# 3
# A Chaos Based Robust and Secure Image Hashing Framework

Recently, several perceptual hashing schemes have been proposed in the literature, however, most of the schemes are unable to capture the robustness against strong geometric attacks [Tang *et al.*, 2014b]. To overcome robustness issue, a chaos based robust and secure hashing technique is proposed in this chapter. The problem is addressed using the geometric moments which provides an effective solution for geometric invariance. For this purpose image normalization procedure is employed to achieve the geometric invariance. The core idea lies behind normalizing is that image becomes invariant averse to orientation, scale, shearing in x and y-direction respectively. After normalization, the coefficients are transformed into frequency domain using block based DCT transformation. A non-linear chaotic map is deployed in the randomly block section process and then selected blocks are decomposed using singular value decomposition. Then, a Hessian matrix is constructed based on left and right singular vectors which produce the final hash value.

## 3.1 PROPOSED CHAOS-BASED HASH GENERATION PROCESS

In this section, the core idea used in the design of a robust hashing framework have been discussed. For this purpose, consider a gray-scale image $F$ of size $M \times N$ as the input to the hash function and the resultant hash value is a binary sequence of length $\ell$.

### 3.1.1 Image Pre-processing: Image Normalization

Image normalization is a process to make an image invariant against different geometric and general manipulations. It essentially transforms an image into another image such that it retains relevant information of the original image. This can be achieved by computing the basic geometric and central moments which are considered as the input parameter for the image normalization. Mathematically, the geometric moments of $(p+q)^{th}$ order of a gray-scale image ($I$) can be defined as.

$$m_{pq} = \iint_C x^p y^q I(x,y)dxdy \qquad p,q = 0,1,2,\ldots \tag{3.1}$$

where $C$ is the support of the image $I(x,y)$. In contrast, the central moments $\mu_{pq}$ can be defined as

$$\mu_{pq} = \iint_C (x-t_\alpha)^p (y-t_\beta)^q I(x,y)dxdy \quad p,q = 0,1,2,\ldots \tag{3.2}$$

where $t_\alpha = \frac{m_{10}}{m_{00}}$ and $t_\beta = \frac{m_{01}}{m_{00}}$ are the centroid of the image $I(x,y)$. On the basis of these geometric and central moments, the normalization process for a given image can be summarized as follows.

1. Translation invariance: For normalization, the input image $I(x,y)$ is translated by transforming the current position $p^{(1)} = (x_1, y_1)$ to the new position $p^{(2)} = (x_2, y_2)$ as follows.

   $$(x_2, y_2)^T = (x_1 - t_\alpha, y_1 - t_\beta)^T \tag{3.3}$$

   This step essentially eliminates translation, if any, by centring the image. Let the centred image is denoted by $I_1(x,y)$.

2. Shearing invariance: Apply the shearing transform to the image $I_2(x,y)$ in both $x$ and $y$-directions to generate a shearing invariant image $I_3(x,y)$. Mathematically, it is denoted as follows.

$$I_2(x,y) = \begin{bmatrix} 1 & \beta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix} I_1(x,y) \tag{3.4}$$

where $\beta$ and $\gamma$ can be computed by the following equations.

$$\beta^3\mu_{03} + 3\beta^2\mu_{12} + 3\beta\mu_{21} + \mu_{30} = 0 \tag{3.5}$$

$$\gamma\mu_{20} + \mu_{11} = 0 \tag{3.6}$$

The cubic equation in $\beta$ has maximum three roots, which may be complex or real or mixed.

3. Anisotropic scaling invariance: Scale the image $I_3(x,y)$ in both $x$ and $y$-directions with scale operator as

$$I_3(x,y) = \begin{bmatrix} \alpha & 0 \\ 0 & \delta \end{bmatrix} I_2(x,y) \tag{3.7}$$

where $\alpha$ and $\delta$ are the scaling parameters.

