ON IMPROVING THE COLLISION PROPERTY OF ROBUST HASHING BASED ON PROJECTIONS

Regunathan Radhakrishnan, Wenyu Jiang and Claus Bauer

Dolby Laboratories Inc, 100 Potrero Ave San Francisco CA 94103

ABSTRACT

In this paper, we study the collision property of one of the robust hash functions proposed in [1]. This method was originally proposed for robust hash generation from blocks of image data and is based on projection of image block data on pseudo-random matrices. We show that collision performance of this robust hash function is not optimal when used to extract hash bits from a moment invariants feature matrix for video fingerprinting. We identify that the collision performance of this hash extraction method could be improved if the pseudo-random matrices are selected carefully. We propose two methods that use an offline training set to improve the collision property. Both of the methods attempt to select the matrices that minimize cross-correlation among the projected features. The first method uses an iterative procedure to select the matrices that satisfy a cross-correlation threshold. The second method used Singular Value Decomposition (SVD) of the feature covariance matrix and hence the cross-correlation of the projected values is zero. We show the improved collision performance of both these methods on the same dataset. Also, we interpret the projection matrices obtained through the SVD procedure and show that they capture appearance and motion information from the moment invariants feature matrix.

Index Terms— Robust Visual Hash, Scalable Fingerprinting

1. INTRODUCTION

A Media fingerprint extraction method generally has the following two steps: (i)Robust Feature Extraction (ii)Robust Hash Extraction. The first step ensures that features that are representative of underlying perceptual content and also invariant under various processing operations, are extracted. The second step ensures that such features are converted into signature bits in a robust fashion i.e small changes in feature values do not result in drastic changes in extracted hash bits i.e for every $(x \sim y)$, $H(x) \sim H(y)$ with very high probability. (Here $\sim$ denotes similarity). This requirement disqualifies the use of normal cryptographic hash functions to convert the feature values into signature bits. The second step also serves to provide a compact representation of the features so that these signature bits can be stored and searched efficiently. Another property that is important for this robust hash extraction step is the collision property. A robust hash function is said to have good collision property if for every $(x \neq y)$, $H(x) \neq H(y)$ with very high probability.

Consider a content identification application with a large database of media fingerprints. Any media fingerprint that is extracted from query media is compared against this database of media fingerprints during identification. As the size of database in terms of number of hours of media increases, it is desirable that the uniqueness of fingerprint codewords is not reduced. The uniqueness property of the fingerprint codewords would make the fingerprint database scale to a larger number of hours. Instead if certain fingerprint codewords are more likely to occur than others, then as the database size grows the uniqueness reduces. This results in more computations to perform the matching. To see this, let us consider a hash-table based searching method for matching the query fingerprints against the fingerprints in the database. The database is indexed using the individual fingerprint codewords. Each fingerprint codeword in the hash table links to the location in a fingerprint file/media where that fingerprint codeword is present. The number of links per fingerprint index in the hash table will be referred to as number of collisions. If a fingerprint codeword is unique, one can quickly find its match in the database. As the uniqueness reduces, one has to perform more look-ups and pick the best match in terms of smallest distance from the query fingerprint. Thus, the fingerprints that have a small number of average collisions per fingerprint codeword will result in shorter search duration. Such fingerprints are scalable for searching through a larger database of fingerprints than fingerprints for which the average number of collisions is higher.

In this paper, we study the collision property of one of the robust hash functions proposed in [1]. This method was originally proposed for robust hash generation from blocks of image data and is based on projection of image data on pseudo-random matrices. We had used this hash bit extraction procedure for extracting hash bits from robust features for video fingerprinting in [2]. We first show that the collision performance is not optimal if the pseudo-random matrices are not selected carefully. We propose two methods that use an offline training set to improve the collision property of this robust hash extraction method. Both of the methods attempt to select the matrices that de-correlate the projected

features. The first method uses an iterative procedure to select the matrices and the second method used Singular Value Decomposition (SVD) of the feature covariance. We show the improved collision performance of both these methods on the same dataset. Also, we interpret the projection matrices obtained through the SVD procedure and show that they capture appearance and motion information from the moment invariants feature matrix.

