VIDEO COPY DETECTION USING FINGERPRINTING WITH FAST IMAGE PROCESSING

Anita Kumari yadav[1], Mr. Pratap Singh Patwal[2]

[1]IET College Alwar, PG Student, Raj. [2]IET College Alwar, Associate Professor, Raj

Abstract: Nowadays, Users are uploading thousands of videos to the internet and every day they are shared. In these uploaded videos, several amounts of videos are manipulated and illegal replicas of present media. So it becomes difficult for copyright management to manage media not to be stealing on internet. In recent times, they have turn into thoughtful concerns for accretive online video sources copyright infractions and data piracy. There are two methods to detect infringements. The First method is centered on watermarking as well as second method is founded on Content Based Copy Detection (CBCD). These disadvantages are overcome in suggested Temporally Informative Representative Images-Discrete Cosine Transform (TIRI-DCT) system.In suggested TIRI-DCT method; two quick searching methods are used for matching of fingerprint. These two approaches are inverted file based similarity search and cluster based similarity search. Such as for a small section of video TIRI holds spatial and temporal. Originated on TIRIs; an effective fingerprinting algorithm is suggested to call as TIRI-DCT and equated with Discrete Cosine Transform (DCT).

Keywords – Cluster-based similarity search, 3D-DCT, Inverted file-based similarity search, Fingerprinting system,TIRI.

1. INTRODUCTION

Nowadays, Users are uploading thousands of videos to the internet and every day they are shared. In these uploaded videos, several amounts of videos are manipulated and illegal replicas of present media. So it becomes difficult for copyright management to manage media not to be stealing on internet. In recent times, they have turn into thoughtful concerns for accretive online video sources copyright infractions and data piracy. Somebody who is not copyright owner duplicates the "appearance" of an effortful work then copyright infringement take place. Then how the idea is expressed is secure when the idea or information behind the work is not safe. The terms piracy and stealing are frequently related to the Copyright infringement. The illegal manufacturing and vending of works in copyright is called Piracy. There are two methods to detect infringements. The First method is centered on watermarking as well as second method is founded on Content Based Copy Detection (CBCD).

The procedure of inserting a definite part of massage called as watermark into interactive media data together with text data, images, audio or videos [1], is called Digital watermarking.CBCD is used to overcome limitations of watermarking. Content Based Copy Detection (CBCD) is a developing and full of life research region because there are numerous enhancements observed in multimedia in addition to communication technology, like acceptance of additional [2]. Multimedia fingerprinting (also known as robust hashing) has been recently proposed for this purpose [1]. A fingerprint is a content-based signature derived from a video (or other form of a multimedia asset) so that it specifically represents the video or asset. To find a copy of a query video in a video database, one can search for a close match of its fingerprint in the corresponding fingerprint database (extracted from the videos in the database). Closeness of two fingerprints represents a similarity between the corresponding videos; two perceptually different videos should have different fingerprints.

1.1 Properties of Fingerprints

Preferably, a scheme of a video fingerprint must have the subsequent features that contain true for a bulky amount of different video content:

Robustness: The robustness of a fingerprint requires that it vary as tiny as possible while the matching video is subjected to content-preserving processes, such for example, transcoding, format change and content editing. The variations that are applied to the video accidentally or purposefully by customers are Content-preserving attacks. These alterations can consist of signal processing operations, added noise, cropping, compression, format changes, rotation, changes in brightness/contrast , logo insertion etc.

Discriminant: The video fingerprints for dissimilar video content should be definitely dissimilar. Briefly two perceptually dissimilar videos have two diverse fingerprints.

Easy to compute: The fingerprint should be easy to compute. There should be capability in fingerprinting algorithm to extract the signatures specifically for online applications.

Compact: An algorithm which demands high computation is not fit for online applications, where simultaneously numerous videos must to be tested to find likely copyright infringements that's why fingerprints must be compact. Searching a match for a video in a DB can convert in a slow process if a fingerprint is not compact. Thus fingerprint must be compact.

Secure: The fingerprinting method should be secured, therefore as to prevent an opponent from damaging it. Security is definitely essential for copy-detection applications.

1.2 Types of Fingerprints

Color-space based fingerprints: This is the first method for extracting fingerprint [3]. Inside the video, Color-space based fingerprints are resulted from colors histograms in time and/or space. Meanwhile when different video formats are used then color features will be different [4], that's why this type of features is not very popular. One more disadvantage of this feature is that they aren't valid for black and white videos. Because of that, maximum video fingerprinting systems put on to the frame's gray scale representation.

Individualities of a video sequence with time are used to take *temporal fingerprints* [5]. For lengthy video clips these features commonly work finely, but don't carry out fine for small video clips because they don't hold enough temporal information. For this reason, small video sequences conquer a big portion of on-line video DB; temporal fingerprints single-handedly are not fit for on-line uses.

