Desynchronization Resilient Video Fingerprinting via Randomized, Low-rank Tensor Approximations

Mu Li and Vishal Monga

Department of Electrical Engineering, The Pennsylvania State University
University Park, PA, 16802, USA
Email: mql5192@psu.edu, vmonga@engr.psu.edu

***Abstract*—The problem of summarizing videos by short fingerprints or hashes has garnered significant attention recently. While traditional applications of video hashing lie in database search and content authentication, the emergence of websites such as YouTube and DailyMotion poses a challenging problem of anti-piracy video search. That is, hashes or fingerprints of an original video (provided to YouTube by the content owner) must be matched against those uploaded to YouTube by users to identify instances of "illegal" or undesirable uploads. Because the uploaded videos invariably differ from the original in their digital representation (owing to incidental or malicious distortions), *robust* video hashes are desired. In this paper, we model videos as order-3 tensors and use multilinear subspace projections, such as a reduced rank parallel factor analysis (PARAFAC) to construct video hashes. We observe that unlike most standard descriptors of video content, tensor based subspace projections can offer excellent robustness while effectively capturing the spatio-temporal essence of the video for discriminability. We further randomize the construction of the hash by dividing the video into randomly selected overlapping sub-cubes to prevent against intentional guessing and forgery. The most significant gains are seen for the difficult attacks of spatial (e.g. geometric) as well as temporal (random frame dropping) desynchronization. Experimental validation is provided in the form of ROC curves and we further perform detection-theoretic analysis which closely mimics empirically observed probability of error.**

***Index Terms*—video fingerprinting, hashing, digital signatures.**

I. Introduction

With the emergence of video websites like YouTube and DailyMotion in the past few years, content tracking and copy right protection of multimedia files has gained increasing importance. Video hashing or fingerprinting has hence assumed significance as an algorithmic tool that can be used for anti-piracy search of video content. Video hashing algorithms constitute randomized video dimensionality reduction which maps a video to a short digest called as its *hash vector*. Unlike traditional hashes [1], video hashes are desired to be *robust*, i.e. they must respond to "perceptual" changes to video and not just its digital representation [2]. At the same time for the hash to be meaningful it must reliably discriminate between distinct video sequences. Besides discriminability and robustness, security against adversarial attacks is desired in video hashes which mandates that algorithmic techniques for computing hashes from video sequences be randomized with the help of a secret key akin to message authentication codes (MACs) in cryptography [1].

The existing literature on video hashing comprises of two main categories: The first class of methods are frame based video hashing methods, that is each frame of the video is treated as an image and state of the art image hashing techniques[1] are employed. The image hashes are then concatenated or combined in a similar fashion to form a video hash [4]–[6]. Lee *et al.* proposed a hashing method based on Centroid of Gradient Orientations (CGO) whose central idea is to extract direction (angle) information from the difference of adjacent pixel values [4], [5]. Roover *et al.* proposed RAdial projection based haSHing (RASH) [6] wherein they compute the variance of pixel values on a set of lines articulated around the center of the frame followed by a one-dimensional DCT to get final hash. To make the video hash more economical, they do the aforementioned procedure on "key" frames of the video. Frame based hashing methods while simple to compute, are well-known [7] to suffer from temporal attacks such as (uniform/non-uniform) frame rate change - either incidental, e.g. in video transcoding or intentional, i.e. deliberate removal of parts of video to defeat hash algorithms. Key frame based video hashing techniques [6] can further be attacked to loose discriminability - since the resulting hash vectors will be exactly the same as long as the attacked frames don't coincide with the frames that are used to calculate the video hash. The value of temporal evolution of the video content is hence critical for hashing, and recent effort has concentrated on spatio-temporal video hashes [7]–[9]. A representative method of this type is based on three-dimensional DCT which selects the low frequency three-dimensional DCT coefficients as hash vectors [7]. While the 3-D DCT based hash is spatio-temporal and can offer benefits over frame-based hashing such as increased robustness to temporal attacks, the geometric attack (e.g. rotation, cropping) vulnerability of DCT and similar transform domain (e.g. 3-D DWT) hashes [7] is a concern.

