Northumbria Research Link

Citation: Khelifi, Fouad and Bouridane, Ahmed (2017) Perceptual Video Hashing for Content Identification and Authentication. IEEE Transactions on Circuits and Systems for Video Technology. ISSN 1051-8215 (In Press)

Published by: IEEE

URL: https://doi.org/10.1109/TCSVT.2017.2776159 <https://doi.org/10.1109/TCSVT.2017.2776159>

This version was downloaded from Northumbria Research Link: http://nrl.northumbria.ac.uk/32873/

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: http://nrl.northumbria.ac.uk/policies.html

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)

www.northumbria.ac.uk/nrl

northumbria UNIVERSITY NEWCASTLE

Perceptual Video Hashing for Content Identification and Authentication

Fouad Khelifi, *Member, IEEE*, and Ahmed Bouridane, *Senior Member, IEEE*

***Abstract*—Perceptual hashing has been broadly used in the literature to identify similar contents for video copy detection. It has also been adopted to detect malicious manipulations for video authentication. However, targeting both applications with a single system using the same hash would be highly desirable as this saves the storage space and reduces the computational complexity. This paper proposes a perceptual video hashing system for content identification and authentication. The objective is to design a hash extraction technique that can withstand signal processing operations on one hand and detect malicious attacks on the other hand. The proposed system relies on a new signal calibration technique for extracting the hash using the discrete cosine transform (DCT) and the discrete sine transform (DST). This consists of determining the number of samples, called the normalizing shift, that is required for shifting a digital signal so that the shifted version matches a certain pattern according to DCT/DST coefficients. The rationale for the calibration idea is that the normalizing shift resists signal processing operations while it exhibits sensitivity to local tampering (i.e., replacing a small portion of the signal with a different one). While the same hash serves both applications, two different similarity measures have been proposed for video identification and authentication, respectively. Through intensive experiments with various types of video distortions and manipulations, the proposed system has been shown to outperform related state-of-the art video hashing techniques in terms of identification and authentication with the advantageous ability to locate tampered regions.**

***Index Terms*—Video hashing, Robustness, Identification, Authentication, Forgery detection.**

I. Introduction

THE field of perceptual image and video hashing (also referred to as fingerprinting) has witnessed an impressive growth over the last decade. This is mainly attributed to the increasing amount of visual data being easily conveyed, broadcast or browsed via digital devices. Perceptual hashing has emerged as an effective way to verify the authenticity of digital data [1][2] and this keeps attracting developers with interest in monitoring multimedia websites and detecting copied or pirated videos over the internet [3]. In addition to security related applications, perceptual hashing also finds applications in image registration and retrieval [4][5]. This paper is concerned with perceptual video hashing where the design requires that two completely different videos provide uncorrelated hashes while two visually similar videos give highly correlated hashes. It is meant here by two visually similar videos that one video is derived from another via commonly used video processing operations including low pass filtering, lossy compression, noise addition, contrast increasing/decreasing, minor geometric alterations and temporal distortions. It is however worth noting that in the case of video authentication, which is also considered in this paper, the hash should be sensitive to forgery and malicious manipulations [6].

In the literature, there has been a growing body of research on robust video hashing. Oostveen *et al.* have used the spatiotemporal Haar filters on block means to extract the hash for video identification [7]. The same authors have proposed a video fingerprinting technique which extracts the hash from differential luminance block means in both the spatial and temporal directions [8]. In [9], an image hashing technique based on radial projections has been proposed. It has then been extended to video data where the hash is extracted from key-frames. Coskun *et al.* have proposed two 3-D transform-based video hashing techniques [10]. The authors have investigated the randomness and robustness of the proposed techniques through experimental analysis and have shown that 3-D DCT-based video hashing is more robust when compared to video hashing based on the 3-D random bases transform. However, this comes at the expense of lower security. In [11], a robust video fingerprinting scheme has been proposed for video identification. The fingerprint is extracted from each frame by using the centroid of gradient orientations computed from non-overlapping blocks. An improved version of the technique using the orientation of luminance centroids has been proposed in [12]. Key frames have also been used in [13] to extract robust features for duplicate and similar video copy detection. Speeded up robust feature points have been adopted in [14] and [15] for video fingerprinting. In another related work [16], the Hessian-Affine region detector and the SIFT descriptor have been used to extract robust features from the key frames of the video. In [17], the problem of detecting a query video segment in a database under different spatio-temporal variations is formulated as a partial matching problem in a probabilistic model. In [18], a feature selection algorithm called Symmetric Pairwise Boosting (SPB) has been proposed for robust fingerprinting. It mainly selects appropriate filters and quantifiers from a class of candidate filters and quantifiers in such a way that perceptually similar pairs of video clips are correctly classified. Xu *et al.* have proposed a video copy detection scheme where the selected low and middle frequency DCT-coefficients of each key-frame are used as a signature [19]. More recently, Esmaeili *et al.* have formed temporally informative representative images from the video sequence, referred to as *TIRIs*, in order to extract a binary fingerprint from the features that can be obtained in the DCT domain of the overlapping blocks of the *TIRI* frames [20]. In [21], an idea for generating weights for a given

Fouad Khelifi and Ahmed Bouridane are with the Department of Computer and Information Sciences, Northumbria University, UK. (e-mail: fouad.khelifi@northumbria.ac.uk; ahmed.bouridane@northumbria.ac.uk).

hash based on visual saliency has been proposed for efficient matching. Li and Monga have used multi-linear subspace projections with a reduced rank factorization to extract the fingerprint as a summarized version of the video [22]. In a more recent paper, the authors have proposed to use structural graphical models to encode the temporal evolution of the video content [23]. The real-valued hash is then projected onto a randomly generated space whose components are drawn $i.i.d.$ from a normal $N(0,1)$ distribution. Finally, a 1-bit adaptive quantizer [24] has been adopted to obtain the final hash in the binary form. The technique has been shown to outperform recent hashing systems when the bit budget of the fingerprint is low. In [25] the authors have presented a solution to the problem of temporal de-synchronization which occurs when the positions of deleted and/or inserted frames in a video are unknown. In [26], sparse coding is adopted to represent non-overlapping blocks in each frame of the normalized video where the matching pursuit decomposition method is used to extract edge and texture features. However, because of the large features size, the authors applied the SVD in two stages to reduce the feature space dimensionality and obtain the fingerprint. In [27], each set of frames, determined by a central key frame, is clustered into two categories depending on their temporal relationships with the central key frame. The grouped surrounding frames are then used to generate a binary code describing the temporal context of the key frame. In [28], the authors substituted the orientation gradient in the Weber Local Descriptor (WLD) by a local textural descriptor, namely Binarized Statistical Image Features (BSIF) to create a histogram-based fingerprint of the video. The technique, which was referred to as Weber Binarized Statistical Image Features (WBSIF), has been shown to outperform other textural features such as WLD, the Local Binary Patterns (LBP), and the Local Phase Quantization (LPQ). More recently, the authors in [29] presented a framework in which the video is viewed as a high-order tensor consisting of different features. Then, a comprehensive feature that results from fusing the video features is constructed via the Tucker model to form the video fingerprint.

Since the common approach for digital content authentication consists of watermarking a signature at one end and analyzing the retrieved signature at the other end, little research has been devoted to video authentication with perceptual hashing. The aforementioned systems are particulary suitable for content-based video identification and video copy detection. In fact, the design of such systems is inspired by the idea of representing the input video by a short data string which makes it difficult to detect small object insertions/removals because such video distortions cause the same effect on the hash as other tolerated changes do (i.e., compression, noise, filtering, etc.). This has motivated researchers to develop hashing-based video authentication systems with the primary aim to detect and locate malicious attacks. For example, the video fingerprinting system proposed in [30] is meant to authenticate MPEG-4 surveillance videos. Su *et al.* have proposed in [6] a video authentication scheme sensitive to malicious visual changes and robust to H.264 video compression. To generate the authentication code, the authors adopted a vector quantization method to encode textured blocks and a scalar quantization method to encode uniform blocks for each frame. In [31], a combination of robust fingerprinting and cryptographic hashing has been adopted. The proposed video authentication system has been shown to withstand transcoding and transrating operations. However, the evaluation of the system's performance on maliciously manipulated videos was not considered. More recently, Kroputaponchai and Suvonvorn [32] proposed an authentication scheme based on a two-dimensional (2D) version of the Histogram of Gradient (HOG) by further considering the temporal dimension as an extension of the conventional HOG. This was, however, tested on a few videos only.

To the best of our knowledge, there has not been any established research conducted on perceptual video hashing to target both the applications of content identification and authentication with a single system. We acknowledge a related work on still images using feature points [33] where the system has been shown to deliver better authentication results when compared to transform-based hashing techniques (DWT and DCT). The identification results, however, have been outperformed by other techniques, including the wavelet-based hashing technique, as demonstrated in [34] on attacked images. Furthermore, the system assumes that the tampered regions provide a mismatch of several extracted feature points. However, if the tampering process consists of just replacing a smooth image region with another[1], the feature-point detector may not find any points and therefore such manipulations cannot be detected. Our objective here is to design a hashing system that can withstand signal processing operations and small geometric distortions on one hand, and detect and locate malicious attacks on the other hand. Here, by malicious attacks it is meant manipulations that aim to alter the semantic content of the video via object insertion and/or removal. The authentication process should however tolerate transcoding operations such as lossy compression (transrating) and frame resolution change (transsizing).

This paper develops a robust video hashing system for content identification and authentication. We propose a new hash extraction technique based on a signal calibration idea using the discrete cosine transform (DCT) and the discrete sine transform (DST). The idea consists of determining the number of samples, called the normalizing shift, that is required for shifting a digital signal so that the shifted version matches a certain pattern according to DCT/DST coefficients. The reason behind the calibration idea is that the normalizing shift does not vary significantly under signal processing operations such as filtering, compression and noise addition while it exhibits sensitivity to local tampering (i.e., replacing a small portion of the signal with a different one). The contributions of this work can be summarized as follows. (i) Unlike the traditional approach where hashing systems are designed to target a specific application, our system serves in both video content identification and authentication by using the same hash. (ii) A new shift-based signal calibration technique upon which

[1]The feature point-based technique extracts the 64 most robust features representing corners and edges.

the hash extraction stage is based. As will be illustrated, the proposed hash exhibits robustness against signal processing attacks on one hand and sensitivity to malicious manipulations on the other hand. These two aspects make the system suitable for video content identification and authentication. It is worth noting here that the proposed calibration idea can also be used in other applications such as image registration and signal alignment. (iii) A new similarity measure for the proposed hash-based video content identification. This enables the system to overcome the issue of synchronization caused by temporal distortions. (iv) A new segment-based video authentication measure. This exploits the temporal redundancy of malicious manipulations and enables the system to differentiate transcoded videos from forged ones. (v) The proposed video hashing system has been shown to outperform recent state-of-the-art techniques specifically designed for video content identification. On the other hand, the superiority of the system over related work in authenticating videos and detecting forgeries has been demonstrated.

The rest of the paper is structured as follows. In section II, the problem of video identification and authentication is described. Section III describes the proposed video hashing scheme. Section IV discusses a matching methodology adopted for identification and authentication. Section V provides an experimental evaluation of the system in comparison with recent and related techniques. Section VI summarizes and concludes the paper.

II. Problem formulation

Let Υ_i and Υ_j be two digital videos, respectively. Denote by Ω the hash function that maps the video Υ_i to a hash h_i, i.e., $h_i = \Omega(\Upsilon_i)$. For content-based video identification, the following requirements are normally considered in the literature.

(i) $\forall\ \Upsilon_i, \Upsilon_j$; if $\Upsilon_i \wr \Upsilon_j$ then $D(h_i, h_j) \geq T_{id}$
(ii) $\forall\ \Upsilon_i, \Upsilon_j$; if $\Upsilon_i \approx \Upsilon_j$ then $D(h_i, h_j) < T_{id}$

where $\wr$ stands for visually different. On one hand, the first requirement suggests that the distance D between two hashes corresponding to any two completely different videos Υ_i and Υ_j should be larger than a certain identification threshold T_{id}. This basically ensures the capability of differentiating between two videos that are distinct visually. On the other hand, *(ii)* ensures that two visually similar videos produce close hashes h_i and h_j where the identification distance is less than T_{id}.

For video authentication, let us denote by $\tilde{\Upsilon}_i$ a transcoded version of Υ_i whereas $\bar{\Upsilon}_i$ represents its forged version. It is meant here by video forgery the process of locally inserting or removing an object from frames as well as the substitution of a number of frames in the video by different ones. Hashing-based authentication systems consider the following requirements.

