ROBUST PERCEPTUAL IMAGE HASHING VIA MATRIX INVARIANTS

Suleyman S. Kozat

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801 USA
Email:*kozat@ifp.uiuc.edu*

Ramarathnam Venkatesan, M. Kıvanç Mıhçak

Cryptography and Anti-Piracy Group,
Microsoft Research, One Microsoft Way,
Redmond, WA 98052-6399
Email: {*venkie,kivancm* }*@microsoft.com*

ABSTRACT

In this paper we suggest viewing images (as well as attacks on them) as a sequence of linear operators and propose novel hashing algorithms employing transforms that are based on matrix invariants. To derive this sequence, we simply cover a two dimensional representation of an image by a sequence of (possibly overlapping) rectangles R_i whose sizes and locations are chosen randomly[1] from a suitable distribution. The restriction of the image (representation) to each R_i gives rise to a matrix A_i. The fact that A_i's will overlap and are random, makes the *sequence* (respectively) a redundant and non-standard representation of images, but is crucial for our purposes.

Our algorithms first construct a *secondary image*, derived from input image by pseudo-randomly extracting features that approximately capture semi-global geometric characteristics. From the secondary image (which does not perceptually resemble the input), we further extract the final features which can be used as a hash value (and can be further suitably quantized). In this paper, we use spectral matrix invariants as embodied by Singular Value Decomposition. Surprisingly, formation of the secondary image turns out be quite important since it not only introduces further robustness (i.e, resistance against standard signal processing transformations), but also enhances the security properties (i.e. resistance against intentional attacks). Indeed, our experiments reveal that our hashing algorithms extract most of the geometric information from the images and hence are robust to severe perturbations (e.g. up to %50 cropping by area with 20 degree rotations) on images while avoiding misclassification. Our methods are general enough to yield a watermark embedding scheme, which will be studied in another paper.

1. INTRODUCTION

We propose a new pseudo-random (PR) signal representation scheme for images, where we view images consisting of a series of linear operator representatives (i.e., matrices). In this case, an image I is represented by $[A_1, \ldots, A_p]$, where each A_i is a matrix that corresponds to a PR location chosen from I.

We use matrix decompositions on $\{A_i\}$, whose outputs are approximately invariant under reasonable perturbations (i.e., "matrix invariants"). We experimentally show that the resulting signal representation domain has favorable robustness properties for various multimedia anti-piracy applications, such as hashing and mark embedding. Furthermore, we conjecture that it provides desired security properties for such applications, i.e., it is hard for the attacker to obtain information about the secret key (which is used as the seed of a secure PR number generator in the algorithms) due to the randomness introduced in the signal representation. In this paper, we apply our approach to the problem of image hashing and present several promising results. Moreover, we believe that our approach (or possibly a variant of it) can be extended to address hashing and mark embedding problems for video and audio as well, with little or no straightforward modifications.

In terms of matrix invariants, we particularly focus on *Singular Value Decomposition* (SVD). We recollect that SVD has some provable optimality properties: "Best" lower-dimensional (say K-dimensional) approximation to a matrix (say rank N, $N \geq K$) in the sense of Frobenius norm is produced by the first K singular vectors and the corresponding singular values. Accordingly, we observe that the essence of the semi-global features and the geometric information of images are compactly captured by the significant components of the SVD of these images. Our experiments reveal that such components are approximately invariant under intentional or unintentional disturbances as long as the image of interest is not altered too severely. Although SVD and similar type of transformations are previously used to generate hashing and mark embedding algorithms [1], here we differ in that we apply SVD to pseudo-randomly-chosen semi-global regions of images and employ the singular vectors to extract robust features in crucial steps of the hashing algorithms (instead of the usage of singular values of the whole image in [1]). Hence, the structure of the proposed transformation is PR (i.e., the choice of the semi-global regions dependent on a secret key, which may be used as the seed of a secure PR number generator). This increases the security of the overall system. In our experiments, we observed that SVD components of PR regions meet the desired goals, provided that there are sufficiently many of these regions and they are large enough.

