ROBUST VIDEO FINGERPRINTS BASED ON SUBSPACE EMBEDDING

Regunathan Radhakrishnan and Claus Bauer

100 Potrero Ave, San Francisco CA 94103
regu.r@dolby.com, cb@dolby.com

ABSTRACT

We present a novel video fingerprinting method based on subspace embedding. The proposed method is particularly robust against frame-rate conversion attacks and geometric attacks among other attacks including compression and spatial scaling. Using a sliding window, we extract fingerprints from a group of subsequent video frames. For the generation of the fingerprints, we first calculate the basis vectors of a coarse representation of this group of frames using a Singular Value Decomposition (SVD). Then, we project the coarse representation of the video frames onto a subset of the basis vectors. Thus, we obtain a subspace representation of the input video frames. Finally, we extract the 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 would result in a corresponding change in estimated basis vectors thereby preserving the subspace representation. We present experimental results on 250hrs of video to show the robustness and sensitivity of the proposed signature extraction method.

Index Terms— Singular Value Decomposition, Robust Video Fingerprints, Geometric attacks

1. INTRODUCTION

Robust fingerprinting methods create a compact bitstream representation of the underlying content so as to enable content identification applications. The extracted fingerprints should remain similar even after the original content has been modified by common signal processing operations.Geometric attacks and frame rate conversion attacks are the most challenging of all the modifications that a video fingerprint method should be robust against.

Past work on geometric attacks for image fingerprinting is mostly based on a transform that is invariant to geometric attacks. In [1], Swaminathan et al propose a image hash that is based on a discrete polar Fourier Transform. The method derives features invariant to geometric attacks based on the fact that the magnitude of Fourier coefficients is independent of translation and - when expressed in polar coordinates - the coefficients are shifted if the underlying image is rotated. In [2],[3], an image hash algorithm was proposed based on the properties of the Radon transform. In [2], Principal Components Analysis (PCA) is used to extract features from Radon transform coefficients. In [3], the Radon transform is combined with a an autocorrelation step and 2 dimensional FFT. The fingerprints are extracted based on the sign of local energy differences between the coefficients. In [4], the authors propose an image hashing method resilient to image rotation based on matrix invariants derived from the Singular Value Decomposition (SVD) of pseudo-random tiles in the image. In [5], a mesh based image hashing method is proposed. The robustness of this method relies on the robust detection of Harris corners to form the mesh. Aforementioned geometric attack invariant image hash methods could be applied to individual decoded video frames as well. However, for resilience against frame rate conversion attacks it is desirable to create fingerprints from group of video frames rather than from individual video frames. Further, the application of these methods to the creation of video fingerprints would not be practical due to the high complexity of the aforementioned methods.

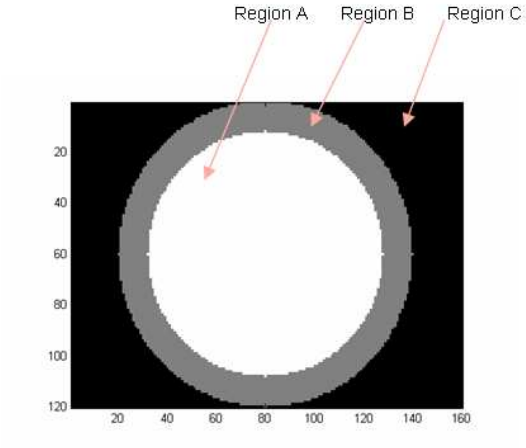

Fig. 1. Region used for fingerprint extraction

In this paper, we propose a novel video fingerprint extraction method based on subspace embedding. Using a sliding window, we extract fingerprints from a group of subsequent video frames. For the generation of the fingerprints, we first calculate the basis vectors of a coarse representation