(iii) $\forall\ \Upsilon_i\ ;\ D(h_i, \bar{h}_i) \geq T_{auth}$
(iv) $\forall\ \Upsilon_i\ ;\ D(h_i, \tilde{h}_i) < T_{auth}$

In *(iii)*, the distance between two hashes corresponding to a video Υ_i and its forged version $\bar{\Upsilon}_i$ should be larger than a certain authentication threshold T_{auth}. This guarantees the detection of forgeries and malicious manipulations. As for *(iv)*, it ensures the robustness of the system against transcoding operations. It is worth mentioning here that both *(ii)* and *(iv)* are in favor of the robustness property. Practically, most existing hashing-based video identification schemes meet *(i)* and *(ii)* to some extent whereas hashing-based authentication schemes meet *(iii)* and *(iv)*. The main challenge, however, resides in meeting all the requirements simultaneously because, visually speaking, forged videos look similar to the original ones and this conflicts with *(ii)*. To overcome this issue, our work relies on two main and complementary contributions. The first contribution consists of a new hash function that is robust against signal processing operations on one hand and sensitive to malicious attacks on the other hand. In the second contribution, the identification distance is made different from the authentication distance to take into account the difference between the concept of dissimilarity in identification and that in authentication. The proposed requirements become as follows.

(i) $\forall\ \Upsilon_i, \Upsilon_j$; if $\Upsilon_i \wr \Upsilon_j$ then $D_{id}(h_i, h_j) \geq T_{id}$
(ii) $\forall\ \Upsilon_i, \Upsilon_j$; if $\Upsilon_i \approx \Upsilon_j$ then $D_{id}(h_i, h_j) < T_{id}$
(iii) $\forall\ \Upsilon_i\ ;\ D_{auth}(h_i, \bar{h}_i) \geq T_{auth}$
(iv) $\forall\ \Upsilon_i\ ;\ D_{auth}(h_i, \tilde{h}_i) < T_{auth}$

where D_{id} and D_{auth} represent the identification and authentication distances, respectively. Finally, it is worth noting that learning-based hashing, which has been widely used in image retrieval and object tracking applications [35][36][37], differs from the hashing considered here in terms of design requirements and objectives.

III. Proposed video hashing scheme

The proposed system for generating a video hash is composed of three main stages as illustrated by Fig 1. First, the input video is pre-processed to reduce the effect of temporal distortions and color changes. The differences of luminance block means are then computed to describe the video information in each direction. Finally, a new shift-based signal calibration technique is used to obtain the final hash. Note that two similarity measures are introduced for content identification and authentication.

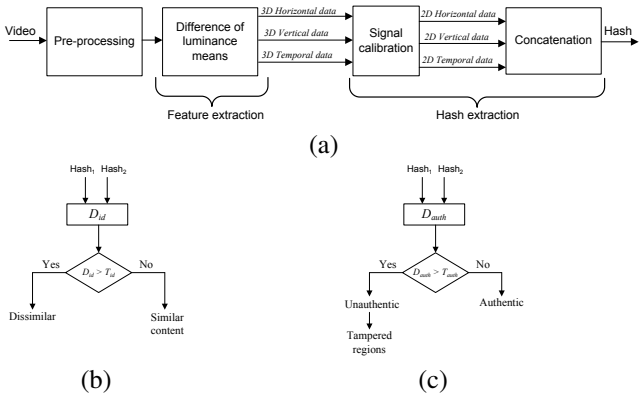


Fig. 1. Proposed Video hashing scheme for content identification and authentication. (a) Hash generation. (b) Identification. (c) Authentication.

A. Pre-processing

To eliminate sensitivity to color manipulations, the luminance component is extracted from the input video. One of the straightforward pre-processing techniques to reduce the effect of temporal distortions consists of re-sampling the video at a small frame rate [10][12][20]. In the proposed system, the video is re-sampled at 5 frames per second. However, if the video undergoes significant temporal translations, this might not be sufficient and, hence, a new similarity measure is presented later.

B. Feature extraction

In order to extract robust features, the idea of extracting Differential Luminance Block Means (DLBM) in the spatial domain is borrowed from [8]. However, our technique differs from [8] in that DLBM are also extracted in vertical and temporal directions. The use of these features is justified by their efficiency in representing textured areas including edges and contours at low computational cost. Indeed, the derivation of DLBM can be thought of as a process of calculating the gradient of a down-scaled version of the original image. It is worth noting that this paper is concerned with the perceptual hashing of short clips (clips of a few seconds long) and hence the hashing of long videos can just be an extension of this work with a further consideration of optimized similarity search techniques in high dimensional databases. As this is beyond the scope of this paper, the reader can be acquainted with more details by referring to [8][20][38].

First, a three-dimensional (3-D) array is formed by computing the mean of non-overlapping blocks in each frame. The size of the blocks is set so that each frame is split into $M \times N$ blocks. Denote by $A(i,j,k)$ the obtained array with $0 \leq i \leq M-1$, $0 \leq j \leq N-1$, and $0 \leq k \leq K-1$ where K is the number of frames in the pre-processed video. Next, three (3-D) arrays of the same size are derived from A by calculating the differences in the horizontal (*H*), vertical (*V*) and temporal (*T*) directions, respectively as

$$H(i,j,k) = \begin{cases} A(i,j+1,k) - A(i,j,k) & if \quad j < N-1 \\ A(i,0,k) - A(i,j,k) & if \quad j = N-1\,. \end{cases} \tag{1}$$

$$V(i,j,k) = \begin{cases} A(i+1,j,k) - A(i,j,k) & if \quad i < M-1 \\ A(0,j,k) - A(i,j,k) & if \quad i = M-1\,. \end{cases} \tag{2}$$

$$T(i,j,k) = \begin{cases} A(i,j,k+1) - A(i,j,k) & if \quad k < K-1 \\ A(i,j,0) - A(i,j,k) & if \quad k = K-1\,. \end{cases} \tag{3}$$

C. Hash extraction

Once the horizontal, vertical and temporal features are computed as described in the previous section, a new signal calibration technique is used to extract the hash. We first propose a shift invariant signal normalization method upon which the calibration idea is based.

1) DCT/DST-based Signal Normalization: Denote by $x(n)$ with $n = 0, 1, \cdots, L-1$ a signal obtained by traversing one of the previous 3-D arrays in one direction (as will be explained in subsection III-C2). Let us first define the family of transforms which will be used in this work. The DCT of $x(n)$ is given by

$$\begin{aligned} X^C(m) &= \sum_{n=0}^{L-1} x(n) \cos\left(\frac{\pi m (n+\frac{1}{2})}{L}\right); \\ m,n &= 0, 1, \cdots, L-1. \end{aligned} \tag{4}$$

The DST of $x(n)$ is expressed as

$$\begin{aligned} X^S(m) &= \sum_{n=0}^{L-1} x(n) \sin\left(\frac{\pi (m+1) (n+\frac{1}{2})}{L}\right); \\ m,n &= 0, 1, \cdots, L-1. \end{aligned} \tag{5}$$

We first propose a normalization technique which provides the same sequence even if the input sequence has undergone a circular translation. Consider the sequence of samples

$$x_0 = \{x(0), x(1), \cdots, x(L-1)\}, \tag{6}$$

and its shifted version by one sample

$$x_1 = \{x(1), x(2), \cdots, x(L-1), x(0)\}. \tag{7}$$

It can be shown that [39]

$$\begin{aligned} X_1^C(m) &= \cos\left(\frac{\pi m}{L}\right) X_0^C(m) \\ &\quad + \sin\left(\frac{\pi m}{L}\right) X_0^S(m-1) \\ &\quad + x(0) \cos\left(\frac{\pi m}{2L}\right) ((-1)^m - 1)\,, \end{aligned} \tag{8}$$

and

$$\begin{aligned} X_1^S(m-1) &= -\sin\left(\frac{\pi m}{L}\right) X_0^C(m) \\ &\quad + \cos\left(\frac{\pi m}{L}\right) X_0^S(m-1) \\ &\quad + x(0) \sin\left(\frac{\pi m}{2L}\right) (1 - (-1)^m)\,. \end{aligned} \tag{9}$$

In this work, only even values of m ($m = 2, 4, 6, \cdots, 2p$) are considered[2]. It follows

$$\begin{aligned} X_1^C(m) &= \cos\left(\frac{\pi m}{L}\right) X_0^C(m) \\ &\quad + \sin\left(\frac{\pi m}{L}\right) X_0^S(m-1), \end{aligned} \tag{10}$$

and

$$\begin{aligned} X_1^S(m-1) &= -\sin\left(\frac{\pi m}{L}\right) X_0^C(m) \\ &\quad + \cos\left(\frac{\pi m}{L}\right) X_0^S(m-1). \end{aligned} \tag{11}$$

Hence, the DCT and DST coefficients of a shifted sequence by one sample can be expressed as a function of the DCT and DST coefficients of the original sequence. Let $w = \frac{\pi m}{L}$. For $i = 0, 1, \cdots, L-1$, we obtain the following recursive equations

$$X_{i+1}^C(m) = \cos(w)\, X_i^C(m) + \sin(w)\, X_i^S(m-1) \tag{12}$$

and

$$X_{i+1}^S(m-1) = -\sin(w)\, X_i^C(m) + \cos(w)\, X_i^S(m-1) \tag{13}$$

[2]Note that odd values of m cannot lead to the findings of this work.

Observe that

$$\begin{aligned}\left(X_0^C(m)\right)^2+\left(X_0^S(m-1)\right)^2 &= \left(X_1^C(m)\right)^2+\left(X_1^S(m-1)\right)^2\\ \cdots &= \left(X_{L-1}^C(m)\right)^2+\left(X_{L-1}^S(m-1)\right)^2 .\end{aligned} \quad (14)$$

This describes the shift invariance property of the magnitude of the Discrete Fourier Transform (DFT). In view of (12) and (13), it can be shown that the DCT and DST coefficients of a shifted sequence by i samples can be expressed using the DCT and DST coefficients of the original sequence (see Appendix A) as

$$\begin{aligned}X_i^C(m) &= \sqrt{(X_0^C(m))^2+(X_0^S(m-1))^2}\\ &\quad\times\cos\left(w\ i-\arctan\left(\frac{X_0^S(m-1)}{X_0^C(m)}\right)\right),\end{aligned} \quad (15)$$

and

$$\begin{aligned}X_i^S(m-1) &= \sqrt{(X_0^C(m))^2+(X_0^S(m-1))^2}\\ &\quad\times\cos\left(w\ i-\arctan\left(-\frac{X_0^C(m)}{X_0^S(m-1)}\right)\right)\\ &= \sqrt{(X_0^C(m))^2+(X_0^S(m-1))^2}\\ &\quad\times\cos\left(w\ i-\arctan\left(\frac{X_0^S(m-1)}{X_0^C(m)}\right)+\frac{\pi}{2}\right).\end{aligned} \quad (16)$$

That is, the DCT and DST coefficients of repetitively shifted versions of x_0 follow a cosine function with the same magnitude. From (15) and (16), one can deduce that the DCT and DST coefficients of a shifted sequence reappear at a shifting rate equal to $\frac{2L}{m}$. For $m=2$, there is only one period of the cosine function against the variable i in $[0, L-1]$ which means that the pair of coefficients $(X_i^C(2), X_i^S(1))$ occurs only once in $[0, L-1]$. In the rest of the paper, m is set to 2. The proposed normalization idea consists of determining an amount of samples by which the signal can be shifted to provide the same DCT/DST coefficients. To elaborate more, the problem is described as follows. There are L possible sequences shifted from each other by one sample where each sequence corresponds to a unique pair of coefficients $(X_i^C(2), X_i^S(1))$. Given a pair of coefficients $(X_{i^*}^C(2), X_{i^*}^S(1))$, the problem can simply be thought of as finding the corresponding sequence which is referred to as the normalized one. In view of (15), this can be obtained by using a reference angle α in $[0, 2\pi[$ so that the normalizing shift i^* can be found as

$$\left(w\ i^*-\arctan\left(\frac{X_0^S(m-1)}{X_0^C(m)}\right)\right)=\alpha\,. \quad (17)$$

With $m=2$, it follows

$$\frac{2\pi}{L}\ i^*=\arctan\left(\frac{X_0^S(1)}{X_0^C(2)}\right)+\alpha\,. \quad (18)$$

Finally,

$$i^*=\left\lfloor\frac{L\arctan\left(\frac{X_0^S(1)}{X_0^C(2)}\right)+L\alpha}{2\pi}\right\rfloor \mod L\,. \quad (19)$$

It is worth noting that $\arctan\left(\frac{X_0^S(1)}{X_0^C(2)}\right)$ takes its value in $[0, 2\pi[$ depending on the sign of $X_0^S(1)$ and $X_0^C(2)$. This makes the normalized sequence unique with a single normalizing shift in $[0, L-1]$. Also, observe that the normalizing shift i^* requires only the calculation of one DCT coefficient $X_0^C(2)$ and one DST coefficient $X_0^S(1)$. Once the normalizing shift is obtained, the new sequence is nothing but a shifted version of the input sequence by i^*

$$x_{i^*}=\{x(i^*), x(i^*+1), \cdots, x(L-1), x(0), \cdots, x(i^*-1)\}\,. \quad (20)$$

Regardless of the input sequence, the normalized sequence corresponds to the unique pair of coefficients

$$X_{i^*}^C(2)=\sqrt{\left(X_0^C(2)\right)^2+\left(X_0^S(1)\right)^2}\ \cos(\alpha)\,, \quad (21)$$

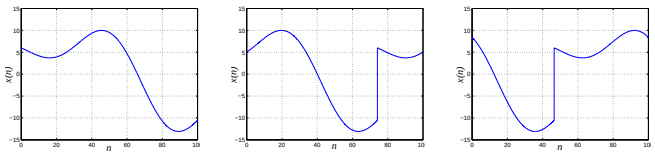
and

$$X_{i^*}^S(1)=\sqrt{\left(X_0^C(2)\right)^2+\left(X_0^S(1)\right)^2}\ \cos(\alpha+\frac{\pi}{2})\,. \quad (22)$$

The algorithm of shift invariant normalization can be summarized as follows

(i) Input original sequence (see (6)).
(ii) Set α in $[0, 2\pi[$.
(iii) Calculate $X_0^C(2)$ and $X_0^S(1)$ using (4) and (5).
(iv) Determine the normalizing shift i^* using (19).
(v) Output normalized sequence using (20).