In our algorithms, we also use DCT (discrete cosine transform) and DWT (discrete wavelet transform) jointly with SVD. We observe that such a joint usage enhances the performances of our schemes. In all the randomized steps of our algorithms, a secret key κ is used as the seed of a secure PR number generator (e.g., RC4).

Our hashing algorithms consist of two major stages. In the first stage we derive robust feature vectors from PR semi-global regions via matrix invariants; these features are termed "intermediate features". In the second stage, the intermediate features are used to construct a PR secondary image, which is then used to extract the final feature vectors, which constitute the hash value of the image. We observe that the second stage increases robustness against attacks (e.g., compression, rotation, cropping, etc.) and conjecture that it limits information leakage about κ to the adversary. By extracting the hash value from the secondary image, we effectively reduce the effects of the geometric attacks, which are well-pronounced in the intermediate features [2]. Recall that, the

[1]Our algorithm will use a cryptographically secure random generator for the random choices it makes, so all our random choices are in fact pseudo-random.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

idea of extracting features from pseudo-randomly selected regions for hashing has been considered before [3, 4, 5, 6]. A similar approach was followed in [7] to design an image watermarking algorithm. In contrast with the prior robust image hashing work, in this paper we employ SVD components to capture essential characteristics of images rather than PR linear statistics. In summary, the main contribution of this work is to view PR image portions as linear operator representatives (i.e., matrices) and to use matrix invariants on them, in particular SVD, to extract robust PR features and to employ them to derive robust hash values. Thus, we effectively propose a new PR signal representation and illustrate its usage within the context of the image hashing problem.

The organization of the paper is as follows. In Section 2, we give a general description of the image hashing problem and then give an algorithmic description of our hashing approach. In Section 3, we continue with our candidate hashing algorithms, where we give a brief description of each one. In Section 4 we test our algorithms under various attacks and report the performance results.

2. HASHING

A robust image hash function for security purposes has two inputs, an image I and a secret key κ and produces a short binary vector $\vec{h} = H_\kappa(I)$ from a set $\{0,1\}^h$ (i.e., h bits long). The hash function should possess perceptual properties: Hash values for all perceptually "approximately-the-same" images are desired to be equal with high probability; in contrast, perceptually different images should produce independent hash values with high probability. Obviously, such a hash function is a many-to-one mapping. We formulate these requirements as follows:[2]

1) Randomization : For any given input I, its hash value should be approximately uniformly distributed among all possible 2^h outputs:

$$\forall h \in \{0,1\}^h, \quad \Pr\{H_\kappa(I) = \vec{h}\} \approx 2^{-h}.$$

2) Pairwise Independence : The hash outputs for two perceptually different images (say I_1 and I_2) should be approximately independent:

$$\forall h_1, h_2 \in \{0,1\}^h, \quad \Pr\{H_\kappa(I_1) = \vec{h}_1 | H_\kappa(I_2) = \vec{h}_2\}$$

$$\approx \Pr\{H_\kappa(I_1) = \vec{h}_1\}.$$

3) Invariance : For all possible acceptable disturbances, the output of the hash function should remain approximately invariant. Let I and $\hat{I}$ be perceptually similar images. Then,

$$\forall h \in \{0,1\}^h, \quad \Pr\{H_\kappa(I) = \vec{h}\} \approx Pr\{H_\kappa(\hat{I}) = \vec{h}\}.$$

The notion of acceptable disturbances is not precise, and in this version it can be taken to mean that we consider two images to be similar when there are unnoticeable visual distortions between them in terms of human perception. We will address this issue in our future work. Next, we present the algorithmic description of our generic hashing scheme:

