THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS INC.
SIGNAL PROCESSING
ALGORITHMS, ARCHITECTURES, ARRANGEMENTS, AND APPLICATIONS
SPA 2017
September 20-22nd, 2017, Poznań, POLAND

A Novel Inverted Index File based Searching Strategy for Video Copy Detection

Mengyang Liu*, Lai-man Po*, Yasar Abbas Ur Rehman*, Xuyuan Xu† and Yuming Li‡ Litong Feng§
*Department of Electronic Engineering
City University of Hong Kong, Kowloon, Hong Kong SAR, China
Email: {lmyleon2014, edmond.po, and yasir.abbas42}@gmail.com
†Tencent Video, Tencent Holdings Limited, Shenzhen, China
Email: xuyuanxu2@gmail.com
‡Autonomous driving lab, Tencent Holdings Limited, Beijing, China
Email: lymhust@gmail.com
§SenseTime Group Limited, Hong Kong SAR, China
Email: lightedfeng@gmail.com

***Abstract*—The demand of video copy detection system is growing rapidly, as the development of online video uploading and sharing. In the past two decades, the research concentrates on the video information extraction or feature building rather than the fast searching strategy. However, as the number of videos is growing, fast searching in video copy detection systems has become a big issue. In this paper, we propose a novel fast searching strategy for Inverted Index File (*IIF*) based video copy detection system by using fingerprinting technology. The proposed searching approach consists of two parts - fingerprint matching and video fragment matching. To speed up the fingerprint matching process, a table lookup operation is utilized that rely on the counting of matched sub-fingerprints instead of Hamming Distance metric. For video fragment matching, all fingerprint candidate is used to propose more than one matched video candidate with a different similarity score. The proposed fast searching strategy is tested on experimental content-based video copy detection system with different fingerprinting methods, distortion types and video database scale. Experimental results show that the proposed searching approach achieve high accuracy, and is around 10 times faster compared with the conventional IIF method. Moreover, with database upscaling, the searching rate of the proposed approach is faster than the conventional IIF methods that further make it a potential candidate to be used in large-scale video copy detection systems.**

I. INTRODUCTION

With the rapid advancement in Internet and mobile devices (especially for smartphones), the video capturing, processing, storage, and sharing has become much easier and cheaper. There is a plethora of streaming websites, like YouTube, Facebook, and iQIYI in China that provide video uploading and sharing services to millions of users. These websites have billions of video resources, and the number of active users sharing and watching these resources has been progressively increasing. Due to the availability of these massive video resources, a problem has been emerged and become a critical challenge for video professionals - *Video copyright protection*. To tackle this problem, video copy detection techniques are always required to determine whether one video is a copy of another video. This kind of technique can also be used to video content tracing, video content retrieval, and video propagation analysis. In this paper, we follow [1] to define the copied video, "The copied video is close to the original video but different in file formats, compression parameters, photometric variation, editing operations, different lengths, and frames adding or losing."

The content-based and the watermark-based copy detection are two popular methods for video copy detection. Content-based video Copy Detection (CBCD) systems utilize the contents of a video, which is more efficient and compatible as compared to systems based on watermarking [2]–[4]. Thus, in this paper, we only discuss the CBCD system. In general, the CBCD systems consists of two parts. The first part consists of a system for extracting features of a video and the latter part is designed for searching the matched video in a large data space. Video fingerprinting is a common technique for extracting features of a video in CBCD systems. It uses a long binary string or a sequence of binary strings to represent the video [5]–[12]. The binary string is a compatible data structure, which can be easily applied on the different platform. In this paper, we choose video fingerprinting as the feature extraction method in the first part of CBCD system.