In this paper, we propose a new video hashing algorithm based on modeling videos as tensors and using sub-space projections of tensors, such as low-rank tensor approximations via PARAFAC [10]. Formally, a tensor is a multidimensional array. An order-1 tensor is a vector, an order-2 tensor is a matrix. A video clip which has both image content in 2-D and temporal evolution in the third dimension can be well modeled by an order-3 tensor. Multilinear sub-space projection

[1]For a thorough review of image hashing, we point the reader to [3].

978-1-4577-1434-4/11/$26.00 ©2011 IEEE

techniques such as parallel factor analysis (PARAFAC) enable a rank-r approximation of the tensor as a sum of rank-1 tensors. For an order-3 video tensor, each rank-1 tensor is made of an outer product of three vectors. We observe that when r is really small, e.g. $r = 1$, these representations are very robust to perturbations on the original tensor. Further, the vectors in the outer product are interpretable in that they correspond to the spatial as well as temporal components of the tensor. To yield a secure hash algorithm, we perform a random 3-D tiling of the video via overlapping sub-cubes and compute low-rank tensor approximations (LRTAs). Our randomization strategy makes a departure from existing spatio-temporal video hashing techniques such as 3-D DCT [7] which create a random basis on which to project the video. Our randomization technique in fact can capture local video components - hence guarding against adversarial temporal desynchronization attacks.

The rest of the paper is organized as follows. Section II-A provides background on low-rank tensor approximations (LRTAs), in particular PARAFAC. The randomized LRTA video hash algorithm is subsequently proposed in Section II-B. Section III gives both empirically estimated Receiver Operating Characteristics (ROCs) curves for probability of error and also provides a customized detection theoretic analysis which shows remarkable agreement with experimental ROC curves.

II. Hashing Based on Low-rank Tensor Approximations

A. Low-rank Tensor Approximation

PARAFAC tensor factorization, which is a typical low-rank tensor approximation method, can be considered as an extension of matrix Singular Value Decomposition (SVD) [10]. Specifically, rank-r PARAFAC factorization of an order-3 tensor $\mathcal{Y} \in \mathbb{R}^{I\times J\times K}$ is:

$$\mathcal{Y} \approx [\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \triangleq \sum_{i=1}^{r} \lambda_i \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i \tag{1}$$

where $\mathbf{a}_i \in \mathbb{R}^I$, $\mathbf{b}_i \in \mathbb{R}^J$, $\mathbf{c}_i \in \mathbb{R}^K$ are component vectors which are usually normalized such that the ℓ_2 norm is equal to one, $\mathbf{A} \in \mathbb{R}^{I\times r}$ is a matrix whose columns are $\mathbf{a}_i$ for $i = 1 \dots r$. $\mathbf{B}$ and $\mathbf{C}$ are defined similarly. $\circ$ denotes outer product of vectors, which means that (1) is equivalent to

$$y_{ijk} \approx \sum_{l=1}^{r} \lambda_l \cdot a_{il} \cdot b_{jl} \cdot c_{kl} \tag{2}$$

If the order-3 tensor $\mathcal{Y}$ represents a video, then the vectors $\mathbf{a}_i$ and $\mathbf{b}_i$, which correspond to decomposition results of frontal slices of $\mathcal{Y}$, will summarize the spatial information of the video. Simultaneously, the vectors $\mathbf{c}_i$ will encode how that spatial information evolves over the temporal dimension [11].

Computing the PARAFAC factorization is equivalent to solving the following optimization problem:

$$[\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \arg \min_{\lambda_i, \mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i} \| \mathcal{Y} - \sum_{i=1}^{r} \lambda_i \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i \|_2 \tag{3}$$

where $\|\|_2$ denotes the norm of a tensor which is defined analogously to the Frobenius norm of a matrix:

$$\| \mathcal{Y} \|_2 = \sqrt{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} y_{ijk}^2} \tag{4}$$

The problem in (3) is non-convex, hence obtaining the global minima is hard. A rich body of numerical algorithms exists for computing the PARAFAC representation, *viz.* the Alternating Least Squares (ALS) algorithm [12], as well as its many variants and improvements [13].

B. Video Hashing Based on Low-rank Tensor Approximations

The formal statement of our algorithm is given as follows:

Algorithm 1 LRTA Video Hash

Input:

Query video $\mathcal{V}_q$.

