A Review of Video Fingerprints Invariant to Geometric Attacks

Regunathan Radhakrishnan[a], Wenyu Jiang[a] and Claus Bauer[a]

[a]Dolby Laboratories Inc, 100 Potrero Ave, San Francisco, CA, USA;

ABSTRACT

Video fingerprints can help us identify a large amount of video on the Internet and enable interesting services to the end user. One of the main challenges for video fingerprints is for them to be robust against intentional/unintentional geometric modifications on the content such as scaling, aspect ratio conversion, rotation and cropping. In this paper, we review a number of fingerprinting methods proposed in literature that are particularly designed to be robust against such modifications. We also present two approaches that we adopted. One that is based on estimation of Singular Value Decomposition (SVD) bases from a window of past video frames (Approach 1) and another that is based on extraction of moment invariant features from concentric circular regions and doesn't require any specific transform (Approach 2). While both approaches provide the desired robustness against geometric modifications, Approach 1 is computationally more intensive than Approach 2 as the SVD bases are updated for every input frame at 12fps. It also requires a longer query clip than Approach 2 for reliable identification. We present results comparing the performance of both of these approaches for a 150hr video database.

Keywords: Video fingerprinting, Singular Value Decomposition, Moment Invariants, Robust Perceptual Hashing

1. INTRODUCTION

A video fingerprint is a compact bitstream representation of the underlying content that is robust against many signal processing operations on the content and can be used to uniquely identify that content. One of the main challenges of a video fingerprinting method is to provide robustness against geometric attacks such as rotation, aspect ratio change and cropping. These modifications are particularly difficult as there is either a loss of information (e.g cropping) or a loss of registration (e.g rotation,cropping) between modified content and original content. There have been lot of approaches proposed in literature to tackle these challenges. In this paper, we review the proposed methods in video fingerprinting literature under two broad approaches: (i) Video fingerprints based on transform domain features (ii) Video fingerprints based on local features. In the first approach, the input video is first transformed to a new domain (e.g Radon transform domain[1]) to obtain a geometric attack invariant representation. Then, features are extracted from this domain and are converted to a bitstream representation. In the second approach, robust local feature points (e.g Harris corners[2]) are first detected in the spatial domain, spatio-temporal domain or any transform domain. Then, these features are converted to fingerprint bits. These two approaches lead us to the following two questions: (i) can we obtain a geometric attack invariant representation of the video data using a data-dependent transform? (ii) can we obtain a geometric attack invariant representation using a semi-global representation of video data ?. In this paper, we present two approaches that address these questions. Approach 1 is based on estimation of Singular Value Decomposition (SVD) bases from a window of past video frames and Approach 2 that is based on extraction of moment invariants from concentric circular regions of the video frame. We present results comparing the performance of both of these approaches for a 150hr video database. We show that while both approaches provide the desired robustness against geometric modifications, Approach 1 is computationally more intensive than Approach 2 as the SVD bases are updated for every input frame at 12fps. Approach 1 also requires a longer query clip than Approach 2 for reliable identification.

Further author information: (Send correspondence to Regunathan Radhakrishnan)
E-mail: regu.r@dolby.com, Telephone: 1 415 558 0104

Media Forensics and Security, edited by Edward J. Delp III, Jana Dittmann, Nasir D. Memon, Ping Wah Wong, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 7254, 725407 · © 2009 SPIE-IS&T · CCC code: 0277-786X/09/$18 · doi: 10.1117/12.805627

The rest of the paper is organized as follows. In section 2, we present a review of some the representative work under the two broad approaches in literature for invariance against geometric modifications. In section 3, we present a description of our SVD based video fingerprint extraction method as an example of data-dependent transform based fingerprinting. In section 4, we present a description of our proposed moment invariants based video fingerprint as an example of semi-global representation of video content. In section 5, we present experimental results comparing the proposed two approaches on a 150hr video database.