2. PROPOSED METHOD

2.1. Motivation

The first step in media fingerprint extraction is to extract robust features from audio and video that are invariant to various processing operations on the content. Let us represent the extracted robust features by a matrix Q. For all the experiments in this paper, Q is a matrix with $G \times N$ elements that attempts to capture appearance and motion information from the input video. The input video is first temporally downsampled to 12 fps. Then, every frame is spatially downsampled to $120 * 160$ ($Height \times Width$) resolution after letterbox detection and removal. The N columns in the feature matrix Q correspond to N video frames in a buffer including the current frame. We use a buffer of 3s ($N = 12 \times 3$) to capture motion information. The G rows correspond to a set moment invariants extracted from each frame in the buffer. We extract 7 moment invariants from 5 concentric circular regions in each frame ($G = 5 \times 7$). Moment invariants are global measures of an image surface that are invariant to translation, rotation and scaling and were originally proposed for text pattern recognition in [3]. Now, each column of the matrix Q attempts to capture the appearance of the corresponding frame of video in the buffer by measuring how the 7 moment invariants change over the 5 regions. The second step is to extract hash bits from the feature matrix Q using a robust hash function.

In [1] proposed one such robust hash bit extraction procedure from blocks of image data. Given an image block B, the following steps are performed to extract robust hash bits. Using a secret key K the method generates N random matrices with entries uniformly distributed in the interval $[0, 1]$. Then, a low-pass filter is repeatedly applied to each random matrix to obtain L random smooth patterns. All patterns are then made DC-free by subtracting the mean from each pattern. Considering the block and the pattern as vectors, the image block B is projected on each pattern P_i , $i = 1, 2, ...L$ and the projected value is compared with a threshold Th to obtain L bits b_i. If $B.P_i < Th$, then $b_i = 0$ else $b_i = 1$.

We applied this method of robust hash bit extraction for the moment invariants based feature matrix Q described earlier and studied collision property. We extracted 36 bit fingerprint codewords from a 25 min video clip at 12 fps. Let us denote this collection of fingerprint codewords using S. We computed the average number of collisions per fingerprint codeword for this collection of codewords (S). The average number of collisions was found to be 3.59. Then, we added about 3 hrs of fingerprint codewords extracted from unique content that is not related to this 25 min clip. Again, we compute the average number of collisions per codeword in the set S. Now, the average number of collisions for this dataset increased to 7.37. Ideally, the average number of collisions should remain unchanged after the inclusion of fingerprint codewords from unrelated content. However, it did not remain the same. This means that the collision property (i.e. for every $(x \neq y)$, $H(x) \neq H(y)$ with very high probability) is not optimal for this hash function. In the following two subsections, we describe the two proposed methods to improves the collision property of robust hash in [1] by selecting the projection matrices P_i $(i = 1, 2...L)$ carefully.

2.2. Iterative Procedure for Selecting Random Matrices:*iter*

This procedure is motivated by the fact that the pseudo-random matrices $P_1, P_2, ..P_L$ should capture different aspects of the feature matrix Q through their corresponding projections $H_1, H_2, ...H_L$. Instead if the information they capture about the feature matrix Q is correlated, then certain hash bits would tend to vary together. This means that certain codewords are more likely than others. The following are the steps to choose the L pseudo-random matrices:

Step 1: Obtain a training dataset of media content and extract robust features $Q^1, Q^2, ...Q^M$. Here M is the number of training instances for selecting the pseudo-random matrices.

Step 2: Initialize iteration index $j = 1$ and a corresponding set of pseudo-random matrices for this iteration $P_{1,j}, P_{2,j}, ...P_{L,j}$

Step 3: Project each Q^i onto $P_{1,j}, P_{2,j}, ...P_{L,j}$ to obtain the projected values $H^i_{1,j}, H^i_{2,j}, ...H^i_{L,j}$, $i = 1, 2...M$. We would like each of pseudo-random matrices to capture different aspects of the feature matrix. Therefore, we should select a set of pseudo-random matrices whose projections $H^i_{1,j}, H^i_{2,j}, ...H^i_{L,j}$ have low correlation among themselves. In the following step, we compute the cross-correlation matrix C of the projected values using the training set containing M instances of the feature matrix Q.