Output:

Hash vector $\mathbf{h}_K\ (\mathcal{V}_q) \in \mathbb{R}^{(M+N+P)\cdot r}$.

1: [**Normalization**] To make sure that all videos are handled similarly, akin to existing video hashing methods [7], we experimentally pre-determine a desired video size of $I \times J$ pixels per frame with a total of K frames. Then, given any query video $\mathcal{V}_q$, temporally re-sample the video so that it has exactly K frames. Perform spatial re-sizing of each frame to $\mathbb{R}^{I\times J}$, resulting in a normalized video $\mathcal{V} \in \mathbb{R}^{I\times J\times K}$.

2: [**Randomization**] Randomly select the locations of Q overlapping sub-cubes $\mathcal{V}_i \in \mathbb{R}^{M\times N\times P}$, $i = 1 \dots Q, M < I, N < J, P < K$, so that they approximately cover the entire video $\mathcal{V}$ - see Figs. 1(a), 1(b).

3: [**Rank-r PARAFAC factorization**] For each sub-cube representing a sub-video $\mathcal{V}_i$, calculate its rank-r PARAFAC tensor factorization as in (3). This results in 3 sets of r vectors corresponding to the two spatial and one temporal dimension(s). Via concatenation, collect the spatial components into vectors $\mathbf{x}_i \in \mathbb{R}^{M\cdot r}$, $\mathbf{y}_i \in \mathbb{R}^{N\cdot r}$. Similarly, the temporal components are collected in a vector $\mathbf{z}_i \in \mathbb{R}^{P\cdot r}$ for $i = 1 \dots Q$.

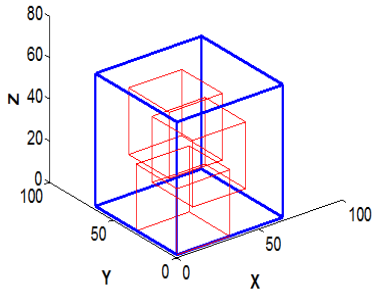
4: [**Arithmetic averaging**] The final hash $\mathbf{h}_K$ is as follows:

$$\mathbf{h}_K = \left[\ \frac{\sum_{i=1}^{Q}\mathbf{x}_i}{Q}; \frac{\sum_{i=1}^{Q}\mathbf{y}_i}{Q}; \frac{\sum_{i=1}^{Q}\mathbf{z}_i}{Q}\ \right] \in \mathbb{R}^{(M+N+P)\cdot r}$$

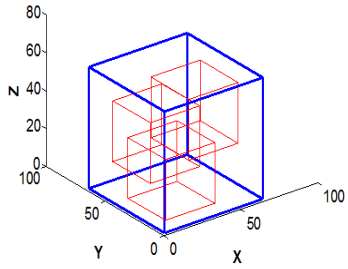
Characteristics of the Proposed Algorithm:

- We guard against intentional guessing and forgery of the hash by computing the LRTA hash from randomly selected overlapping sub-cubes or sub-videos which altogether roughly cover the whole video. That is, the locations as well as sizes of these sub-cubes[2] are determined by a pseudo-random number generator (PRNG). The

[2]For notational ease, the sizes of each sub-video/cube in Step 2 are defined to be the same - they may be varied in practice to increase randomization.



(a) Randomization with secret key K_1



(b) Randomization with secret key K_2

Fig. 1. Randomly distributed overlapping sub-cubes in a video as determined using two different secret keys K_1 and K_2.

secret key K serves as the seed to this PRNG. Because the key is unknown to the attacker, the randomized locations and sizes of these sub-videos impart the necessary entropy to the hash. Further, by defining smaller sub-videos from which to compute the hash, we capture local components of the video. This in turn greatly improves the resilience of the hash against adversarial attacks of temporal desynchronization where selected frames are removed from the video to defeat the hash algorithm. One example of the distribution of these sub-cubes under two different keys is shown in Fig. 1, where 8 sub-cubes $\in \mathbb{R}^{32\times32\times32}$ are selected randomly. For ease of visualization, only 3 sub-cubes are shown in each video $\in \mathbb{R}^{64\times64\times64}$. Sub-cubes are overlapping because related research in image hashing [3] shows that overlap can make the hash more robust against estimation attacks.