Spatial fingerprints are the features those are resulted from every frame or key frame. These are broadly used for both video as well as image fingerprinting. There are enormous bodies of research in image fingerprinting area [6]. Spatial fingerprints are of two types global and local fingerprints. Global fingerprints represent the frame's global features or a section of that frame (e.g., image histograms), whereas local fingerprints frequently signify local information of frame's certain interest points (e.g., corners, edges, etc.).

These interest points are usually used for object recovery purposes. On the other hand, with suitable post-processing also they can be useful for multimedia copy detection. SIFT features are used in these workings [7] which are very robust to distortions and this is effectively used in retrieval process. Robustness is provided by SIFT features to content changing that are not possible with global features. Though, there are number of SIFT features for a small section of a video. For that it is unfeasible for memory management purpose, when it is applied to a big video DB. Additionally, these methods are less attractive for copy detection purposes because they required high amount of post-processing. Consequently, in this work, we use global features, as they don't have the above restrictions.

1.3 Summary

One limitation of spatial fingerprints is that it can't capture the temporal information of a video. *Spatio-temporal fingerprints* hold spatial as well as temporal information of a video then it is expected to do better than other fingerprints because they uses only either spatial or temporal fingerprints. So we have implemented spatio-temporal fingerprints because of their comprehensiveness. Certain spatio-temporal algorithms take video by means of a 3D matrix and extract 3D transform based features. To produce fingerprints, other algorithms use spatio-temporal descriptors of specific interest-points. When a 3D transform apply to a video, it will be a computationally challenging method and may create complications in on-line applications.

In this paper, we also address the search time of our proposed fingerprinting algorithm. We propose two fast search methods. The first is a generalization of [8] and the second is based on a novel approach involving clustering of the fingerprints in the database. We also show that the second method yields superior results. Our final proposed fingerprinting system, i.e., the TIRI-DCT method along with the search algorithm specifically developed for it introduces a fingerprinting system that is robust, discriminant, and fast. Fig-1 shows the overall structure of this fingerprinting system.

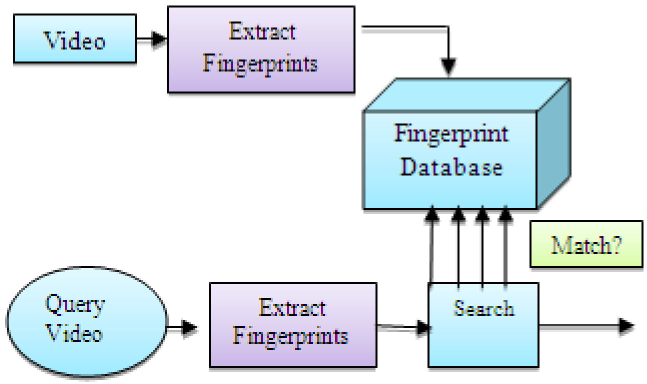


Fig-1 Schematic of a complete fingerprinting system.

2. EXTRACTING ROBUST AND DISCRIMINANT FINGERPRINT

In this portion, we concisely describe a well-known fingerprinting algorithm [9], which will be used such as the base of the evaluations in our simulations. We formerly describe how we create a Temporally Informative Representative Image (TIRI), after that discuss on the key points of our suggested fingerprinting algorithm. First we pre-develop the video signal and then extract the fingerprint. Due to this reason the copies of the similar video but dissimilar frame sizes also with different frame rates generally are there in the similar video DB. Hence a fingerprinting algorithm should be robust for variations in frame size along with frame rate. Robustness can be increased by Down-sampling to these variations of a fingerprinting algorithm. As presented in Fig-2, in time and space at every single video is down-sampled.

To avoid aliasing in both directions a Gaussian smoothing filter is used after down sampling. A fixed size ($W \times H$ pixels) and fixed rate (F frames/second) video signal is generated as result of down-sampling method results.

W =144, H =176 is selected (this is the size of QCIF series broadly used in the video processing), and F =4. The values for this parameter experimentally have been chosen. After finishing the preprocessing process, the down-sampled frames of video are distributed into fixed-length *overlapping* segments, every segment has J frames. After that these segments are process by the fingerprinting algorithm. The amount of overlapping is chosen experimentally and it will be 50%. The sensitivity of fingerprints is decreased by overlapping for time shift attacks.

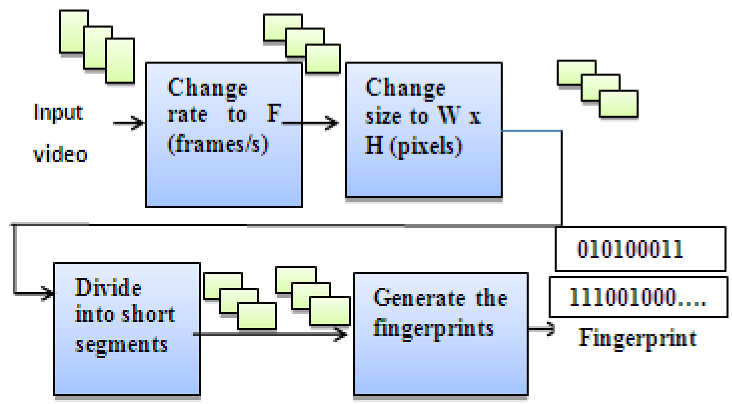


Fig-2: Pre-processing applied to the videos before fingerprinting.