Step 1: Let the $n \times n$ input image be $I \in \mathbb{R}^{n \times n}$.
Step 2: From I, pseudo-randomly form p possibly overlapping rectangles (each of them of size $m \times m$): $A_i \in \mathbb{R}^{m \times m}$, $1 \le i \le p$.
Step 3: Generate a feature vector $\vec{g}_i$ from each rectangle A_i via the transformation $\vec{g}_i = T_1(A_i)$.
Step 4: Construct a secondary image J by using a PR combination of intermediate feature vectors $\{\vec{g}_1, \ldots, \vec{g}_p\}$.
Step 5: From J, pseudo-randomly form r possibly overlapping rectangles (each of them of size $d \times d$): $B_i \in \mathbb{R}^{d \times d}$, $1 \le i \le r$.
Step 6: Generate a final feature vector $\vec{f}_i$ from each rectangle B_i via the transformation $\vec{f}_i = T_2(B_i)$.
Step 7: Combine $\{\vec{f}_1, \ldots, \vec{f}_r\}$ to form the final hash vector.

The choice of the transformations T_1 and T_2 is crucial for the performance of the scheme; we propose to use SVD for T_1, or T_2, or both. In principle, other approximately-invariant matrix decompositions may also be employed. Furthermore, the formation of the secondary image J improves the results considerably. In each step of the algorithm, pseudo-randomization is achieved via a secure PR number generator by using the same key, or part of a common key as the seed; this key is unknown to the attacker. Note that, in this paper, we particularly focus on robust PR feature generation from images; in the formation of the final hash values, further dimensionality reduction can, in principle, be achieved via quantization with PR lattices or PR projection to lower-dimensional appropriately-chosen subspaces.

3. HASHING ALGORITHMS

In general, the choice of transformations is a significant and nontrivial task: We would like to capture the essence of the geometric information while having dimensionality reduction and introducing enough randomness. In case of conventional transforms (such as DCT or DWT, which have proven to be effective for traditional applications), the image is projected onto a fixed set of basis vectors. It is still an open question, however, which mappings (if any) from DCT/DWT coefficients preserve the essential information about an image for hashing and/or mark embedding applications. In addition, from a security point of view, it may be beneficial if the basis vectors of the transform of interest are pseudo-randomly adapted to the image to minimize information leakage to an adversary.

Unlike DCT/DWT-type fixed basis transforms, SVD selects the optimal basis vectors in L_2 norm sense such that, for any $m \times m$ real matrix A, $\forall k,\ 1 < k \le m$, we have

$$(\sigma_k, \vec{u}_k, \vec{v}_k) = \arg \min_{a, \vec{x}, \vec{y}} \left| A - \sum_{l=1}^{k-1} \sigma_l \vec{u}_l \, \vec{v}_l^T - a\vec{x} \, \vec{y}^T \right|_F^2, \quad (1)$$

where $a \in \mathbb{R}$, $\vec{x}, \vec{y} \in \mathbb{R}^m$, $\sigma_1 \ge \sigma_2 \ldots \ge \sigma_m$ are the singular values, $\{\vec{u}_i\}$ and $\{\vec{v}_i\}$ are the corresponding singular vectors and $(\cdot)^T$ is the transpose operator. As an analogy, if an image is represented as a vector in some high-dimensional vector space, then the singular vectors give the optimal "directional information" about the image in the sense of (1), while the singular values give the distance information along this "direction". Consequently, the singular vectors that correspond to large singular vectors are naturally prone to any scaling attack and other small conventional signal-processing modifications.

By using SVD, we view an image as a two dimensional surface in a three dimensional space. When DCT-like transformations are applied to an image (or surface), the information about any particularly distinctive (hence important) geometric feature of the image is dispersed to all coefficients. As an example, a surface with strong peaks (e.g., very bright patches in a dark background) will be dispersed to all transform in case of DCT. By using SVD, we preserve both the magnitude of these important features (in singular values) and also their location and geometry in the singular vectors. Hence, the combination of the "top" left and right singular vectors (i.e., the ones that correspond to the largest singular values) capture the important geometric features in an image in the L_2 norm sense. We utilize this observation in our algorithms.

Here, we propose several hashing algorithms by using different combinations of transforms in Steps 1 and 2 (i.e., different

[2]Here, the probability measure is defined by the secret key

choices for T_1 and T_2) of the generic hashing algorithm given in Sec. 2. We name each method according to the order of the employed transformations.