- The averaging operator in Step 4 prepares vectors that consolidate the spatio-temporal information from each of the sub-cubes while keeping the hash to be a manageable length for practical applications. Randomized linear combinations (with the help of the secret key K) could also be performed in this step which can facilitate a robustness vs. security trade-off.

(a) One frame of an original video

(b) The same frame after compression and AWGN

Fig. 2. Corresponding frames from an original video and its distorted version

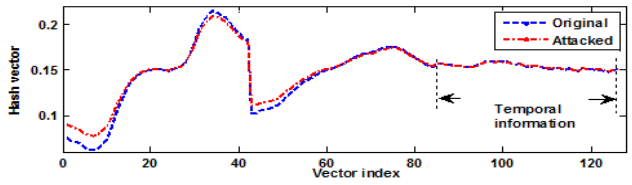
III. Experimental Results and Discussion

A. Illustration of hash robustness

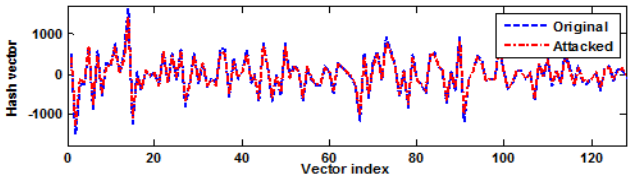
A preliminary experiment is provided here which illustrates the ability of the proposed hash algorithm to effectively capture robust and informative spatio-temporal video features. The video frame shown in Fig. 2(a) undergoes compression and white noise addition resulting in the video frame in Fig. 2(b). We address the video hashing methods in [7], [4], [6] as DCT, CGO and RASH, respectively. All videos are normalized to $\mathbb{R}^{64\times64\times64}$ based on which DCT, CGO and RASH hash vectors are computed. For the frame based methods, i.e. CGO, RASH, the number of features per frame (or key frame) was calibrated such that each of these hashes $\in \mathbb{R}^{128}$. Likewise, a robust set of 128 low-frequency 3-D DCT values as described in [7] (barring the DC coefficient) was chosen. To maintain approximate consistency in hash lengths across algorithms, in our proposed algorithm we compute PARAFAC with rank $r = 1$ on 8 (randomly selected) sub-videos $\in \mathbb{R}^{42\times42\times42}$ and therefore synthesize an LRTA video hash $\in \mathbb{R}^{126}$. Figs. 3(a) through 3(d) plot the actual hash vectors extracted from the videos in Figs. 2(a) and 2(b) corresponding to the four hash algorithms above against the index of the hash vector. The correlation between the hash vectors of original vs. distorted video is worth investigating in Figs. 3(a) through Fig. 3(d). In particular, for the proposed LRTA video hash (Fig. 3(a)) this correlation is particularly well pronounced for indices 85 through 126 (the final 42 entries) because they largely encode temporal information while the applied distortion is a spatial attack applied to each frame.

Analogously we apply a temporal attack (with no additional spatial distortion to the individual frames), i.e. temporal sub-sampling[3] by a factor of 4 to another video and visually examine correlation between original and distorted hash vectors in Figs. 4(a)-4(d). It may be inferred from Fig. 4(a) that now the first 84 (spatial) coefficients of the LRTA video hash are virtually indistinguishable from its attacked version. We note that such characteristics are not exhibited by the hash algorithms and corresponding plots in Figs. 4(b)-4(d). This ability of the LRTA hash to discriminatively capture the spatial and temporal information present in the video makes it robust

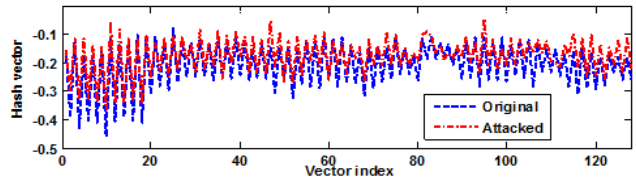
[3]The missing frames are estimated by linear temporal interpolation.