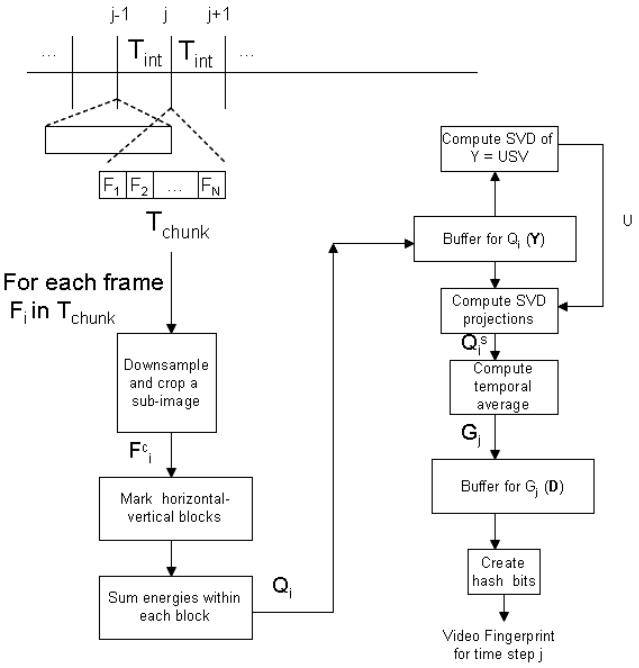

Fig. 2. Video Fingerprint Extraction based on SVD

of this group of frames using a Singular Value Decomposition (SVD). Then, we project the coarse representation of the video frames onto a subset of the basis vectors. Thus, we obtain a subspace representation of the input video frames. Finally, we extract the fingerprint bits by projecting a temporal average of these representations onto pseudo-random basis vectors. 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 would result in a corresponding change in estimated basis vectors thereby preserving the subspace representation. Since performing SVD for each time step is computationally intensive, we update the SVD basis for the next time step using an incremental SVD update procedure proposed in [6].

2. PROPOSED METHOD

Figure 2 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 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 [7].

- We divide the video in intervals of length T_{int} and associate a fingerprint 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.

- For each time interval T_{int} (see Figure 2), 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 $N = \frac{T_{chunk}}{T_{int}}$.

- 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 1. 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:

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

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

  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 $Q_i(k,l)$.

- Q_i computed in the previous step 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 that is invariant to geometric attacks, we represent Q_i in a subspace using basis vectors that are estimated from the sequence $(Q_1, Q_2, .., Q_N)$ 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. Let us represent the basis vectors that span the set $(Q_1, Q_2, .., Q_N)$ as $(B_1, B_2, .., B_N)$. Now, let us obtain the coordinates of Q_i in the new space spanned by $(B_1, B_2, .., B_N)$ by projecting Q_i 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 is a vector of dimension $M_1 * M_2$ 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 $(Q_1, Q_2, .., Q_N)$. Therefore, if the original video is rotated, each Q_i 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 $Q_1, Q_2, .., Q_N$ for each time step. We create a matrix Y in which each column (j) represents a frame Q_j. The number of rows is $(M_1 * M_2)$ (same as the number of elements in Q_j and scanned row by row). The dimensions of this matrix are $(M_1 * M_2) \times N$ ($N << (M_1 * M_2)$). 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 as USV. By 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 in the new space $Q_i^s = U' Q_i^v$.Here 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.

Performing SVD of Y for each time step is computationally expensive and its time complexity is $O(((M_1 * M_2)^2 N) + (N^2 (M_1 * M_2)) + N^3)$ [6]. 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 [6] has a time complexity of $O((M_1 * M_2) N * N)$. Since N (e.g 36) is usually very small compare to $M_1 * M_2$ (e.g 44*60) this reduces computationally complexity to a large extent.

- 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

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

$$l = 1, 2...N$$

  In order to do so, 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.

- Finally, we create K fingerprint bits from the matrix D by projecting it onto random basis vectors as in [8]. This hash bit extraction method proposed in [8] 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.

3. EXPERIMENTAL RESULTS

In this section, we present experimental results on proposed video fingerprint extraction method to show its robustness against geometric attacks, frame rate conversion, compression and spatial scaling. We created a video database of 250 hrs and extracted fingerprints for the original, MPEG-2 encoded video. T_{chunk}, T_{int} M_1, M_2,N,K,L,R and the pseudo-random matrices are fingerprint extraction parameters that need to be the same for both the original and modified video. They were set to be the following: $T_{chunk} = 3s$,$T_{int} = \frac{1}{12}$,$M_1 = 44$, $M_2 = 60$, (dimensions of Q_is),$N = 36$,$R = 17$, $L = 26$, (dimensions of the matrix D),$K = 36$ (the number of fingerprint bits per time interval). In a second step, the content was re-compressed, spatially scaled and rotated. Apart from these modifications, we also changed the original frame rate to 24fps. Then, we derived for the fingerprints for the modified video files. A sequence of 144 fingerprints corresponding to 12s of query video clip was compared with a corresponding 12s in the original video clip using a hamming distance measure. We record the percentage of fingerprint bits that flip (or the Bit Error Rate(BER)) for each comparison.