3.1. SVD-SVD Hashing Algorithm

In the SVD-SVD method, we first find the SVD of each subimage (i.e., rectangle) in step 3 of the generic algorithm:

$$A_i = U_i S_i V_i^T, \quad 1 \leq i \leq p,$$

where U_i, V_i are the $m \times m$ real left and right singular vector matrices and S_i is the real $m \times m$ diagonal matrix consisting of the singular values along the diagonal. Next, we collect the "first" left and right singular vectors (i.e., the singular vectors that correspond to the largest singular value) of each subimage. Let $\Gamma = \{\vec{u}_1, \ldots, \vec{u}_p, \vec{v}_1, \ldots, \vec{v}_p\}$, where $\vec{u}_i$ (resp. $\vec{v}_i$) is the first left (resp. right) singular vector of the i-th subimage. Then, we form a PR smooth image J (i.e., secondary image), such that the elements of Γ form the columns of J in an appropriately-designed PR order: We initially pseudo-randomly select an element of Γ as the first column of J. Then, J is formed in an iterative fashion; at step i, the i-th column of J is selected from Γ such that it is closest to the $(i-1)$-th column of J in L_2 norm sense, under the constraint that it has not been chosen before in any of the previous $i-1$ steps. Hence, after $2p$ steps all the elements of Γ are pseudo-randomly re-ordered to form J (size $m \times 2p$). Note that, the L_2 metric can be replaced by any other suitable metric in the formation of J so that continuity and smoothness are achieved. The smooth nature of J is important to gain robustness against geometric attacks. Also note that, instead of this simple PR re-ordering of vectors, it is possible to apply other (possibly more complex) operations to generate J to introduce more randomness. Having formed J, we proceed with re-applying SVD to each subimage B_i in step 6 of our generic algorithm. We keep the first left and right singular vectors from each B_i, $1 \leq i \leq r$; combining these r vectors, we obtain the image hash.

3.2. DCT(DWT)-SVD Hashing Algorithm

As a variant of our previous approach, we first use 2D-DCT as the initial transform (i.e., T_1) in step 3 of the generic hashing algorithm; which yields the DCT-SVD method[3]. Similar to DCT-SVD, a DWT-SVD method can be derived using the DWT transformation instead of the DCT in step 3 of the generic algorithm. Due to space limitations, here we only focus on DCT-SVD method and report that DWT-SVD approach yields similar results experimentally. Let D_i be the 2D-DCT of each sub-image A_i, $1 \leq i \leq p$. After computing $\{D_i\}_{i=1}^p$, we keep the coefficients that correspond to low-to-mid band frequencies from each D_i; see Fig. 1. Note that, the selection of f_{min} and f_{max}, that determines the selected frequency band, is crucial. Although we believe that there is no particular frequency band of DCT coefficients that can be clearly considered as more important than the others, we observed that the coefficients of low-to-mid band frequencies carry more descriptive and distinctive information about images. By selecting $f_{min} > 0$, we avoid near DC frequencies which are more sensitive to simple scaling or DC level changes. By selecting a relatively small value of f_{max}, we avoid using coefficients of higher frequencies, which can be altered significantly by noise addition, smoothing, compression, etc. In general, depending on the problem specifications, suitable values of f_{min} and f_{max} can be chosen.

[3]It is shown in [2], that by using DCT instead of SVD in step 3 of the generic algorithm, an optimization problem can be formulated to derive robust image mark embedding algorithms.

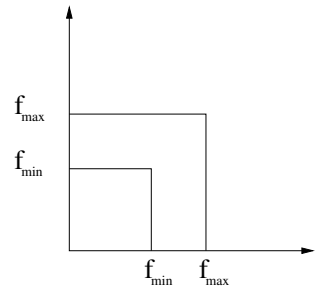

Fig. 1. Selection of low-to-mid band frequncies in the 2D-DCT domain for the construction of the secondary image in the DCT-SVD method.