The resultant image $I_3$ is the normalized image, which is essentially invariant to the translation, rotation and scaling manipulations. More precisely, any translation of the affine attack is eliminated by centring the possibly distorted image, the shearing effect (including the rotation) is eliminated by the step 2, and finally the scaling distortion is eliminated by forcing the image to desired size. It can be observed that each step of the normalization process is invertible and thus the original image can be reconstructed, as per requirement.

### 3.1.2 Hash Sequence Construction

The main steps of the hash generation process can be summarized as:

1. Apply normalization process to the image $F$ as described in the Section 3.1.1. Let $F^{(N)}$ denote the normalized image.

2. Partition the normalized image $F^{(N)}$ into non-overlapping blocks $\{B_j | j = 1,2\ldots\ell\}$ of size $z \times z$, where $\ell = (M \times N)/z^2$.

3. Generate a random sequence $S$ using a secret key $S_{key}$ and a non-linear chaotic map.

$$S = \{S_k | k = 1,2\ldots L \leq \ell\} \text{ where } S_k \in \{0,1\} \tag{3.8}$$

4. Create an array $S_R$ from $S$ for block selection as follows.

$$S_R = \lfloor (S * 2^{12}) \rfloor \bmod \ell \tag{3.9}$$

5. Select the random blocks from $S_R$ as follows.

$$B_s = \{B_j | j = 1,2\ldots S_k \text{ and } k = 1,2\ldots L\} \tag{3.10}$$

6. Apply DCT transformation to the block $B_s$.

$$B_s^{(f)} = DCT\{B_s\} \tag{3.11}$$

7. Perform singular value decomposition on transformed coefficients $B_s^{(F)}$.

$$B_s^{(f)} = U_s^{(f)} S_s^{(f)} V_s^{T(f)} \tag{3.12}$$

8. Obtain a vector $h$ using left and right singular vectors corresponding to highest singular value as $h = \left[U_h^{(f)}, V_h^{(f)}\right]$.

9. Construct a 2D-array $H$ by stacking the elements of $h$ and apply the SVD on it.

$$H = U^{(H)} S^{(H)} V^{H(T)} \tag{3.13}$$

10. Create Hessian matrix $Q$ using $U^{(H)}$ and $V^H$ at each position $(x,y)$ as follows.

$$Q_m(x,y) = \begin{bmatrix} q_{11}^m & q_{12}^m \\ q_{21}^m & q_{22}^m \end{bmatrix} \tag{3.14}$$

where $m \in \{U^{(H)}, v^{(H)}\}$ and coefficients $q_{11}^m, q_{12}^m, q_{21}^m, q_{22}^m$ are defined as:

$$q_{11}^m = m(x+1,y) + m(x-1,y) + m(x,y) \tag{3.15}$$

$$q_{12}^m = q_{21}^m = m(x+1,y) + m(x,y-1) + m(x,y) \tag{3.16}$$

$$q_{22}^m = \frac{1}{4}[m(x+1,y+1) - m(x+1,y-1) - m(x-1,y+1) + m(x-1,y-1)] \tag{3.17}$$

11. Obtained a binary matrix as follows:

$$F_m(x,y) = \begin{cases} 1, & |Q_{U^{(H)}}| \leq |Q_{V^{(H)}}| \\ 0, & \text{otherwise} \end{cases} \tag{3.18}$$

12. Stack the elements of matrix $(F_m)$ into a vector $F_H$.

13. The hash vector $F_H$ is randomly permuted to generate the final hash value.

### 3.2 EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the performance of the proposed hashing framework scheme is extensively evaluated using MATLAB platform. For this purpose, hamming distance is computed between the hash pairs and then normalized it with respect to length of the hashes. Mathematically, the normalize hamming distance (NHD) can be computed as follows:

$$NHD = d(H_1, H_2) = \frac{\sum_{j=1}^{N} |(H_1(j) - H_2(j))|}{N} \tag{3.19}$$

where $H_1(j)$ and $H_2(j)$ denote $j^{th}$ element of the hash $H_1$ and $H_2$ respectively. The estimated NHD can be divided into two sets: $I_1 = [0, \lambda)$ and $I_2 = [\lambda, b]$ where $b > \lambda$ and $\lambda > 0$ is prefixed threshold value. If NHD belongs to set $I_1$, the image pairs are considered as perceptually similar whereas NHD belonging to set $I_2$ indicate that image pairs are not identical or image is maliciously modified. In general, an ideal hashing scheme should have the ability to classify the similar or non-similar images. Also, the perceptual hashing scheme must be completely secure. In proposed scheme, a random block selection process is employed in the hash generation process to further