Attack type	BER
Rotate by 2^o & R	0.029854
Rotate by 3^o & R	0.0307
Rotate by 10^o & R	0.028626
Rotate by 45^o & R	0.029325
Rotate by 45^o, R & 30fps to 24fps	0.036313
SS & R	0.020244
SS, R & AS	0.042938

Table 1. Robustness of Proposed Video Fingerprint for various Signal Processing Operations; SS: Spatial scaling, R: Recompressed at 1Mbps, AS: Aspect ratio change from 4:3 to 16:9

Table 1 presents the BER results for several modifications. Note that the BER is not a function of the amount of rotation. Only 3% of bits flip on average for rotation attacks from 2 degrees to 45 degrees. Spatial scaling combined with aspect ratio change and recompression causes the BER to go upto 4.29%.

Figure 3 illustrates the sensitivity of the proposed video fingerprints. We compare the CDFs of BERs for two cases in this figure. Case 1 pertains to the scenario when we compare fingerprints of the original video and against the fingerprints of the modified video. The modifications include compression, spatial scaling, rotation and frame-rate conversion (denoted as comparison between video A and A' in the figure). Case 2 pertains to the scenario when we compare fingerprints of two different video files (denoted as comparison between video A and B in the figure). For case 1, the probability that the BER between fingerprints of video A and A' ≤ 0.05 is about 0.96. For case 2, the probability that the BER between fingerprints of two different videos A and B ≤ 0.05 is about 0.05. This shows that for a chosen BER threshold of 0.05 we will be able to correctly identify modified video with probability 0.96 and the probability of declaring a different video as original (false alarm) is about 0.05.

4. CONCLUSION

We proposed a novel video fingerprint extraction method based on subspace embedding. A fingerprint is extracted for each time interval in the video from a group of frames around that time interval. we first calculate the basis vectors of a coarse representation of this group of frames using a Singular Value Decomposition (SVD). Then, we project the coarse representation of the video frames onto a subset of the basis vectors. Thus, we obtain a subspace representation of the input video frames. Finally, we extract fingerprint bits by projecting a temporal average of these representations onto pseudo-random basis vectors. We use an incremental SVD update procedure to estimate the SVD basis for each time interval in the video. The makes the fingerprint extraction method run in real-time. Since the video data is represented

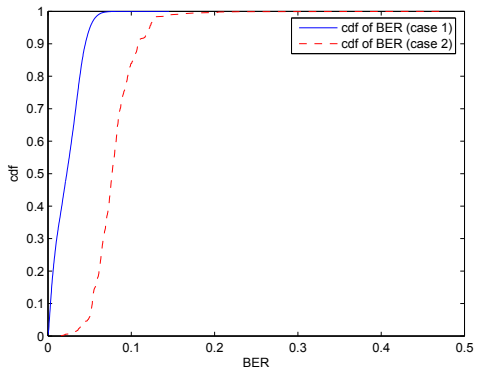

Fig. 3. Comparison of Cumulative Distribution Functions (CDFs) of BER for two cases; case 1: Comparing video A and A'; case 2: Comparing video A and B

using a set of basis vectors estimated from the input video data itself, the extracted features are particularly resilient to geometric attacks. The proposed fingerprint have been shown to be robust against other attacks as well including compression, frame-rate conversion, aspect ratio change, spatial scaling. We have shown the robustness and sensitivity of proposed fingerprints based on experiments on a 250hr database.

5. REFERENCES

[1] A.Swaminathan, Y. Mao and Min Wu, "Image hashing resilient to geometric and filtering operations," *Proc. of MMSP*, 2004.

[2] F. Lefbvre, B.Macq and J.D. Legat, "Rash: Radon soft hash algorithm," *Proc of EUSIPCO*, 2002.

[3] J.S.Seo, Jaap Haitsma, T.Kalker and C.D.Yoo, "A robust image fingerprinting system using the radon transform," *IEEE Signal Processing Journal: Image Communication*, vol. 19, pp. 325–339, 2004.

[4] S.S.Kozat, R. Venkatesan and M.K.Mihcak, "Robust perceptual image hashing via matrix invariants," *Proc. of ICIP*, 2004.

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

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

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

[8] J.Fridrich and M.Goljan, "Robust hash functions for digital watermarking," *ITCC*, 2000.