Although most of research work focused on CBCD systems, there have been very little work avilable on the searching strategy for CBCD systems. The Inverted Index File (*IIF*) is a commonly used data format for building the binary strings searching algorithm. Following the original *IIF* based design [5], [10], [13]–[15], a binary string (fingerprint) is converted into a decimal number as a hash key in the Look-Up Table, and the information about the fingerprint is stored in the bucket that is pointed by the corresponding hash key. One advantage of *IIF* based algorithm is that the computation complexity of locating the position of a certain fingerprint is *O(1)*. While it is desired in most of the systems, the drawback is that this basic algorithm can only locate the same fingerprint without any bit error. It is impossible to generate the same fingerprint of the original video and for a copied video with a little bit distortions. Because of this, many attempts concentrate on the

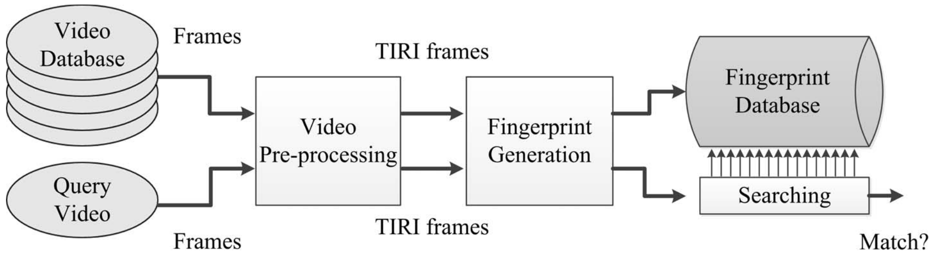

Fig. 1. The framework of video copy detection system

IIF based similarity searching. Hamming Distance (*HD*) is a well-known measure of similarity which is applied to the original *IIF* based design. However, the *HD* calculation is not efficient and accurate. Thus, it needs extra operations to lower down the computational load. Esmaeili M.M et al. [10] use cluster-based approach and Kulis B. et al. [16] use kernelized approach to reducing the *HD* calculation times. In these type of methods, the data should be grouped into several chunks at the pre-processing stage. At the searching stage, the query's fingerprint for a corresponding group is found firstly, and then, the similarity measurement is applied to the candidates of that group. The number of candidates in a group can be much smaller than the number of candidates in the whole database, so the times of similarity measurement has been reduced. However, this type of method depends on the data distribution. If the data cannot be divided into groups well, the accuracy decreases a lot and the searching time for the different group is unstable. Some groups may contain a large number of memeber, which leads to a long searching time. Moreover, cluster-based approaches are not suitable for adding new videos which leads to an unbalanced problem. There are other works concentrated on enhancing the searching algorithm. Hao et al. [17] design the beacon guided search (BGS), which only consider the candidates with high occurrence rate to lowering the number of candidates. However, the accuracy of this method is not acceptable, because the fingerprints with low occurrence rate can be significant. Haitsma et al. [18] define the "weak bits" which are the bits that can be easily influenced by the noise or distortion and generate a list of the most probable alternative fingerprints as candidates for one fingerprint. However, it is hard to define the "weak bits" which is determined by the fingerprinting algorithm and the distortion type. Esmaeili M.M. et al. [19] use pseudo sub-fingerprint with the different number of error bit to search the matching candidates and give the candidates with "0" bit error the highest weights, the candidates with "1" bit error the second highest weights, and so on. This method can get rid of the similarity measurement, however, when the length of a fingerprint is long and the number of bit error is large, the number of pseudo sub-fingerprints will increase too much. For example, the 9-bit sub-fingerprint with 3-bits error means $\binom{9}{3} = 84$ pseudo sub-fingerprints.

The main contribution of this paper is that we introduce the number of matched sub-fingerprint candidates to lower down the computation cost caused by the *HD* calculation, which can achieve faster searching in CBCD system. The other advantage of our algorithm is that it can find more than one candidate with the different similarity score to further increase the accuracy of the video copy detection. The remainder of this paper is organized as follow. Section II introduces the basic structure of the whole system and our proposed searching algorithm. Section III is the experimental setting and results. The last section mainly talks about the conclusion and further work.