In the following section, we concisely explain the algorithm suggested in [9], mentioned to as the 3D-DCT.

2.1 Spatio-temporal fingerprinting using 3D-DCT

A video is considered as a 3D- matrix of its luminance values by Coskun *et al.*[9]. Subsequently the pre-processing phase is described above; for extracting spatio-temporal features, in this method a three dimensional discrete cosine transform (3D-DCT) is applied to video. After that the thresholding is performed on low frequency coefficients of the transform to derived Binary fingerprints, as shown in Fig-3. The median value of the chosen coefficients is called threshold. Such as an outcome, the same numbers of 0's and 1's is generated by fingerprint (hash). Using this property we can make the maximum no. of different fingerprints that from a binary vector falls from 2^L to $(\frac{L}{L/2})$ of L length. By this property robustness can be increases of the fingerprint, but reduces its discrimination capability. L is responsible for the robustness as well as discrimination of a fingerprint and from the size of the low-frequency block it can be determined. As shown in [9], the 3D-DCT was unaffected from the many kinds of attacks occurred on video signals, added Gaussian noise, brightness/contrast changes, spatial/temporal shift, rotation, and frame loss.

2.2 Generating temporally informative representative images (TIRIs)

This method computes a *weighted average* to produce a *representative image* of the frames. A blurred image is resulted that holds all information about the motions present in a video series. Following is the generation of TIRI:

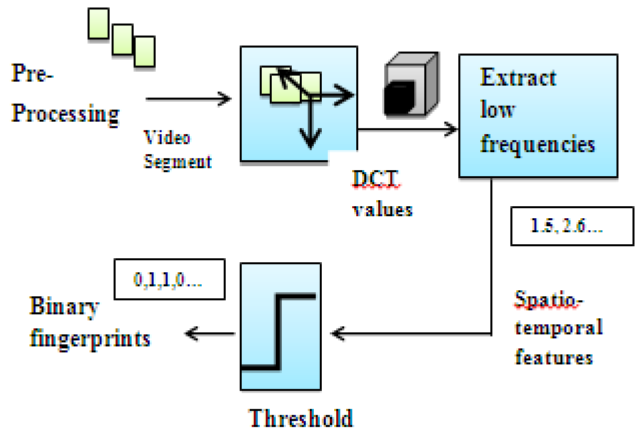
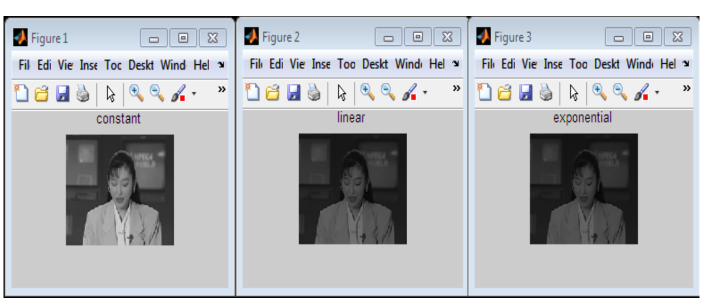


Fig. 3 Schematic of the 3D-DCT algorithm proposed.

let $l_{m,n,k}$ is the luminance value of pixel $(m,n)^{th}$ of the k^{th} frame in a set of J frames. Earlier the pixels of TIRI are found such as:

$$l'_{m,k} = \sum_{k=1}^{J} w_k l_{m,n,k}$$

The different weight factors are tested (constant, linear, and exponential) and we saw that *exponential weighting factor* generate images those finely capture the *motion*. Our resultant TIRIs which is obtained by the three different weighting functions, presented in Fig- 4(a) to (c): constant, linear and exponential. As per talk about earlier, Figure-4 illustrate the exponential weighting function yields perceptually superior results. Therefore for producing TIRIs the exponential weighting function $w_k = \gamma^k$ is preferred.



(a) (b) (c)

Fig-4: Resultant TIRI those have different weighting functions: (d) $w_k = 1$ (constant), (e) $w_k = k$(linear), (f) $w_k = 1.2^k$ (exponential).

2.3 TIRI-DCT Algorithm

1. Generate TIRIs from each segment of J frames after preprocessing of input video. TIRIs are generated using .$\omega_k = \gamma^k$

2. Segment each TIRI into overlapping blocks of size 2ω*2ω, using

$$B^{i,j} = \{l'_{x,y} \mid x \in i\,\omega \pm \omega\ ,\ y \in j\,\omega \pm \omega\}$$

Where i∊{0,1,2,..ω/ω-1} and j∊{0,1,2,...H/ω-1}
When indexes are outside of the boundary then TIRI image is padded with 0's.