Step 4: The entries of the cross-correlation matrix C are computed as given below

$$C^j(k,l) = \frac{\frac{1}{M}\sum_{i=1}^{M}(H^i_{k,j} - E(H_{k,j}))(H^i_{l,j} - E(H_{l,j}))}{\sqrt{\sigma_{k,j}}\sqrt{\sigma_{l,j}}}$$

$$E(H_{k,j}) = \frac{1}{M}\sum_{i=1}^{M} H^i_{k,j}$$

$$\sigma_{k,j} = \frac{1}{M}\sum_{i=1}^{M}(H^i_{k,j} - E(H_{k,j}))^2$$

Here $C^j(k,l)$ represents the cross correlation coefficient between the projected values $H_{k,j}$ and $H_{l,j}$ for j^{th} set of pseudo-random matrices and $k, l, n = 1, 2...L$. $E(H_{k,j}))$ and $\sigma_{k,j}$ represent the mean and variance of $H_{k,j}$ estimated

from this training set. Note that the cross-correlation matrix is a matrix of dimension $L \times L$ whose main diagonal elements are equal to the value 1.0. Once we have computed the cross-correlation matrix in this manner, we select pseudo-random matrices for which the cross-correlation of projections with other pseudo-random matrices is smaller than a chosen threshold.

Step 5: After we have selected random matrices that meet our criteria, we replace the matrices that did not meet our criteria with new ones and proceed with the next iteration.

Step 6: We continue the iterations until we find all L pseudo-random matrices that capture different aspects of the feature matrix. The cross-correlation matrix after the last iteration would have all terms less than the chosen threshold (say 0.2) except for the main diagonal elements.

2.3. SVD based Selection of Projection Matrices:*svd*

In the previous subsection, we described an iterative procedure to design the L pseudo-random matrices $P_1, P_2, ..P_L$. These pseudo-random matrices ensure that the cross-correlation coefficient between any two projected values ($C(k,l)$) is smaller than a chosen threshold (say 0.2). Here $C(k,l)$ represents the cross-correlation between H_k (projection of Q onto P_k) and H_l (projection of Q onto P_l). In this section, we describe a method that designs L matrices $\phi_1, \phi_2, ...\phi_L$ such that the cross-correlation between any two projected values ($C^{svd}(k,l)$) is equal to zero. Here $C^{svd}(k,l)$ represents the cross-correlation between H_k^{svd} (projection of Q onto ϕ_k) and H_l^{svd} (projection of Q onto ϕ_l). The matrices $\phi_1, \phi_2, ...\phi_L$ are the eigenvectors of the covariance matrix of the feature values from the training set.

Step 1: Obtain a training dataset of media content and extract robust features $Q^1, Q^2, ...Q^M$. Here M is the number of training instances. Let the dimension of each feature matrix Q^i be $R = G \times N$.

Step 2: Then, we compute the covariance matrix C^{feat} (dimension R*R) of the features from the training set $Q^1, Q^2, ...Q^M$ as given by $C^{feat}(k,l) = \frac{1}{M}\sum_{i=1}^{M}(Q_k^i - E(Q_k))(Q_l^i - E(Q_l))$. Here $E(Q_k) = \frac{1}{M}\sum_{i=1}^{M} Q_k^i$ and $k, l = 1, 2...R$.

Step 3: Once we compute the covariance matrix of the features from the training set, we compute the eigenvectors of the matrix C^{feat} that satisfy the relation, $V^{-1}C^{feat}V = D$, using PCA (Principal Components Analysis). Here the columns of V (dimension R*R) are the eigenvectors of the covariance matrix C^{feat} and are represented as $\phi_1, \phi_2, ...\phi_R$. D is a diagonal matrix with eigenvalues $(E_1, E_2...E_R)$ as its main diagonal elements and zero elsewhere.