Fig. 2 illustrates an example of signal normalization with $\alpha=\frac{\pi}{3}$. Fig. 2(a) shows an original signal. In Fig. 2(b), the



(a) (b) (c)

Fig. 2. (a) Original. (b) Shifted. (c) Normalized signal from (a) and (b).

original signal has undergone a circular shift distortion; both of these signals yield the same normalized sequence as shown in Fig. 2(c).

2) Signal calibration: Given a pre-defined value of α, signal calibration in this work consists of determining the normalizing shift for a digital signal. This represents the number of samples by which the signal can be shifted so that the shifted version follows a certain pattern by satisfying (17). Recall from (1), (2) and (3) that three 3-D arrays are constructed by DLBM in the horizontal, vertical and temporal directions. Next, each array is used to create a 2-D matrix of normalizing shifts by calibrating individual signals obtained in each corresponding direction. That is, the horizonal array is traversed horizontally, the vertical array is traversed vertically and the temporal array is traversed temporally (see Fig. 3). This is motivated by the fact that the information in each 3-D array captures the video content in the direction that DLBM are computed. The aim of the signal calibration idea is threefold. First, reducing the size of the feature vector since one feature only will be extracted from each DLBM signal. Second, increasing the robustness of the features when compared to complete DLBM signals as will be illustrated

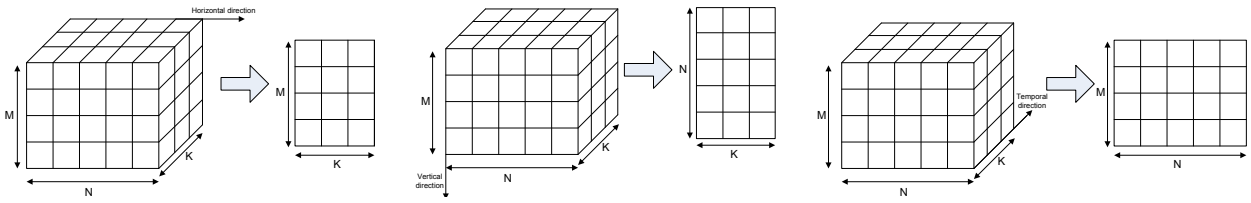


Fig. 3. Calibration of 3-D arrays. (a) Horizontal calibration. (b) Vertical calibration. (c) Temporal calibration.

later in experiments. Third, creating a hash that is sensitive to video tampering. The obtained 2-D arrays are denoted by H_{2D}, V_{2D}, and T_{2D}, respectively as

$$H_{2D}(i,k) = \left\lfloor \frac{L \arctan\left(\frac{DST_{\{H_{i,k}\}}(1)}{DCT_{\{H_{i,k}\}}(2)}\right) + N\alpha}{2\pi} \right\rfloor \mod N\,. \tag{23}$$

$$V_{2D}(j,k) = \left\lfloor \frac{L \arctan\left(\frac{DST_{\{V_{j,k}\}}(1)}{DCT_{\{V_{j,k}\}}(2)}\right) + M\alpha}{2\pi} \right\rfloor \mod M\,. \tag{24}$$

$$T_{2D}(i,j) = \left\lfloor \frac{L \arctan\left(\frac{DST_{\{T_{i,j}\}}(1)}{DCT_{\{T_{i,j}\}}(2)}\right) + K\alpha}{2\pi} \right\rfloor \mod K\,. \tag{25}$$

where $H_{i,k} = \{H(i,0,k), \cdots, H(i,N-1,k)\}$, $V_{j,k} = \{V(0,j,k), \cdots, V(M-1,j,k)\}$, and $T_{i,j} = \{T(i,j,0), \cdots, T(i,j,K-1)\}$ while $DCT_{\{\vartheta\}}(\cdot)$ and $DST_{\{\vartheta\}}(\cdot)$ represent the DCT and DST of ϑ, respectively. The concatenation of these arrays constitute the final hash. Let $x(n)$ be one of the aforementioned DLBM signals (supposedly of length L) in $\{H_{i,k}, V_{j,k}, T_{i,j}\}$ and $x^{'}(n)$ be its distorted version. That is

$$x^{'}(n) = x(n) + d(n)\,. \tag{26}$$

In view of (19), the linearity of the DCT suggests that the normalizing shift for $x'(n)$ becomes

$$i^{'*} = \left\lfloor \frac{L \arctan\left(\frac{DST_x(1)+DST_d(1)}{DCT_x(2)+DCT_d(2)}\right) + L\alpha}{2\pi} \right\rfloor \mod L\,. \tag{27}$$

Note that a significant change in $DST_d(1)$ and/or $DCT_d(2)$ plays a key role in the change of the normalizing shift. To illustrate the rationale for using the normalizing shift as a feature being robust against transcoding operations on one hand and sensitive to tampering on the other hand, Fig. 4 shows a signal and its altered versions with the respective normalizing shift denoted by i^* and displayed accordingly for each sequence with $\alpha = \frac{\pi}{3}$. As can be seen, the normalizing shift does not get affected by the addition of the white Gaussian noise considered in the example whereas the replacement of a portion of the signal with a different pattern leads to a clear change in the normalizing shift although the Signal-to-Noise Ratio (SNR) is larger. This can be justified

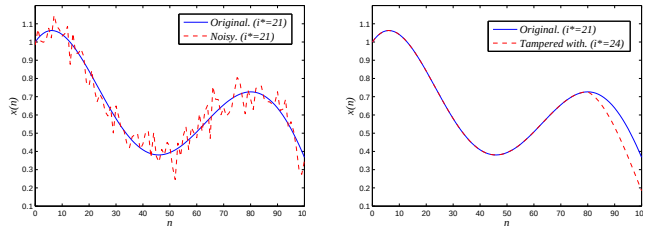


Fig. 4. Normalizing shift. (a) noise addition with SNR=20.77 dB. (b) Tampering with the signal by replacing a small portion with another pattern SNR=22.20 dB.

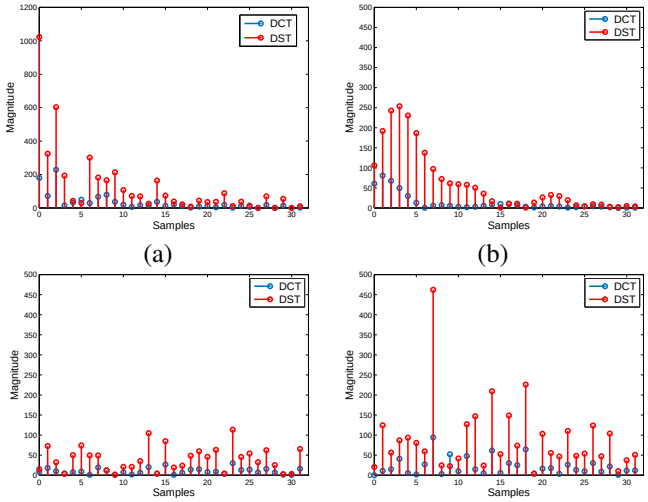
by the following analysis. According to (27), if the distortion is rich in frequency (i.e., a noise-like pattern), its energy is normally spread over the full range of frequencies. On the other hand, if the distortion is of low frequency content (e.g., due to content replacement), its energy is mostly packed in a few low-frequency DCT/DST coefficients. As a result, the low-frequency DCT/DST coefficients of the noise, in particular $DST_d(1)$ and $DCT_d(2)$, tend to be smaller in magnitude when compared to those of a distortion caused by malicious manipulations. For the sake of demonstration, DLBM signals have also been analyzed on a test video that has undergone tampering as well as transcoding operations as depicted in Fig. 5. Fig. 6 shows the magnitude of the DCT/DST coefficients of a DLBM signal (corresponding to $V_{j,k}$ with $(j,k) = (1,1)$) as well as those of distortions due to tampering and transcoding. As can be seen, both the DLBM signal and the distortion due to tampering are of low frequency content whereas transcoding distortions are rich in frequency since the corresponding DCT and DST coefficients are spread over the full range of frequencies. This makes $DST_d(1)$ and $DCT_d(2)$ in (27) more significant in the case of content tampering when compared to the transcoding operations (i.e., compression and resizing). Consequently, the normalizing shift undergoes a smaller change under transcoding than that caused by malicious manipulation.

It is worth mentioning that for $m > 2$, the normalizing shift becomes less robust since it would depend on a higher frequency content of DLBM. Furthermore, for each DLBM signal, there would be at least two normalizing shifts corresponding to the same reference angle α in the full signal length. Hence, to ensure that a unique normalizing shift can be obtained, only a fraction of the signal length should be used. As a result, the discriminative power of the hash would be significantly reduced.

Finally, in view of (23), (24), and (25), note that the

(a) (b) (c) (d)

Fig. 5. Original video and its distorted versions. (a) Original video with 480×854, 30 fps. (b) Forged. (c) Compressed by MPEG-4 at 128 kbps. (d) Resized to 240×320, 30 fps.



(a) (b) (c) (d)

Fig. 6. DCT/DST coefficients of the original DLBM signal and transcoding distortions. (a) Original signal. (b) Forgery distortion. (c) Distortion due to compression. (d) Distortion due to resizing.

properties of the hash do not depend on the value of the reference angle α as long as it is used as a constant parameter in the system. The proposed video hashing process is a low computational complexity algorithm and can be summarized as follows.

Input: Video.
Output: Hash.
Parameters: Reference angle α, the number of vertical splits M, and the number of horizontal splits N.

1. Re-sample the video at 5 fps. K is the number of frames in the re-sampled video.
2. Extract 3D DLBM arrays in three directions using (1), (2), and (3), respectively.
3. Compute the normalizing shift for each signal in all 3D directional DLBM arrays using (23), (24), and (25). This gives three 2D normalizing shift arrays.
4. Concatenate the normalizing shift arrays to form the final hash. This results in a hash length of $M\ N+M\ K+N\ K$.

The computational complexity can be analyzed in three steps. For a color video with η pixels, the color to grey-level conversion requires 3 multiplications and 2 additions per pixel whereas the re-sampling involves a few multiplications and additions per pixel depending on the length of the low pass filter used. As a result, the pre-processing stage requires $O(\eta)$. The computation of DLBM in each of the three directions can be performed in $O(\eta + \frac{\eta}{MNK})$. Finally, the hash extraction stage has a complexity of $O(MNK)$.

IV. Hash Matching

At the matching stage, the hash is assumed to be in the form of three 2-D arrays prior to concatenation; i.e., $h = \{H_{2D}, V_{2D}, T_{2D}\}$. This can be easily obtained by just reversing the process of concatenation. Recall that the values in each matrix are bounded since they represent the normalizing shifts. That is, $0 \leq H_{2D}(i,j) \leq N-1$, $0 \leq V_{2D}(i,j) \leq M-1$, and $0 \leq T_{2D}(i,j) \leq K-1$. Two similarity measures are defined: *identification measure* and *authentication measure*.