3. Extract DCT coefficient from each TIRI block. These are first horizontal and first vertical DCT coefficient. First vertical frequency $\alpha_{i,j}$ can be found for $B^{i,j}$ as

$$\alpha_{i,j} = v^T B^{i,j}$$

Where $v=[\cos(0.5\pi/2\omega),\cos(1.5\pi/2\omega),......\cos(1-0.5\pi/2\omega)]^T$

And 1 is a column vector of all ones. Similarly first horizontal frequency $B^{i,j}$ can be found for $B^{i,j}$ as

$$B^{i,j} = 1^T B^{i,j} v$$

4. Concatenate all coefficients to form feature vector f.

5. Find median m, using all elements of f.

6. Generate binary hash h, using f

$$h_k = \begin{cases} 1, & f_k \geq m \\ 0, & f_k < m \end{cases}$$

Fig-5 TIRI-DCT algorithm

3. FAST MATCHING OF FINGERPRINTS

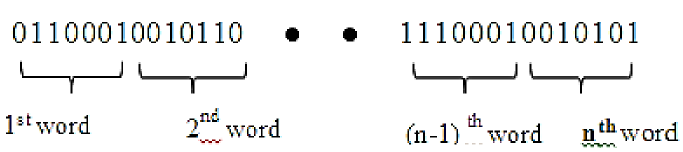
3.1 Inverted file-based similarity search

Formerly the Hamming distance and the query are calculated. If hamming distance of a fingerprint is less than threshold, it will be the match. If the exact match is found, the same process is applied for the fingerprints that have exactly the equal second word. Continuous this process is run until a match is found or last word is tested. At last, till no matched word is found, it means that the requested query doesn't' appropriate to the DB.

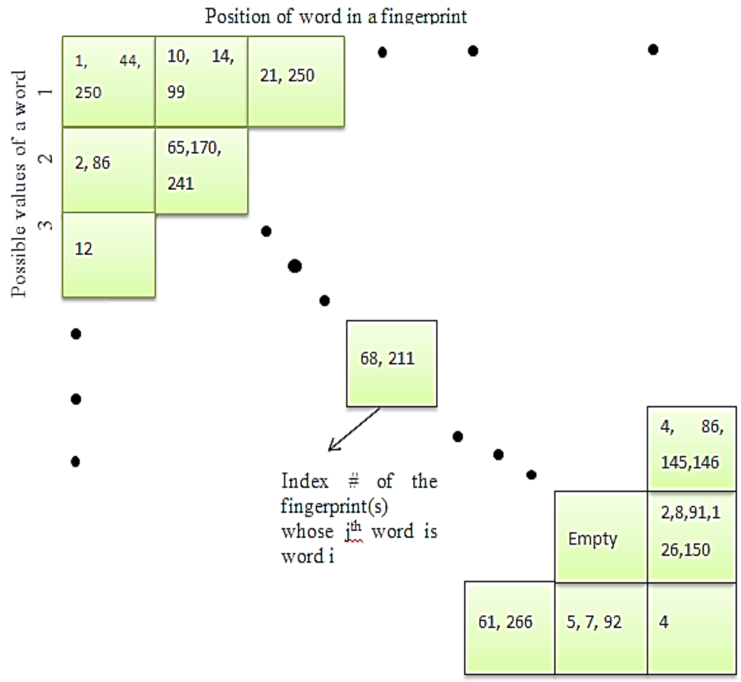
Supposing that there are no couple of fingerprints those are closer in the database comparing with a Hamming distance of th, the algorithm is guaranteed to get the right match, if $th < n(=\frac{L}{m})$. Though, if $th \geq n$, so at that point the algorithm might produce false negatives because there is possibility that a disparity is there inside every word.

Create the table, we should start with the first word of every fingerprint (Fig-6(a)), and the index of the fingerprint should be add to the entry of the first column that will match the value of word. We carry on this procedure for entire words in all fingerprint and all columns in the inverted file table.

An example of inverted file is presented in Fig-6(b). Entry (i,j)shows the list of the indices in table of all fingerprints so that their j^{th} word has the value i. For finding a query fingerprint in DB, the fingerprint should be divided in n words (of m bits). The query is matched with all the fingerprints that begin with the matching word. From the corresponding entry in the first column the indices are getting.



(a)



(b)

Fig-6 (a) Dividing the fingerprint into words. (b) Sample inverted file for the fingerprint database.