II. Methodology

The framework of the proposed approach is illustrated in Figure 1. The video pre-processing part includes frame rate changing and frame size changing to make the system suitable for the various frame rate and size, and the *TIRI* frame [20] generation is used to get the tempo information among video frames. The fingerprinting generation part is used to extract the features of an input video and to build the fingerprint. The video which will be added to the database and the query video use the same technique in these two parts. For the video which is added in the database, the fingerprints are used as the hash key to locate the bucket. The video ID and the frame index is stored as the hash value. For the query, the fingerprint sequences are used to find the matched video.

In this section, the proposed Inverted Index File (*IIF*) based searching strategy will be introduced. The conventional *IIF* based searching strategies [10] mainly depend on the Hamming Distance (*HD*) that is used to be a measure of similarity. For binary strings (fingerprints), *HD* is equal to the number of "1" (population count) in "*XOR*" operation between two strings. For one fingerprint pair, if *HD* is lower than a certain threshold, they are considered as the matched fingerprint pair. This process is the same in the video segment similarity measurement but with a different threshold. To use the single scale threshold, the Normalized Hamming Distance (*NHD*) is used, which is based on$NHD = HD/l$, where the *HD* is the hamming distance and the l is the total length of the binary strings. *NHD* is a good tool to represent the similarity among binary strings, however, it is time-consuming and its flexibility is low in video copy detection. To find the matched fingerprint or the matched video segments in a large video database, the conventional searching strategies use many *HD* calculations. Using "*XOR*" and "counting" operations too many times lower

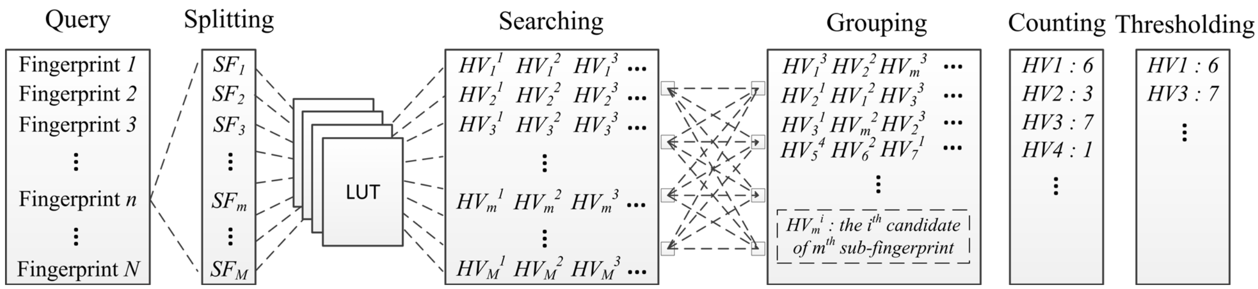

Fig. 2. Basic operations of fingerprint matching step in proposed searching method using n^{th} fingerprint in query as an example. The operation applied on each query's fingerprint is the same and independent. **Splitting**: the fingerprint will be split into *M* nonoverlapping sub-fingerprints. **Searching**: finding the matched sub-fingerprints' candidates from the corresponding *LUT* of each sub-fingerprint. Each candidate is represented by the Hash Value (*HV*). **Grouping**: putting the candidates with the same *HV* together, which come from the same fingerprint in video database. **Counting & Thresholding**: calculating the number of sub-fingerprint of each group and deleting the group whose number of candidates is not larger than the threshold (N_{sf}). The remained *HVs* are considered as the fingerprint candidates.

the efficiency of these systems. The conventional method can get a matched result quickly only when a matched candidate can be found for that query. However, this raises a new problem: the searching process is slow when the query is not in the video database because the system will go through all fingerprints' sub-fingerprints' candidates.

The method proposed in this paper consists of two steps: fingerprint matching and video segment matching. We have found that many of the *HD* computations can be avoided if the number of equal sub-fingerprints between fingerprints is considered in the fingerprint matching step. As in the *IIF* based data structure, the computation complexity for finding a sub-fingerprint's candidates is *O(1)*. A lot of computations will be saved if the system only counts the number of sub-fingerprint's candidates rather than calculating the *HD* for all of them. The proposed method assumes that "*The probability for two fingerprints, to be considered as, matched is high if they have many same sub-fingerprints*". This assumption is the same with the *NHD* case. The only change is that changing the number of "1" bit for binary strings to the number of equal sub-fingerprint. In the fingerprint matching step, therefore, we only consider the fingerprints with the number of equal sub-fingerprints above a pre-defined threshold (N_{sf}) as the matched fingerprint. As this step can dramatically decrease the number of fingerprint candidates, in the next step (video fragment matching), we still use *NHD* to represent the similarity among query and video segment candidates. Following is the details of these two steps.