A. Identification Measure

For content identification purposes, the similarity measure must be as small as possible if two videos Υ_1 and Υ_2 are derived from each other. Denote by $\{H^1_{2D}, V^1_{2D}, T^1_{2D}\}$ and $\{H^2_{2D}, V^2_{2D}, T^2_{2D}\}$ their corresponding hashes, respectively. Because the normalizing shifts are determined in a forward direction only, one should consider the case where a change in the video slips the normalizing shift to the beginning of the sequence as it exceeds the boundary. Hence, we define a distance D as

$$\begin{aligned} D \;=\; & \sum_{i=0}^{M-1}\sum_{j=0}^{K-1} \min\{|H^1_{2D}(i,j) - H^2_{2D}(i,j)|, \\ & |H^1_{2D}(i,j) - H^2_{2D}(i,j) - N|, |H^1_{2D}(i,j) - H^2_{2D}(i,j) + N|\} \\ + & \sum_{i=0}^{N-1}\sum_{j=0}^{K-1} \min\{|V^1_{2D}(i,j) - V^2_{2D}(i,j)|, \\ & |V^1_{2D}(i,j) - V^2_{2D}(i,j) - M|, |V^1_{2D}(i,j) - V^2_{2D}(i,j) + M|\} \\ + & \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \min\{|T^1_{2D}(i,j) - T^2_{2D}(i,j)|, \\ & |T^1_{2D}(i,j) - T^2_{2D}(i,j) - K|, |T^1_{2D}(i,j) - T^2_{2D}(i,j) + K|\} \end{aligned} \quad (28)$$

It is expected that the use of blocks to compute DLBM makes the normalizing shifts robust to small rotations and spatial translations. However, in view of (28), the hash remains sensitive to temporal translations. Indeed, a shift of Υ_1 in the temporal direction will produce a video Υ_2 with a hash corresponding to horizontally translated versions of H^1_{2D} and V^1_{2D} in addition to an increase/decrease of the values in T^1_{2D} by the same amount of translation. To overcome this limitation, a full search among the possible shifted versions of H^2_{2D} and V^2_{2D} with the corresponding increase/decrease of T^2_{2D} would accurately determine the shift that minimizes D. However, this is computationally expensive as it requires K computations of D for each video comparison. To address this issue, we use only two rows from each of the horizontal and vertical arrays (H^1_{2D}, V^1_{2D}, H^2_{2D}, V^2_{2D}) to estimate a shift $q^* \in [0, K-1]$

that minimizes the difference as follows.

$$q^* = \arg\min_q \{d_{H^1_{2D},H^2_{2D}}(q) + d_{V^1_{2D},V^2_{2D}}(q)\}, \quad (29)$$

where

$$d_{H^1_{2D},H^2_{2D}}(q) = \sum_{j=0}^{K-1} \left|H^1_{2D}(\lfloor M/2 \rfloor, j) - H^2_{2D}(\lfloor M/2 \rfloor, (j-q) \mod K)\right| + \sum_{j=0}^{K-1} \left|H^1_{2D}(\lfloor M/2 \rfloor + 1, j) - H^2_{2D}(\lfloor M/2 \rfloor + 1, (j-q) \mod K)\right|, \quad (30)$$

and

$$d_{V^1_{2D},V^2_{2D}}(q) = \sum_{j=0}^{K-1} \left|V^1_{2D}(\lfloor N/2 \rfloor, j) - V^2_{2D}(\lfloor N/2 \rfloor, (j-q) \mod K)\right| + \sum_{j=0}^{K-1} \left|V^1_{2D}(\lfloor N/2 \rfloor + 1, j) - V^2_{2D}(\lfloor N/2 \rfloor + 1, (j-q) \mod K)\right|. \quad (31)$$

Then, the proposed identification distance becomes

$$\begin{aligned} D_{id} = & \sum_{i=0}^{M-1}\sum_{j=0}^{K-1} \min\{|H^1_{2D}(i,j) - H^2_{2D}(i,j-q^*)|, \\ & |H^1_{2D}(i,j) - H^2_{2D}(i,j-q^*) - N|, \\ & |H^1_{2D}(i,j) - H^2_{2D}(i,j-q^*) + N|\} \\ & + \sum_{i=0}^{N-1}\sum_{j=0}^{K-1} \min\{|V^1_{2D}(i,j) - V^2_{2D}(i,j-q^*)|, \\ & |V^1_{2D}(i,j) - V^2_{2D}(i,j-q^*) - M|, \\ & |V^1_{2D}(i,j) - V^2_{2D}(i,j-q^*) + M|\} \\ & + \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \min\{|T^1_{2D}(i,j) - T^2_{2D}(i,j) - q^* + K|, \\ & |T^1_{2D}(i,j) - T^2_{2D}(i,j) - q^*|, \\ & |T^1_{2D}(i,j) - T^2_{2D}(i,j) - q^* + 2K|\}. \end{aligned} \quad (32)$$

In practice, two videos are said to be similar if D_{id} does not exceed a certain threshold T_{id}. Otherwise, the videos are considered dissimilar. T_{id} can be found empirically by using the Neyman-Pearson criterion such that the false negative probability is minimized, subject to a fixed false positive probability [40].

B. Authentication Measure

In content-based video authentication, the similarity measure should produce a sufficiently large distance when the video undergoes forgery operations such as object insertion or removal. On the other hand, the distance is expected to be insignificant under common video transcoding operations including transsizing and transrating. Let us define two matrices D_h and D_v characterizing the spatial difference between two videos Υ_1 and Υ_2 as

$$D_h(i,k) = |H^1_{2D}(i,k) - H^2_{2D}(i,k)|; \quad i \in \{0,\cdots,M-1\},\ k \in \{0,\cdots,K-1\}. \quad (33)$$

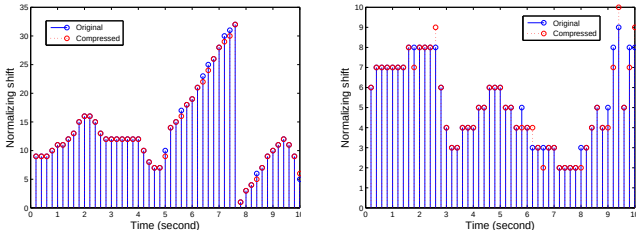
and

$$D_v(j,k) = |V^1_{2D}(j,k) - V^2_{2D}(j,k)|; \quad j \in \{0,\cdots,N-1\},\ k \in \{0,\cdots,K-1\}. \quad (34)$$

If a video is maliciously manipulated, the corresponding horizontal and vertical DLBM get affected in tampered regions causing a change in horizontal and vertical normalizing shifts (i.e., hash values) accordingly. Thus, both D_h and D_v are likely to be different from zero at the location of tampered regions. Let Φ be an array of size $(M \times N \times K)$ and defined as

$$\Phi(i,j,k) = \begin{cases} 1 & if\ D_h(i,k) > 0 \bigwedge D_v(j,k) > 0 \\ 0 & Otherwise. \end{cases} \quad (35)$$

This can detect changes in individual frames $k = 0,\cdots,K-1$ but false detections could also occur under transcoding operations. To overcome this issue, we propose a segment-based forgery detection measure exploiting the redundancy of distortions caused by object insertion and/or removal across the temporal dimension. Note that it is unlikely that the transcoding process creates such a uniform distortion, i.e., a distortion with similar effect and location through successive frames. Fig. 7 shows samples of the normalizing shifts extracted in the horizontal and vertical directions at the same spatial location over time from a video at a resolution of 480×640 and its MPEG-4 compressed version at 256 kbps. Observe that the variations of the normalizing shift over time



(a) (b)

Fig. 7. Temporal variation of the spatial hash values under MPEG-4 compression. (a) Horizontal normalizing shifts. (b) Vertical normalizing shifts.

are normally discontinued. On the other hand, the proposed idea relies on the fact that, in digital forgery, tampered regions should be visually observable in a video for a certain period of time to change or influence the viewers' perception. Changing the viewer's perception is actually the key objective of the malicious attacker and, hence, inserted or removed objects are not expected to move extremely fast unnoticeably. Therefore, it is reasonable to assume, in forged videos, that the malicious content is naturally observable as any other content but its detection requires some kind of decision making in an automated way. Each hash is therefore divided into Q short segments in which the forgery detection is performed. Denote by Φ'_q $(q = 1,\cdots,Q)$ the proposed segment-based forgery detection measure corresponding to the q^{th} segment as follows

$$\Phi'_q(i,j) = \begin{cases} 1 & if\ \sum_{k=(q-1)\times s}^{(q\times s)-1} \Phi(i,j,k) = s \\ 0 & Otherwise. \end{cases} \quad (36)$$

where s is the length of the segment. Eq. (36) serves as a measure to detect tampered regions in a video. In fact, without loss of generality, let us assume that the original video size is $(M' \times N' \times K')$ where M', N', and K' are multiples of M, N, and K, respectively [3]. Each value in Φ'_q corresponds to a cube of $\frac{M'}{M} \times \frac{N'}{N} \times \frac{K'}{K}\ s$ pixels in the video because of the down-sampling process and the use of blocks to calculate H_{2D} and V_{2D}. Obviously, the size of the cube represents the precision of the system in locating tampered regions[4]. Finally,

[3]This is because of the downsampling process described in subsection III-B. If M' and N' are not multiples of M and N respectively, the video frames can be resized accordingly.

[4]It is worth noting here that the precision in locating tampered regions is different from the authentication performance.

the proposed segment-based authentication distance is given by

$$D_{auth}(q) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \Phi'_q(i,j)\,. \tag{37}$$

A threshold T_{auth} can be used to decide whether a video segment is authentic, given the hash of the original video. This can also be determined empirically via the Neyman-Pearson criterion [40]. Now, because the videos are first resampled at 5 fps, the forged moving content should be detected except if the distance between the initial and current position, recorded in a short period of time equal to $s/5$, is larger than a certain threshold. If $s = 5$, the time frame would correspond to one second. Given the fact that the tampering takes place in a reasonably large spatial region to attract the viewer's attention, such a motion speed could be considered high and unlikely to occur. For the sake of illustration, Fig. 8 shows a moving object (in red) in its initial position on left side and its next position on the right side within a time frame $t = s/5$. Because

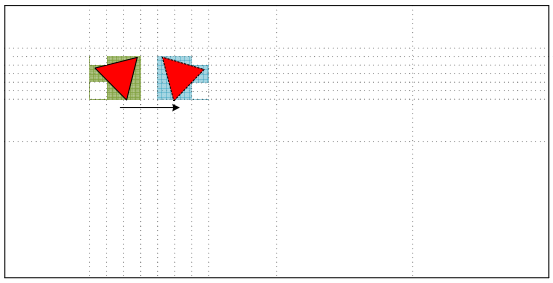


Fig. 8. Minimum distance traversed by an inserted/removed object in $s/5$ seconds so that tampering could not be detected by the system.

the calculation of DLBM involves two consecutive blocks in each direction, the tampering can be missed by the system only if distance between all the tampered blocks in the current frame (colored in green) and those in the next one (colored in blue) after t is more than the block size.

V. Experimental results

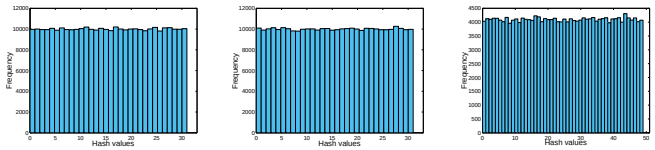
The performance of the proposed hashing technique has been assessed by conducting a number of experiments on a dataset of 200 various MPEG-2 color video clips with two different frame rates 25 and 30 fps and seven frame resolutions as depicted in Table I. Note that the dataset includes 170 Standard Definition (SD) videos and 30 High Definition (HD) videos. These videos have been collected from academic and public websites that cover the practically used video types (i.e., format, resolution, frame rate, etc.). This consists of the Open video Project [41], ReefVid [42], Youtube, and 30 HD videos from the Videvo website [43]. Each video clip is 10 seconds long. The shift-invariant normalization algorithm has been used with $\alpha = \frac{\pi}{20}$. The values of M and N have been empirically set to 32 for a good trade off between identification and authentication. With this setting, the hash for a 10 seconds long video consists of $(32+32)\times 5\times 10+(32\times 32) = 4224$ integer values. Note that the horizontal and vertical shifts can be encoded with only 5 bits each while the temporal shifts can be encoded with 6 bits.

TABLE I
The set of videos used in experiments.