3.2 Cluster-Based Similarity Search

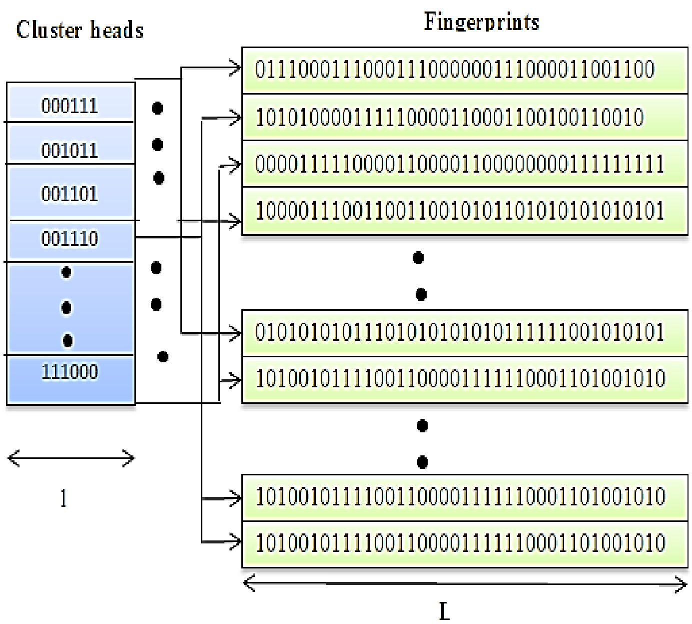
Here, one more similarity search algorithm is proposed for binary fingerprints. The main purpose to use this algorithm is to reduce the no. of requests tested in the DB. Here only single cluster is allocate to each fingerprint (from the K clusters), the fingerprints will be clustered K into non-overlapping groups in the database.

To do accordingly, we select a centroid for every cluster, called the *cluster head*. If a fingerprint is nearby to cluster's head then it will be allocated to cluster k [see Fig-7(a)]. On the way to know that if a request fingerprint matches with a fingerprint then a nearby cluster head should be found.

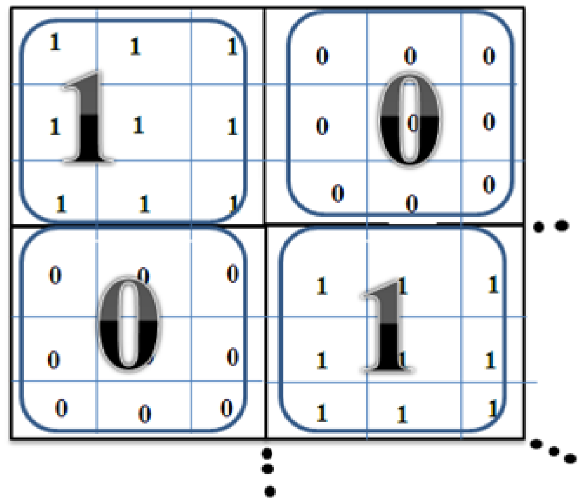
At that time, fingerprints related to this cluster are searched to find out a match, i.e., fingerprint with least Hamming distance $(< th)$.A cluster which is second nearby to the query is tested, if a match is not found. This process execute until a match is not found or the outermost cluster is not tested. At last, it is confirmed that the query is out of DB.

We should select a cluster heads such as a minor variation in fingerprint doesn't affect the fingerprint which is allocated to a different cluster. So, we select cluster heads such that entirely binary vectors have length $l \ll L$. First the fingerprint is separated into words with length $m = L/l$ before a fingerprint is allocating to a cluster. Then in cluster head of *l*-bit, Every word is denoted by a single bit, depending on the majority of bit values; for example, it is denoted by 1, if there are more than/equal to $m/2$ 1's else it is denoted by 0.

As earlier we sated that if the query is not matched in a cluster with any fingerprint, the algorithm keep searching with other clusters which is closer ones. The Aimed of this algorithm is to examine all fingerprints. So not like the inverted-file-based search, it assured to yield a match if present.



(a)



(b)

Fig-7(a) Clustering the fingerprints for TIRI-DCT. (b) Expanding a cluster head to compare it with a fingerprint.

4. SIMULATION RESULTS

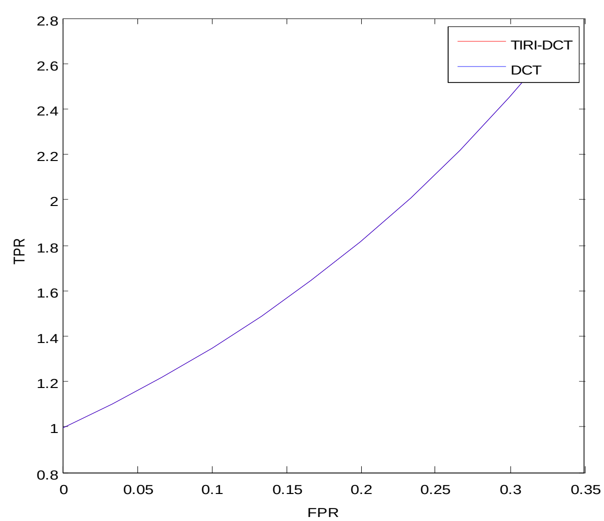


Fig-8: the Receiver Operating Characteristic (ROC) curve for different values of *w* ($4 \leq w \leq 32$) when g = 0.65.

Fig-8 shows the graphical scheme which provides the performance of the binary classifier system. ROC is the curve which is plotting between false positive rate (FPR) and true positive rate (TPR), called ROC curve.

Figure 9(a) it observed that even if the noise is increased from 10 to 70, F-Score value is not randomly decreased. For noise range from 10 to 70 F-Score value is closer to 1 which indicates better performance of the TIRI-DCT system than 3D-DCT.