A. Fingerprint Matching

If candidates of all sub-fingerprints in one Query's fingerprint are considered, the number of candidates becomes too large for a particular system. The conventional *IIF* methods deal with all these candidates when the system cannot find a matching result, which makes the searching time quite long and it increases nonlinearly as the scale of the database increasing. We found that if the number of matched sub-fingerprint's candidates has been introduced and a certain threshold (N_{sf}) is used to filter out a bunch of candidates, a lot of computation can be saved. Figure 2 is the illustration of this process. After pre-processing, the query video has been converted to a sequence of fingerprints. The following operations applied on each fingerprint is equal and independent, so the n^{th} fingerprint has been chosen as an example. Firstly, the fingerprint is split into *M* nonoverlapping sub-fingerprints. Each sub-fingerprint is equal in length and non-overlaping. Secondly, using each sub-fingerprint as a hash key to find their candidates from the corresponding Look Up Table (*LUT*). Normally, the number of *LUT* is equal to the number of sub-fingerprint and all the data in one bucket can be found using *LUT*. After this step, all sub-fingerprints' candidates are found. All candidates are represented by Hash Value (HV_m^i) which means the i^{th} candidate of the m^{th} sub-fingerprint. As the *LUT* is an efficient binary searching strategy, the computational cost of this step is *O(1)*. Thirdly, the candidates with the same *HV* are put together. Since each *LUT* is linked to a certain sub-fingerprint, therefore the candidates in this step follow this rule:

$$HV_m^i \neq HV_m^j, i \neq j, \forall m \in [1, M] \tag{1}$$

It means that any two candidates belong to one sub-fingerprint cannot have the same *HV*, while the two candidates come from different sub-fingerprints can have equal *HV* and these two sub-fingerprints' candidates are considered as coming from the same fingerprint in the video database. Fourthly, how many candidates of each *HV* should be counted. The last step is thresholding. The *HV* with the number of sub-fingerprints' candidates less than a certain threshold (N_{sf}) should be deleted. In practice, most of *HVs* will be deleted and only a small number of the *HVs* (fingerprint candidates) remained.

B. Video Fragment Matching

The second step is the video fragment matching, which uses the results of fingerprint matching step (fingerprint candidates). Each fingerprint candidate is a unique Hash Value, which contains two types of information (Video ID (*VID*) and Video Frame Index (*VFI*)). The "*VID*" points to the video which is corresponding to the fingerprint candidate. The

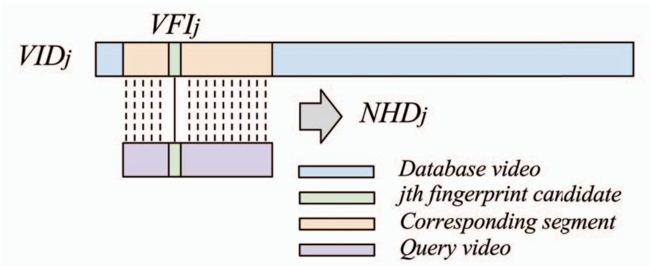

Fig. 3. Basic operations in video fragment matching step using j^{th} fingerprint candidates as an example.