(a) (b) (c) (d) (e)

**Figure 3.1 :** Experimental Images: (a) Cameraman, (b) Lena, (c) Barbara, (d) Goldhill, (e) Boat.

strengthen the security. For this purpose, mid-points of all the sides of the normalized images are connected forming a diamond shaped structure. This diamond-shape is then partitioned into non-overlapping blocks followed by the random block selection to construct the hash sequence. In this manner, diamond shaped region is served as the domain (regardless of the nature of the attack) for the block selection. Finally, the block selection has been done on the basis of non-linear chaotic map. Therefore, the initial seed will be stored to ensure precise selection of the blocks even after different image processing and geometric attacks. The performance of the proposed scheme is examined using robustness, sensitivity, discriminative capability and key dependence analysis. The descriptions of these analysis can be summarized as given below.

### 3.2.1 Robustness Analysis

The robustness of the proposed scheme is evaluated considering various content preserving operations such as Gaussian noise addition, salt & pepper, speckle noise, Gaussian blur, cropping, rotation, scaling and JPEG compression. For this purpose, five standard images namely Cameraman, Lena, Barbra, Goldhill and Boat of size $512 \times 512$ are considered as the test images, which are shown in Fig. 3.1. In first experiment, the robustness of the scheme is measured against rotation attack. In image rotation, the original position of the pixel is changed due to rotational transformation with appropriate rotation angle. Hence, test images are rotated by the angle $10^\circ$, $20^\circ$, $30^\circ$, $40^\circ$ and $50^\circ$ respectively, then normalized hamming distance is computed between the rotated and original images. The estimated NHD are shown in Fig. 3.2(a). From the figure, it can be observed that maximum and minimum hamming distance are 0.1 and 0.01 corresponding to Barbara and Boat image respectively.

The performance of the scheme is also analyzed against image scaling operation. Image scaling is a geometric operation, in which size of the image is increased or decreased according to requirement. In this experiment, size of the test image is decreased by scale factor 0.75, 0.85, 0.95 and increased by 1.05, 1.15, 1.25 respectively and after this estimate the NHD between the scaled and original test images which is depicted in Fig. 3.2(b). The maximum and minimum NHD is 0.08 and 0.005 corresponding to Lena and cameraman image respectively. The effectiveness of the scheme is also tested against additive Gaussian noise. For this purpose, zero mean Gaussian noise is applied on test images with different variance 0.02, 0.04, 0.06, 0.08 and 0.10 respectively. The NHD is determined between the original and noisy test images as shown in Fig. 3.2(c). The maximum and minimum NHD against AGN is 0.083 and 0.023 corresponding to Barbara and Goldhill image respectively. In addition, the performance is also investigated against JPEG compression. Data compression is fundamental operation and widely used in day to day life to reduce size the digital data. Hence, JPEG compression is employed on the test images by reducing the respective size 15%, 30%, 45%, 60%, 75% and 90%. The NHD is computed between the original and compressed test images as depicted in Fig. 3.2(d). The maximum and minimum NHD against JPEG compression is 0.077 and 0.03 with respect to Lena and Boat image. Further, the validity of the pro-