2. REVIEW OF PRIOR ART

2.1 Video Fingerprints based on Transform Domain Features

In this approach, in order to be robust against geometric operations on the video, robust features are extracted from a transform domain that is invariant to geometric operations on the video. First, we give two examples of video fingerprinting methods that are based on Radon transform representation.The radon transform domain representation of an image is especially useful in dealing with geometric operations on the image including translation, scaling and rotation. All of these operations can be tracked in the radon transform domain and can be accounted for in a better manner than they can be handled in the original spatial domain. Seo et al propose an image fingerprinting method based on Radon transform of the input image.[1] The Radon transform of an image $f(x, y)$ denoted as $g(s, \theta)$, is defined as its line integral along a line inclined at an angle θ from the y-axis and at a distance s from the origin. The proposed method uses three properties of the Radon transform to extract fingerprints. First, translation of the input image causes the radon transform to be translated as well in the direction of s. In order for the feature to be translation invariant, the authors perform normalized autocorrelation. Second, scaling of the input image causes the corresponding Radon transform to be scaled by same factor. In order for the feature to be scale invariant, the authors perform log-mapping which makes the scaling operation manifest as a shift in the direction of s. Third, rotation of the input image causes the Radon transform to be shifted by the same amount in the direction of θ. In order to provide rotation and scale invariance for the features, the authors perform a 2D-FFT after autocorrelation and log-mapping. The shifts in the direction of s and θ due to scaling and rotation are captured in the phase information after 2D-FFT. The hash is derived by thresholding local energy differences of the absolute values of the 2D-FFT. Since the the hash bits are extracted from a representation that is invariant to rotation, it doesn't matter by how many degrees the input image is rotated. Lefebvre et al propose an image hashing method based on Radon transform and Principal Components Analysis (PCA).[2]

Second, we give two example methods that are based on Discrete polar 2D-FFT representation of the image data. This transform domain representation is also useful for extracting features that are invariant to translation and rotation and is computationally less demanding than the Radon transform. Swaminathan et al propose an image hashing method based on the discrete polar FFT of the input image.[3] The input image is first transformed to the Fourier domain. Since the magnitude of the fourier transform is invariant to the translation of the input image, only the magnitude of the coefficients is retained for further analysis. Then, the magnitude fourier coefficients are expressed in polar coordinates. A rotation attack would only cause a shift in θ direction of the polar coordinates representation. To obtain a feature vector invariant to rotation, the fourier coefficients along the θ axis are summed at particular radius. Finally, robustness is achieved by retaining only significant low-frequency coefficients. The resulting feature vector is quantized to obtain a bitstream representation. Mavandadi et al propose an image feature from the same representation domain that is invariant to rotation attacks.[4]

2.2 Video Fingerprints based on Local Features

In this section, we review video fingerprint extraction methods that are based on detection of robust local features ("significant" feature points). Then, each video frame is represented as a collection of these feature points. These feature points are indexed using the descriptors of their local neighborhood. Any global geometric transform such as rotation or aspect ratio conversion or cropping would preserve a large number of these "significant" feature points. Therefore, the robustness is provided by the redundancy introduced by representing the video frame as a collection of these "significant" feature points. Perceptually similar images would result in the same set of "significant" feature points. In fact, these features have been successfully used in computer vision for view-independent object recognition and also for image registration. Although, this method would increase the

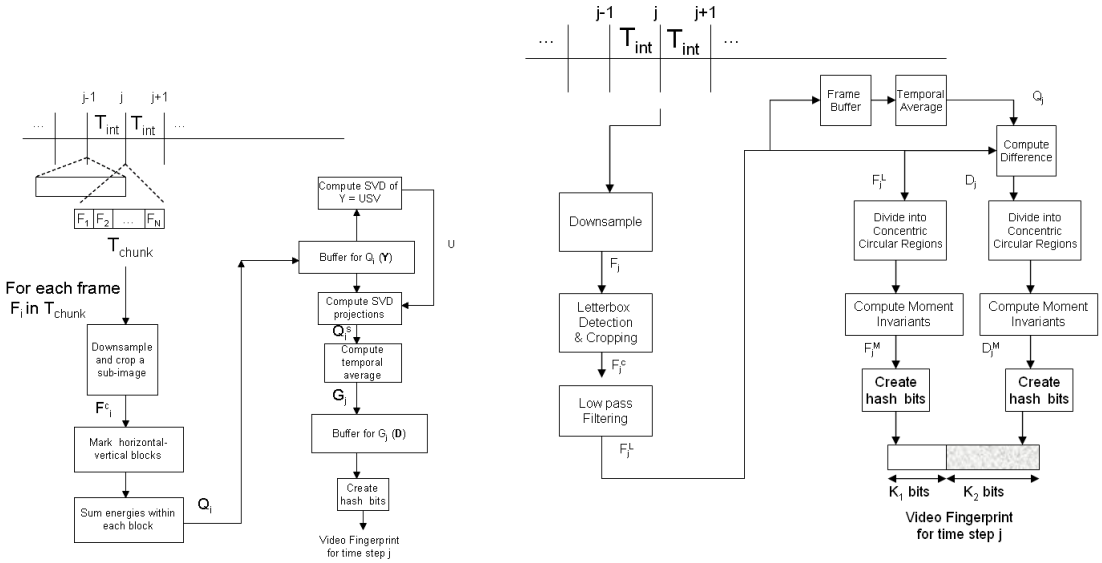


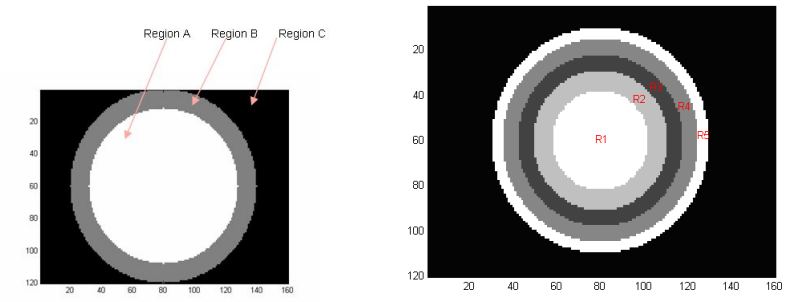
Figure 1. (a) Video Fingerprint Extraction based on SVD: Approach 1; (b) Video Fingerprint Extraction based on Moment Invariants: Approach 2 .