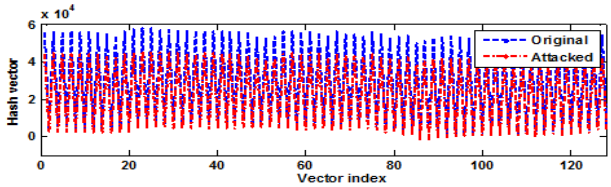
(a) LRTA



(b) 3-D DCT

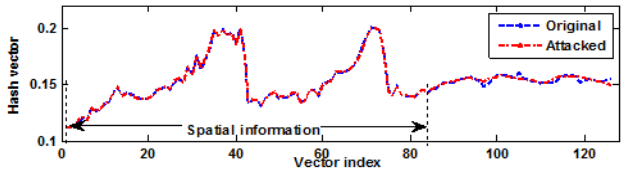


(c) CGO

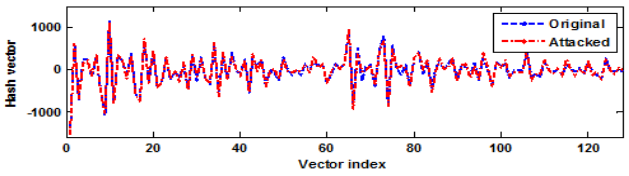


(d) RASH

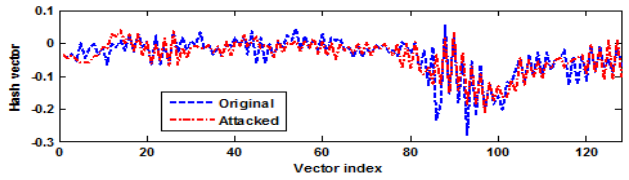
Fig. 3. Plot of hash vectors (vs. index) extracted from an original video and its attacked version. The spatial attack involved compression and noise addition applied to each video frame.



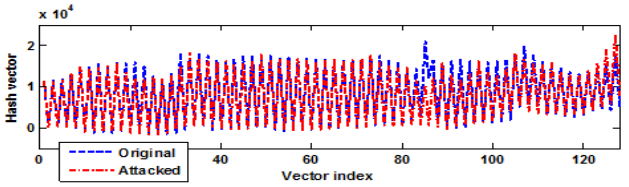
(a) LRTA



(b) 3-D DCT



(c) CGO



(d) RASH

Fig. 4. Plot of hash vectors (vs. index) extracted from an original video and its attacked version. The temporal attack involved temporal sub-sampling by a factor of 4. No other spatial attack was applied to individual video frames.

against a wide variety of incidental as well as adversarial attacks as formally validated in the next Section.

B. Quantitative Measures of Hash Deviation

We extensively test the LRTA video hash's performance under most typical distortions/attacks applied to video data. This includes:

- **Incidental signal and image processing attacks:** E.g. compression, random contrast changes, blurring and addition of white Gaussian noise (AWGN)
- **Geometric attacks:** E.g. Individual frame rotation, cropping and rescaling.
- **Temporal attacks:** E.g. Frame rate change, temporal desynchronization via random frame deletion.

Hashes were computed from a database of $n = 1000$ original videos obtained from YouTube and their respective distorted versions. To facilitate evaluation, we define a quantitative measure

$$D = \frac{\frac{\sum_{i=1}^{n} \|\mathbf{h}_K(\boldsymbol{\mathcal{V}}_i) - \mathbf{h}_K(A(\boldsymbol{\mathcal{V}}_i))\|}{n}}{\frac{\sum_{i=1}^{n}\sum_{j=1}^{i-1} \|\mathbf{h}_K(\boldsymbol{\mathcal{V}}_i) - \mathbf{h}_K(A(\boldsymbol{\mathcal{V}}_j))\|}{\frac{n\cdot(n-1)}{2}}} \tag{5}$$

Here, $\boldsymbol{\mathcal{V}}_i$ denotes the ith video in the database of size n, $A(\boldsymbol{\mathcal{V}})$ denotes the attacked version of a video $\boldsymbol{\mathcal{V}}$. Since D represents the ratio of the Euclidean distance between visually similar videos and between visually different videos, the smaller D is, better the performance of the algorithm. The characterization of various attacks and corresponding quantitative results (values of D) across the 4 hash algorithms discussed in Section III-A are provided in Tables I and II respectively. From Table II, it is readily apparent that the proposed LRTA video hash is either the best or in the worst case the second best across all the attacks. It is worth emphasizing that RASH owing to its choice of radial features [6] exhibits excellent robustness under rotation.

