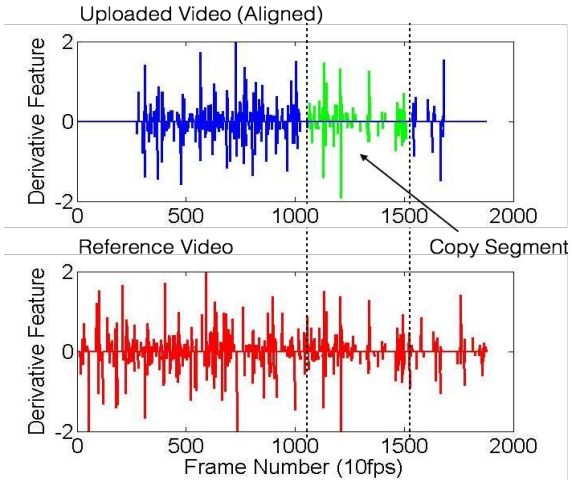

Figure 4. Final result of comparing the uploaded video and the selected reference video. Green: matched segment; Blue: unmatched segment; Red: reference sequence.

To plot the precision-recall curves, we vary the threshold of the minimum copy-segment length Φ. We set $\alpha = 1.7\%$ and $N = 8$ for all our experiments.

To compare our results with the current state of the art, we implemented the best-performing approach on the VCDB benchmark, Tan *et al*. [33]'s temporal network approach (TNP). Our implementation reproduced the results reported in [14]. Furthermore, we increased the reference library size by adding an additional 50,000 YouTube videos (4,300 total hours) to the VCDB dataset. Fig. 7 demonstrates the consistent performance of our approach as the reference library size increases. Our approach yields an *18.2% improvement in recall rate compared to TNP [33] at the same precision of 90.0%*.

5.1. Robustness to Transformations

Naturally, our accuracy depends on the characteristics of the pirated content; the more illumination changes in the original material, the better. To further evaluate the performance of our method, we grouped the videos in the VCDB dataset by category. The results based on this categorization are shown in Fig. 5. For comparison, we also evaluated the accuracy of the baseline approach [33] considering the different grouping of videos. We observe that the approach of [33] has slightly better performance on only a handful of newscast-type videos (breakdown not shown).

To better demonstrate the strength of the proposed approach, we elaborate on specific samples from the VCDB dataset where we succeed but the current state of the art performs poorly. Through our empirical evaluations, we found that the TNP [33] largely fails to detect copies that are modified by spatial flipping, brightness adjustments, and insertion of captions, which are all important real-world

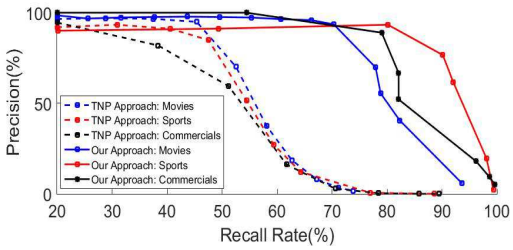

Figure 5. The Precision-Recall curve for TNP versus our approach considering different genres of videos.

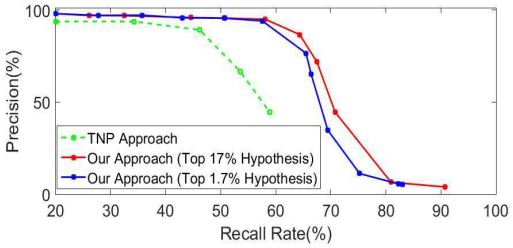

Figure 6. Precision-Recall curve of the TNP approach versus our approach with different accuracies.

classes of spatial transformations to which our approach is robust. Sensitivity to spatial flipping could be dealt with in the method of [33] by flipping the uploaded video before the matching process is performed, but the other types of modifications still pose significant challenges to any frame-retrieval-based framework. (Table 1 lists the occurrence of each on these transforms in the VCDB dataset.) Specifically, the currently widely used spatial features such as SIFT and SURF provide weak performance against illumination changes [16]. We also observed a severe limitation of existing methods to distinguish between different types of inserted captions. Specifically, captions copied into a pirated video inevitably lead to false detections, as they are incorrectly matched to random text in the reference library.

5.2. Runtime Performance

While precision and recall rates are important considerations, the computational complexity of a content-based detection system is equally important, particularly given the fact that as much as 300 hours of videos are uploaded to UGC platforms per minute. In general, the computational cost of any copy detection algorithm depends on its required accuracy, the length of the uploaded video, and the size of the reference dataset.