number of fingerprint bits generated per video frame, it is the most robust method against cropping and other geometric attacks. The robustness is governed by the number of local features indexed per video frame. We describe two methods in literature that are examples of this method and use different methods to detect these significant feature points. In,[5] Lu et al propose a robust mesh-based image hashing method. The first step in this method is to detect Harris corner from a coarse representation of the input image. The coarse representation is obtained from LL subband of a wavelet transform. Then, they use Delaunay tessellation to decompose the image into a set of disjointed triangles. Each triangle is the smallest unit of the whole mesh on the image. Fingerprint bits are extracted from each triangle. In,[6] use a local spatio-temporal feature detector and index the video as a collection of these feature points based on the description of the spatio-temporal neighborhood of each feature point. One can also use (Scale Invariant Feature Transform) SIFT as local features and index them.[7]

3. ROBUST VIDEO FINGERPRINTS BASED ON DATA-DEPENDENT TRANSFORM: APPROACH 1

In this section, we describe a robust video fingerprint extraction method based on a data-dependent transform.[8] This method is an example of an approach that derives a geometric attack invariant representation based on a data-dependent transform. This doesn't use any specific transform such as Radon transform, Fourier Mellin or Discrete Polar 2D-FFT but derives the bases for the data-dependent transform adaptively from a neighboring group of video frames for each time step. Using the basis vectors obtained from Singular Value Decomposition (SVD) of this group of frames, we first obtain subspace representation for the frames. Then, we extract fingerprint bits by projecting a temporal average of these representations onto pseudo-random basis vectors. Since the subspace is estimated from the input video data itself, any global attack on video such as rotation or cropping would result in a corresponding change in estimated basis vectors thereby making the subspace representation robust against these modifications. One of the challenges of this method is that it is computationally expensive to adaptively estimate the SVD bases for each time step. To deal with this, instead of performing batch SVD for each time step, we update the SVD basis for the next time step using an incremental SVD update procedure proposed in.[9]

Figure 1(a) illustrates the steps in the proposed method for video fingerprint extraction. The first two steps in the proposed method ensure resilience against frame rate conversion attacks by first downsampling the input video to a reference frame rate (say 12fps) and then by extracting fingerprints by summarizing information from



(a) (b)

Figure 2. (a) Region used for fingerprint extraction based on SVD: Approach 1; (b) Concentric Circular Regions for Moment Invariants: Approach 2.

a group of frames rather than from individual frames. Also, extracting fingerprints at a reference rate (12fps) makes comparing fingerprints of two videos at different frame rates easier than resorting to dynamic programming for comparing signature sequences of different lengths as in.[10]

Step 1: We divide the video in intervals of length T_{int} and associate a fingerprint codeword with each time interval. T_{int} is derived from the smallest frame rate conversion that we would like the fingerprint be robust against. For example, if the original video is at 30 fps and we would like the fingerprint to be robust for frame rate conversion down to 12fps, then we extract fingerprints every $T_{int} = \frac{1}{12}$ of a second.

Step 2: For each time interval T_{int} (see Figure 1), we select a group of frames around the time interval with a combined length of $T_{chunk} > T_{int}$. We refer to this group of frames as $(F_1, F_2, .., F_N)$. Here $N = \frac{T_{chunk}}{T_{int}}$.

Step 3: Then, for every frame F_i we perform the following two steps to ensure resilience against addition of graphics and logos along the boundaries of video frames and cropping during rotation attacks. First, we downsample the image to a reference 120×160 resolution and crop out a circular region for fingerprint extraction as shown in Figure 2(a). This ensures that input images are in a common subspace of dimension 120×160. From each image F_i, only region A is used for signature generation. This circular region is the only region of the image that would survive all rotations of the input image. Region C is excluded as rotation would cause pixels in those regions to go out of the viewing area. Region B is excluded so as to allow for text overlay in the bottom one tenth of the original picture and to allow for placement of logo or graphics around the corners. We denote this cropped out image as F_i^c. Second, we obtain a coarse representation (Q_i) of F_i^c by dividing F_i^c into $M_1 * M_2$ blocks and averaging the intensity within each block as given below:

$$Q_i(k,l) = \frac{1}{W_x * W_y} \sum_{m=(k-1)W_x}^{kW_x} \sum_{n=(l-1)W_y}^{lW_y} F_i^c(m,n) \tag{1}$$

$$k = 1, 2...M_1; l = 1, 2...M_2$$

Here 'm' and 'n' represent the indices for the horizontal and vertical dimensions for the image F_i^c and $F_i^c(m,n)$ is the intensity at that location. 'k' and 'l' represent the indices of the coarse image Q_i. $W_x * W_y$ is the number of pixels in each of the blocks that are averaged to get one element $Q_i(k,l)$.

