VIDEO FINGERPRINTING BASED ON MOMENT INVARIANTS CAPTURING APPEARANCE AND MOTION

Regunathan Radhakrishnan and Claus Bauer

Dolby Laboratories Inc, 100 Potrero Ave San Francisco CA 94103

ABSTRACT

In this paper, we propose two video fingerprinting methods that are robust to both geometric and non-geometric modifications on content. Both of the proposed methods are based on computation of moment invariants as features from concentric circular regions. The two methods differ in the way they capture appearance and motion information from video. In one method, we capture motion information by computing a difference image between the current video frame and a temporal average video frame computed from a past window of video frames. This method captures appearance by computing moment invariants from concentric circular regions of a video frame. In the second method, we capture appearance and motion by projecting features onto two sets of basis functions and explicitly capture how the moment invariants change over the regions and over time. We present experimental results on both of these video fingerprinting comparing their performance in terms of robustness against attacks and sensitivity to content.

Index Terms— Moment Invariants, Video fingerprinting, SVD

1. INTRODUCTION

Video fingerprinting methods that are robust against both non-geometric and geometric attacks has been an active area of research in the recent years. Past work in this area can be broadly classified into two approaches. The first class of approaches transform the input video data into a transform domain (e.g Radon transform, Fourier Mellin Transform) that is invariant under geometric operations before extracting robust features from this domain[1],[2],[3]. The second class of approaches represent the video frame as a collection of local features such as SIFT [4]. The robustness of this class of methods is due to the redundancy introduced by the multiple fingerprint codewords derived per video frame from corresponding local features. For instance, cropping attack may get rid of some of the local features but there is sufficient information in the remaining local features for successful identification.

In this paper, we propose two video fingerprinting methods that are based on moment invariants. The two methods differ from one another in the way they capture appearance and motion information. Moment invariants are global measures of the image surface that are robust against translation, rotation and scale change attacks and were originally proposed by Hu in 1962 for recognition of characters [5].The proposed methods differ from [1],[2],[3] in the fact that they do not perform any specific transform on input video. Unlike the local features based method proposed in [4], the proposed methods derive only one fingerprint codeword per video frame. We present experimental results on a 150hr database and compare their performance in terms of robustness and sensitivity.

2. PROPOSED METHODS BASED ON MOMENT INVARIANTS

In this section we describe the proposed two video fingerprinting methods that capture appearance and motion in different ways. Both of the methods are based on moment invariants as features extracted from concentric circular regions. We will first describe the parts of the processing blocks that are common to both of these methods. Figure 1 illustrates the processing blocks in the proposed methods.

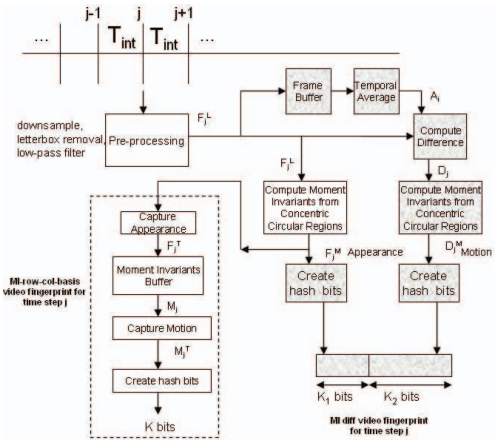


Fig. 1. Proposed Video Fingerprinting Methods.

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'. The signature for time step 'j' is then extracted based on features from F_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.

Step 3: A sub-image is cropped out 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.

These are the preprocessing steps common to both of the proposed methods. In the following subsection, we describe the remaining steps (**5a-8a**) in the first proposed method "MI diff". This method captures captures appearance by computing moment invariants as features from concentric circular regions. It captures motion by computing a difference image between the current frame and a temporal average frame computed from a past window of frames.

2.1. Method 1:MI diff

Step 5a: The low-pass filtered image F_j^L captures the appearance of the current frame. In order to capture information about motion of objects in the video, we create a difference image D_j. The difference image is derived according to: $D_j = A_j(k,l) - F_j^L(k,l); k = 1,2...H; l = 1,2...W$;Here W is the width and H is the height of F_j^L. In our implementation, $W = 160$ and $H = 120$. And A_j is the temporal average image obtained by averaging pixels from a window of past 'T' decoded frames. A_j is computed according to $A_j(k,l) = \frac{1}{T}\sum_{i=j-T}^{j} F_i^L(k,l)$. The motivation for computing D_j as a difference between the current frame (F_j^L) and a temporally averaged image (A_j) is the following. If the difference is computed from just one previous decoded frame then the values of A_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 6a: In this step, we create regions in the two matrices (F_j^L and D_j) before extracting features from each of the regions. Figure 2(a) 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 7a: 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 that are robust against translation, rotation and scale change attacks. Please see [5] for details on moment invariants.

