

IMAGE HASHING RESILIENT TO GEOMETRIC AND FILTERING OPERATIONS

Ashwin Swaminathan, Yinian Mao and Min Wu

ECE Department, University of Maryland, College Park, U.S.A.

ABSTRACT

Image hash functions provide compact representations of images, which is useful for search and authentication applications. In this work, we have identified a general three step framework and proposed a new image hashing scheme that achieves a better overall performance than the existing approaches under various kinds of image processing distortions. By exploiting the properties of discrete polar Fourier transform and incorporating cryptographic keys, the proposed image hash is resilient to geometric and filtering operations, and is secure against guessing and forgery attacks.

1. INTRODUCTION

The wide availability of multimedia tools has created a gamut of possibilities to manipulate visual data. Hence, there is a need to design techniques to ensure image integrity. Robust image hashes and semi-fragile watermarks have been used for this purpose [1, 2, 3]. Some studies have shown that the image hashes have an upper hand as they can be used to authenticate the “visual content” of the image, rather than the exact representation [4, 5].

Cryptographic hash functions map input data to short binary strings and have been traditionally used in database searches [6]. However, most cryptographic hashes are very sensitive to the changes in every bit of the data. In applications such as in image databases and watermarking, it is often necessary that the hash be tolerant to a set of authentic modifications of the image. In these situations, the image hash function must be invariant to common filtering operations (such as averaging and median filtering), geometric distortions (such as rotation, scaling and translation or RST), JPEG compression and luminance non-linearity. Further, visually similar images need to be mapped to similar hash values and dissimilar images to different hash values.

Many of the existing images hashing schemes follow a three-step framework to generate the hash [2, 4]: (1) generate a key-dependent feature vector from the image, (2) quantize the feature vector, and (3) further compress the quantized vector. The most challenging part of this framework has been to generate a feature vector that is representative of the image yet resilient to minor distortions. A typical

approach is to identify some essential image features that can be considered invariant to the common image processing operations. These features are then used to generate the hash function.

The use of image edge information was initially proposed to generate the hash function, but the resulting hash would not be rotation invariant. To overcome the problem, invariant moments were employed and key-dependent linear or nonlinear combination [7] of image blocks were used to generate the hash. But the problem with these methods is that the invariant moments are publicly known and can be purposely modified to thwart the invariance [5].

An iterative key-dependent scheme based on repeated thresholding and spatial filtering was proposed in [1]. Although this scheme produces a hash invariant to filtering distortions, it was sensitive to rotation, translation and other affine transformations. In [4], principal values calculated from the wavelet transform of the image blocks were used to generate a hash invariant to general gray scale operations. The resultant statistics vector was then quantized using randomized quantization algorithms and then compressed by passing it through the decoding stage of a Reed-Muller error-correcting code [8]. A distributed coding approach using the *Wyner Ziv* encoder was presented in [2].

The algorithms described above perform well under additive noise and common filtering operations, but not under desynchronization and geometric operations. To counter these geometric operations, the RAdon Soft Hash algorithm (*RASH*) was proposed based on the properties of the Radon transform [3]. While this scheme performs well for images that have undergone RST distortions, it does not perform well for filtering and de-noising.

Ideally, a robust image hashing scheme should authenticate the image content and be invariant to common image processing operations. Directly modeling this distinguishing capability should take into account the human perception and understanding of image. In this paper, we approximate this distinguishing capability by explicitly considering a number of distortions such as filtering and geometric operations, and derive Fourier-domain image features that are resilient against these operations. We also incorporate cryptographic keys into the hash to achieve security against guessing and forgery. As will be shown in the experiment section, the resulting image hashing scheme achieves a better overall performance compared with the existing schemes.

The authors can be contacted via {ashwins, ymao, minwu}@eng.umd.edu. This work has been supported by NSF CAREER and URI grants.