C. ROC Curves

In this Section, we provide detailed statistical evaluation for a few carefully selected and particularly strong attacks. Receiver operating characteristic (ROC) curves are commonly used to compare the performance of different hashing algorithms. We employ the widely used energy detector, i.e. the ℓ_2 norm of the difference of hash vectors for making inferences. Given a query video, we want to judge whether it is a visually different

TABLE I
INCIDENTAL AS WELL AS ADVERSARIAL VIDEO ATTACKS

Attack	Parameter Setting
Compression	Motion JPEG with quality factor 10
Contrast enhancement	Increase contrast to 100(AVS7.1)
Blurring	Circular averaging filter with radius 10
AWGN	σ_N=110
Frame rotation	5 deg counterclockwise
Frame cropping	25% cropping
Frame rate change	Subsampling by a factor of 2
Frame rate change	Subsampling by a factor of 4
Frame rate change	Subsampling by a factor of 8
Random frame deletion	Deletion of 16 frames

TABLE II
NORMALIZED HASH DEVIATION UNDER DIFFERENT ATTACKS

Attack	LRTA	DCT	CGO	RASH
Compression	0.0097	0.0126	0.1281	0.0091
Contrast enhancement	0.4208	0.4693	0.4766	1.1265
Blurring	0.1293	0.0289	0.4137	0.3096
AWGN	0.3487	0.2138	0.5247	0.6023
Frame rotation	0.2003	0.4031	0.8924	0.1048
Frame cropping	0.6114	0.9103	0.6591	0.3526
Frame rate change (Factor 2)	0.1043	0.0810	0.2463	0.1770
Frame rate change (Factor 4)	0.1773	0.1857	0.3429	0.2855
Frame rate change (Factor 8)	0.2721	0.4089	0.4503	0.5000
Random frame deletion	0.1301	0.1719	0.2635	0.1600

video with the reference video or just an attacked version of reference video. Clearly, this is a binary hypothesis testing problem where we define hypothesis H_1 to be that the query video is similar with reference video while hypothesis H_0 to be that the query video is different with reference video. In this setting, the miss and false alarm probability (respectively P_M and P_F) are defined as follows:

$$P_M(\tau) = Pr(\| \mathbf{h}_K(\boldsymbol{\mathcal{V}}) - \mathbf{h}_K(A(\boldsymbol{\mathcal{V}})) \| > \tau) \quad (6)$$

$$P_F(\tau) = Pr(\| \mathbf{h}_K(\boldsymbol{\mathcal{V}}) - \mathbf{h}_K(A(\boldsymbol{\mathcal{V}}')) \| < \tau) \quad (7)$$

where $A(\boldsymbol{\mathcal{V}})$ denotes the attacked version of a video $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{V}}'$ represents a completely different video.

We evaluate 4 distinct hash algorithms: the DCT, CGO and RASH hash algorithms [7], [4], [6] and the proposed LRTA video hash. The parameters for these algorithms are as described in Section III-A so they are all approximately the same length. A database that consists of 1000 good quality videos was obtained from YouTube and used in our experiments. Each video was normalized to $\mathbb{R}^{64\times64\times64}$ prior to hashing. Our proposed algorithm implemented PARAFAC tensor factorization using tensor toolbox provided by Sandia National Labs [14]. The MATLAB code for all algorithms, and finally the evaluation scripts for ROC Curves are available at:*http://signal.ee.psu.edu/VideoHashing.htm*

We implemented and applied two severe geometric (spatial) desynchronization attacks: 1.) random local bending, and 2.) a composite attack comprising of severe MPEG4 compression, random contrast adjustment, and rotation by 5 deg counterclockwise. Example video frames before and after the attack are shown in Figs. 5(a)-5(c). The corresponding ROC curves for the 4 hash algorithms are in Figs. 6(a) and 6(b). It may

(a) One frame of the original video (b) Frame after random bending (c) Frame after MPEG 4 compression, rotation and random contrast adjustment

Fig. 5. Video frame under spatial desynchronization (geometric) attacks

be inferred from Fig. 6(a) that for the local random bending attack (which involves several local rotations), RASH and LRTA are the most competitive with RASH being mildly better owing to its inherent robustness to rotation which has been observed in past work [6] as well. Since CGO is based on angle information, it is very vulnerable against random local geometric bending [4]. Under the tough composite attack (Fig. 6(b)) , the LRTA video hash outperforms the others.