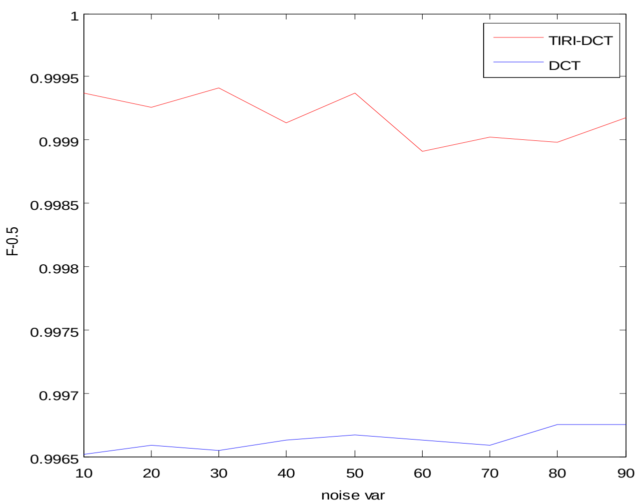


Fig-9(a): F-Score v/s Noise

Fig-9(b) shows that when brightness is increased from -0.6 to 0 then F-Score increases up to 1 and in case of 3D-DCT, F-sore increases later at -0.3. Once F-Score reaches to 1 and brightness of video is again increased then F-Score performance is not degraded. So TIRI-DCT is faster in this case.

Fig-9(c) Indicates that for Contrast Range from 0.2 to 2 F-Score is nearly remains constant for both TIRI-DCT and 3D-DCT but F-Score value is closer to 1 for TIRI-DCT. So TIRI-DCT is better.

From Fig-9(d) it observed that if video frames are not rotated then both methods maintain very good performance in terms of the F - Score. But if frames are rotated in negative or positive degree then performance degrades slightly but in 3D-DCT, immediately performance degrades which is less than TIRI-DCT in terms of F-score.