The coefficients in this frequency band are then stored as a vector $\vec{d}_i \in R^{f_{max}*f_{max}-f_{min}*f_{min}}$ for each rectangle A_i. The ordering of the elements of $\{\vec{d}_i\}$ is user dependent and can possibly be used to introduce extra randomness. Then, we proceed with a similar method to the one given in Sec. 3.1: First we form the set $\Gamma = \{\vec{d}_1, \ldots, \vec{d}_p\}$ and then form the PR smooth secondary image J; the method of formation of J given Γ is the same as the one explained in Sec. 3.1. Next, we find the SVD of the secondary image J: $J = USV^T$ and store the first left and right singular vectors $\vec{u}_1$ and $\vec{v}_1$ as the hash value of the image, i.e., $h = \{\vec{u}_1, \vec{v}_1\}$.

4. EXPERIMENTAL RESULTS

To asses the performance of each method, we apply them both to standard test images and an image database of about 5000 images. Here, we present some preliminary results under compression, rotation and cropping attacks. We note that the practical choice of algorithmic parameters can further be optimized in order to improve the results. In experiments involving standard test images, we consider the 512×512 grayscale Lena image and compare its hash value with those of the attacked Lena and Goldhill images. In all simulations, $\vec{h}_1$, $\vec{h}_2$, and $\vec{h}_3$ denote the hash values of the original Lena, attacked Lena and Goldhill respectively. Although we experimented with a wide range of attacks, including benchmark attacks, here we report results for only a few classes of attacks for the purposes of illustration.

We start our experiments with the SVD-SVD method introduced in Sec. 3.1. The algorithmic parameters are chosen as, $p = 200$, $m = 256$, $r = 200$, $d = 150$; the secondary image J is of size 256×400. As an attack, we crop 50 percent of the image by area, rotate it 20 degrees and JPEG compress it with quality factor (QF) 5. In Figs. 2(a) and 2(b), we show the original and attacked Lena respectively. In Fig. 2(c) (resp. Fig. 2(d)), we plot the difference between the left (respectively the right) singular vectors corresponding to the same subimages. We observed similar results with all of the other standard test images; the SVD-SVD method produces favorable results.

Next, we apply the SVD-SVD method to an image database of about 5000 images. We randomly select an image from the database, and compare its hash value with the hash values of both the attacked version of the initially-selected image and several other randomly-chosen perceptually-different images. As for the attacks, we use combinations of cropping (between 10 to 60 percent by area), rotation (1 to 20 degrees) and JPEG compression (QF between 5 and 60). In Fig. 3, each instance of the attack combination is represented by a diamond, where x-axis (resp. y-axis) is the total difference between left (resp. right) singular vectors from the hash

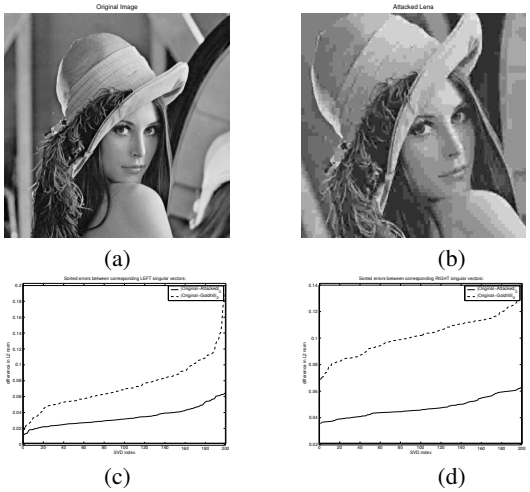

Fig. 2. SVD-SVD hashing algorithm. (a) The original Lena image, (b) Attacked Lena image: Crop = 50 percent, Rotation angle = 20 degrees, QF=5 (c) The difference between the first left singular vectors from the hash values of the original and attacked Lena (solid), the difference between the original Lena and Goldhill (dashed) (d) The difference between the first right singular vectors from the hash values of the original and attacked Lena (solid), the difference between original Lena and Goldhill (dashed)

values of the original and attacked image. In the same figure, each instance of the choice of a randomly-chosen perceptually-different image from the database is represented by a circle, where x-axis (resp. y-axis) is the total difference between the left (resp. right) singular vectors from the hash values of the original and other image. In Fig. 4, we show results for the DCT-SVD method under the same experimental setup. We observed that the performances of the SVD-SVD and DCT-SVD methods are comparable in general.