Step 4: Let us create a vectorized representation of Q_i and denote it as Q_i^v. Q_i^v is a vector of dimension $(M_1 * M_2) \times 1$ which is obtained from the matrix Q_i of dimension $M_1 \times M_2$ by scanning the entries of the matrix in row by row to convert it into a vector. Q_i^v is sensitive to geometric attacks and hence cannot be directly used for extracting fingerprint bits. For instance, if the original video is rotated, the values of $Q_i(k,l)$ will change. In order to obtain a representation of Q_i^v that is invariant to geometric attacks, we represent Q_i^v in a subspace using basis vectors that are estimated from the sequence $(Q_1^v, Q_2^v, .., Q_N^v)$ itself. For instance, in case of a rotation, this ensures that the basis vectors are rotated accordingly thereby preserving the subspace representation of Q_i^v. Let us represent the basis vectors that span the set $(Q_1^v, Q_2^v, .., Q_N^v)$ as $(B_1, B_2, .., B_N)$. Now, let us obtain

the coordinates of Q_i^v in the new space spanned by $(B_1, B_2, .., B_N)$ by projecting Q_i^v onto each of the basis vectors. Let us represent these projections as $\overrightarrow{Q_i^s} = (Q_i^{s,1}, Q_i^{s,2}.., Q_i^{s,N})$. Note that Q_i^v is a vector of dimension $M_1 * M_2 \times 1$ and is now represented by $\overrightarrow{Q_i^s}$ a vector of dimension N in the new space spanned by $(B_1, B_2, .., B_N)$. The new representation $\overrightarrow{Q_i^s}$ is invariant to geometric attacks. This is because the basis vectors $(B_1, B_2, .., B_N)$ are estimated from $(Q_1^v, Q_2^v, .., Q_N^v)$. Therefore, if the original video is rotated, each Q_i^v is rotated accordingly and so are the basis vectors obtained from them.

Now, we describe how we obtain the basis vectors $(B_1, B_2, .., B_N)$ from $Q_1^v, Q_2^v, .., Q_N^v$ for each time step. We create a matrix Y in which each column (j) represents a frame Q_j^v. The number of rows is $(M_1 * M_2)$ (same as the number of elements in Q_j^v). The dimensions of this matrix are $(M_1 * M_2) \times N$ (Here $N << (M_1 * M_2)$). The rank of matrix Y is utmost N and we obtain the basis vectors $(B_1, B_2, .., B_N)$ using the SVD of $Y = USV$. By the definition of the SVD, the columns of U that span the column space of Y are the basis vectors $(B_1, B_2, .., B_N)$.

Here U is of dimension $(M_1 * M_2) \times N$, S is of dimension $N \times N$ and V is of dimension $N \times N$. S is the diagonal matrix with singular values in the order of decreasing magnitudes. The columns of V form the basis vectors which span the rows of Y. Using the basis vectors $(B_1, B_2, .., B_N)$ obtained through SVD, we obtain the coordinates of Q_i^v in the new space $Q_i^s = U' Q_i^v$.

Performing SVD of Y for each time step is computationally expensive and its time complexity is $O(((M_1 * M_2)^2 N) + (N^2(M_1 * M_2)) + N^3)$.[9] Recall that Y is of size $(M1 * M2) \times N$ and each column of Y has elements of Q_i $(i = 1, 2...N)$. This means that for the next time interval the first column of Y gets removed and a new column is added. Therefore, instead of computing the SVD of Y for the current time step, we could incrementally update the matrices U,S and V obtained from the previous time interval. The incremental procedure in[9] has a time complexity of $O((M_1 * M_2)N * N)$. Since N (e.g $36 = 3 \times 12$, if we use a 3s window) is usually very small compare to $M_1 * M_2$ (e.g 44*60) this reduces computationally complexity to a large extent.

