



# LWT-DCT based image hashing for image authentication via blind geometric correction

Ram Kumar Karsh<sup>1</sup>

Received: 8 July 2020 / Revised: 4 October 2021 / Accepted: 2 June 2022 /

Published online: 13 June 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Image authentication based on robust image hashing has been paid large attention by researchers. However, most of the existing methods are unable to authenticate, if the image is processed through geometric transformations and tampered. In this paper, we have proposed a blind geometric correction approach, which eliminates the effect of geometric transformation, including rotation-scaling-translation (RST). We have incorporated Lifting Wavelet Transform (LWT) and Discrete Cosine Transform (DCT) to construct a short hash. Furthermore, an algorithm to generate an image map from the hash is proposed to detect the tampered regions. The main objective is to keep the hash length short with better performance, i.e., perceptually robust to content-preserving operations and image tampering detection. Based on the difference of image maps obtained from “source image” and “query images”, tampering regions have been localized. The proposed method can detect tampering, even if tampering and composite RST geometric transformations occur simultaneously, due to blind geometric correction. The experimental results show that the proposed image authentication method outperforms the state-of-the-art techniques.

**Keywords** Image hashing · LWT · DCT · Geometric transformation · Tamper detection

## 1 Introduction

The growth of advanced image editing tools forced us to think about sophisticated authentication mechanisms [22]. Distinguishing original images from fake ones and localizing the tampered area is a challenging issue for industries and academia. Recently, an image hashing-based approach has been widely used for image authentication as well

---

✉ Ram Kumar Karsh  
ram@ece.nits.ac.in

<sup>1</sup> Speech and Image Processing Group, Electronics and Communication Engineering Department, National Institute of Technology, 788010 Assam, India

as tampered area localization. In this technique, an image is represented by an image hash or a digital signature, which is a visual representation of image contents. An image hash should be robust to content-preserving operations (CPOs) such as compression, rotation, translation, etc., and reactive to change in content manipulations [31]. The image hashing approach has been used in different applications like image authentication [24, 38], image retrieval [4, 37], tampering detection [3, 12–14, 25], and some other applications [16, 17]. An image hashing technique was first introduced by Venkatesan et al. [33]. In this method, wavelet coefficients are extracted to form an image signature, which is robust in compression, geometric distortions, but sensitive to some CPOs. To authenticate and localize the area of tampering, Roy and Sun [27] first introduced a block-based image hashing approach. This method is robust to compression, rotation, but translation has not been explored. Ahmed et al. [2] considered the LL band of DWT from the  $16 \times 16$  sizes of image blocks to construct hash. This method can detect tampering and robust against filtering operation as well as compression, but the hash length is very high. To achieve better authentication and tampering localization approach, Pun et al. [25] combined global and local features to construct a hash. This method is robust to compression, filtering, and rotation up to 5 degrees, but limited in the case of combined RST transformation.

There are some image alignment techniques [3, 20, 21, 35] to eliminate the impact of geometric transformations, which have been employed to authenticate images and to localize tampering. These image alignment methods require too long hash to reconstruct the original image from the geometric translated one. Lu et al. [21] introduced the theory of scale space and the Radon transform to obtain the parameters of geometric distortion and reconstructed the image. In another work, Lu and Wu [20] improved [21] using SIFT features. Battiato et al. [3] presented a new approach using a voting procedure in the space model. The performance of the geometric correction in [3] is improved compared to [20, 21], but the major limitation is too long hash length, i.e., 1000 digits. Yan et al. [35] represented an image alignment approach based on a quaternion Fourier-Mellin transform. In this method, the estimation of geometric parameters has been affected by tampering operations. In another study, Karsh et al. [14] introduced image alignment based on a furthest non-zero pixel, but this method is limited only to the positive angle of rotations.

Yan et al. [36] represented an image hashing based on multi-scale, where an image is divided into multiple rings. This method is robust to most CPOs, but sensitive to composite RST attacks. In another work, Yan et al. [34] presented tamper detection based on multi-scale difference map fusion. This method is robust against some CPOs, but sensitive to translation. The hashing techniques [8, 9, 11, 23, 26, 28–30] are applied for image authentication, but fail to locate tampering, if composite RST and tampering occur simultaneously.

It can be observed from the literature that most of the existing image authentication methods are sensitive to composite RST transformation. Also, the tampered area may not be localized, if tampering and composite RST transformation occur simultaneously. Based on the limitation of the existing methods, the main contribution of the proposed work can be encapsulated as follows.

The main contributions of the work are as follows.