Theoretically, the bottleneck of our approach lies in the sub-sequence matching process, where we evaluate the most likely copy sub-sequence correspondences. The more correspondences we test, and the longer the uploaded video, the more computation time is required. Specifically, our proposed approach has computational complexity $O(HL)$, where H is the number of tested frame correspondences and L is the length of the uploaded video. In practice, this is also affected by other factors such as peak-rate.

Speed versus Accuracy The design of our approach allows us to explore the tradeoff between accuracy and speed. The parameter that has the largest influence on accuracy is α, the number of hypotheses that are tested during the copy detection phase. To achieve a good compromise between recall, precision and computational cost, we examined a range of values for α. The results are shown in Fig. 6.

Our proof of concept consists of two components: the local feature correspondence retrieval and the sub-sequence estimation. The correspondence retrieval is implemented through the VLFeat library for the K-d tree search, and the sub-sequence estimation is implemented using Matlab. To study the computational overhead of our approach, we use a six-core computer with 2.40GHz CPU and 24GB memory. Our results indicate that to obtain near optimal precision, the top 17% of the hypotheses would have to be tested, which results in an average evaluation time of 70 ms per frame with a standard deviation of 4.5 ms.

Notice that the results depicted in Fig. 6 show that we can improve performance by only testing the top 1.7% of the hypotheses while reducing the recall rate by about 3%. When α is set to 1.7%, the average detection time drops to 8.6 ms per frame with a standard deviation of 0.8 ms. These results validate that the sub-sequence estimation component of our proposed approach is linearly related to the number of tested frame correspondences.

Length of Queried Video Given that the entire query video has to be checked for potential infringement, it is obvious that the computational cost is related to the length of the uploaded video. The performance evaluation in Fig. 8 shows that when the query length increases, the computational cost of our techniques increases modestly from 14 s for a 40 s video to 64 s for a 20-minute video. This is a dramatic improvement over the state-of-the-art system [33], whose computational time ranges from 8 minutes to nearly 4.7 hours — making it impractical for real world usage.

Type	%	Description
Illumination Adjustment	20.0%	While local spatial features such as SIFT and SURF provide some degree of brightness invariance, they are still sensitive to illumination adjustments.
Spatial Flipping	0.4%	The SIFT/ SURF feature is sensitive to flipping.
Caption	8.9%	Local spatial features cannot distinguish between different captions. As a result, the caption frames in the video are all considered similar with high probability, leading to false detections

Table 1. Transformations that caused failures in the TNP approach. 20% of the failures can be attributed to illumination adjustments made by the pirates.

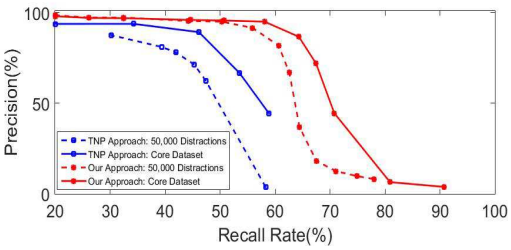

Figure 7. The Precision-Recall curve of our approach and TNP on the VCDB core dataset with an additional 50,000 videos.

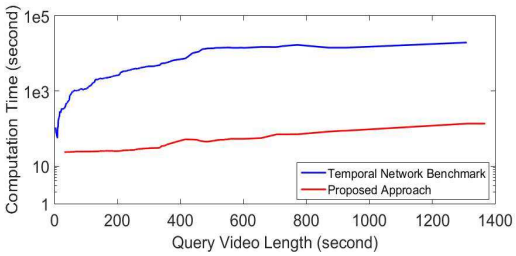

Figure 8. The influence of query length on computational cost.

It is prudent to note that our implementation of the benchmark approach follows the guidelines presented by [14] and is consistent with the computation time of the semi-brute-force matching used by Wu *et al*. [37]. The significantly higher computation time of the baseline approach can be attributed to the fact that it requires a frame retrieval stage that takes place twice for every second in the uploaded video. Tan *et al*. [32] suggested that one can consider only the most likely candidate keyframe correspondences, which would boost the frame retrieval process by a factor of 58 at the expense of a minor loss of precision. However, even considering this improvement, our approach still outperforms theirs by at least an order of magnitude.