2. IMAGE HASHING ALGORITHM BASED ON DISCRETE POLAR FOURIER TRANSFORM

2.1. Discrete Polar Fourier Transform Domain

Consider an image $i(x, y)$ and its rotated, scaled and translated version of the image $i'(x, y)$. We can relate them as

$$i'(x, y) = i(\sigma(x\cos\alpha + y\sin\alpha) - x_0, \sigma(-x\sin\alpha + y\cos\alpha) - y_0), \quad (1)$$

where the RST parameters are α , σ and (x_0, y_0) , respectively. The magnitude of the 2-D Fourier transform of $i'(x, y)$ can be written as

$$|I'(f_x, f_y)| = |\sigma|^{-2} |I(\sigma^{-1}(f_x\cos\alpha + f_y\sin\alpha), \sigma^{-1}(-f_x\sin\alpha + f_y\cos\alpha))|. \quad (2)$$

Consider now a transformation using polar coordinates $f_x = \rho\cos\theta$ and $f_y = \rho\sin\theta$, where $\rho \in [0, 1]$ is the normalized radius and $\theta \in [0, 2\pi)$ is the angle parameter. The equation (2) becomes

$$|I'(\rho, \theta)| = |\sigma|^{-2} |I(\rho\sigma^{-1}, \theta - \alpha)|. \quad (3)$$

The magnitude of the Fourier transform is independent of the translational parameters (x_0, y_0) . Further, we sum up $|I'(\rho, \theta)|$ along the θ -axis to make the result independent of the rotational parameter α [10, 11].

2.2. Basic Steps of the Proposed Algorithm

We first preprocess the image by applying a low-pass filter, scale the image $i(x, y)$ to a predetermined size, and perform histogram equalization. We apply a Fourier transform on the resulting preprocessed image and obtain $I(f_x, f_y)$. We then convert the transformed image to polar co-ordinates using an inexpensive linear interpolation technique as discussed in [11] to obtain $I'(\rho, \theta)$. We sum up $I'(\rho, \theta)$ along the θ -axis on K equidistant points in the range of $[0, 2\pi)$, i.e. for $\theta \in \{\frac{\pi}{K}, \frac{3\pi}{K}, \dots, \frac{(2K-1)\pi}{K}\}$, to obtain the image feature vector h_ρ :

$$h_\rho = \sum_{i=0}^{K-1} |I'(\rho, (2i+1)\pi/K)|, \quad (4)$$

where $K = 360$ is used in our implementation. We retain the low-frequency components of h_ρ as they capture the essential details of the image. Some high-frequency components corresponding to ρ close to 0.5 are discarded for better compression. We discretize (in ρ) the real part of h_ρ to form a statistic vector. We then quantize the resulting statistics vector and compress it to obtain the final hash. Techniques such as the *Wyner-Ziv* or the *Reed-Muller* decoder can be used to further compress the resulting hash [2, 8].

2.3. Securing the Hash

In applications involving image databases, image authentication and watermarking, it is often necessary that the hash

function be secure and dependent on a secret key. This key can be a unique number associated with a database to limit its access or a number associated with an author for proof of ownership. An attacker who does not know the key cannot generate the correct hash. In principle, the dependence on the secret key can be introduced in any of the three stages in the framework. Here we propose two different methods to incorporate the key in the feature extraction stage.

Scheme 1: We obtain $|I'(\rho, \theta)|$ as in equation (3) and do a weighted sum along the θ -axis to obtain the j^{th} hash value:

$$h_j = \sum_{i=0}^{K-1} \beta_{\rho_j, i} |I'(\rho_j, (2i+1)\pi/K)|, \quad (5)$$

where $\{\beta_{\rho_j, i}\}$ are key-dependent pseudo-random numbers normally distributed with mean and variance 1.

Scheme 2: We first use the secret key to generate random sets of radii $\{\Gamma_j\}$. We then use $|I'(\rho, \theta)|$ obtained in equation (3) and do a summation along the θ -axis for each radii in this set. A weighted sum of the resulting summations gives the j^{th} hash value. This can be represented as

$$h_j = \sum_{\rho \in \Gamma_j} \beta_\rho \sum_{i=0}^{K-1} |I'(\rho, (2i+1)\pi/K)|, \quad (6)$$

where β_ρ are key dependent pseudo-random numbers normally distributed with mean and variance 1.