Number of videos	Type	Resolution and frame rate
4	SD	288×360, 25 fps
17	SD	288×384, 25 fps
12	SD	288×512, 25 fps
87	SD	480×640, 25 fps
50	SD	480×854, 30 fps
19	HD	1080×1920, 30 fps
9	HD	1080×1920, 25 fps
2	HD	720×1280, 25 fps

A. Hash analysis

As discussed earlier, hash values consist of the normalizing shifts that are obtained in different directions of the DLBM features. Ideally, these hash values should be equally distributed over the full range of possible integers, i.e., corresponding to maximum information capacity in the hash, to ensure the discriminative capability of the system [44]. This is because the presence of hash values with higher probability than others would increase the likelihood that two hashes of visually distinct videos match by chance. In our first experiment, we have analyzed the distribution of the normalizing shifts in each of the directions on the aforementioned set videos. Note that the horizontal, vertical and temporal normalizing shifts take values in $\{0,1,\cdots,M-1\}$, $\{0,1,\cdots,N-1\}$ and $\{0,1,\cdots,K-1\}$, respectively. Fig. 9 shows the histogram of actual data in each of the directions, respectively. It can



(a) (b) (c)

Fig. 9. Distribution of the video hash values. (a) Horizontal hash extraction. (b) Vertical hash extraction. (c) Temporal hash extraction.

be seen that the extracted hashes follow a uniform distribution. This gives us a good indication of the rich and well balanced information contained within the extracted hashes for representing digital videos. Recall from (32) that the proposed identification distance is composed of three parts where each captures information in one direction, i.e., horizontal, vertical and temporal. Although the variables used for the hash and distances are integers, one can use a theoretical analysis on continuous data given the large number of features and video samples in our experiments. If the compared hashes in (32) correspond to two videos that are completely independent and visually distinct, the terms $|H^1_{2D}(i,j) - H^2_{2D}(i,j-q^*)|$, $|H^1_{2D}(i,j) - H^2_{2D}(i,j-q^*) - N|$, and $|H^1_{2D}(i,j) - H^2_{2D}(i,j-q^*) + N|$ follow a triangular distribution in $\{0,1,\cdots,N-1\}$, $\{1,2,\cdots,2N-1\}$, and $\{1,2,\cdots,2N-1\}$, respectively. This is because they represent the absolute value of two independent and uniformly distributed variables. However, the use of the $\min$ function involving an adjustment with $\pm N$ will approximately produce a normally distributed variable in an interval reduced to half of the original size, i.e., $\{0,1,\cdots,N/2\}$. Consequently, under the assumption that the normalizing shifts

are uncorrelated and according to the central limit theorem, the horizontal part can be viewed as a sum of $M \times K$ independent identically distributed variables and, hence, it follows a normal distribution with mean $M \times K \times N/4$. Likewise, both the vertical and temporal part follow a normal distribution with mean $N \times K \times M/4$ and $M \times N \times K/4$, respectively. Finally, one can deduce that the identification distance, which is the sum of these three normal variables, follows a normal distribution centered at $\mu_{D_{id}}$ that is given as

$$\mu_{D_{id}} = \frac{3MNK}{4} \tag{38}$$

Given 200 distinct videos, 19900 identification distances have been computed using (32). The distribution of the distance is illustrated by Fig. 10. As can be seen, the actual distance

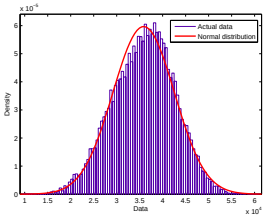


Fig. 10. Distribution of the proposed identification distance computed on visually distinct videos.

follows a normal distribution. Given the setting of $M = 32$, $N = 32$, and $K = 50$, the theoretical statistical mean is 3.84×10^4 whereas the empirical mean is found to be 3.59×10^4. This validates our theoretical analysis of the identification distance.

B. Identification performance

In this set of experiments, the ability of the proposed system to identify videos of similar content is assessed. It is also important to measure the system's ability to differentiate between the videos of different content. To this end, we have adopted ROC curves which display the True Positive Rate TPR (i.e., correct detection of similar video contents) against the False Positive Rate FPR (i.e., false detection of similar video contents). Ideally, these measures should correspond to $TPR = 1$ and $FPR = 0$. To illustrate our contributions in relation to video content identification, the following tests have been conducted. (i) The entire DLBM features are used as a fingerprint without the hash extraction stage[5] to assess the gain of the proposed hash extraction technique (i.e., the signal calibration technique). (ii) The proposed hashing system is also assessed using the Euclidean Distance (EC) as a similarity measure to evaluate the gain offered by the proposed identification distance D_{id} (see (32)). This is denoted by 'Proposed/EC'. Note that theoretically speaking, D_{id} differs from EC in the sense that it addresses two adversary effects on the hash when the video undergoes content-preserving changes. First, D_{id} takes into account the case where a change in video content slips the normalizing shift to the beginning of the sequence as it exceeds the boundary. Second, it compensates the hash changes that may be caused by temporal translations of the video. (iii) Recent state-of-the-art video hashing systems have also been applied on the same test videos for the purpose of comparison. Four well known techniques have been adopted in our comparative study: Centroid of Gradient Orientations (CGO) hashing [11], Temporally Informative Representative Images (TIRI) hashing [20], Weber Binarized Statistical Image Features (WBSIF) hashing [28], and Structural Graphical Models (SGM) based hashing [23]. We have used our own implementation of [11], [28], and [20] and the authors' implementation of [23] which has been available in [45]. The same parameters setting of the aforementioned systems has been adopted here as suggested by their authors. A number of content-preserving attacks have been considered in order to compute TPR in the ROC curves. As depicted in Table II, the attacks consist of spatial, geometric, and temporal distortions. It is worth mentioning that these attacks have been applied on the grey scale version of the videos because the hashing systems are expected to withstand color changes. Fig. 11 illustrates visual distortions caused by some spatial and geometric attacks on a test video frame. The robustness results of the proposed hashing system and the aforementioned techniques are shown in Fig. 12-14. The results are in perfect agreement with those reported in [23] in the sense that CGO is significantly outperformed by the TIRI and SGM hashing techniques. It can be seen that CGO cannot withstand the applied attacks. Note that these attacks are more significant than the ones reported in [11]. This suggests that CGO can only be used when the videos undergo minor distortions. Likewise, WBSIF fails to provide good performance because most attacks affect the differential excitation or the BSIF code which are both used to extract the final hash. The results also show that the proposed technique and TIRI perform equally well under spatial signal processing attacks whereas SGM exhibits a slightly lower performance (see Fig. 12). It has been reported that the strength of SGM lies in its ability to identify the video content when the bit budget of the hash is low [23]. However, with sufficiently large fingerprints, TIRI seems to perform slightly better than SGM according to our experimental results.

TABLE II
DIFFERENT ATTACKS USED TO ASSESS THE IDENTIFICATION PERFORMANCE.

Type	Attack	Parameters
Spatial	Brightness increase	adding 80% of the frame mean
	Brightness decrease	subtracting 80% of the frame mean
	Contrast increase	$[60, 180]$ to $[0, 255]$
	Contrast decrease	$[0, 255]$ to $[60, 180]$
	AWGN	$\sigma = 56$, $\mu = 0$
	Median filter	(11×11)
	MPEG-4 compression	SD videos: 128 kbps HD videos: 500 kbps
Geometric	Cropping and resizing	90% and 85%
	Rotation	2 and 5 degrees
	Shifting	$[5, 5]$ and $[10, 10]$ pixels.
Temporal	Frame dropping	50%.
	Frame insertion	50% via interpolation.
	Shifting	Temporal Shift: 10% and 20% .
	Shifting in time + Content replacement	Circular shift: 20% + Replacement of shifted content.

[5]Note that DLBM as described in our paper cannot be used as a fingerprint in practice because they are too large in size (153600 real valued features for a 10 seconds video) and involve high computational complexity at the matching stage.

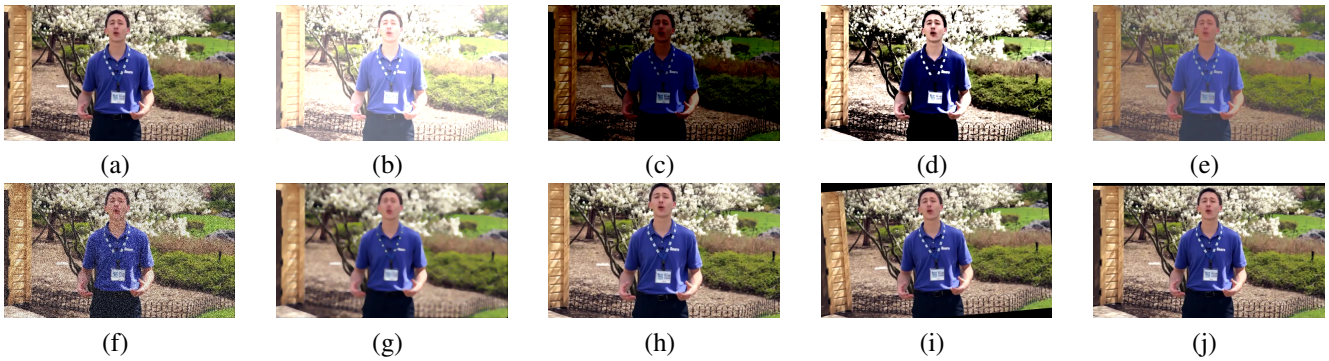


Fig. 11. Visual changes caused by different attacks (a) Original video. (b) brightness increase. (c) brightness decrease. (d) contrast increase. (e) contrast decrease. (f) additive white Gaussian noise with $\sigma = 56$. (g) median filter (11×11). (h) cropping 85% of the central part and resizing. (i) rotation by 5 degrees. (j) Shifting by $[10, 10]$.

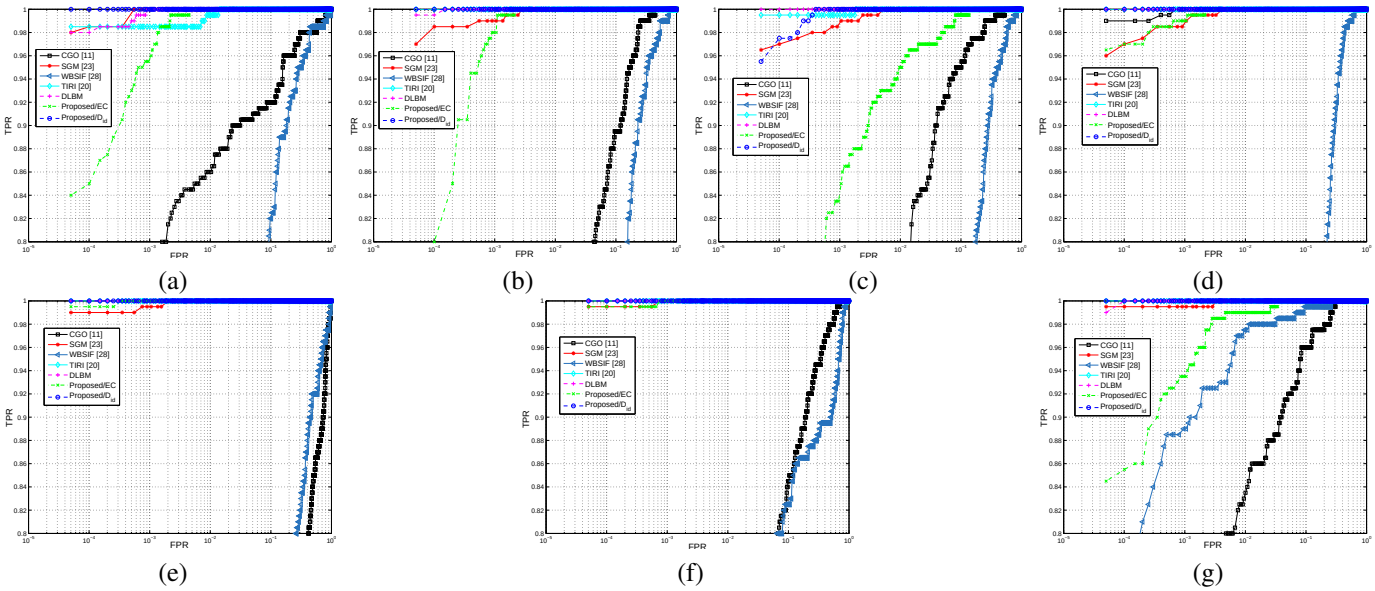


Fig. 12. Robustness results under spatial signal processing attacks. (a) Brightness increase. (b) Brightness decrease. (c) Contrast increase. (d) Contrast decrease. (e) Additive White Gaussian noise. (f) 11×11 Median filtering. (g) MPEG-4 compression.