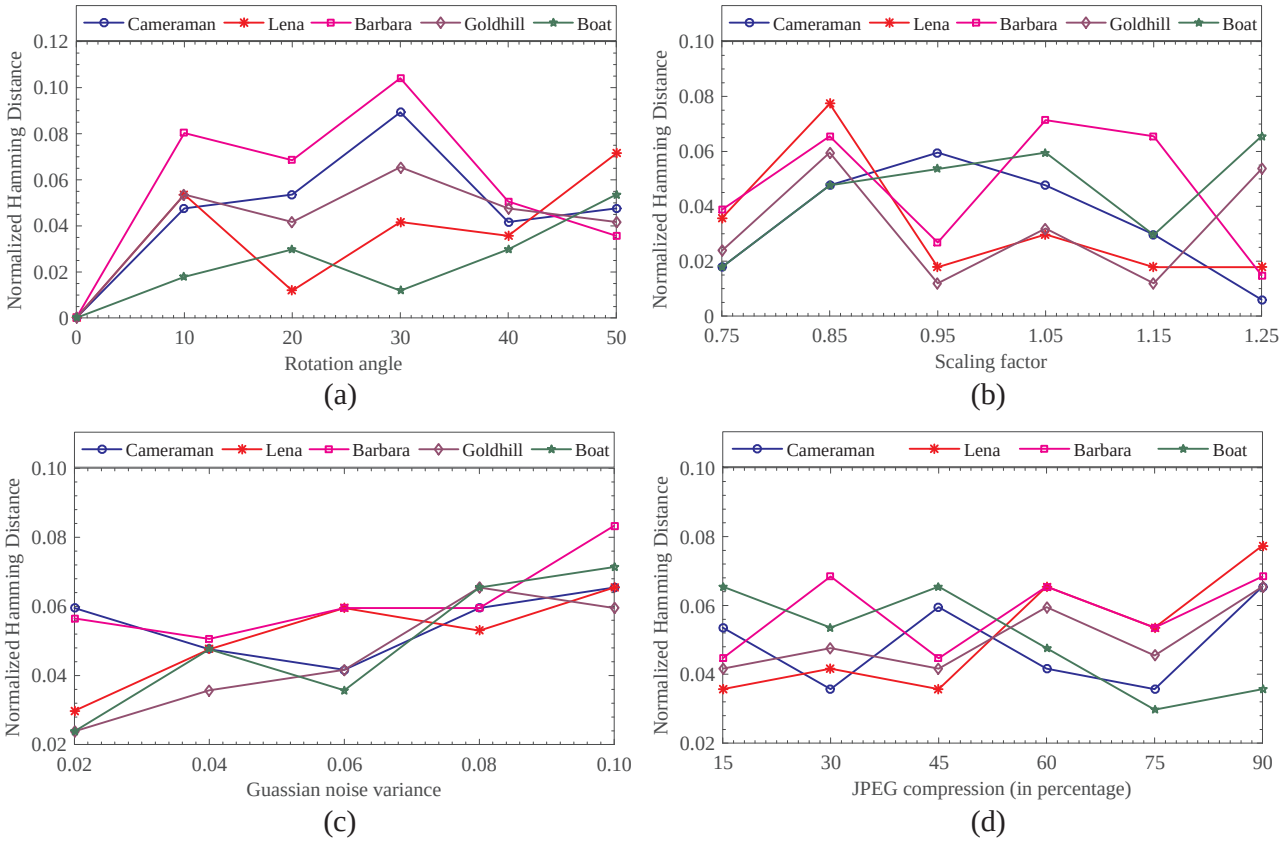


**Figure 3.2 :** Distribution of normalized hamming distances: (a) Rotation, (b) Scaling, (c) Gaussian noise addition and (d) JPEG compression

posed system is illustrated through salt & pepper (30%), speckle noise (30%), Gaussian blur $(5 \times 5)$, average filter $(5 \times 5)$, median filter $(5 \times 5)$, cropping (30%), shearing and translation. The NHD is determined for the experimental images and corresponding average are depicted in Table 3.1.