Thus, we can now transform any input feature vector Q to a L dimensional space by projecting it onto the first L eigenvectors $\phi_1, \phi_2, ...\phi_L$ to obtain $H_1^{svd}, H_2^{svd}, H_L^{svd}$ respectively. Here H_k^{svd} is the projection of Q onto ϕ_k. Now the cross-correlation between any two projected values ($C^{svd}(k,l)$) is guaranteed to be equal to zero.

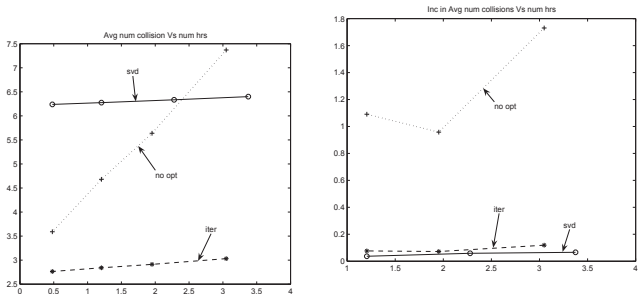
Unlike the iterative procedure in the previous section, we obtain the $\phi_1, \phi_2, ...\phi_L$ by directly computing the eigenvectors of the feature covariance matrix, C^{feat}. $\phi_1, \phi_2, ...\phi_L$ are pairwise orthogonal i.e ($< \phi_i.\phi_j >= 0\ i \neq j$) whereas $P_i.P_j$ need not to identically equal to 0 for $i \neq j$. Also, the elements of P_i are uniformly distributed between random numbers between -0.5 and 0.5 whereas the elements of ϕ_i are obtained in a data-driven fashion and are not drawn from a specific distribution. In the case of the projections obtained from projecting Q onto $\phi_1, \phi_2, ...\phi_L$ ($H_1^{svd}, H_2^{svd}, ...H_L^{svd}$), there is an inherent order of significant projections that corresponds to the significance of the basis functions (ϕ_1 more significant than ϕ_2 , ϕ_2 more significant than phi_3 ...so on). This means that one could derive varying number of fingerprint bits from each of the projections $H_1^{svd}, H_2^{svd}, ...H_L^{svd}$. Obviously, one would like to derive more number of bits from H_1^{svd} than from H_L^{svd}. In the case of projections obtained from projecting Q onto $P_1, P_2, ..P_L$, there is no specific order of significance and therefore we usually derive equal number of bits from each of the projected values $H_1, H_2, ...H_L$.

3. EXPERIMENTAL RESULTS

In this section, we present experimental results on collision performance of the iterative procedure as well as the SVD based procedure for selecting the projection matrices. The feature matrix Q for all experiments in this paper is a matrix of dimensions 35×36 ($G \times N$) where 36 corresponds to a 3s buffer of frames at 12fps and 35 corresponds to the 7 moment invariants computed from 5 concentric circular regions of each frame in the buffer. First, we extract 22 bit fingerprint codewords (L = 22) from a 25 min video clip using three sets of projection matrices: (i) pseudo-random matrices (*no-opt*) (ii) pseudo-random matrices selected using the iterative procedure (*iter*)(iii) projection matrices selected using the SVD procedure (*svd*). For selecting the projection matrices in case of (ii) and (iii), we used the same offline training set of 3 hrs of content. In the case of (i), the pseudo-random matrices are not selected carefully using either procedure proposed in this paper and this method acts as a baseline method. In the case of (ii), the pseudo-random matrices were selected ensuring that the cross-correlation between the projected values is always less than 0.2. Let us denote the collection of fingerprint codewords extracted from the 25 min clip using projection matrices from (i) as S^{no-opt}. And let those extracted using projection matrices from (ii) be denoted as S^{iter} and those extracted using projection matrices from (iii) be denoted as S^{svd}. Then, we compute the average number of collisions for the fingerprint codewords in each of the three sets namely S^{no-opt},S^{iter} and S^{svd}.