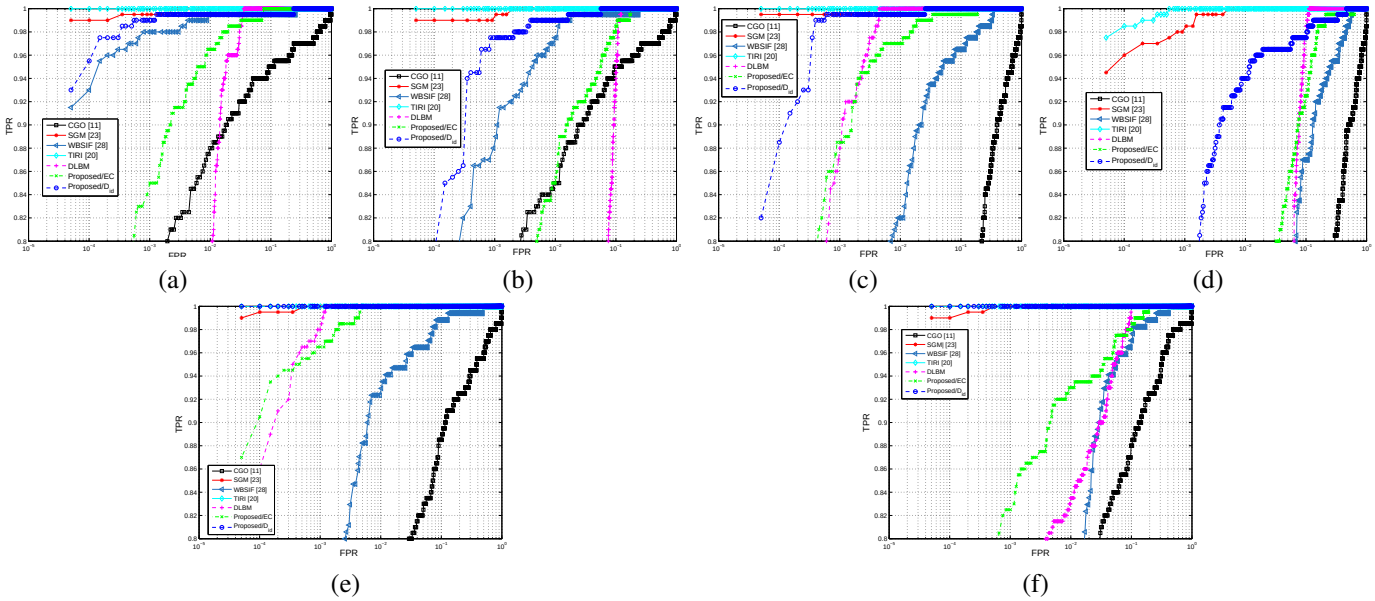


Fig. 13. Robustness results under geometric distortions. (a) Frame cropping by 90% and resizing. (b) Frame cropping by 85% and resizing. (c) Rotation by 2 degrees. (d) Rotation by 5 degrees. (e) Shifting $[5, 5]$. (f) Shifting $[10, 10]$.

Note that SGM uses a dimensionality reduction technique to obtain an approximate of video segments in the spatial domain prior to the feature extraction which is conducted in the pixel domain too (a technique similar to K-means clustering). This explains the lower performance under spatial signal processing operations when compared to TIRI and our hashing system. As expected, DLBM show robustness against attacks that tend to maintain the low frequency content of the videos such as the brightness change and low-pass filtering. The robustness of the proposed system against spatial signal processing attacks is mainly attributed to the efficiency of DLBM.

Our system performs reasonably well under small geometric distortions such as rotation by 2 degrees, cropping by 90% and spatial shifting (Fig. 13). This is partly attributed to the use of block-based features at the feature extraction stage. However, the performance deteriorates as the strength of such attacks increases because of synchronization-related changes in DLBM. Observe, however, that the hash extraction stage brings significant improvements over DLBM when the proposed identification distance is used. The justification for this is twofold. First, the distortion caused by geometric attacks corresponds to a noise-like pattern of DLBM changes, i.e., rich in frequency whereas the difference between two dissimilar videos represents a signal of low frequency content characterizing the DLBM changes (see for instance sub-section III-C2 on resizing). This makes the hash more sensitive to video content dissimilarity than small geometric distortions. Second, the proposed identification distance deals with the problem of synchronization efficiently as can be supported by the lower performance of the system when the Euclidean distance was used. This confirms the suitability of the identification distance for this particular application. The reason that TIRI exhibits good efficiency under geometric distortions is partly due to the fact that the DCT-based features are extracted from overlapping blocks which have been shown to offer more robustness than non overlapping blocks in [34]. It can be seen from Fig. 14 that the proposed hashing system outperforms its competitors under temporal distortions. This is mainly attributed to the efficient description of the video content by the normalizing shifts as well as the identification distance which takes into account any possible translations. In presence of temporal distortions, the competing techniques suffer from a synchronization problem. It is, however, worth mentioning that SGM surpasses TIRI and CGO because it encodes the temporal information of the video efficiently using the normalized cuts graph partitioning technique. The overall ROC curve for all the aforementioned attacks is plotted in Fig. 14(f). As can be seen, the proposed system outperforms all the competing techniques for a TPR higher than 0.94. Beyond this range, the system's performance drops rapidly due to the effect of geometric distortions on the overall performance. Note also that SGM performs slightly better than TIRI according to the overall results. Finally, the Equal Error Rate (EER) which corresponds to the point where FPR is equal to the False Negative Rate FPR$= 1-$TPR is depicted in Table III.

TABLE III
EER (%) FOR DIFFERENT SYSTEMS.

CGO	SGM	WBSIF	TIRI	DLBM	Proposed/EC	Proposed/D_{id}
14.65	1.86	11.91	2.17	7.35	6.53	**0.82**

C. Authentication performance

In this subsection, the system's performance is evaluated in terms of authentication. A reliable authentication system should detect video forgeries on one hand and tolerate transcoding on the other hand. To this end, widely used transcoding operations have been conducted on the test videos as depicted in Table IV. In this experiment, 250 forged videos have been created from the original ones. These forgeries include object insertion/removal and video embedding (See Fig. 15). Although the tampered regions vary in size from a video to another, it is worth mentioning that the tampering process affects no more than 8% of the original videos. The tampering was conducted using the 'Adobe After Effects' software. In the first part of experiments, the hashing system has been used with the proposed authentication distance D_{auth} on video segments of different lengths ($s = 2 \cdots, 6$). Here, ROC curves display the correct forgery detection rate (TPR), computed on forged videos, against the false forgery detection rate (FPR) which is measured on transcoded videos. The results are shown in Fig. 16. Note that the performance of the system increases with the number of frames in the segment, $s \in \{2, 3, \cdots, 6\}$, up to $s = 5$, i.e., the longer the video segment, the better the detection. This clearly shows the advantage of grouping individual frames by the proposed segment-based authentication measure to exploit the redundancy of malicious manipulations across the temporal dimension. For $s = 6$, however, the performance drops slightly suggesting that the tampered regions in some videos change position in a time shorter than 6 frames of the re-sampled video. In the rest of the paper, $s = 5$ is adopted.

TABLE IV
TRANSCODING OPERATIONS USED TO ASSESS THE AUTHENTICATION PERFORMANCE.

Transcoding operation	Description	Parameter
Transrating	MPEG-4 compression	SD videos: 128 kbps HD videos: 500 kbps
		SD videos: 256 kbps HD videos: 1000 kbps
		SD videos: 500 kbps HD videos: 1500 kbps
	Frame dropping	20%
Transsizing	Resizing by MPEG-4	240×320

Next, we include three video hashing techniques that have been used in authentication, namely SVD-based hashing [6], 3D DCT hashing [10], and 2D HOG hashing [32]. Similar to the previous identification experiments, the system has also been used with the Euclidean distance to illustrate the efficiency of the proposed authentication distance D_{auth}. The SVD-based hashing technique down-samples each video frame by calculating the average values of 4×4 blocks. Then, sub-blocks of size 4×4 are factorized with the Singular Value

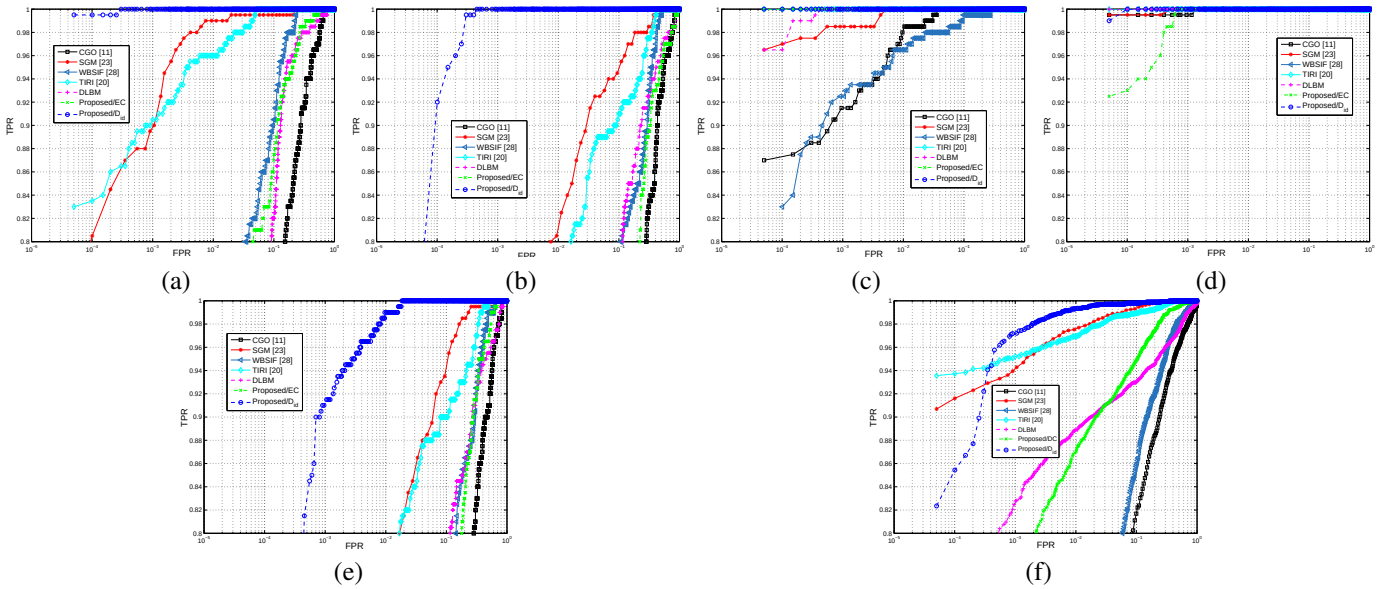


Fig. 14. Robustness results under temporal attacks and overall performance. (a) Temporal shifting by 10%. (b) Temporal shifting by 20%. (c) Frame drop with 50%. (d) Frame insertion with 50%. (e) Temporal shifting by 20% and video content replacement (20%). (f) Overall performance with all the aforementioned attacks.

Fig. 15. Samples of tampered test videos. First row: Original videos. Second row: Forged videos.

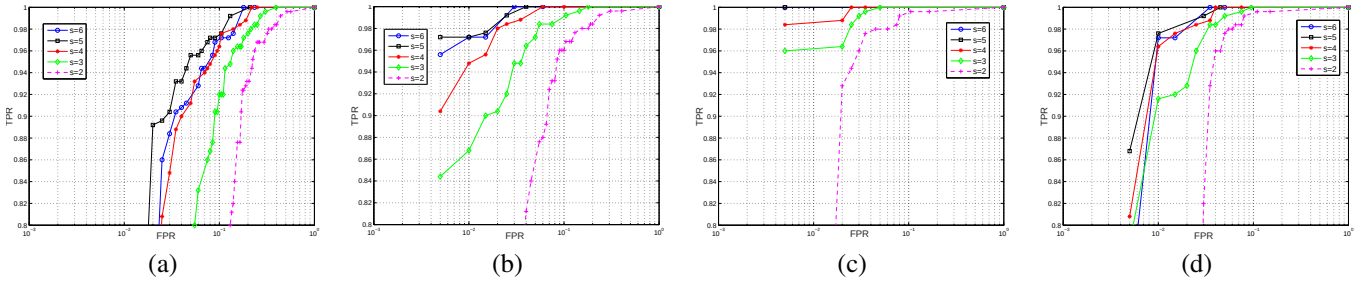


Fig. 16. Authentication performance. (a) Compression: SD videos at 128 kbps and HD videos at 500 kbps. (b) Compression: SD videos at 256 kbps and HD videos at 1000 kbps. (c) Compression: SD videos at 500 kbps and HD videos at 1500 kbps. (d) Transsizing.