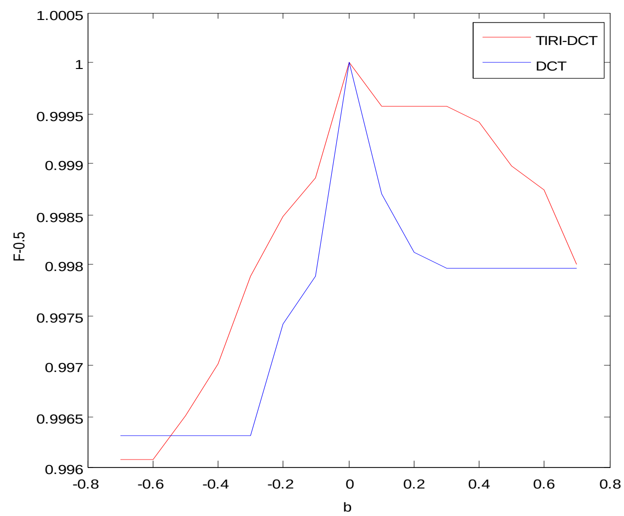


Fig-9(b): F-Score v/s Brightness

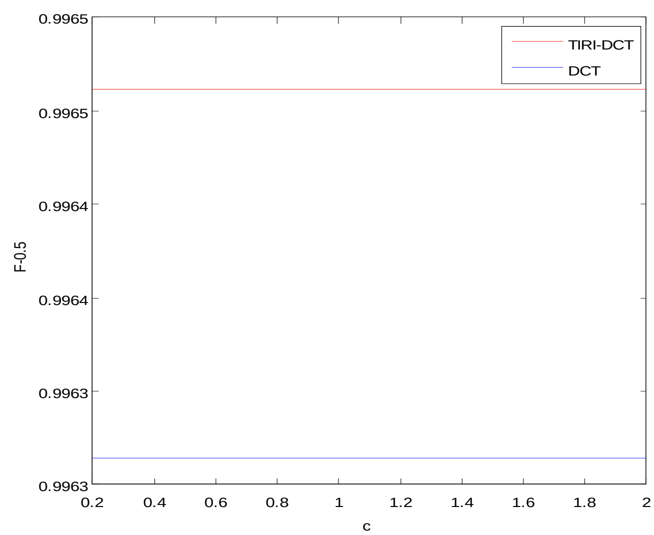


Fig-9(c): F-Score v/s Contrast

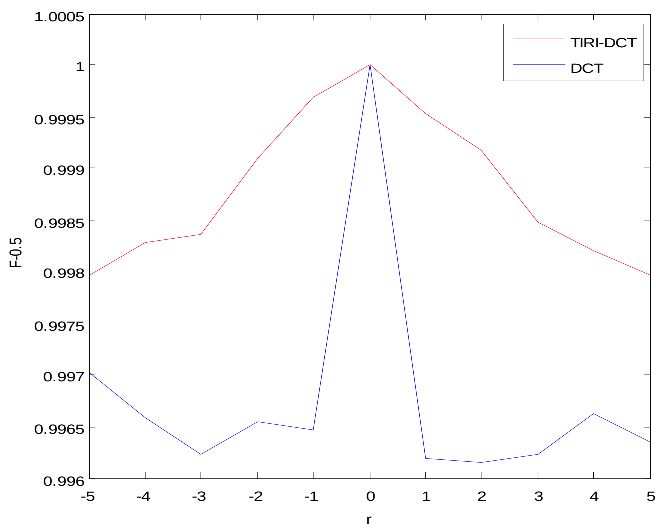


Fig-9(d): F-Score v/s Rotation

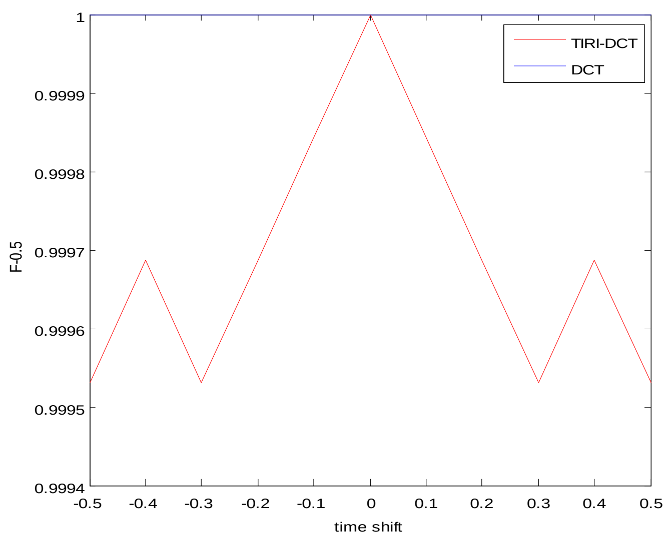


Fig-9(e): F-Score v/s Time Shift

Fig-9(e) shows that if the video is shifted by some second from 0 to 0.5 seconds or from 0 to −0.5 seconds then F-Score value is slowly decreased but still it is closer to 1, representing good performance of TIRI-DCT. If video is not shifted in time, means the starting of query video is precisely aligned with starting of reference video then F-Score value reaches to 1.But there is no effect of time shift on 3D-DCT method. it remains at F-sore 1 and give better performance than TIRI-DCT.

Fig-9(f) concludes that if a video frame is shifted by -4 (%) to 4 (%) right and-4 (%) to 4 (%) down then 3D-Dct maintain good performance than TIRI-DCT.

Fig-9(g) shows that both methods maintain good performance for up and down shift and give F-score 1 which indicate that if video frames are shifted up/down then in this case both methods perform best.

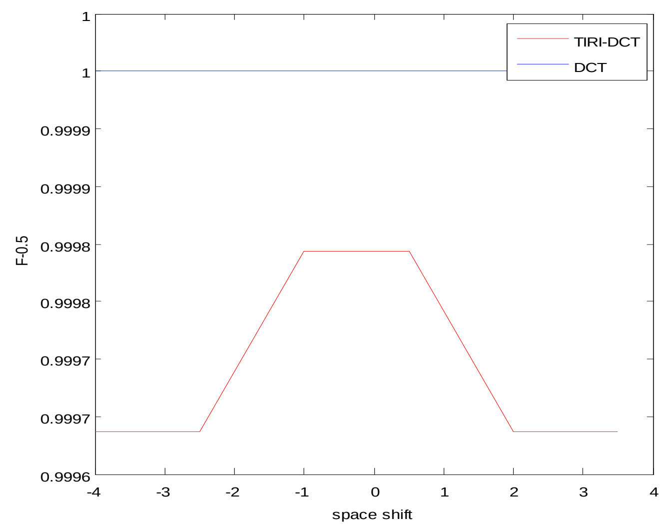


Fig-9(f): F-Score v/s Space Shift

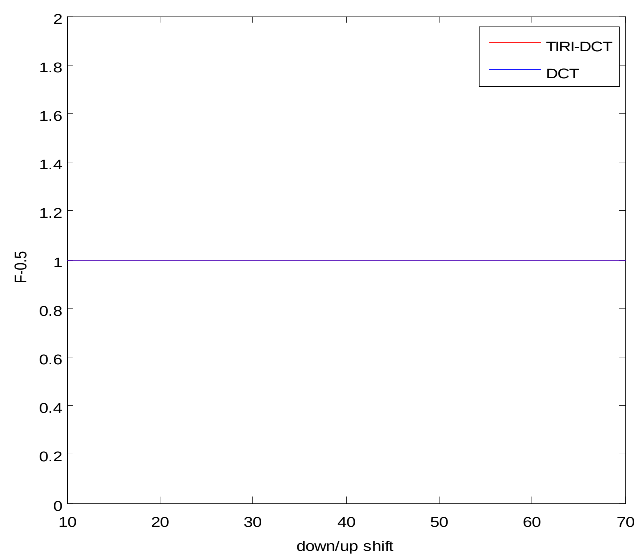


Fig-9(g): F-Score v/s Down Shift

Fig-9(a)-(g) F-score of TIRI-DCT and 3D-DCt for different attack parameters: (a) Noise addition; (b) brightness; (c) Contrast (d) rotation; (e) time shift; (f) space shift; (g) down Shift.