Step 5: In the previous step, by computing $\overrightarrow{Q_i^s}$ we have obtained a representation for Q_i which is invariant to geometric attacks. In this step, we compute the temporal average of the new coordinates $(\overrightarrow{Q_i^s})$. This step ensures that extracted features are not dependent on values from individual frames but are derived from a group of frames thereby providing robustness against frame rate conversion. The temporal average G of $(Q_1^s, Q_2^s, ...Q_N^s)$, is computed as shown below

$$G(l) = \frac{1}{N} \sum_{i=1}^{N} Q_i^s(l) \tag{2}$$
$$l = 1, 2...N$$

We select the top L values of G for the recent R time intervals and store this in a buffer D. Then, D is a matrix (of size $R \times L$) which summarizes how the L projections vary over R time intervals.

Step 6: Finally, we create K fingerprint bits from the matrix D by projecting it onto random basis vectors as in.[11] This hash bit extraction method proposed in[11] was originally applied for generating robust hashes from images. First, we create the K random basis vectors $(P_1, P_2, ...P_K)$ that have the same dimension as D. Then, we compute the mean of these random vectors and subtract them from the respective vectors. Finally, the matrix D is projected onto this set of K vectors. The fingerprint bits are then derived by comparing the K projections to a threshold defined as their median.

4. ROBUST VIDEO FINGERPRINTS BASED ON MOMENT INVARIANTS: APPROACH 2

In this section, we propose a video fingerprint extraction method which extracts moment invariants as statistics from concentric circular regions in the decoded video frame that are particularly robust against such geometric operations on the content. This method is an example of an approach that provides robustness against geometric attacks without performing any specific transform. This method just uses moment invariants as semi-global features derived from concentric circular regions for generating the fingerprints. The extracted fingerprint captures

how these statistics change as one proceeds from an inner region to an outer region. The following are steps involved in this fingerprint extraction method. Figure 1(b) illustrates the steps in the proposed method.

Step 1: The input video is first temporally downsampled to a reference frame rate. This makes comparison of signatures easier when the original video and the processed video do not have the same frame rate. This means that we extract signature at certain time interval (T_{int}) only.Let F_j represent the closest video frame for time step 'j'.

Step 2: The frame F_j is downsampled to reference spatial resolution say, (120*160). This helps in dealing with spatial resolution changes. The registration between the original video and spatially scaled video is not disturbed as long as the aspect ratio is not changed. Then, we perform letterbox detection and removal from the input frame F_j.Once we detect the letterbox, we remove it and upsample the active region of the frame to the chosen spatial resolution (120*160)($Height \times Width$).

Step 3: In this step, a sub-image is cropped out as shown in Figure 2(b) from the downsampled F_j image. This region is selected so as to allow for text overlay in a portion of the original picture and to allow for placement of logo or graphics around the corners. The fingerprint of the processed video will not be affected as long as the selected region does not contain any new graphics content. Let us represent this image as F_j^c.

Step 4: A low-pass filtering operation is performed on F_j^c to improve the robustness of extracted features. A simple low-pass filter using the average of $3*3$ neighborhood of the current pixel could be used. Let us represent the low pass filtered image by F_j^L.

Step 5: The low-pass filtered image F_j^L captures the spatial configuration of pixels (objects in the current video frame). In order to capture information about motion of objects in the video, we create a difference image D_j. The difference image is computed according to: $D_j = Q_j(k,l) - F_j^L(k,l); k = 1, 2...H; l = 1, 2...W$;Here W is the width and H is the height of F_j^L. In our implementation, they are set to 160 and 120 respectively. And Q_j is the temporal average image obtained by averaging pixels from a window of past 'T' decoded frames. Q_j is computed according to $Q_j(k,l) = \frac{1}{T}\sum_{i=j-T}^{j} F_i^L(k,l)$. The motivation for computing D_j as a difference between the current frame (F_j^L) and a temporally averaged image (Q_j) is the following. If the difference is computed from just one previous decoded frame then the values of Q_j are susceptible to change under frame rate conversion attacks. The temporal averaging operation prevents dependence on just one frame. At the end of this step, we have two matrices from which features are to be extracted: one capturing the appearance of the current decoded frame (F_j^L) and another capturing the motion aspects of the current frame (D_j).

Step 6: We create regions in the two matrices (F_j^L and D_j) before extracting features from each of the regions. Figure 2 shows an example of a case where we have five concentric regions $R_1, R_2...R_5$ with radii $r_1, r_2, ...r_5$ respectively. We pick the inner region R_1 and select the other regions such that area increments are equal to the area of region R_1. One advantage of these concentric circular regions is that the content of the image within each region stays the same irrespective of the amount of rotation.

Step 7: In this step, we extract 'M' features from each of the 'N' regions demarcated in the previous step. The feature matrix derived from the regions of F_j^L is denoted as F_j^M. F_j^L is of dimension $H \times W$ and F_j^M is of dimension $N \times M$ where H and W represent the height and width of the downsampled video frame and N and M represent the number of regions and the number of features extracted from each region. Similarly, the 'M' features are extracted from 'N' regions of the matrix D_j to create D_j^M. The robustness of the extracted fingerprints depend on the robustness of the M features extracted from each of the regions.