Decomposition (SVD) where the first eigenvalue is used to classify uniform and non-uniform blocks. Uniform blocks are represented by Scalar Quantization (SQ) indices while Vector Quantization (VQ) is used to encode the first left-singular and right-singular vectors of non-uniform blocks [6][6]. The 3D DCT hashing technique applies a Gaussian filter in all directions (i.e., temporal and spatial directions) followed by a downsampling process and thresholding based on the median value to get a compact binary hash [10]. Finally, the 2D HOG hashing system resizes the video frames to 320×240 and selects a DCT coefficient from each 8×8 block to form a new smaller 3D array. Then, three directional gradient filters are applied to create three gradient arrays from which the authors calculate a magnitude and two angles at each sample location. These magnitude and angle arrays will then serve to extract the 2D HOG as a hash [32].

These techniques have been implemented in this work and

[6]The codebook size has been set to 256 as this was shown to yield the best performance in [6].

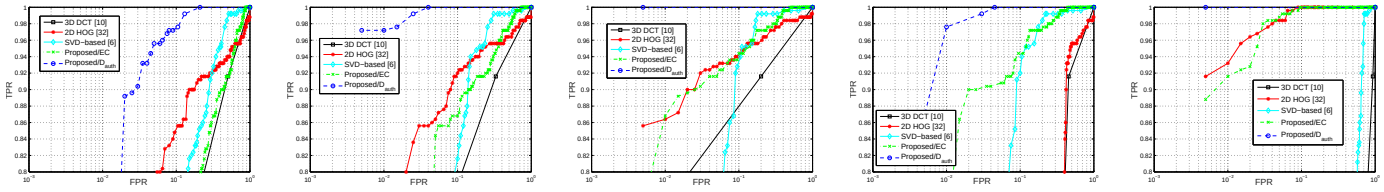


Fig. 17. Performance comparison. (a) Compression: SD videos at 128 kbps and HD videos at 500 kbps. (b) Compression: SD videos at 256 kbps and HD videos at 1000 kbps. (c) Compression: SD videos at 500 kbps and HD videos at 1500 kbps. (d) Transsizing. (e) Frame drop at a rate of 20%.

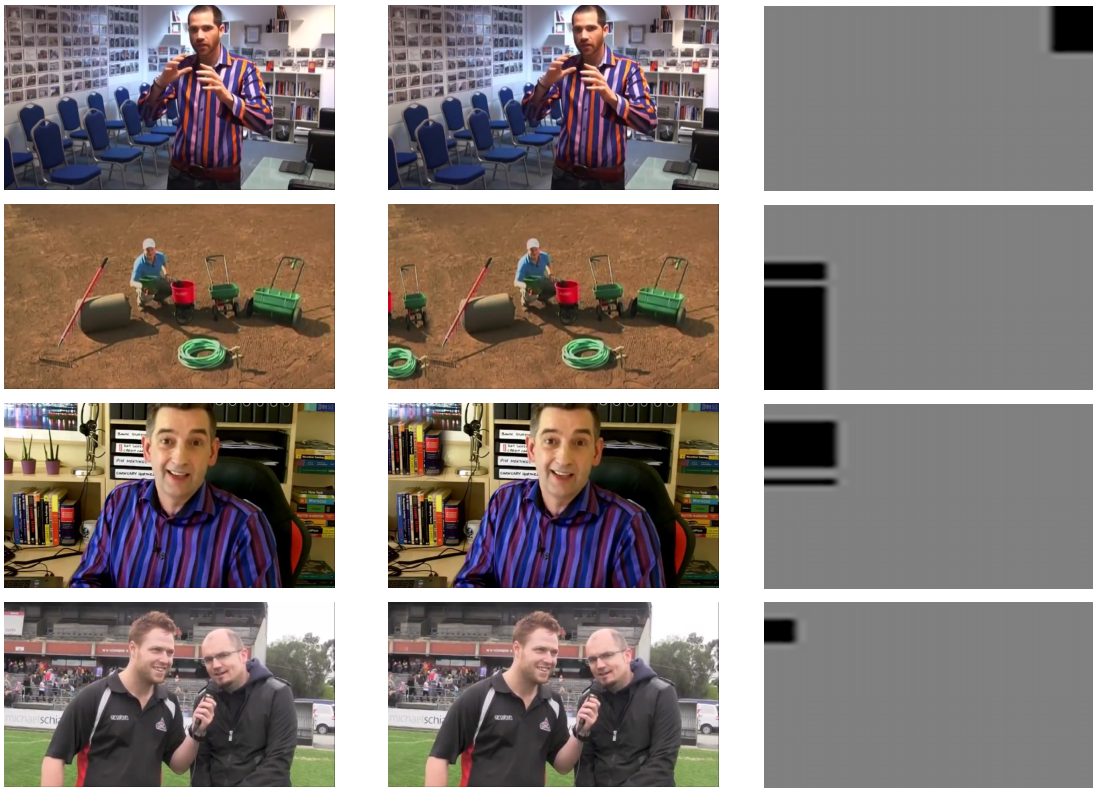
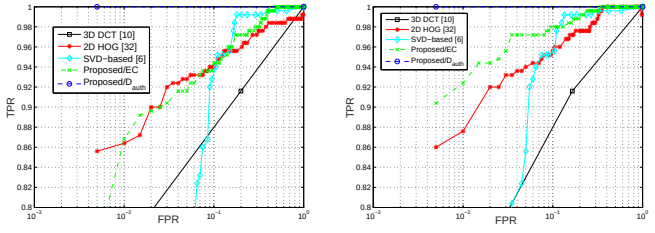


Fig. 18. Detection of tampered regions. First column: original videos. Second column: forged videos. Last column: detected tampering.

applied on the same test videos. It is worth mentioning that the 3D DCT hashing technique was mainly designed for video identification but its results were not reported earlier because it has already been outperformed by TIRI according to [20]. Unlike the proposed and other competing authentication techniques, the SVD-based hashing technique suffers from a synchronization problem when the video undergoes transsizing operations because it operates on individual frames using a fixed block size. For the sake of comparison, however, the transsized videos are resized back to their original size before applying the SVD-based authentication technique. It is also worth noting that the distortion caused by MPEG-4 compression is more significant than the one reported in [6] since our test videos are larger in size. The results are shown in Fig. 17. It can be seen that the proposed hashing system outperforms its competitors significantly. The 3D DCT hashing system completely fails to detect forged videos because its main design relies on a 3D transform which tends to summarize the video in a compact hash. As a result, tampered regions that affect a small portion of the video do not cause significant changes in the extracted hash. The other competing techniques appear unable to tolerate video distortions caused by low bit-rate compression while detecting video forgeries. Indeed, for the SVD-based hashing technique, low bit-rate compression seems to cause significant distortions in textured/edged blocks affecting the corresponding left-singular and right-singular vectors of the SVD and this leads to incorrect codeword representations in the codebook. As expected, the 2D-HOG technique produces a poor performance when transsizing was used. This can be explained by the sensitivity of the gradient orientation to resizing. Finally, one can clearly see that the proposed authentication distance fits well in the overall system when compared to the Euclidean distance. The results validate our claim on the sensitivity of the proposed hash to malicious manipulations on one hand and its robustness against transcoding operations on the other hand. Fig. 18 illustrates the detection of tampered regions in forged videos. The detected regions take the form of rectangular blocks because of the block-based locating process as described by (35). Finally, we have conducted similar experiments using another video cod-

ing standard, i.e., H264. This is to verify whether the format change affects the performance of the aforementioned systems. We show below the results of H.264 compression at the rate of 500 kbps for SD videos and 1500 kbps for HD videos as compared to the results of MPEG-4 compression at the same rate. Fig. 19 illustrates the corresponding performance.



(a) (b)

Fig. 19. Authentication performance under MPEG-4 and H.246 compression at the same bit-rate, respectively. (a) MPEG-4. (b) H.264.

Interestingly, the same observation can be made regarding the performance of the systems with a slight enhancement of accuracy due to the better video compression with H.264 when compared to MPEG-4.

D. Complexity analysis

In the proposed hashing scheme, most of the computational cost is caused by the pre-processing stage, i.e., the re-sampling process. The computational complexity of the aforementioned competing techniques is evaluated here for the sake of comparison. The average run time on a 10 second test video with the frame size of 480×854 and frame rate of 30 fps is measured. All the source codes were implemented in MATLAB and run on a platform of an Intel Core Duo i7 $-$ 4770 CPU 3.40GHz with 16 GB of memory. Note that Matlab is a high level programming language and the reported results could be significantly improved using a low level programming language such as C or C++. We used the authors' implementation of SGM [45] whereas our own implementation is used for other techniques. The results in milliseconds (*ms*) are depicted in Table V. The computational cost of the proposed hashing system is low when compared to its competitors but the identification stage requires a considerably higher cost than that of other techniques. This is because the measure has been adjusted with some extra calculations to take into account the changes that might occur on the hash due to temporal video operations. It is, however, worth mentioning that some parallelism can explored to run the directional distances of (32) (i.e., vertical, horizontal and temporal) simultaneously. As for the authentication measure, the run time for our technique is reasonably fast since this is a verification problem involving only a one-to-one matching to reach the decision on authenticity.

TABLE V
RUN TIME IN MILLISECONDS (*ms*) OF THE HASHING AND MATCHING STAGES WITH DIFFERENT TECHNIQUES.

Technique	Hashing stage	Matching stage	
		Identification	Authentication
CGO	1153	0.0033	$--$
SGM	1441	**0.0024**	$--$
WBSIF	911	0.0027	$--$
TIRI	887	0.0056	$--$
Proposed	**774**	0.0261	2.375
3D DCT	5633	$--$	**0.0023**
2D HOG	4720	$--$	0.1195
SVD-based	73950	$--$	18.16

VI. CONCLUSION

A perceptual video hashing system has been presented in this paper. The system exhibits an interesting feature in that it can serve in both the applications of video content identification and authentication using the same hash. Compared to the traditional approach, i.e., using a separate system for each application, this concept brings the advantage of reducing the computational complexity and saving the storage space. The key idea relies on a new shift-based signal calibration technique using DCT and DST coefficients. Through theoretical and experimental analysis, this technique has been shown to offer efficient hash information which can withstand signal processing operations such as noise and low pass filtering on one hand and detect malicious manipulations on the other hand. The system has been experimentally assessed in the two applications and its superiority over state-of-the-art techniques has been demonstrated.

APPENDIX A
PROOF OF (15) AND (16)

Let x_0 be a discrete time signal and x_1 be its shifted version as described by (6) and (7), respectively. The common z-transform of x_0 is given by

$$ZT(x_0) = \sum_{n=0}^{L-1} x(n) z^{-n}\,, \tag{39}$$

where z is a complex number. It can be shown that

$$ZT(x_1) = z \times ZT(x_0) - x(0)\left(z - z^{-(L-1)}\right)\,. \tag{40}$$

Now, recall that (12) and (13) can be expressed in a matrix form. That is, for each value of m we have

$$Y_{i+1} = A\ Y_i\,, \tag{41}$$

where Y_i is given by

$$Y_i = \begin{bmatrix} X_i^C(m) \\ X_i^S(m-1) \end{bmatrix}\,, \tag{42}$$

and A is described as

$$A = \begin{bmatrix} \cos(w) & \sin(w) \\ -\sin(w)\cos(w) \end{bmatrix}\,, \tag{43}$$

where $w = \frac{\pi m}{L}$. The goal is to mathematically express Y_i as a discrete time function of variable i. In view of (40) and (41), the z-transform gives

$$(z\ I - A)\ ZT(Y) = I\ \left(z - z^{-(L-1)}\right) Y_0\,, \tag{44}$$

where I is the identity matrix of size 2×2. It follows

$$ZT(Y) = (z\ I - A)^{-1}\ I\ \left(z - z^{-(L-1)}\right) Y_0\,. \tag{45}$$

We obtain

$$ZT(Y)=\begin{pmatrix} \frac{z-\cos(w)}{1-2\cos(w)z+z^2} & \frac{\sin(w)}{1-2\cos(w)z+z^2} \\ -\frac{\sin(w)}{1-2\cos(w)z+z^2} & \frac{z-\cos(w)}{1-2\cos(w)z+z^2} \end{pmatrix} \times \left(z-z^{-(L-1)}\right) Y_0\,. \quad (46)$$

