
Copyright 2010 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.
Robust Image Hashing Using Higher Order Spectral Features

Brenden Chen and Vinod Chandran
Queensland University of Technology
Brisbane, Australia
e-mail: brenden_chen@msn.com

Abstract—Robust image hashing seeks to transform a given input image into a shorter hashed version using a key-dependent non-invertible transform. These image hashes can be used for watermarking, image integrity authentication or image indexing for fast retrieval. This paper introduces a new method of generating image hashes based on extracting Higher Order Spectral features from the Radon projection of an input image. The feature extraction process is non-invertible, non-linear and different hashes can be produced from the same image through the use of random permutations of the input. We show that the transform is robust to typical image transformations such as JPEG compression, noise, scaling, rotation, smoothing and cropping. We evaluate our system using a verification-style framework based on calculating false match, false non-match likelihoods using the publicly available Uncompressed Colour Image database (UCID) of 1320 images. We also compare our results to Swaminathan’s Fourier-Mellin based hashing method with at least 1% EER improvement under noise, scaling and sharpening.

Keywords—image hashing, high order spectra

I. INTRODUCTION

The rapid growth of digital communication and proliferation of digital image capture has led to a surge in the volume of digital images stored and transmitted. The increased use of computer-based image tools has also made it easy for the average individual to modify and edit these images. This creates a two-fold problem, firstly how to authenticate image data between two untrusting parties and secondly how to index images based on their visual content rather than user selected filenames or meta-data.

In traditional internet applications such as text or data transmission cryptographic hash functions and digital signature schemes are used to validate content integrity. However, these methods are designed for deterministic inputs where every bit remains identical and are not suitable for image hashing. This is because human image perception is robust to pixel value changes, for example JPEG compression can alter a significant number of pixels in an image without altering its perceptual content. Similarly, image retrieval based on a bit-by-bit (or pixel-by-pixel) comparison would be both computationally expensive and inaccurate when presented with certain distortions such as scaling or cropping.

As a result, a number of robust image hash algorithms have been proposed and a good hash function should display a number of invariance and security properties. We represent an image hash function as $H(.)$, the secret-key as $S$ and the resulting hash ($G$) is obtained from an input image ($I$) via $G = H(I, S)$.

1. Robust – The hash ($G$) produced from the image ($I$) should be invariant to common and minor image transformations such as noise, compression, geometric distortions, filtering and basic contrast changes.
2. Secure – The hash generation process ($H(., .)$) should be key-dependent so that without knowledge of the secret key ($S$) the correct hash ($G$) cannot be produced even if the image is known.
3. Non-invertible – Given the secret key ($S$) and full knowledge of the hashing algorithm ($H(., .)$) the original image ($I$) should not be reproducible from its hash value. Likewise, given the original image ($I$) and full knowledge of $H(., .)$ the secret key ($S$) should also be unrecoverable.
4. Collision resistant – Although it is not possible for hash functions to be collision free (since hashing is a many-to-one function), it should be computationally hard to find these collisions. That is, it should be infeasible to find a set of images ($I_1, I_2, I_3, \ldots$) where $I_1 \neq I_2 \neq I_3 \ldots$ but $H(I_1, S) = H(I_2, S) = H(I_3, S)$.

In this paper we describe a new robust image hash function that is non-linear, non-invertible and iterative. It is based on extracting Higher Order Spectral features from a 1D input using a technique first proposed by Chandran et al. [2]. The method has previously been shown to be robust to Gaussian noise, translation, DC offset and amplification. In order to serve as a hash function the method has been modified to be iterative and non-invertible and has been used for biometric cryptography [3] and template security [4]. The Radon transform is used to convert a 2D image into a set of 1D vectors suitable for the HOS feature extraction process as was done in [5]. We compare our technique against that of Swaminathan’s well known and high performing Fourier-Mellin technique [1]. We benchmark using a verification-style framework based on comparing equal error rates using the raw un-quantized features. We treat each image (and its modified versions) as a distinct class and use a Euclidean distance metric to compute intra and inter-class separation. This was chosen over the usual normalized hamming distance metric for two reasons, firstly the most challenging part of robust hash function is feature extraction [6] which is mostly independent of the quantization process.
Secondly, similar to biometric verification, hash functions need to be both robust and discriminative and equal error rates allow both to be compared simultaneously whereas a normalized hamming distance measures robustness alone.