Size of Reference Library An increase in the size of the reference library will affect the computational performance of any CBCD approach. For our approach, a larger dataset influences efficiency in two ways: (1) a larger dataset results in a longer time to retrieve the local feature correspondences, and (2) a larger dataset necessitates more local feature correspondences to be evaluated for copied sub-sequences, leading to higher sub-sequence estimation times. To test the impact of increased library size, we ran our proposed approach with the above described 50,000 additional videos YouTube added to the VCDB dataset and α remaining at 1.7%. The results show that the average computational time increased from 8.6 ms to 410 ms per frame, against an increase of database size from 27 hours to 4300 hours. Regarding the scalability of our method, we believe our prototype implementation can be greatly improved in terms of efficiency by, *e.g*. fine-tuning the query operation for large-scale datasets, rewriting our implementation in C instead of Matlab, and parallelizing the sub-sequence estimation. Fast pre-processing of frame content could also greatly speed up our method by decreasing the number of potential video matches that need to be analyzed. For example, a deep neural network such as SSD [26] could be employed to semantically identify objects in the query video frame, allowing database entries lacking similar objects to be preemptively ruled out.

6. Limitations

There are potential modifications that skilled adversaries could apply to degrade our copy detection rates. In particular, a pirate could temporally smooth a video's gradient features to hide the sudden illumination changes used by our method. Furthermore, our approach assumes a piecewise-consistent temporal transformation; an adversary may attempt to constantly change the frame rate of a video to undermine the sub-sequence detection stage and thereby impair the retrieval process. While these techniques could potentially evade our detection, it is unclear whether they are acceptable avenues in the real world. Transformations such as temporal smoothing (where video segments are arbitrarily sped up or slowed down significantly) can seriously reduce the video's visual quality, which would be at odds with a pirate's intent of profiting from video popularity. Indeed, we found no such modifications in the data we analyzed.

7. Conclusion

We have introduced a novel method for efficiently and robustly detecting pirated video content on user-generated content platforms. Our method is robust against the frequently used spatial and temporal transformations observed in the wild. The techniques we propose for enhanced sub-sequence matching of video-based content provide significantly improved precision and substantially better computational performance and scalability than previous approaches. Importantly, our technique narrows the detection gap in the important area of temporal transformations applied by would-be pirates. Our large-scale evaluation on real-world data shows we can successfully detect infringing content from movies, commercials, and sports clips with 90.0% precision at a 75.1% recall rate, while at the same time outperforming the state of the art *by an order of magnitude* in terms of speed. This is an important step forward at a time where the losses from failure[5] to take down infringed works is estimated at over a billion dollars [12, 34].

[5] For example, YouTube was previously sued by Twentieth Century Fox, Magnolia Studios, and Viacom for losses in the hundreds of millions.

References

[1] T. B. Bartholomew. Death of fair use in cyberspace: Youtube and the problem with content id, the. *Duke L. & Tech. Rev.*, 13:66, 2014.

[2] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[3] B. Brown. Fortifying the Safe Harbors: Reevaluating the DMCA in a Web 2.0 World. In *Berkeley Technology Law Journal*, volume 23, Jan. 2008.

[4] L. Chen and F. Stentiford. Video sequence matching based on temporal ordinal measurement. *Pattern Recognition Letters*, 29(13):1824–1831, 2008.

[5] J.-H. Chenot and G. Daigneault. A large-scale audio and video fingerprints-generated database of tv repeated contents. In *Content-Based Multimedia Indexing (CBMI), 2014 12th International Workshop on*, pages 1–6. IEEE, 2014.

[6] C.-Y. Chiu, C.-S. Chen, and L.-F. Chien. A framework for handling spatiotemporal variations in video copy detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(3):412–417, 2008.

[7] C.-Y. Chiu, C.-C. Yang, and C.-S. Chen. Efficient and effective video copy detection based on spatiotemporal analysis. In *Multimedia, 2007. ISM 2007. Ninth IEEE International Symposium on*, pages 202–209. IEEE, 2007.

[8] Digital Citizens Alliance. Good Money Gone Bad: Digital Thieves and the Hijacking of the Online Ad Business. A Report on the Profitability of Ad-Supported Content Theft. Feb. 2014.

[9] M. M. Esmaeili, M. Fatourechi, and R. K. Ward. A robust and fast video copy detection system using content-based fingerprinting. *Information Forensics and Security, IEEE Transactions on*, 6(1):213–226, 2011.

[10] N. Gengembre, S.-A. Berrani, and P. Lechat. Adaptive similarity search in large databases-application to image/video copy detection. In *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pages 496–503. IEEE, 2008.

[11] A. Hampapur, K. Hyun, and R. M. Bolle. Comparison of sequence matching techniques for video copy detection. In *Electronic Imaging 2002*, pages 194–201. International Society for Optics and Photonics, 2001.

[12] L. Hilderbrand. Youtube: Where cultural memory and copyright converge. 2007.