Step 8a: 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 R_1 to the outer region R_N. 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 K_1 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:

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

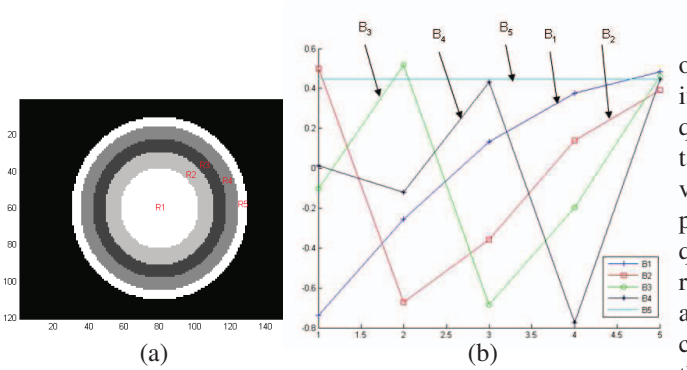
Here, the matrix Q is either F_j^M or D_j^M and M is the number of features per region and N is the number of regions. The signature bits are then derived by thresholding the K_1 projections [6].

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.

2.2. Method 2:MI-row-col-basis

In the previous subsection, we outlined the processing steps that capture appearance and motion for "MI diff" method. Here we describe the remaining processing steps (**5b-7b**) that capture appearance and motion for "MI-row-col-basis" method.

Step 5b: In order to capture appearance explicitly from F_j^L, we first compute M moment invariants from N concentric circular regions to obtain F_j^M. F_j^M is a $N \times M$ matrix with



(a) (b)

Fig. 2. a: Concentric regions from a frame b: SVD basis patterns for capturing appearance

N rows representing the N regions and M columns representing the M moment invariants. In our implementation $N = 5$ and $M = 7$. In a second step, we transform the columns of this matrix using a set of Basis vectors $(B_1, B_2, ...B_N)$ to explicitly capture how the moment invariants change as we proceed from the inner region (R_1) to the outer region (R_N). The Basis vectors $(B_1, B_2, ...B_N)$ are obtained using SVD (Singular Value Decomposition) of a training dataset offline. Notice from Figure 2(b), that the basis vectors capture different patterns. For example B_1 captures a monotonically increasing pattern of the feature. Therefore, if a feature in F_j^M matrix has this monotonically increasing pattern then its projection onto this basis B_1 would be the highest and all other projections would be small. Thus by transforming the columns of the matrices F_j^M using such a basis, we capture the patterns of feature variation from region R_1 through R_N directly. Let us represent the transformed Matrix by F_j^T.

Step 6b: In order to capture motion information explicitly, we first vectorize F_j^T and fill a buffer M_j containing vectorized F_j^T from a window of 'G' past video frames. M_j is $G \times 35$ where the $35(5 \times 7)$ elements are from vectorized F_j^T and there are G of them from corresponding G frames. In a second step, we transform the rows of this matrix using a set of Basis vectors $(V_1, V_2, ...V_G)$ to explicitly capture how these 35 features vary over frames 1 through G. Similar to the vectors $(B_1, B_2, ...B_N)$, the Basis vectors $(V_1, V_2, ...V_G)$ are obtained using SVD of a training dataset offline.Let us represent the transformed matrix by M_j^T.

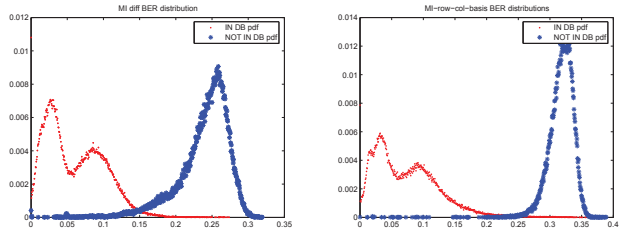
Step 7b: Finally, we create hash bits from M_j^T by projecting it onto $(K = 36)$ pseudo-random matrices $(P_1, P_2, ...P_K)$ and thresholding the projections. This step is similar to the last step in "MI-diff" method[6].