**Table 3.1 :** NHD for the set of different image processing attacks.

| Attacks | Proposed | [Neelima and Singh, 2016] | [Monga and Evans, 2006] |
|---|---|---|---|
| Salt & pepper noise (30%) | 0.0242 | 0.1768 | 0.2517 |
| Speckle noise (30% ) | 0.0390 | 0.1437 | 0.2112 |
| Gaussian Blur | 0.0535 | 0.1711 | 0.2397 |
| Average Filter $(5 \times 5)$ | 0.0223 | 0.1664 | 0.2463 |
| Median Filter $(5 \times 5)$ | 0.0369 | 0.2115 | 0.3282 |
| Cropping (30% ) | 0.0634 | 0.2654 | 0.3541 |
| Shearing | 0.0582 | 0.2103 | 0.3228 |
| Translation | 0.0401 | 0.3615 | 0.4214 |
| Rotation (25°) | 0.0625 | 0.4165 | 0.5499 |
| Scaling (30% ) | 0.0315 | 0.0578 | 0.0316 |

The performance analysis of the proposed hashing scheme is compared with the existing schemes given in [Monga and Evans, 2006; Neelima and Singh, 2016]. For this purpose, average NHD is computed through hashes obtained for the content preserving operations of the experimental images and compared with that of existing schemes. The average NHD is estimated against rotations, scaling operations, Gaussian noise and JPEG compression as depicted in Fig. 3.3(a-d). From the figure, it can be observed that estimated average NHD is minimum in the comparison of the other existing schemes [Monga and Evans, 2006; Neelima and Singh, 2016] against rotations, scale factors and with different noise variances. However, average NHD is almost equivalent to scheme [Neelima and Singh, 2016] and lesser than [Monga and Evans, 2006] for JPEG compression. In addition, the performance is also compared against other standard operations by computing average NHD and are shown in Table 3.1. The results validate the efficiency against operations such as cropping and Gaussian blurring. Hence, overall performance of the proposed hashing scheme is better than the existing schemes against different intentional/unintentional operations.

### 3.2.2 Discriminative Capability

Discriminative Capability is another key factor that defines the strength and efficiency of the hashing system. In principle, a perfect hashing system has ability to discriminate between the perceptually similar or maliciously modified image. The main reason is that the estimated hashes corresponding to original and perceptually similar images have better similarity, whereas more dissimilarity is found in the hashes of the maliciously modified image. Therefore, the performance of the scheme is measured against malicious or inauthentic manipulation. For this purpose, the test image shown in Fig. 3.4(a) is edited by the copy-paste operation. In this operation, a particular object in the image is copied and pasted at one or more locations in the same image. The resultant modified images are shown in Fig. 3.2(b-c). The pairwise NHD is computed between the hashes