[13] X.-S. Hua, X. Chen, and H.-J. Zhang. Robust video signature based on ordinal measure. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 1, pages 685–688. IEEE, 2004.

[14] Y.-G. Jiang, Y. Jiang, and J. Wang. Vcdb: A large-scale database for partial copy detection in videos. In *European Conference on Computer Vision (ECCV)*. 2014.

[15] A. Joly, O. Buisson, and C. Frelicot. Content-based copy retrieval using distortion-based probabilistic similarity search. *Multimedia, IEEE Transactions on*, 9(2):293–306, 2007.

[16] L. Juan and O. Gwun. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.

[17] J. Karaganis. Copyright infringement and enforcement in the US: A research note. November 2011.

[18] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate detection and sub-image retrieval. In *ACM Multimedia*, volume 4, page 5, 2004.

[19] O. Küçüktunç, M. Baştan, U. Güdükbay, and Ö. Ulusoy. Video copy detection using multiple visual cues and mpeg-7 descriptors. *Journal of Visual Communication and Image Representation*, 21(8):838–849, 2010.

[20] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Robust voting algorithm based on labels of behavior for video copy detection. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 835–844. ACM, 2006.

[21] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Vicopt: a robust system for content-based video copy detection in large databases. *Multimedia systems*, 15(6):337–353, 2009.

[22] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 371–378. ACM, 2007.

[23] J. Law-To, A. Joly, and N. Boujemaa. Muscle-vcd-2007: a live benchmark for video copy detection, 2007. `https://www.rocq.inria.fr/imedia/civr-bench/`.

[24] L. Leister. Youtube and the law: A suppression of creative freedom in the 21^{st} century. In *Thurgood Marshall Law Review*, number 305, 2011.

[25] P. N. Leval. Toward a fair use standard. *Harvard Law Review*, pages 1105–1136, 1990.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016.

[27] S. McArthur. How to beat a YouTube ContentID Copyright Claim — what every gamer and MCN should know. `http://www.gamasutra.com/blogs/StephenMcArthur/20140624/219589/How_to_Beat_a_YouTube_ContentID_Copyright_Claim__What_every_Gamer_and_MCN_Should_Know.php`, June 2014.

[28] S. Poullot, O. Buisson, and M. Crucianu. Z-grid-

based probabilistic retrieval for scaling up content-based copy detection. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 348–355. ACM, 2007.

[29] A. R. Simon. Contracting in the dark: casting light on the shadows of second level agreements. In *William & Mary Business Law Review*, Feb. 2014.

[30] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 321–330. ACM, 2006.

[31] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 423–432. ACM, 2011.

[32] D. S. Tan, P. Keyani, and M. Czerwinski. Spy-resistant keyboard: More secure password entry on public touch screen displays. In *Proceedings of the 17th Australia Conference on Computer-Human Interaction*, 2005.

[33] H.-K. Tan, C.-W. Ngo, R. Hong, and T.-S. Chua. Scalable detection of partial near-duplicate videos by visual-temporal consistency. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 145–154. ACM, 2009.

[34] US Government Accountability Office, Report to Congressional Committees. Intellectual Property: Observations on the efforts to qauntify the economic effects of counterfeit and pirated goods. April 2010.

[35] S. Wei, Y. Zhao, C. Zhu, C. Xu, and Z. Zhu. Frame fusion for video copy detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(1):15–28, 2011.

[36] G. Willems, T. Tuytelaars, and L. Van Gool. Spatio-temporal features for robust content-based video copy detection. In *Proceedings of the 1st ACM Int. Conf. on Multimedia information retrieval*, pages 283–290. ACM, 2008.

[37] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *Proceedings of the 15th international conference on Multimedia*, pages 218–227. ACM, 2007.

[38] Y. Xu, J.-M. Frahm, and F. Monrose. Watching the watchers: Automatically inferring tv content from outdoor light effusions. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 418–428. ACM, 2014.

[39] M.-C. Yeh and K.-T. Cheng. A compact, effective descriptor for video copy detection. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 633–636. ACM, 2009.

[40] Youtube. Statistics from youtube website. `https://www.youtube.com/yt/press/statistics.html`, 2015. Accessed: 2015-04-28.

[41] J. Yuan, L.-Y. Duan, Q. Tian, and C. Xu. Fast and robust short video clip search using an index structure. In *Proceedings of the ACM SIGMM international workshop on Multimedia information retrieval*, pages 61–68. ACM, 2004.

[42] D.-Q. Zhang and S.-F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 877–884. ACM, 2004.