"*VFI*" represents the location of the corresponding fingerprint candidate in the video. This step is similar to the conventional *IIF* based searching strategy, while the main difference is that the system considers all fingerprint candidates rather than stopping when one matching result is found. This provides an advantage that the system can found more than one video matching candidates to improve the possibility of correct detection. Figure 3 shows the basic operations of this step using a fingerprint candidate as an example and the operations applied on other candidates are the same. For this fingerprint candidate, which is the j^{th} fingerprint's candidate of the query. The *VID* of this candidate can locate the corresponding video candidate and the *VFI* can locate the matching video fragment in this video. After localization, the *NHD* between the candidate of video fragment's fingerprint and the query's fingerprint can be obtained. Thus, one fingerprint candidate can get one corresponding *NHD* value. Finally, the candidate with the smallest *NHD* value, and if that value is smaller than the threshold (T_v), is considered as the matching result. If the smallest *HD* value is still larger than the T_v, the result is "Not in database".

III. Experimental Setting and Results

Both the conventional method and the proposed method should be applied on the CBCD system with fingerprint, so the searching strategy with different fingerprinting methods have been implemented and compared. We try three different fingerprinting methods which include aHash [11], pHash [12], and DCT-2ac Hash [10]. To simulate the real case when the video is transmitting, storing, and displaying, we add four common types of distortion (Table I), which include noise, compression loss, brightness changing, and the combination of those types of distortions. In addition, the well-known video database, TRECVID 2011 [21], is used to generate the video database. We download the videos from Internet which include movie/TV series fragments, MV fragments, and advertisement fragments, to build the queries which are outside the database.

TABLE I
Types of video distortion and their corresponding parameters

Types of Distortion	Descriptions
Original	Without any distortion
Gaussian Noise	Mean = 0; Variance = 0.01
JPEG Compression	JPEG compression with 50% Quality
Illuminance change	Illuminance change -25%
Combination	Combine above all

A. Accuracy and Searching Time Comparison

In this subsection, we evaluate the performance of the searching strategy in terms of accuracy and searching speed. We build a 500 videos database with a total length of 14.95 hours. The 500 videos are randomly selected from the TRECVID 2011 video database. For the queries inside the database, we choose 90 videos randomly from the 500 videos database and cut them into clip with fixed length (30s). For the queries which are outside the database, we build 37 queries with 30s length from the videos we download from the Internet. Table II, Table III and Table IV shows the results with "aHash", "pHash", and "DCT-2ac Hash", specifically. The F-score (F_β) [22] is designed for binary classification which is suitable for copy detection system, which is calculated by the function (2), (3), and (4). F-score gives Precision and Recall different weights, which makes F-score flexible for the different situation. The higher F-score means higher accuracy with 1 is the best and 0 is the worst. The searching time is the average searching time which is calculated by function (5), where the *NQ* is the total number of queries and the ST_i is the Searching Time of i^{th} query.

$$F_\beta = (1+\beta^2)\frac{precision \times recall}{\beta^2 precision + recall} \tag{2}$$

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{3}$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{4}$$

$$Searching\ Time = \frac{\sum_{i=1}^{N} ST_i}{NQ} \tag{5}$$

The results show that the proposed searching method can perform slightly better accuracy for three different fingerprinting methods and the proposed searching method can get 10 times faster searching than the conventional IIF searching method. During the experiments, we choose the fixed system setting ($M = 12, N_f = 7, and\ T_V = 16\%$) for different fingerprinting methods. Each fingerprint will be divided into 12 sub-fingerprints and the fingerprint candidates with more than 7 matched sub-fingerprints can be considered as the matched fingerprint. At last, the video segment with *NHD* lower than $T_V = 16\%$ can be considered as the final matching results. For the data we used, this setting can get good performance in searching time and accuracy. During the experiments, we also found that the larger *M* cause higher accuracy but the searching time is longer. The lower N_f can get higher accuracy but the searching time is also longer. These are caused by the number of subfingerprint candidates depend on these two settings. More candidates lead to higher accuracy but longer processing time. From the results, the "aHash" group can get the most accuracy detection while the searching