In order to study the collision performance of these three sets, first, we add fingerprint codewords extracted from unique content (content that is unrelated to the 25 min clip) to the database. Then, we compute the average number of collisions as we increase the number of fingerprint codewords of

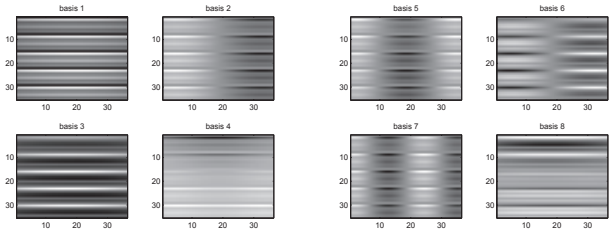
unrelated content from 1 hr to upto 3hrs. Ideally, there should be no change in the average of number of collisions as we add unrelated content to the database. Figure 1(a) illustrates the collision performance for the three sets of projection matrices. The average of number of collisions for S^{no-opt} increases from 3.59 to 7.37 as we add fingerprint codewords from 3hrs of unrelated content. On the otherhand, the average number of collisions for S^{iter} increases from 2.76 to 3.03 for the same increase in number of fingerprint codewords from unrelated content. The average number of collisions for S^{svd} increases from 6.23 to 6.40. The projection matrices obtained through the SVD procedure have the smallest slope (smallest increase in average number of collisions for inclusion of an hour of unrelated content) among the three sets of projection matrices. This is as expected because the SVD procedure ensures that the cross-correlation between the projected values is zero. The iterative procedure only ensures that the cross-correlation is always less than 0.2. Figure 1(b) shows the increase in the average number of collisions for every additional hour of unrelated content included. The slopes for S^{no-opt},S^{iter} and S^{svd} are 1.5,0.1 and 0.05 respectively.



(a) (b)

Fig. 1. (a) Avg. num of collisions vs num hrs of unrelated content (b)increase in Avg. num of collisions vs num hrs of unrelated content;

Figure 2 shows the first 8 projection matrices obtained through the SVD procedure. Each of the projection matrices are of the same dimension as the feature matrix Q (35×36). Recall that the number of rows correspond to 5×7 moment invariants extracted from each frame in the buffer and the number of columns correspond to the number of frames in the 3s buffer (3×12). Projection matrices 1 & 3 (on the top & bottom left of Figure 2(a)) can be interpreted as those capturing appearance information alone from the feature matrix Q. This is so because the values along the columns of the projection matrices 1 & 3 are similar for every row. Also, note that the patterns along rows are repeated 5 times. This is so because the 7 moment invariants computed from the 5 concentric circular regions are correlated and hence the projection matrix values are also correlated. The other projection matrices shown in the Figure 2 capture both appearance and motion information as can be seen from the changing patterns across the columns (time) for these projection matrices.



(a) (b)

Fig. 2. (a) Projection matrices 1-4 from SVD procedure (b) Projection matrices 5-8 from SVD procedure;

4. CONCLUSION

In this paper, we have shown that the collision performance of the robust hash function proposed in [1] is not optimal for extracting hash bits from moment invariants based video fingerprint extraction. The robust hash function proposed in [1] is based on projection of the features onto a set of pseudo-random matrices. We propose two methods to improve the collision performance of this robust hash function by carefully selecting a set of projection matrices that minimize the cross-correlation between projected values. Both of the proposed methods design the projection matrices using a training dataset offline. One method is based on a iterative procedure that selects pseudo-random matrices that satisfy a cross-correlation threshold. Another method is based on the SVD of the feature covariance matrix and hence the cross-correlation between the projected values is set to zero. We showed that both of the proposed methods improved the collision performance when compared to an approach that does not carefully select the projection matrices. We also interpret the projection matrices obtained through the SVD procedure and show that they capture appearance and motion information from the moment invariants feature matrix. Also, the SVD based procedure outperformed the iterative procedure both in terms of average number of collisions and training time to select the projection matrices.

5. REFERENCES

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

[2] R.Radhakrishnan, W.Jiang and C.Bauer, "A review of video fingerprints invariant to geometric attacks," *to appear in Proc. of SPIE*, 2009.

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