By applying the inverse z-transform on both terms of (46) and by considering $0 \leq i \leq L-1$, Y_i can be derived as

$$Y_i = \begin{bmatrix} X_0^C(m)\cos(w\ i) + X_0^S(m-1)\sin(w\ i) \\ -X_0^C(m)\sin(w\ i) + X_0^S(m-1)\cos(w\ i) \end{bmatrix}. \quad (47)$$

Finally, (47) can be written as follows

$$Y_i = \begin{bmatrix} \sqrt{\left(X_0^C(m)\right)^2 + \left(X_0^S(m-1)\right)^2}\cos\left(w\ i - \arctan\left(\frac{X_0^S(m-1)}{X_0^C(m)}\right)\right) \\ \sqrt{\left(X_0^C(m)\right)^2 + \left(X_0^S(m-1)\right)^2}\cos\left(w\ i - \arctan\left(-\frac{X_0^C(m)}{X_0^S(m-1)}\right)\right) \end{bmatrix} \quad (48)$$

APPENDIX B
DETERMINATION OF THE IDENTIFICATION AND AUTHENTICATION THRESHOLDS

Given a random variable D_{id}, the problem can be formulated as a binary decision.

$$\begin{aligned} D_{id} &\geq T_{id} \Rightarrow H_0 \\ &< T_{id} \Rightarrow H_1\,, \end{aligned}$$

where H_0 represents the hypothesis that the compared videos are visually distinct whereas H_1 is the hypothesis of visually similar videos. As seen in subsection V-A, the identification distance can be modeled by a normal distribution when the videos are visually distinct. To obtain the threshold, the Neyman-Pearson criterion is used in such a way that the missed similarity detection probability is minimized, subject to a fixed false alarm probability [40][46]

$$\begin{aligned} P_{FA} &= Prob(D_{id} < T_{id}|H_0)\,, \\ &= \int_{-\infty}^{T_{id}} f_{D_{id}}(t|H_0)dt\,, \end{aligned} \quad (49)$$

where $f_{D_{id}}(t|H_0)$ is the pdf of D_{id}. Since D_{id} follows a normal distribution under H_0, it follows

$$T_{id} = \sqrt{2\sigma_{D_{id}}^2}\ erfc^{-1}(2-2P_{FA}) + \mu_{D_{id}}\,, \quad (50)$$

where $efrc$ is the complementary error function. $\sigma_{D_{id}}$ and $\mu_{D_{id}}$ are the statistical mean and standard deviation of the identification distance, respectively. Likewise, one can deduce the authentication threshold T_{auth}. Assume that H_0 is the hypothesis of genuine videos whereas H_1 is the hypothesis of forged videos. Hence, the decision is

$$\begin{aligned} D_{auth} &\leq T_{auth} \Rightarrow H_0 \\ &> T_{auth} \Rightarrow H_1\,. \end{aligned}$$

In this case, the false alarm probability can be represented as

$$\begin{aligned} P_{FA} &= Prob(D_{auth} > T_{auth}|H_0)\,, \\ &= \int_{T_{auth}}^{\infty} f_{D_{auth}}(t|H_0)dt\,, \end{aligned} \quad (51)$$

where $f_{D_{auth}}(t|H_0)$ is the pdf of D_{auth} under hypothesis H_0. Under the assumption that $f_{D_{auth}}(t|H_0)$ follows a normal distribution, we obtain

$$T_{auth} = \sqrt{2\sigma_{D_{auth}}^2}\ erfc^{-1}(2P_{FA}) + \mu_{D_{auth}}\,, \quad (52)$$

where $\sigma_{D_{auth}}$ and $\mu_{D_{auth}}$ can be computed from the empirical distance between the original videos and their transcoded versions.

REFERENCES

[1] M. Schneider and S. F. Chang, "A robust content-based digital signature for image authentication," in *Proc. IEEE Int. Conf. On Image Processing*, Lausanne, Switzerland, Sep. 1996.

[2] C. Y. Lin and S. F. Chang, "A robust image authentication method surviving JPEG lossy compression," in *Proc. SPIE Storage and Retrieval of Image/Video Database*, San Jose, USA, Jan. 1998, vol. 3312.

[3] J. Lu, "Video fingerprinting for copy identification: from research to industry applications," in *Proc. of SPIE - Media Forensics and Security*, San Jose, CA, USA, Jan. 2009, vol. 7254.

[4] H. J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *IEEE Computational Science and Engineering*, vol. 4, pp. 10–21, Oct.-Dec. 1997.

[5] A. P. Gueziec, X. Pennec, and N. Ayache, "Medical image registration using geometric hashing," *IEEE Computational Science and Engineering*, vol. 4, pp. 29–41, Oct.-Dec. 1997.

[6] P.-C. Su, C. C. Chen, and H. M. Chang, "Towards effective content authentication for digital videos by employing feature extraction and quantization," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 19, pp. 668–677, May 2009.

[7] J. Oostveen, T. Kalker, and J. Haitsma, "Visual hashing of digital video: applications and techniques," in *Proc. of SPIE Applications of Digital Image Processing*, San Diego, CA, USA, Jul. 2001.

[8] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Proc. of the 5th International Conference on Recent Advances in Visual Information Systems*, Taiwan, Mar. 2002.

[9] C. De Roover, C. De Vleeschouwer, F. Lefbvre, and B. Macq, "Robust video hashing based on radial projections of key frames," *IEEE Trans. On Signal Processing*, vol. 53, pp. 4020–4037, Oct. 2005.

[10] B. Coskun, B. Sankur, and N. Memon, "Spatiotemporal transform based video hashing," *IEEE Trans. On Multimedia*, vol. 8, pp. 1190–1208, Dec. 2006.

[11] S. Lee and C. D. Yoo, "Robust video fingerprinting for content-based video identification," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 18, pp. 983–988, Jul. 2008.

[12] S. Lee and Y. H. Suh, "Video fingerprinting based on orientation of luminance centroid," in *Proc. IEEE Int. Conf. On Multimedia and Expo.*, NY, USA, Jul. 2009.

[13] A. Sarkar, P. Ghosh, E. Moxley, and B. S. Manjunath, "Video fingerprinting: Features for duplicate and similar video detection and query-based video retrieval," in *Proc. of SPIE - Multimedia Content Access: Algorithms and Systems II*, San Jose, CA, USA, Jan. 2008, vol. 6820.

[14] Z. Zhang, C. Cao, R. Zhang, and J. Zou, "Video copy detection based on speeded up robust features and locality sensitive hashing," in *Proc. IEEE Int. Conf. On Automation and Logistics*, Hong Kong, Aug. 2010.

[15] G. Yang, N. Chen, and Q. Jiang, "A robust hashing algorithm based on SURF for video copy detection," *Computers and Security*, vol. 31, pp. 33–39, Feb. 2012.

[16] M. Douze, H. Jgou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-filtering," *IEEE Trans. On Multimedia*, vol. 12, pp. 257–266, June 2010.

[17] C-Y. Chiu, C-S. Chen, and L-F Chien, "A framework for handling spatiotemporal variations in video copy detection," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 18, pp. 412–417, Mar. 2008.

[18] S. Lee, C. D. Yoo, and T. Kalker, "Robust video fingerprinting based on symmetric pairwise boosting," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 19, pp. 1379–1388, Sep. 2009.

[19] Z. Xu, H. Ling, F. Zou, Z. Lu, P. Li, and T. Wang, "Fast and robust video copy detection scheme using full dct coefficients," in *Proc. IEEE Int. Conf. On Multimedia and Expo*, NY, USA, Jul. 2009.

[20] M. M. Esmaeili, M. Fatourechi, and R. K. Ward, "A robust and fast video copy detection system using content-based fingerprinting," *IEEE Trans. On Information Forensics and Security*, vol. 6, pp. 213–226, Mar. 2011.

[21] J. Sun, J. Wang, J. Zhang, X. Nie, and J. Liu, "Video hashing algorithm with weighted matching based on visual saliency," *IEEE Signal Processing Letters*, vol. 19, pp. 328–331, Jun. 2012.

[22] M. Li and V. Monga, "Robust video hashing via multilinear subspace projections," *IEEE Trans. On Image Processing*, vol. 21, pp. 4397–4409, Oct. 2012.

[23] M. Li and V. Monga, "Compact video fingerprinting via structural graphical models," *IEEE Trans. On Information Forensics and Security*, vol. 8, pp. 1709–1721, Nov. 2013.

[24] G. Zhu, J. Huang, S. Kwong, and J. Yang, "Fragility analysis of adaptive quantization-based image hashing," *IEEE Trans. On Information Forensics and Security*, vol. 5, pp. 133–147, Mar. 2010.

[25] M. Li and V. Monga, "Twofold video hashing with automatic synchronization," *IEEE Trans. On Information Forensics and Security*, vol. 10, pp. 1727–1738, Aug. 2015.

[26] B. Wu, S. Krishnan, N. Zhang, and L. Su, "Compact and robust video fingerprinting using sparse represented features," in *Proc. IEEE International Conference on Multimedia and Expo*, Seattle, USA, July 2016.

[27] R. B. Wang, H. Chen, J. l. Yao, and Y. T. Guo, "Video copy detection based on temporal contextual hashing," in *Proc. IEEE International Conference on Multimedia Big Data*, Taipei, Taiwan, Apr. 2016, pp. 223–228.

[28] A. Boukhari and A. Serir, "Weber binarized statistical image features (WBSIF) based video copy detection," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 50–64, 2016.

[29] X. Nie, Y. Yin, J. Sun, J. Liu, and C. Cui, "Comprehensive feature-based robust video fingerprinting using tensor model," *IEEE Trans. on Multimedia*, vol. 19, pp. 785–796, Apr. 2017.

[30] A. Pramateftakis, T. Oelbaum, and K. Diepold, "Authentication of MPEG-4 based surveillance video," in *Proc. Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 33–37.

[31] Y. J. Ren, L. O'Gorman, J. Wu, F. Chang, T. L. Wood, and J. R. Zhang, "Authenticating lossy surveillance video," *IEEE Trans. on Information Forensics and Security*, vol. 8, pp. 1678–1687, Oct. 2013.

[32] T. Kroputaponchai and N. Suvonvorn, "Video authentication using spatio-temporal signature for surveillance system," in *Proc. International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Thailand, July. 2015.

[33] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: Performance evaluation and tradeoffs," *IEEE Trans. on Image Processing*, vol. 15, pp. 3453–3466, Nov. 2006.

[34] F. Khelifi and J. Jiang, "Perceptual image hashing based on virtual watermark detection," *IEEE Trans. on Image Processing*, vol. 19, pp. 981–994, Apr. 2010.

[35] L. Chen, D. Xu, I. W.-H. Tsang, and X. Li, "Spectral embedded hashing for scalable image retrieval," *IEEE Trans. on Cybernetics*, vol. 44, pp. 1180–1190, Jul. 2014.

[36] C. Ma and C. Liu, "Two dimensional hashing for visual tracking," *Computer Vision and Image Understanding*, vol. 135, pp. 83–94, 2015.

[37] J. Fang, H. Xu, Q. Wang, and T. Wu, "Online hash tracking with spatio-temporal saliency auxiliary," *Computer Vision and Image Understanding*, vol. 160, pp. 57–72, 2017.

[38] M. L. Miller, "Audio fingerprinting: Nearest neighbour search in high dimensional binary spaces," in *Proc. IEEE Workshop on Multimedia Signal Processing*, Virgin Islands, USA, Dec. 2002, p. 182185.

[39] P. Yip and K. R. Rao, "On the shift property of DCTs and DSTs," *IEEE Trans. On Accoustics, Speech, and Signal Processing*, vol. 35, pp. 404–406, Mar. 1987.

[40] T. Ferguson, *Mathematical Statistics: A Decision Theoretical Approach*, Academic Press, 1967.

[41] "Open video project," https://open-video.org/, 2015, Online, accessed 12-May-2015.

[42] "A resource of free coral reef video clips for educational use," http://www.reefvid.org/, 2015, Online, accessed 12-May-2015.

[43] "Free stock video footage," https://www.videvo.net/, 2015, Online, accessed 12-May-2015.

[44] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1148–1161, Nov. 1993.

[45] http://signal.ee.psu.edu/SGMVideoHashing.htm , Matlab Implementation of Video fingerprinting via structural Graphical Models. Accessed in Feb. 2015.

[46] A. Piva, M. Barni, F. Bartolini, and V. Cappellini, "Threshold selection for correlation-based watermark detection," in *Proc. COST254 Workshop on Itelligent Communications*. 1998, pp. 67–72, Press.