Further, we can secure the quantization and compression stages of the hashing scheme by making them key-dependent. Randomized quantization algorithms [8] can be used to quantize the hash. The compression stage can also be made key-dependent by randomly selecting the order of the *Reed-Muller* decoder for different sub-sections of the hash. Doing this would further enhance the security of the resultant hash vector. The final hash is randomly permuted according to a permutation table generated using the key.

3. EXPERIMENTAL RESULTS

A database of 50 different images was chosen. Each image was then distorted to produce 120 different versions:

- 40 filtered versions, including 10 average filtered, 10 median filtered, 10 wiener filtered and 10 sharpened with filters of different orders;
- 30 geometrically distorted versions, including 10 rotated, 10 sheared and 10 cropped;
- 10 JPEG compressed with different quality factors;
- 20 images with different amounts of noise added;
- 10 images that underwent γ -correction.

Thus the database has around 6000 images. The proposed schemes 1 and 2 were implemented and the results obtained were compared with the existing schemes. The normalized Hamming distance between the hashes was used as a performance metric. The distance is expected to be close to 0 for similar images and close to 0.5 for dissimilar ones.

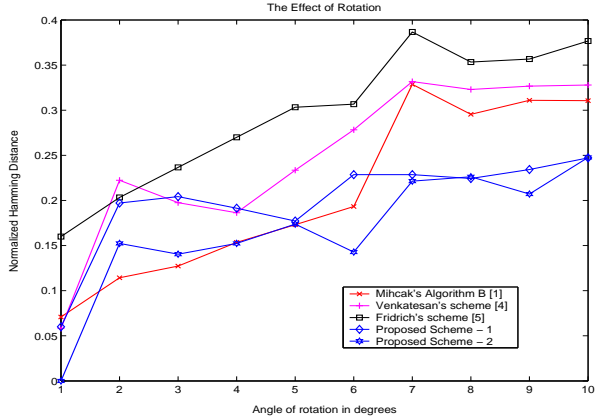


Fig. 1. Performance of various schemes for rotation

Our results indicate that the proposed schemes perform well for desynchronization distortions. The plot of the normalized Hamming distance with the angle of rotation is shown in Fig. 1. We observe that the distance between the quantized feature vectors for both the proposed schemes are smaller than those of the existing schemes, especially in larger rotation angles. This is expected as a rotation in image domain would lead to a rotation by the same amount in the Fourier transform domain, thus by summing along the θ axis, the effects of rotation are reduced. It is also observed that scheme-2 gives better results than scheme-1. This can be attributed to the fact that doing a weighted sum along the θ -axis no longer preserves rotation invariance. The proposed algorithms also perform significantly better than the existing schemes for cropping as shown in Fig. 2. In addition, the proposed algorithms achieve comparable performance with most existing algorithms for shearing operation as shown in Fig. 3, filtering and additive noise as shown in Fig. 4, and for JPEG compression as shown in Fig. 5.

To evaluate the overall effectiveness of the algorithm, we also need to study its performance with dissimilar inputs. For dissimilar inputs, an ideal hashing algorithm should produce hash vectors that are as far apart as possible. Our experiments with a separate set of 100 different images indicate that the mean of normalized hamming distance of the resulting 4950 combinations was around 0.48. To further demonstrate the performance of the proposed scheme, we consider the following model involving three images (a), (b), and (c), where the image (c) is a obtained by combining parts of the two images (a) and (b), and would be classified by many hashing applications as a different image. We perform this test on 25 images and list the normalized Hamming distance between the obtained hash vectors for different algorithms in table 1. We can see from the table that the proposed schemes find the image (c) to have large distance from (a) and (b), and thus correctly declare it inauthentic;