We also tested against a temporal desynchronization attack which is representative of undesirable uploads of copyrighted video content to YouTube. In this attack, half of the video frames, i.e. 16 frames from both ends of the normalized video ($\in \mathbb{R}^{64\times64\times64}$), are randomly deleted. The ROC curves across the 4 hash algorithms are plotted in Fig. 6(c) which clearly establishes the temporal robustness of the LRTA video hash. Unsurprisingly, the frame based hashing methods such as RASH and CGO do rather poorly against such a strong temporal attack. The spatio-temporal approaches, i.e. the 3-D DCT [7] and proposed LRTA video hash do much better with LRTA affording the lowest errors owing to its ability to better preserve spatial information under temporal attacks and localization as introduced by our randomness strategy.

D. Detection Theoretic Modeling and Validation

Video hashing is well modeled as binary hypothesis testing in a detection theoretic setting. Here H_0 represents the hypothesis that the query video has "visually different" content from the reference video $\boldsymbol{\mathcal{V}}$; correspondingly, H_1 represents the hypothesis that the query video is an attacked/distorted version of the reference video $\boldsymbol{\mathcal{V}}$. Let $\mathbf{y}$ be the hash of the query video, $\mathbf{x}$ be the hash of reference video, and $\mathbf{z}$ be the hash of an arbitrary video that is "perceptually" different from $\boldsymbol{\mathcal{V}}$. Then, consistent with (6) and (7), the following binary hypothesis testing problem results:

$$\mathrm{H}_1 : \mathbf{w} = \mathbf{n} \sim f_{\mathbf{w}|\mathbf{x},\mathrm{H}_1}(\mathbf{w}) = f_{\mathbf{w}|\mathrm{H}_1}(\mathbf{w}) \quad (8)$$

$$\mathrm{H}_0 : \mathbf{w} = \mathbf{z} - \mathbf{x} + \mathbf{n} \sim f_{\mathbf{w}|\mathbf{x},\mathrm{H}_0}(\mathbf{w}) \quad (9)$$

where "|" denotes "condition on". $\mathbf{w} \triangleq \mathbf{y} - \mathbf{x}$ and $\mathbf{n}$ denote the noise on the hash vector due to the signal/image processing attack. (8) is valid since noise $\mathbf{n}$ is assumed to be independent of $\mathbf{x}$. Experimentally we determine that $\mathbf{w}$ can be well modeled

by a multivariate Gaussian Mixture Model (GMM) comprising of individual colored Gaussian components, i.e.

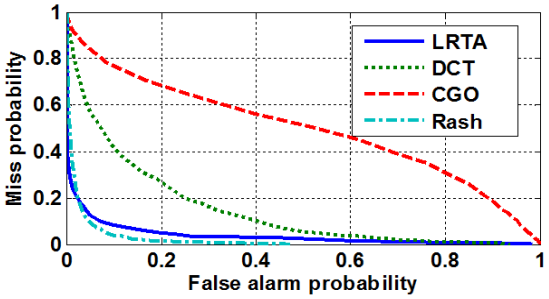
$$f_{\mathbf{w}|\mathrm{H}_1}(\mathbf{w}) = \sum_{i=1}^{M} c_i r_i(\mathbf{w}) \quad (10)$$

$$\sum_{i=1}^{M} c_i = 1 \quad (11)$$

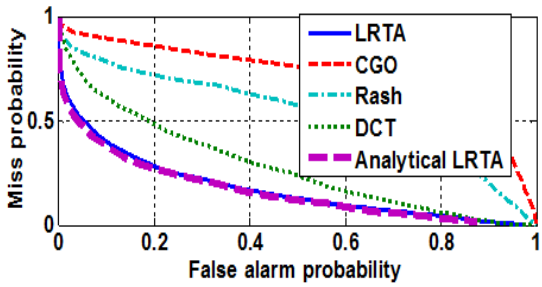
$$f_{\mathbf{w}|\mathbf{x},\mathrm{H}_0}(\mathbf{w}) = \sum_{i=1}^{N} d_i q_i(\mathbf{w}) \quad (12)$$