Figure 6.9 Show the compromise b/w system's performance and security which is represented by TPR, FPR and F-score. The bars in chart indicate the amount of variations in F-score, TPR and FPR. As we see that TPR decreases by 29% FPR decreases by 44% and F-score increases by 28%. So this is understood by the results that security can be attained with small reduction in performance if lager fingerprints are used. But it decreases the speed of detection because when hamming distance is calculated it requires more computation. So overall both algorithms are robust to noise addition, changes in brightness/contrast, and rotation with high F-scores.

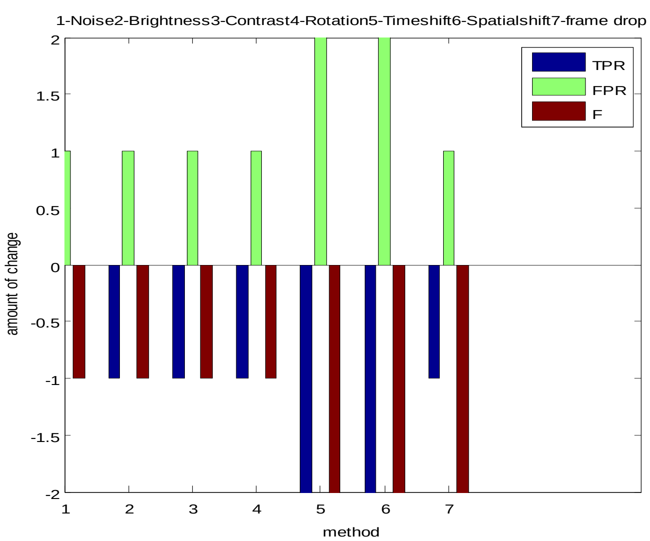


Figure 6.9 Amount of change in the average TPR, FPR, and F0.5 when the security is added to the system.

1. CONCLUSION

This paper proposes a fingerprinting system for video copy detection. It can be used for copyright management and indexing applications. The system consists of a fingerprint extraction algorithm followed by an approximate search method. The proposed fingerprinting algorithm (TIRI-DCT) extracts robust, discriminant, and compact fingerprints from videos in a fast and reliable fashion. These fingerprints are extracted from TIRIs containing both spatial and temporal information about a video segment. We demonstrate that TIRI-DCT generally outperforms the well-established (3D-DCT) algorithm and maintains a good performance for different attacks on video signals, including noise addition, changes in brightness/contrast, rotation, spatial/temporal shift, and frame loss. It is shown experimentally that TIRI-DCT has a high average true positive and a low average false positive rate

We also propose two fast approximate search algorithms. The cluster-based method is experimentally shown to be superior to the inverted-file-based method in terms of the detection performance and the query retrieval time. We have thus adopted the cluster-based method as the search engine of our fingerprinting system.

REFERENCES

[1] P. Chomphoosang, P. Zhang ; A. Durresi ; L. Barolli," Survey of Trust Based Communications in Social Networks", Network-Based Information Systems (NBiS), 2011 14th International Conference on 7-9 Sept. 2011.

[2] Ahmet Saracoglu. "Content Based Copy Detection with Coarse Audio-Visual Fingerprints",2009 Seventh International Workshop on Content-Based Multimedia Indexing, 06/2009.

[3] Jian Lu. Video fingerprinting for copy identification: from research to industry applications. volume 7254, page 725402. SPIE, 2009.

[4] Arun Hampapur, Rudolf M. Bolle. Videogrep: Video copy detection using inverted file indices. Technical report, IBM Research Division Thomas.J.Watson Research Center, 2001.

[5] Li Chen and F. W. M. Stentiford. Video sequence matching based on temporal ordinal measurement. Pattern Recogn. Lett., 29(13):1824–1831,2008. ISSN 0167-8655.

[6] Sunil Lee and C.D. Yoo," Robust video fingerprinting for content-based video identification". *IEEE Transactions on Circuits and Systems for Video Technology*, 18(7):983–988, July 2008. ISSN 1051-8215.

[7] David G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91–110, 2004.

[8] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Proc. Int. Conf. Recent Advances in Visual Information Systems (VISUAL)*, London, U.K., 2002 pp. 117–128, Springer-Verlag.

[9] B. Coskun, B. Sankur, and N. Memon. Spatiotemporal transform based video hashing. IEEE Transactions on Multimedia, 8(6):1190–1208, Dec.2006.

[10] Mani Malek Esmaeili, Mehrdad Fatourechi, and Rabab Kreidieh Ward," A Robust and Fast Video Copy Detection System Using Content-Based Fingerprinting", ieee transactions on information forensics and security, vol. 6, no. 1, march 2011.

[11] Job Oostveen, Ton Kalker, and Jaap Haitsma. Feature extraction and a database strategy for video fingerprinting. In *International Conference on Recent Advances in Visual Information Systems (VISUAL)*, pages 117–128, London, UK, 2002. Springer-Verlag.