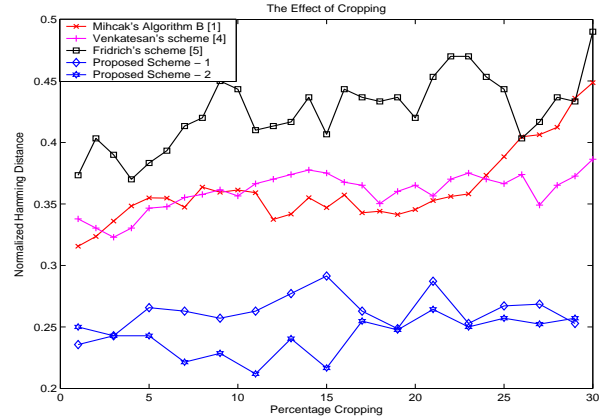


Fig. 2. Performance of various schemes for cropping

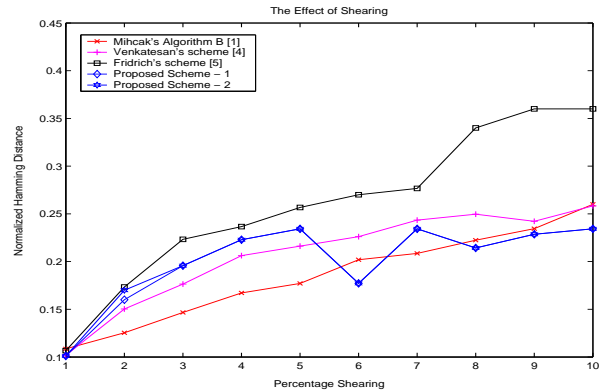


Fig. 3. Performance of various schemes for shearing

on the other hand, the existing algorithms suggest a smaller distance and have lower reliability to distinguish (c) from (a) and (b).

Table 1. Performance of the algorithm for dissimilar images. d_{ab} denotes the distance between images (a) and (b) and so on.

Hashing Method used	d_{ab}	d_{ac}	d_{bc}
Mihcak's Algorithm B [1]	0.49	0.21	0.27
Venkatesan's scheme [4]	0.39	0.16	0.31
Fridrich's scheme [5]	0.40	0.25	0.34
Proposed Scheme 1	0.47	0.29	0.37
Proposed Scheme 2	0.48	0.33	0.39

More generally, the problem of image authentication can be considered as a hypothesis testing problem with (1) H_0 : Image is not authentic; and (2) H_1 : Image is authentic. For each original, unprocessed image in the database, we examine its processed/distorted versions and record the number of them that are correctly classified as originated from the original image, which gives us an estimate of the probability of correct detection (P_D). We also record the number of processed versions of other images that are falsely classi-

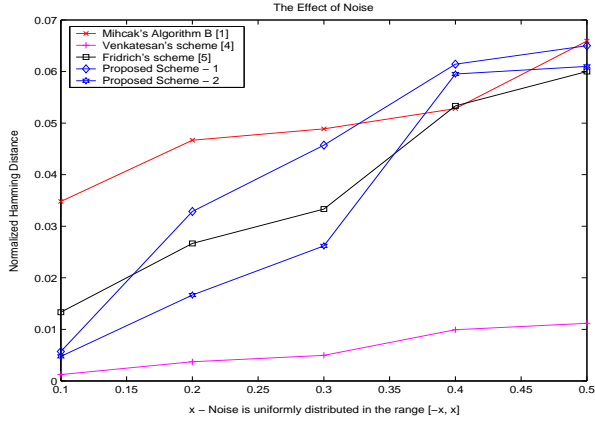


Fig. 4. Performance of various schemes for additive noise

fied and obtain an estimate of the probability of false alarm (P_F). We repeat this process for different decision thresholds and arrive at the Receiver Operating Characteristics as shown in Fig. 6. We can observe from the ROC curves that the proposed schemes attain a $P_D = 0.95$ when the P_F is 0.15, while the other schemes attain the same P_D when P_F close to 0.25. This further demonstrates the advantages of the proposed hashing schemes.

The resulting hash is also secure. We examined the normalized hamming distance between the two hashes of the same image generated using different keys and found it close to 0.5. This indicates that any person who does not know the secret key cannot generate the correct hash and hence cannot use it to search for a particular image in the database.