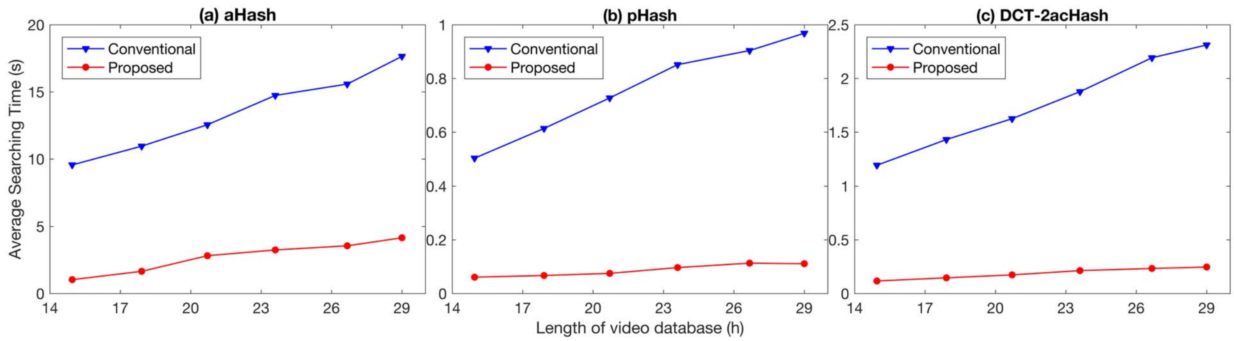

Fig. 4. Searching Time incresing with the video database increasing.

TABLE II
PERFORMANCE EVALUATION RESULTS WITH DIFFERENT DISTORTION (AHASH)

Distortion	Conventional IIF method		Proposed method	
	F-score	Time (s)	F-score	Time (s)
Original	0.995	9.624	0.995	1.222
Gaussian	0.973	10.041	0.991	1.081
JPEG	0.995	9.264	0.995	1.125
Illuminance	0.995	9.436	0.995	1.158
Combination	0.970	9.560	0.988	1.036

TABLE III
PERFORMANCE EVALUATION RESULTS WITH DIFFERENT DISTORTION (PHASH)

Distortion	Conventional IIF method		Proposed method	
	F-score	Time (s)	F-score	Time (s)
Original	0.995	1.223	0.995	0.167
Gaussian	0.988	1.145	0.988	0.120
JPEG	0.993	1.189	0.993	0.158
Illuminance	0.995	1.206	0.995	0.174
Combination	0.986	1.192	0.986	0.117

TABLE IV
PERFORMANCE EVALUATION RESULTS WITH DIFFERENT DISTORTION (DCT-2ACHASH)

Distortion	Conventional IIF method		Proposed method	
	F-score	Time (s)	F-score	Time (s)
Original	0.988	0.451	0.988	0.082
Gaussian	0.976	0.526	0.976	0.062
JPEG	0.978	0.549	0.978	0.078
Illuminance	0.983	0.554	0.983	0.081
Combination	0.970	0.503	0.970	0.061

speed is slower than the other two fingerprint methods. The reason of this is the “aHash” cannot provide a smooth hash key distribution, which introduces more fingerprint candidates. The second step of searching needs more time to deal with all candidates. However, more candidates lead to the higher possibility of finding the right detection.

B. Different Video Database Scale

In this subsection, six different scale video databases (the lengths are 14.95h, 17.91h, 20.71h, 23.6h, 26.67h, and 29h, specifically) are used to evaluate the searching time changing with the increasing of video database scale. It is really hard to simulate the real database scale because it is too large and powerful equipment is needed. However, whether a system can handle the large data scale can be proved through observing the searching time increasing trend with the increasing of data scale. The Figure 4 is the line chart of the searching time changing comparison with “aHash”, “pHash”, and “DCT-2acHash”. It is obviously that the increasing rate of proposed method is smaller than the conventional searching algorithm. Moreover, the “pHash” fingerprinting method with the proposed searching method has the best performance in this case. The increasing rate is the smallest among these three fingerprinting methods. These results indicate that as the video database going to large, the proposed searching method is more suitable with the large video database.