We compute a set of seven moment invariants as semi-global features from these regions. This set of moment invariants are global measures of the image surface that are robust against translation, rotation and scale change attacks. This set of moment invariants were originally proposed by Hu in 1962 for recognition of characters.[12] Please see[12] for details on the expressions for the seven moment invariants. The moment invariants computed from each region of F_j^L forms the rows of the matrix F_j^M. Similarly, the moment invariants computed from each region of D_j forms the rows of the matrix D_j^M. In our particular implementation, H=120, W=160, N=5 and M=7 (5 regions and seven moment invariants).

Step 8: In this step, we have two input matrices F_j^M and D_j^M each representing how the extracted features from each region change as one proceeds from the inner region R1 to the outer region RN. The signature generation procedure is identical for both matrices (F_j^M and D_j^M). Therefore, we explain this bit extraction procedure for one of them, say, F_j^M. In order to generate K1 bits from F_j^M, we first create K_1 vectors $(P_1, P_2...P_{K1})$ that have the same dimension as F_j^M. The matrix F_j^M is projected onto this set of K_1 vectors as shown in the equation below:

$$H_k = \sum_{i=1}^{N} \sum_{j=1}^{M} Q(i,j) * P_k(i,j) \tag{3}$$

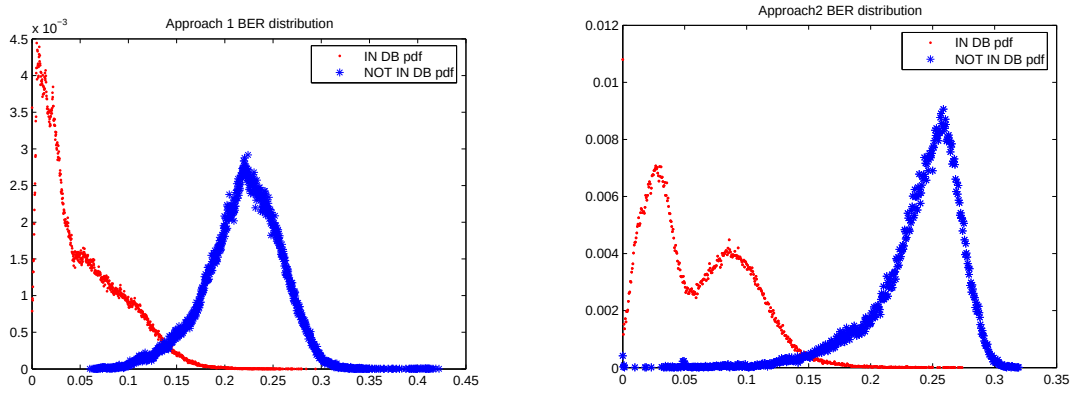
Here, the matrix Q is either F_j^M or D_j^M and M is the number of features per region and N is the number of regions. The signature bits are then derived by thresholding the K_1 projections. It is desirable that the projections based on the set of K_1 vectors $(P_1, P_2...P_{K1})$ capture different aspects of the matrix F_j^M. For example, if any two of the K_1 vectors are similar, then 2 bits out of the K_1 bits will be identical. One possible solution to avoid this problem is to use an orthogonal basis set of K_1 vectors. Another possible solution is to use a set of K_1 pseudo-random vectors. The assumption is that the K_1 pseudo-random vectors are approximately orthogonal to each other. In our particular implementation, we create K_1 hash bits from F_j^M based on projections onto K_1 pseudo-random vectors. Similarly, we create K_2 hash bits from D_j^M bits. Then, finally our fingerprint is of length $(K_1 + K_2)$ bits and is the concatenation of the two sets of hash bits from F_j^M and D_j^M. The pseudo-random matrices for generating K_1 hash bits and the pseudo-random matrices for generating K_2 hash bits are different. In our implementation, K_1 and K_2 were set to be 18.

5. EXPERIMENTAL RESULTS

In this section, we present experimental results on the performance of the proposed two approaches. Approach 1 performs fingerprint extraction using a data-dependent transform representation and is computationally more expensive than Approach 2. Approach 2 performs fingerprint extraction based on moment invariants from concentric circular regions and is computationally less expensive. We compared the performance of these two approaches in terms of robustness against attacks and sensitivity to content. We created a 150hr database of fingerprints from a dataset of reference videos using both methods (Approach 1 and Approach 2). We created modified versions of some of the reference clips to be used as query videos. The modifications included non-geometric attacks such as compression, spatial scaling, frame-rate conversion and also the geometric attacks such as rotation, aspect ratio conversion and cropping. The number of hours of query videos was about 55hrs and was used to illustrate the robustness of the proposed methods against both geometric and non-geometric attacks. We also set aside about 22hrs of query content that was not part of the reference database to illustrate the sensitivity of the proposed methods to content. We implemented the nearest neighbor search based on a 256-ary tree originally proposed for audio fingerprinting search[13] for our experiments.