4. CONCLUSIONS

From a comparative study of the existing image hashing algorithms, we have observed that many algorithms do not simultaneously perform well for geometric distortions and filtering distortions. In this work, we have proposed an image hashing scheme resilient to common geometric and filtering operations. The proposed schemes were tested for many kinds of distortions and the results indicate that the proposed algorithm is able to distinguish authentic modifications from inauthentic ones better than the existing schemes. The resulting hash is also secure against guessing and forgery, and can provide a robust, secure, and compact representation of images for secure image database applications.

5. REFERENCES

[1] M. K. Mihcak and R. Venkatesan, "New Iterative Geometric Methods for Robust Perceptual Image Hashing," *Proc. of ACM Workshop on Security and Privacy in Dig. Rights Mgmt*, PA, Nov 2001.
 [2] M. Johnson and K. Ramachandran, "Dither-based Secure Image Hashing Using Distributed Coding," *Proceedings IEEE International Conference on Image Processing*, Spain, Sep 2003.

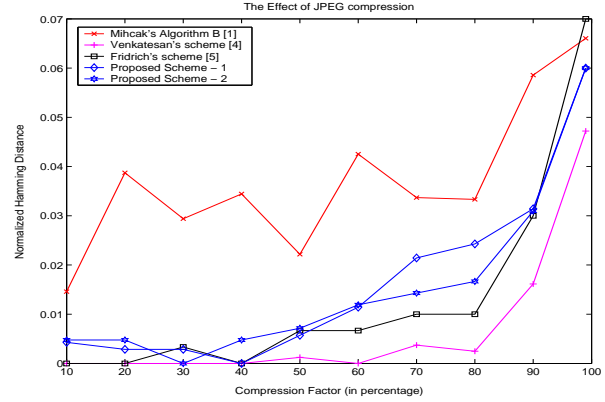


Fig. 5. Performance for JPEG compression

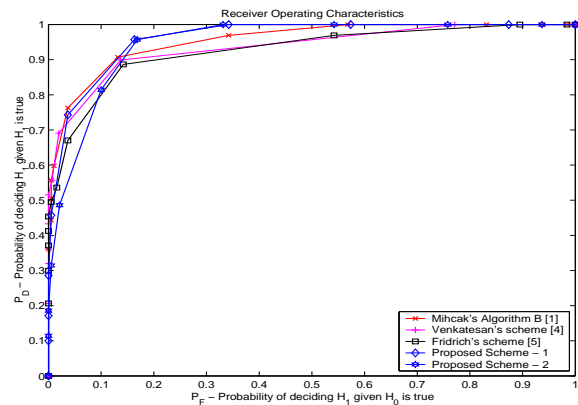


Fig. 6. The ROC of the hypothesis testing problem

[3] F. Lefbvre, B. Macq and J-D. Legat, "RASH: Radon Soft Hash algorithm," *Proc. EUSIPCO*, France, 2002.
 [4] R. Venkatesan, S. M. Koon, M. H. Jakubowski and P. Moulin, "Robust image hashing," *IEEE Proc. International Conference on Image Processing*, vol. 3, pp. 664 - 666, 10-13 Sep 2000.
 [5] J. Fridrich and M. Goljan, "Robust Hash functions for Digital Watermarking," *IEEE Proc. International Conference on Information Technology: Coding and Computing*, pp. 178 - 183, 27-29 Mar 2000.
 [6] A. Menezes, V. Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1998.
 [7] M. Kuei-Hu, "Visual Pattern Recognition by Moment Invariants," *IEEE Trans. on Information theory*, Vol. 8, pp. 179-187, Feb 1962.
 [8] M. K. Mihcak and R. Venkatesan, "A Tool For Robust Audio Information Hiding: A Perceptual Audio Hashing Algorithm," *Proceedings of 4th Intl. Information Hiding Workshop*, PA, Apr 2001.
 [9] R. E. Blahut, *Theory and Practice of Error-Correcting Codes*, Addison Wesley, 1994.
 [10] J. J. K. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Processing*, vol. 66, no. 3, pp. 303-317, 1998.
 [11] C-Y. Lin, M. Wu, J. A. Bloom, M. L. Miller, I. J. Cox and Y-M. Lui, "Rotation, Scale, and Translation Resilient Public Watermarking for Images," *IEEE Trans. on Image Proc.*, vol.10, no.5, May 2001.