The rest of this paper is organized as follows. Section 2 briefly summarizes existing methods with an extended description of Swaminathan’s polar Fourier technique. Section 3 outlines the steps of our HOS based image hash method. Section 4 illustrates the performance using the publicly available UCID database.

II. BACKGROUND

One of the earliest methods to display these properties was proposed by Fridrich et al. [7] who produced an image hash by projecting the image using a set of zero-mean random patterns generated based on the secret-key (usually a password used as the seed for a pseudorandom number generator). This technique was shown to be robust to noise and filtering-based image modifications but was not invariant to scale and rotation. It was also shown in [8] to be collision prone, an arbitrary image could be modified so that it produces the same hash as a known image by statistically modeling the hash bit generation process using Adaboost.

Venkatesen et al. [9] proposed a scheme where the wavelet transform was applied to a random block of the input image to extract robust features. The secret key is used to generate 4 random numbers \((x,y,w,h)\) that are used to describe the location and size of the random block (where \((x,y)\) are the coordinates of the top left corner of the block and \((w,h)\) are the width and height of the block respectively). The method was shown in [10] to be susceptible to a form of the hill-climb attack [11] where an iterative algorithm could be applied to search for a set of random numbers \((x,y,w,h)\) that produced the closest matching hash. Knowledge of \((x,y,w,h)\) is the equivalent of knowing the secret-key \((S)\).

Some other methods also rely on the Radon transform, in particular the Radon Soft Hash (RASH) [12, 13] but these methods use the Radon projection features directly without adding some form of key-dependency or non-invertability to the process. Ou and Rhee [14] also presented a Radon transform based hash function where the 1D-DCT was used to extract features from 40 randomly selected rotation angles. Their evaluation was limited to using a single image (Lena) compared against 500 impostor images. Controlled randomness is introduced by using secret-key dependent permutation of the DCT features. However, since the permutation is applied after feature extraction it changes only the ordering but not the actual feature values.

The use of HOS features, the fourth-order cumulants, was proposed by Weng and Preneel [15]. However their performance evaluation was limited to only 4 images and applicability of the method as an image hash function cannot be verified without more comprehensive testing.

Swaminathan et al. [1] proposed a Fourier-Mellin based image hash function that offered more robustness over a number of previous methods. Swaminathan takes the 2D Fourier Transform of the input image and obtains hash features by circularly summing the magnitude of Fourier coefficients that are a certain radius \((\rho)\) away from the origin (or zero-frequency component) as shown in Figure 1. The radii, \(\rho\) (rho), are dependent on image size and are normalized to be between 0 and 1. Sixty-four different values of \(\rho\), equally spaced between 0 and 0.4 were used by the authors [10] to produce a hash feature vector of length 64 denoted as \(K\).

Security and non-invertability is then achieved by linearly transforming the hash feature vector by applying a random projection using a matrix \(M\). The values in \(M\) are normally distributed random numbers of zero-mean and unit variance obtained using the secret-key \((S)\). Random projection is a well known technique in privacy preserving data-mining and biometric template security. It relies on matrix multiplication as means of data compression and if the elements of \(M\) are chosen using a statistical criterion then the Euclidean distance relationship between inputs can be maintained in the resulting projected space. However since the process is linear there exists a number of methods able to invert the random projection [16]. The authors in [10] show that if enough input/hash pairs produced using the same secret-key \((S)\) are known, then an attacker can recover the secret-key \((S)\). This is similar to Shannon’s concept of Unicity distance in encryption systems where the uncertainty of the encryption key is decreased for every plaintext/ciphertext pair known.