First, we present the results on the sensitivity property of the two methods. From the 55hrs of modified query video fingerprints, we performed a query for every 8s of query video amounting to a total close to 300,000 queries. For every query fingerprint, we record the percentage of bit errors (BER) between the query fingerprint and matching fingerprint in the reference database. Then, we compute the probability distribution of the BER for this dataset. Let us denote this distribution as "IN DB pdf". Now, we perform close to 100,000 queries from the 22 hrs of content that is not part of the reference database of content and again record the BER between the query fingerprint and closest matching fingerprint in the reference database for this experiment as well. Then, we compute the probability distribution of the BER for this dataset. Let us denote this distribution as "NOT IN DB pdf".

Figure 3 compares the "IN DB pdf" and "NOT IN DB pdf" for both Approach 1 and Approach 2. Notice that in both cases the two pdfs have little overlap. The BER of modified versions are smaller than the BER of content that is not part of the reference database. This implies that for any input query, by comparing the BER of the matching fingerprint to a threshold one can declare the query to be a modified version of a reference video clip (if the BER is less than the chosen threshold) or declare the query to be a clip that is not in the



(a) (b)

Figure 3. Comparison of BER distributions for Approach 1(a) and Approach 2(b); x-axis:BER y-axis:density;

reference database (if the BER is greater than the chosen threshold). In the case of Approach 1, by choosing a threshold of 0.15 it is possible to account for 98.31% of modified content. This is the cumulative probability in "IN DB pdf" of Approach 1 that is below 0.15 BER threshold. The remaining 1.69% is equal to the probability of miss (i.e the query is a modified version of a reference clip and we declare it as not present in the database). Similarly, one can compute probability of false alarm (cumulative probability in "NOT IN DB pdf" that is below 0.15) and for Approach 1 it was found to be 5.11%. We compute the probability of miss and probability of false alarm from the BER distributions of Approach 2 in a similar fashion and they were found to be 1.49% and 2.47% respectively. Based on these measures, we can see that Approach 2 slightly outperforms Approach 1 in terms of sensitivity. Note that Approach 1 is computationally more expensive and requires a total of 14s of query video for each query to perform the matching (120 codewords of fingerprint at 12fps and 4s of video for SVD basis estimation and averaging). On the other-hand, Approach 2 is computationally less expensive and requires only 5.5s of query video to perform the matching (54 codewords of fingerprint at 12fps and 1s of video for computing the average difference image).

Second, we present the results on the robustness property of the two methods. From the 55hrs of modified query video fingerprints, we performed close to 300,000 queries for every 8s of video. For every query fingerprint, we record the percentage of bit errors (BER) between the query fingerprint and matching fingerprint in the reference database. Table 1 presents the comparison of the performance of two approaches in terms of average BER for non-geometric attacks (compression, spatial scaling, frame rate conversion), rotation attacks (2,3,10 and 45 degrees of rotation) and aspect ratio conversion attacks (4:3 to 16:9 aspect ratio change)

Table 1. Average BER for Approach 1 and Approach 2 in case of non-geometric attacks, rotation and aspect ratio conversion

	Non-Geometric Attacks	Aspect Ratio	Rotation
Approach 1	0.049657	0.1067	0.05324
Approach 2	0.06037	0.11506	0.0832

Note that in terms of robustness against Non-geometric attacks as well as rotation and aspect ratio conversion, Approach 1 outperforms Approach 2 and has lower average BER in all cases. Also, the average BER in all these attacks is smaller than the chosen threshold of 0.15 for both approaches. Next, we present the performance of both the approaches for another geometric attack (cropping) in Table 2. Here, we record the average BER for increasing amount of cropping from the edges of a picture. In this Table, the cropping amount is expressed in terms of percentage as $\frac{\pi(r_2^2-r_1^2)}{\pi r_2^2}$. Here r_2 is the radius that covers the whole picture and the region between r_2

and r_1 is the region that is cropped out. As one increases the cropping percentage from 0% to 14.44% and all the way up to 55.55%, the average BER for Approach 1 increases gracefully from 2.5% to 7.22% and to 16.76% respectively. For the same amount of increases in cropping percentage, the average BER for Approach 2 also increases gracefully from 4.6% to 12.38% and to 24.87% respectively. However, the average BER for Approach 2 degrades less gracefully than for Approach 1. In otherwords, Approach 1 is more robust against cropping attack than Approach 2. This is as expected because Approach 2 attempts to capture how moment invariants change from the inner circular region to the outer circular region. For the cropped video, the boundaries of these circular regions are at different locations than they were for the original reference video clip.