IV. CONCLUSION

In this paper, we proposed a novel IIF based searching method which is used for the CBCD system. The proposed method contains two steps fingerprint matching and video fragment matching. We introduce the number of matched sub-fingerprint to replace the measure of similarity in the first step, which can achieve the faster fingerprint matching. In the second step, the system considers all fingerprint candidates, which are the results of the first step, to finding the video fragments of each fingerprint candidate, rather than stopping at one matched result. This makes the proposed method can found more than one matched results. The proposed searching method is around 10 times faster than the conventional one with the same accuracy. Furthermore, the experimental results show that the proposed method is more suitable for the large-scale video database. In the future, the *NHD* calculation at the video fragment matching also can be replaced. The searching speed can be further increased.

ACKNOWLEDGMENT

The work described in this paper was substantially supported by Internal SRG Funding from City University of Hong Kong [Project No. 7004609].

REFERENCES

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

[2] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[3] Y.-G. Jiang and J. Wang, "Partial copy detection in videos: A benchmark and an evaluation of popular methods," *IEEE Transactions on Big Data*, vol. 2, no. 1, pp. 32–42, 2016.

[4] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang, "Near-duplicate video retrieval: Current research and future trends," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 44, 2013.

[5] F. Yuan, L.-M. Po, M. Liu, X. Xu, W. Jian, K. Wong, and K. W. Cheung, "Shearlet based video fingerprint for content-based copy detection," *Journal of Signal and Information Processing*, vol. 7, no. 02, p. 84, 2016.

[6] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European conference on computer vision*. Springer, 2008, pp. 304–317.

[7] S. Kim, S. H. Lee, and Y. M. Ro, "Rotation and flipping robust region binary patterns for video copy detection," *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 373–383, 2014.

[8] X. Nie, J. Sun, Z. Xing, and X. Liu, "Video fingerprinting based on graph model," *Multimedia tools and applications*, vol. 69, no. 2, pp. 429–442, 2014.

[9] B. Coskun, B. Sankur, and N. Memon, "Spatio–temporal transform based video hashing," *ieee Transactions on Multimedia*, vol. 8, no. 6, pp. 1190–1208, 2006.

[10] M. M. Esmaeili, M. Fatourechi, and R. K. Ward, "A robust and fast video copy detection system using content-based fingerprinting," *IEEE Transactions on information forensics and security*, vol. 6, no. 1, pp. 213–226, 2011.

[11] B. Yang, F. Gu, and X. Niu, "Block mean value based image perceptual hashing," in *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP'06. International Conference on*. IEEE, 2006, pp. 167–172.

[12] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," 2010.

[13] X. Jie, A. Xie, Y. Peng, and S. Jian, "A novel video retrieval algorithm based on coarse-grained and fine-grained," in *Wireless Communications, Networking and Mobile Computing (WiCOM 2015), 11th International Conference on*. IET, 2015, pp. 1–4.

[14] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *International Conference on Advances in Visual Information Systems*. Springer, 2002, pp. 117–128.

[15] J. Mao, G. Xiao, W. Sheng, Y. Hu, and Z. Qu, "A method for video authenticity based on the fingerprint of scene frame," *Neurocomputing*, vol. 173, pp. 2022–2032, 2016.

[16] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1092–1104, 2012.

[17] F. Hao, J. Daugman, and P. Zielinski, "A fast search algorithm for a large fuzzy database," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 203–212, 2008.

[18] J. A. Haitsma, A. A. C. M. Kalker, C. P. M. J. Baggen, and J. C. Oostveen, "Generating and matching hashes of multimedia content," Jun. 16 2009, uS Patent 7,549,052.

[19] M. M. Esmaeili, R. K. Ward, and M. Fatourechi, "A fast approximate nearest neighbor search algorithm in the hamming space," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 12, pp. 2481–2488, 2012.

[20] M. M. Esmaeili and R. K. Ward, "Robust video hashing based on temporally informative representative images," in *Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on*. IEEE, 2010, pp. 179–180.

[21] M. Ayari, J. Delhumeau, M. Douze, H. Jégou, D. Potapov, J. Revaud, C. Schmid, and J. Yuan, "Inria@ trecvid'2011: Copy detection & multimedia event detection," in *TRECVID*, 2011.

[22] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.