This concept can be practically applied to Random projections because matrix multiplication can be represented as a set of linear equations. It can be easily shown that an unknown matrix \(M\) of size \(NxN\) can be completely obtained from \(N\) independent sets of \(G\) and \(K\) where each \(G = MK\), this matrix multiplication can be written as \(N\) sets of \(N\) linear equations thus \(M\) can be solved with \(N\) sets of \(G\) and \(K\).

III. THE HOS IMAGE HASH

HOS based features extracted from a 1D signal input have a number of useful invariant features [2]. We briefly summarise these below before describing how a 2D image input can be meaningfully converted to a set of 1D vectors that can take advantage of these invariance properties.
1. HOS features have zero expected value for Gaussian noise.
2. Are invariant to both DC-offset and amplitude scaling, which can be trivially achieved through normalisation of the input signal.
3. Are invariant to scaling of the 1D input signal by either decimation or interpolation (shrinking or stretching of the 1D signal along the time-domain).
4. HOS features rely on Fourier phase information, which is a shape-dependent non-linear function of the frequency of the input.

The Radon transform is a well suited method of converting an image into a set of 1D vectors (Figure 2). It can be thought of as the summation of pixel values along each column as the image is rotated about its centre. It can then be easily seen that:

1. An image affected by Gaussian noise will produce 1D projections also affected by Gaussian noise since the summation of Gaussian distributed random variables is also Gaussian.
2. An image affected by uniformly distributed salt & pepper noise produces 1D projections that are DC offset by an amount related to the percentage of noisy pixels and to the ratio of salt to pepper noise.
3. Rotation of the image results in a circular shift of the set of 1D projections.
4. Scaling of the image that preserves aspect ratio (a scaling in both directions) results in a similar scaling of each of the 1D projections.
5. A change in overall image brightness results in a multiplicative amplification of each 1D projection.
6. Smoothing of the image results in a smoothing of the 1D projections, if the smoothing is relatively minor than the overall shape of the 1D projections will remain unchanged.
7. Compression of the image results in changes to pixel values of the image (dependent on compression amount). The 1D projections can remain largely unchanged since they are the summation of pixel values.
8. Cropping of the image by an equal amount along both axes results in a similar cropping of the 1D projections.

The Radon transform when practically applied to an image of size $M \times N$ produces 1D projections of length $L = \sqrt{M^2 + N^2}$ which corresponds to the diagonal length of the image. The projections produced from rotation angles other than those that orientate the image along its diagonal ($\theta = 45^\circ, 135^\circ, 225^\circ, 315^\circ$) will be shorter in length but are zero padded so that all projections are of equal length. Also the Radon transform at angles other than $0^\circ, 90^\circ, 180^\circ$ and $270^\circ$ results in an image that is not rectangular but diamond in shape with a thick centre that tapers to a point at each end (see Figure 3). This results in 1D projections that have an underlying shape dependent not only the image content but also on $\theta$, this is detrimental to the HOS feature extraction which is sensitive to shape (property 4). To overcome this, Radon projections are normalized by the number of pixels in each column. This can be done easily by dividing the original projections by the projections obtained from an identically sized image composed entirely of ones.

Each projection is then permuted using the secret-key ($S$), this allows for $L$-factorial possible different permutations. A moderately sized image of 256x256 has an $L$ of 362 resulting in 362! combinations that an attacker would need to brute-force search. Permuting in such a way can significantly alter the shape and frequency profile of a given input. The resulting HOS features are highly sensitive to such transformations.