5. CONCLUSION

In this paper, we introduce a new pseudo-random signal representation (via viewing images as a sequence of linear operator representatives, i.e., matrices) and apply it to the robust image hashing problem. In particular, we heavily use SVD in our methods, to extract semi-global robust PR features of images. We observe that the results are very promising; our hash methods are robust under severe geometric disturbances. We believe that our methods capture the essence of the geometric structure of images. In our future research, we plan to explore other matrix decomposition methods, that satisfy approximate invariance properties.

6. REFERENCES

[1] R. Sun, H. Sun, and T. Yao,"An SVD and quantization based semi-fragile watermarking technique for image authentication,*2002 6th Annual Conference on Signal Processing*, vol. 2, pp 1592–1595, Aug. 2002.

[2] S. S. Kozat, M. K. Mıhçak and R. Venkatesan, "Robust Hashing and Watermarking via Matrix Invariances, *in preparation for submission to IEEE Trans. on Image Processing.*

[3] M. K. Mıhçak and R. Venkatesan, "New Iterative Geometric Methods for Robust Perceptual Image Hashing," *Proc. of ACM Workshop on Security and Privacy in Digital Rights Management* , Philadelphia, PA, Nov. 2001.

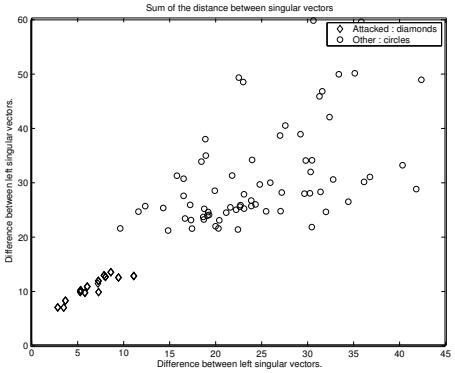

Fig. 3. SVD-SVD algorithm applied when the images are randomly chosen from a database of 5000 images. The difference between the hash values of original and attacked images: diamonds; the difference between the hash values of perceptually-different images: circles

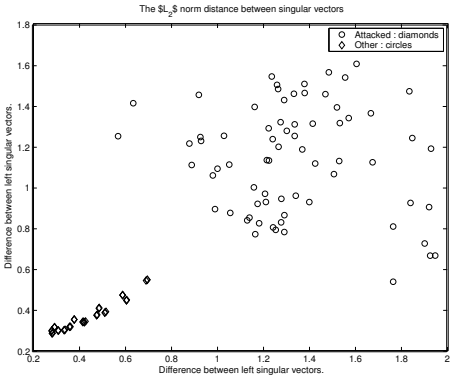

Fig. 4. DCT-SVD algorithm applied when the images are randomly chosen from a database of 5000 images. The difference between the hash values of original and attacked images: diamonds; the difference between the hash values of perceptually-different images: circles

[4] M. K. Mıhçak and R. Venkatesan, "A Perceptual Audio Hashing Algorithm: A Tool For Robust Audio Identification and Information Hiding," *Proc. of 4th International Information Hiding Workshop*, Pittsburgh, PA, April 2001.

[5] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," *Proceedings of ICIP 2000*, vol. 3 , pp. 664–666, Sep. 2000, Vancouver, Canada.

[6] J. Fridrich and M. Goljan, "Robust hash functions for digital watermarking," *Proceedings of International Conf. on Information Tech.: Coding and Computing*, Mar. 2000.

[7] M. K. Mıhçak, R. Venkatesan and M. Kesal, "Watermarking via Optimization Algorithms fr Quantizing Randomized Statistics of Image Regions", *Alerton Conference on Proceedings for the 37th Annual Allerton Conference*