$$\sum_{i=1}^{N} d_i = 1 \quad (13)$$

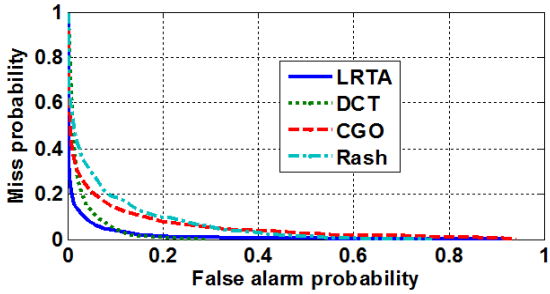
where $r_i(\mathbf{w}) \sim \mathcal{N}(\boldsymbol{\mu}_{1i}, \boldsymbol{\Sigma}_{1i}), i = 1 \ldots M$ and $q_i(\mathbf{w}) \sim \mathcal{N}(\boldsymbol{\mu}_{0i}, \boldsymbol{\Sigma}_{0i}), i = 1 \ldots N$. When using the energy detector as in Section III-C, the detection statistic is a quadratic form of multivariate colored Gaussian mixtures for which, an analytical closed form p.d.f (probability density function) is not known. To circumvent this, we first estimate the GMM parameters $\{c_i, \boldsymbol{\mu}_{1i}, \Sigma_{1i}\}_{i=1}^{M}$ and $\{d_i, \boldsymbol{\mu}_{0i}, \Sigma_{0i}\}_{i=1}^{N}$ using expectation maximization (EM), and subsequently predict the theoretic ROCs using numerical Monte-Carlo techniques. These numerical Monte-Carlo estimates based on our detection theoretic model are plotted for the LRTA Video hash algorithm in Fig. 6(b) alongside the experimental ROC curve and remarkably, their agreement is near perfect.



(a) ROC Curves. Attack: Random Local Bending.



(b) ROC Curves. Attack: MPEG 4 compression, rotation and random contrast adjustment.



(c) ROC Curves. Attack: Temporal desynchronization via random frame dropping.

Fig. 6. Statistical Evaluation of Video Hashing Algorithms Via Receiver Operating Characteristic (ROC) Curves.

REFERENCES

[1] A. Menezes, V. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1998.

[2] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems*, ser. VISUAL '02. London, UK, UK: Springer-Verlag, 2002, pp. 117–128.

[3] V. Monga and K. Mihcak, "Robust and secure image hashing via non-negative matrix factorizations," *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 3, pp. 376 –390, 2007.

[4] S. Lee and C. Yoo, "Robust video fingerprinting for content-based video identification," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 7, pp. 983 –988, 2008.

[5] S. Lee, C. Yoo, and T. Kalker, "Robust video fingerprinting based on symmetric pairwise boosting," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 9, pp. 1379 –1388, 2009.

[6] C. De Roover, C. De Vleeschouwer, F. Lefebvre, and B. Macq, "Robust video hashing based on radial projections of key frames," *Signal Processing, IEEE Transactions on*, vol. 53, no. 10, pp. 4020 – 4037, 2005.

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

[8] J. Oostveen, T. Kalker, and J. Haitsma, "Visual hashing of digital video: applications and techniques," in *Proc. SPIE*, vol. 4472, no. 121, 2001.

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

[10] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, September 2009.

[11] A. Cichocki, R. Zdunek, A.-H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis*. John Wiley, November 2009, pp. 410 –412.

[12] J. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition," *Psychometrika*, vol. 35, pp. 283–319, 1970, 10.1007/BF02310791.

[13] D. Nion and L. D. Lathauwer, "An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma," *Signal Processing*, vol. 88, no. 3, pp. 749 – 755, 2008.

[14] B. W. Bader and T. G. Kolda, "Matlab tensor toolbox version 2.4," http://csmr.ca.sandia.gov/ tgkolda/TensorToolbox/, March 2010.