Features can then be extracted from these 1D projections producing a separate feature vector for each value of $\theta$. Each of these 1D projections is first zero padded to four times its original length to improve spectral resolution. This is then scaled by the magnitude of the largest element and then made zero-mean by subtracting the mean. The $N$-point discrete Fourier transform (DFT) of this normalized vector $x(n)$ is then taken to obtain $X(f)$ from which the magnitude spectrum ($|X(f)|$) can be computed. The magnitude spectrum is then zero padded to length $N$ to produce:

$$y(n) = \begin{cases} 
|X(f)|, & n = 1, 2, ..., N/2 - 1 \\
0, & n = N/2, ..., N - 1 
\end{cases}$$

(1)

The bispectrum ($B(f_1, f_2) = X(f_1)X(f_2)X^*(f_1 + f_2)$) can be estimated from this by Fourier transforming $y(n)$. The phase spectrum is retained thus making the bispectrum complex valued with non-zero imaginary components and is sensitive to asymmetry. The bispectrum is then integrated along radial slices in the bifrequency plane to obtain:

$$V(a) = \int_{f_1=0}^{1/(1+a)} B(f_1, af_1)af_1$$

where $a = \frac{1}{N}, \frac{2}{N}, ..., 1$ (2)
Frequencies are normalized by the Nyquist frequency and the zero frequency component (or average signal) is eliminated from the above computation and \( a \) is the slope of the line in bifrequency \( (f_1,f_2) \) space along which the integral is computed.

The original process is modified to become iterative by feeding back the integrated bispectrum as a complex valued input vector of length \( N \) for the next iteration. The normalization step guarantees the transformation will be stable regardless of the number of iterations taken and also does not tend towards zero output owing to loss of precision alone. From each iteration a measure of change is extracted. This measure is the complex valued inner product (dot product) of the difference between the previous and present outputs with the previous output,

\[
D_i(n) = [x_{i-1}(n) - x_i(n)] \star [x_{i-1}(n)] \\
= \sum_{n=0}^{N-1} [x_{i-1}(n) - x_i(n)] \star [x_{i-1}(n)] \\
= M_i \exp(j\phi_i)
\]

where \( i \) represents the \( i \)-th iteration and \( D \) is the dot product that can be represented as a magnitude (\( M \)) and angle (\( \Phi \)) pair (one pair per iteration). This can be viewed as a method of controlled information loss since it is only the resultant product (magnitude/angle pair) that are stored rather than the integrated bispectrum itself. The nature of the dot product also means that slight changes in the input will result in only slight change in output. These magnitude/angle pairs are used as the transformed features. An \( i \)-length feature vector will be produced for each angle (\( \theta \)) used in the Radon transform. Feature-fusion or score-fusion can be applied to produce a single decision from this set of feature vectors. In this paper we apply the Radon transform at 4 angles (0\(^\circ\), 45\(^\circ\), 90\(^\circ\), 135\(^\circ\)) obtaining 4 separate feature vectors. We match each vector separately and employ score-fusion by using the minimum score obtained.

IV. EXPERIMENTAL RESULTS

In order to benchmark the performance of the HOS based image hash function we use the publicly available UCID database [17] which contain 1320 colour images of size 512x384 (or 384x512) of a wide range of content (see Figure 3). Each image undergoes two minor pre-processing steps:

1. The image is converted to grey-scale and pixels are normalised to have a value between 0 and 1.
2. If the image is 384x512 it is rotated by 90 degrees to be 512x384.

These pre-processed images form the training-set that is hashed using both Swaminathan’s Fourier-Mellin technique (as described in section 2) and our HOS image hash. Each image then undergoes a total of 22 different transformations (summarised in Table 1) and these form the test-set of 29040 images. Biometric style verification can be performed by attempting to verify each image in the test-set with one in the training-set using Euclidean distance as a measure of difference between feature vectors. The resulting false match, false non-match likelihoods are used to produce equal error rate (EER) measurements that can be used to compare both robustness and discriminative ability of the resulting image hashes.

Table 1 summarizes the EER performance of the two methods under image transformations over the entire UCID database. The HOS hash achieves a superior overall EER when larger transformations (variable 2) are applied and has much better performance when image are affected by noise (both Gaussian and salt & pepper), scaling and sharpening. The large improvements for noise and scaling can be attributed to the invariance properties of the HOS features that the hash function relies on. Swaminathan’s method is slightly better when images are filtered (both mean and median), JPEG compressed or have deleted boxes, although this difference is small. Swaminathan’s method does outperform the HOS hash when images are cropped or rotated. Once robust features are extracted any method of optimal quantization can be applied such as DROBA [18].