3. EXPERIMENTAL RESULTS

In this section, we present experimental results comparing the performance of the proposed two approaches ("MI diff" and "MI-row-col-basis") for capturing appearance and motion using moment invariants as features.

We created a 150hr database of fingerprints from a dataset of reference videos using both methods. 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 content that was not part of the reference database to illustrate the sensitivity of the proposed methods to content.

First, we present the results on the sensitivity 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. 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".



(a) (b)

Fig. 3. Comparison of BER distributions for MI diff (a) and MI-row-col-basis(b).x-axis:BER y-axis:density

Figure 3 compares the "IN DB pdf" and "NOT IN DB pdf" for both "MI diff" and "MI-row-col-basis". 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 by selecting a BER threshold one can compute the probability of miss (P_M) and the probability of false alarm (P_F) for these methods. In case of "MI-row-col-basis",for a BER threshold of 0.2, $P_M = 1.98\%$ and $P_F = 0.06\%$. In case of "MI-diff", for a BER threshold of 0.15,$P_M = 1.49\%$ and $P_F = 2.47\%$

Based on these measures, we can see that "MI-row-col-basis" outperforms "MI diff" in terms of sensitivity. This may be attributed to the fact that "MI-row-col-basis" uses $3s$ worth

of video to capture motion information whereas "MI diff" uses only $1s$ worth of video for the same. Both approaches perform in a similar fashion in terms of robustness as their probability of miss is comparable. "MI-row-col-basis" requires a total of $7.5s$ of query video for each query to perform the matching (54 codewords of fingerprint at 12fps and 3s of video for motion information). On the other-hand, "MI diff" requires only 5.5s of query video to perform the matching (54 codewords of fingerprint at 12fps and 1s of video for motion information).

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; A:Non-Geometric Attacks; B:Aspect ratio C:Rotation

	A	B	C
MI-row-col-basis	0.0728	0.1826	0.1323
MI diff	0.0603	0.1150	0.0832

Note that in terms of robustness against Non-geometric attacks as well as rotation and aspect ratio conversion, "MI-row-col-basis" and "MI diff" perform in a similar fashion. Although, "MI diff" has lower average BER in all cases, the BER for "MI-row-col-basis" in all these attacks is smaller than the chosen threshold of 0.2. 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 both approaches does not degrade gracefully. This is as expected because both approaches attempt 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.

Unlike these two methods, if the fingerprint were derived from a list local features identified from a decoded video frame as in [4], then there would be many redundant entries in the database for a particular video frame. This would translate into more robustness against cropping as some of the local features are likely to survive the cropping operation.

Table 2. Degradation of average BER for cropping attacks (C: cropping ratio; M1: MI-row-col-basis; M2: MI diff)

C	0	0.1444	0.3055	0.4375	0.5555
M1	0.0799	0.2199	0.2805	0.2905	0.2736
M2	0.0463	0.1238	0.1807	0.2201	0.2487

4. CONCLUSION

We proposed two video fingerprinting methods that are based on moment invariants computed from concentric circular regions. The methods differ in the way they captured motion and appearance information.In one method, we capture motion information by computing a difference image between the current video frame and a temporal average video frame computed from a past window of video frames. This method captures appearance by computing moment invariants from concentric circular regions of a video frame. In the second method, we capture appearance and motion by projecting features onto two sets of basis functions and explicitly capture how the moment invariants change over the regions and over time. Based on our experimental results, we conclude that both the approaches are robust against geometric and non-geometric modifications on content. Their performance against cropping attack is not as robust as the performance of a local features based method.

5. REFERENCES

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

[2] A.Swaminathan, Y.Mao, and M.Wu, "Robust and secure hashing for images," *IEEE Trans. on Info. Forensics and Security*, vol. 1, pp. 215–230, 2006.

[3] R.Radhakrishnan and C.Bauer, "Robust video fingerprinting based on subspace embedding," *Proc. of ICASSP*, 2008.

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

[5] Hu,M.K, "Visual pattern recognition by moment invariants," *IRE Trans. Info. Theory*, vol. IT-8, pp. 179–187, 1962.

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