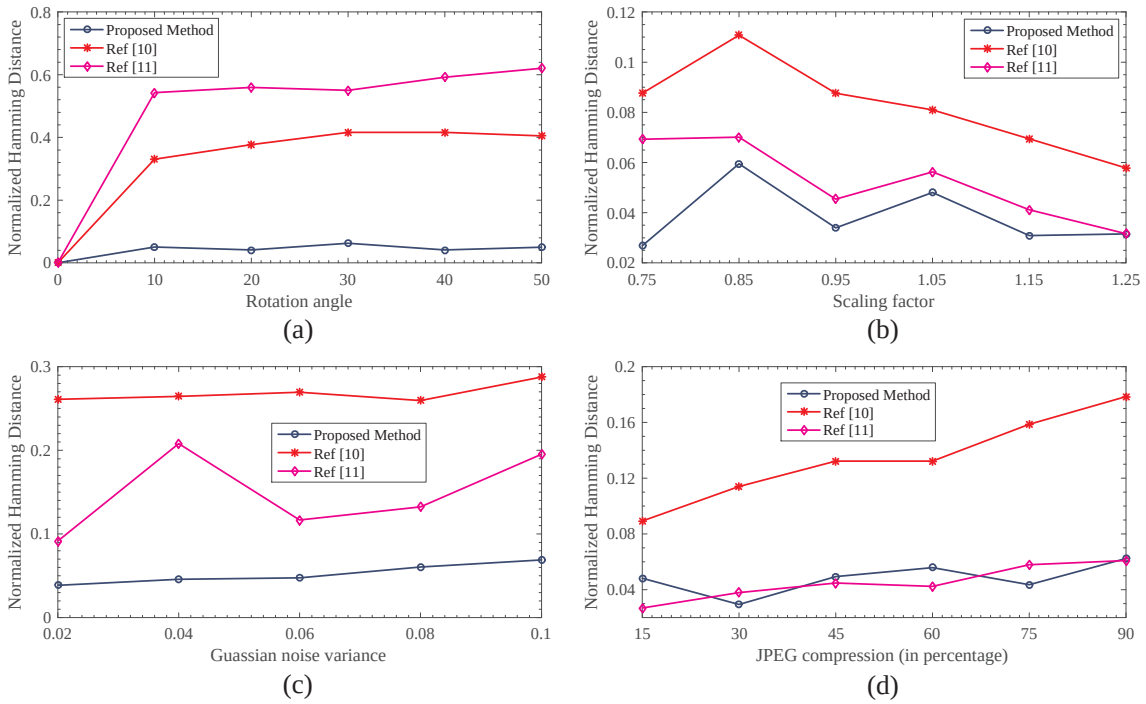


**Figure 3.3 :** Performance of various hashing schemes: (a) Rotation, (b) Scaling, (c) Gaussian noise addition, (d) JPEG compression.

(a) (b) (c) (d)

**Figure 3.4 :** (a) Original image, (b,c,d) Maliciously modified images.

**Table 3.2 :** Normalized hamming distance between image pairs.

| Image Pairs | NHD | Image Pairs | NHD |
|---|---|---|---|
| (a,b) | 0.297 | (b,c) | 0.320 |
| (a,c) | 0.321 | (b,d) | 0.313 |
| (a,d) | 0.333 | (c,d) | 0.351 |

obtained from the test and modified images. The complied NHDs are depicted in Table 3.2. From the table, the obtained NHD is larger than NHD corresponding to content preserving operations. It signifies that either image is not perceptually similar to the original one or image is forged with malicious modification. Hence, proposed hashing scheme has good discriminative capability.

### 3.2.3 Key Dependence

The standard experimental images as described in Section 4.1 are used to validate the key dependence of the proposed hashing system. For this purpose, different secret keys are exploited to generate the hash value for each image and then compute the normalize hamming distance between hash pairs. The obtained results show that all the hamming distance are large enough. For space limitation, the key dependence has been presented considering the 'Cameraman' image.

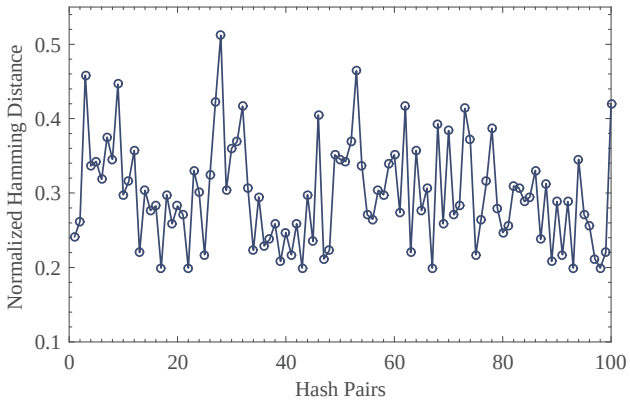


**Figure 3.5 :** Distance between hash using wrong keys

Firstly, consider a key $K_1$ to generate the perceptual hash and then 100 different keys are used to generate the perceptual hash corresponding the same image. During the experiments, all other parameters remain unchanged. The normalized hamming distance between the hash corresponding to key $K_1$ and other 100 hashes were estimated and shown in Fig. 3.5.

## 3.3 SUMMARY

In this work, a new efficient and reliable hashing technique has been developed which provides the solution to the problems of geometric deformation in the image hashing framework. In this technique, the input image is normalized for invariance feature extraction and the corresponding hash value can be used for image indexing and authentication purpose. The performance proposed of the scheme is evaluated against various types of geometric operations as well as signal processing operations. The simulated results demonstrate the better robustness against various type content preserving operations.