V. CONCLUSIONS

This paper has presented a new method of producing robust and secure image hashes. The technique takes advantage of invariance properties of Higher Order Spectral features extracted from Radon projections of the input image. The method is non-linear, iterative through feedback...
and includes a step of robust information loss in the form of a dot product at each iteration. Controlled randomness is introduced by permuting the image-derived Radon projections using a user/transmission dependent secret-key (S). The method is shown to be robust over the relatively large UCID database of real world images and out performs Swaminathan’s Fourier-Mellin image hash under certain transformations (particularly noise and scaling). The non-linear nature of transformation coupled with the novel method of controlled randomness makes the method difficult to analyse and provides increased complexity to brute force attack.

**REFERENCES**


<table>
<thead>
<tr>
<th>Transformation</th>
<th>Variable 1</th>
<th>Variable 2</th>
<th>Variable 1</th>
<th>Variable 2</th>
<th>Variable 1</th>
<th>Variable 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG Compression</td>
<td>80% Quality</td>
<td>10% Quality</td>
<td>0%</td>
<td>0.0056%</td>
<td>0%</td>
<td>0.1340%</td>
</tr>
<tr>
<td>Gaussian Noise</td>
<td>0.01 Variance</td>
<td>0.1 variance</td>
<td>1.23%</td>
<td>16.63%</td>
<td>0.1532%</td>
<td>1.64%</td>
</tr>
<tr>
<td>Salt &amp; Pepper Noise</td>
<td>5% pixels</td>
<td>40% pixels</td>
<td>1.65%</td>
<td>35.61%</td>
<td>0.2217%</td>
<td>3.42%</td>
</tr>
<tr>
<td>Scaling</td>
<td>1.02x</td>
<td>1.1x</td>
<td>1.06%</td>
<td>49%</td>
<td>0.0012%</td>
<td>0.0015%</td>
</tr>
<tr>
<td>Mean filtering</td>
<td>3x3 Window</td>
<td>15x15 Window</td>
<td>0.001%</td>
<td>0.4228%</td>
<td>0.003%</td>
<td>0.8284%</td>
</tr>
<tr>
<td>Median filtering</td>
<td>3x3 Window</td>
<td>15x15 Window</td>
<td>0.2207%</td>
<td>2.057%</td>
<td>0.4454%</td>
<td>2.84%</td>
</tr>
<tr>
<td>Deletion of Columns</td>
<td>3 columns</td>
<td>15 columns</td>
<td>0.0722%</td>
<td>1.26%</td>
<td>0.1516%</td>
<td>0.7378%</td>
</tr>
<tr>
<td>Deletion of Box</td>
<td>3x3 Box</td>
<td>15x15 box</td>
<td>0%</td>
<td>0.0002%</td>
<td>0%</td>
<td>0.2813%</td>
</tr>
<tr>
<td>Cropping of pixels from each border</td>
<td>1 pixel</td>
<td>15 pixels</td>
<td>0.0002%</td>
<td>4.51%</td>
<td>0.1477%</td>
<td>9.72%</td>
</tr>
<tr>
<td>Sharpening</td>
<td>3x3 laplacian</td>
<td></td>
<td>2.41%</td>
<td></td>
<td>0.4545%</td>
<td></td>
</tr>
<tr>
<td>Rotation</td>
<td>2 degrees</td>
<td></td>
<td>2.04%</td>
<td>19.75%</td>
<td>6.63%</td>
<td>28.18%</td>
</tr>
<tr>
<td>All transformations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.24%</td>
<td>17.13%</td>
</tr>
</tbody>
</table>

Table 1: Equal error rate (EER) comparison of the two image hash algorithms given images affected by different transformations. Rotated images are also cropped to remove zero padding after rotation then resized to their original size as was done in [1].