Table 2. Graceful degradation of average BER for cropping attacks (Approach 1 and Approach 2)

Cropped area ratio	no cropping	0.1444	0.3055	0.4375	0.5555
Approach 1	0.02518	0.0722	0.1082	0.1398	0.1676
Approach 2	0.04639	0.1238	0.1807	0.2201	0.2487

5.1 Discussion

Based on our experimental results, we can conclude that both the approaches are robust against geometric and non-geometric modifications on content. For cropping attack, since we only extract one fingerprint codeword from a decoded video frame at 12fps, we can only hope for graceful degradation of BER as one increases the cropping percentage. As one increases the cropping percentage, there is loss of information and any fingerprinting method that one derives just one fingerprint codeword from all the features in a frame would be more sensitive to cropping attack than other attacks. This is the case for both of our approaches. Also, Approach 2 is more sensitive to cropping as the boundaries of the circular regions have to be the same for the attacked video clip and the reference video clip. In the case of Approach 1, the computed SVD basis adapts to the global cropping operation and hence the BER degrades more gracefully than Approach 2. Unlike Approach 1 and Approach 2, if the fingerprint were derived from a list local features identified from a decoded video frame as in,[6] then there would be many redundant entries in the database for a particular video frame. This would translate into more robustness against cropping as atleast some of the local features are likely to survive the cropping operation. Also, the registration between the attacked video and reference video is less important as each frame is only represented as a bag of fingerprint codewords (derived from corresponding local features).[6]

6. CONCLUSIONS

In this paper, we reviewed two major approaches to video fingerprinting methods robust to geometric attacks: (i) Robust feature extraction from a transform domain such as Radon transform or Discrete Polar 2D-FFT; (ii) Robust local feature extraction based. The review led us to the following two questions: (i) can we obtain a video fingerprint robust to geometric attacks based on a data-dependent transform? (ii) can we obtain a video fingerprint robust to geometric attacks based on semi-global features and without using any transform?. To address the first question, we presented a SVD based video fingerprint extraction method (Approach 1). The SVD bases were estimated using data from a group of frames and were updated incrementally for each time step in the video. For any global attack, the SVD bases adapted accordingly and hence the sub-space representation was invariant to geometric attacks. Then, in order to address the second question, we proposed a moment invariants based video fingerprint extraction method (Approach 2). In this approach, each video frame is divided into 5 concentric circular regions and seven moment invariants were computed from each region. These moment invariants are global measures that characterize the image surface in a region and are invariant to scaling, translation and rotation. Finally, we showed the effectiveness of both the approaches in terms of robustness against geometric and non-geometric attacks on a 150hr database. Approach 1 has better robustness against cropping and other geometric attacks than Approach 2 but is computationally more expensive than Approach 2. Also, Approach 1 requires a longer query video clip than Approach 2.

REFERENCES

[1] J.S.Seo, J.Haitsma, T. and C.D.Yoo, "A robust image fingerprinting system using the radon transform," in [*Signal Processing:Image Communication*], *Proc. of ACM MM* **19**, 325–339 (2004).

[2] F.Lefbvre, J. and B.Macq, "A robust soft hash algorithm for digital image signature," *Proceddings of European Signal Processing Conference* (2002).

[3] A.Swaminathan, Y. and M.Wu, "Image hashing resilient to geometric and filtering operations," *Proc. of MMSP* (2004).

[4] S.Mavandadi and P.Aarabi, "Rotation invariance in images," *Proc. of ICASSP* (2007).

[5] C. Lu, C.Y.Hsu, S. and Chang, P., "Robust mesh based hashing for copy detection and tracing of images," *Proc of ICME* (2004).

[6] G.Willems, T.Tuytelaars, L., "Spatio-temporal features for robust content-based video copy detection," *Proc. of ACM MM* (2008).

[7] D.G.Lowe, "Distinctive image features from scale invariant keypoints," in [*IEEE Journal on Computer Vision*], (2004).

[8] R.Radhakrishnan and C.Bauer, "Robust video fingerprinting based on subspace embedding," *Proc. IEEE ICASSP* (2008).

[9] M.Brand, "Fast low rank modifications of the thin singular value decomposition," in [*Linear Algebra and its Applications*], **415**, 20–30 (2006).

[10] X.S.Hua, X. and H.J.Zhang, "Robust video signature based on oridnal measure," *Proc. of IEEE ICIP* (2004).

[11] J.Fridrich and M.Goljan, "Robust hash functions for digital watermarking," *Proc. of ITCC* (2000).

[12] Hu, M., "Visual pattern recognition by moment invariants," *IRE Trans. Info. Theory* **IT-8**, 179–187 (1962).

[13] M. L. Miller, M. A. R. and Cox, I. J., "Audio fingerprinting: nearest neighbor search in high dimensional binary spaces," *J. of VLSI Signal Processing* (2005).