- It has been observed from the literature that most of the existing image authentication methods are sensitive to composite RST transformation. Also, the tampered area may not

be localized, if tampering and composite RST transformation occur simultaneously. Hence, a combination of LWT and a modified image compression approach based on DCT has been proposed to construct a short image hash.

- The image map has been constructed using the proposed modified image de-compression via inverse DCT of hash. The difference of image map from the received hash and received image yields tamper localization. As per our literature survey, this is the first-time hashing based on LWT and modified image compression based on DCT, used for image authentication and tampering area localization, in the proposed work.
- Besides, a modified blind geometric distortion correction approach based on inherent geometric characteristics has been proposed. Due to this, the proposed method is robust to composite RST transformation. Also, the proposed system obtained the tampering location, where tampering and geometric transformation occur simultaneously, which is a major limitation in the state-of-the-art methods.

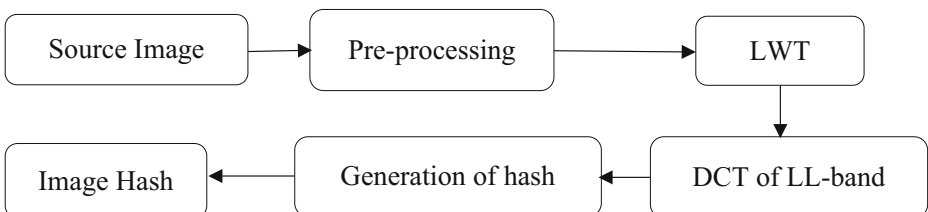
The arrangement of the paper is as follows. Section 2 represents a detailed description of the proposed image hashing methodology. Section 3 demonstrates the proposed blind geometric distortion correction approach, image authentication, and tampering localization. The experimental results and analysis have been discussed in Section 4. Finally, the conclusions and future scopes are mentioned in Section 5.

## 2 Proposed image hashing

The proposed image hashing method is shown in Fig. 1. Brief descriptions of each step of the proposed image hashing are drawn in the subsequent subsections.

### 2.1 Preprocessing

An arbitrary size of the source image is resized into  $p \times p$  using bilinear interpolation. Here, image resizing is necessary to keep the hash length fixed and maintain robustness against scaling operation. If the source image is an RGB image, it is mapped into CIE  $L^*a^*b^*$  color space [6], and the intensity component ( $L^*$ ) has been considered for further processing. The reason behind choosing the CIE  $L^*a^*b^*$  color space is that it is perceptually uniform and reasonably related to human perception space [31]. Hence, features extracted from the  $L^*$  component are stable compared to other color spaces.



**Fig. 1** Depicts a procedure to generate a hash code from an image

### 2.2 LWT

The pre-processed image is decomposed using LWT up to the third level [7], as shown in Fig. 2. By experiment, to maintain trade-off between the length of hash and discrimination performance, the LL3 sub-band, let it  $\mathbf{F}$  a square matrix sized  $q \times q$ , has been selected for further processing, which approximates the pre-processed image. The reason to use LWT after pre-processing is to reduce the dimension, while keeping most information intact. Furthermore, DCT is applied on  $\mathbf{F}$  to get the compressed hash, discussed in the following subsection.

### 2.3 DCT (Discrete Cosine Transform)

The mathematical expressions to obtain DCT of  $\mathbf{F}$  is discussed as follows. Firstly,  $\mathbf{F}$  is divided into non-overlapping blocks of sized  $m \times m$ . For simplicity, let  $q$  be an integral multiple of  $m$ . Hence, the total number of blocks is  $\delta = (q/m)^2$ . Let  $\mathbf{M}_k(x, y); 0 \leq x, y \leq m - 1$ , be the  $k$ -th block indexed from left to right and top to bottom ( $1 \leq k \leq \delta$ ). Then, 2-D discrete cosine transform of  $\mathbf{M}_k(x, y)$  has been obtained as follows:

$$\mathbf{B}_k(u, v) = \mathbf{T}(u, v) \times \mathbf{M}_k(x, y) \times \mathbf{T}'(u, v) \tag{1}$$

Here, three matrices  $\mathbf{T}$ ,  $\mathbf{M}$ , and  $\mathbf{T}'$  are multiplied ( $1 \leq k \leq \delta$ ), where  $\mathbf{T}(u, v)$  is a discrete cosine transform matrix [1, 18], shown in Appendix 1.  $\mathbf{T}'(u, v)$  represents a transpose of  $\mathbf{T}(u, v)$ .  $\mathbf{B}_k(u, v)$  is the DCT of  $k^{th}$  ( $1 \leq k \leq \delta$ ) non-overlapping pixel blocks. The reason to use DCT after LWT is its energy compaction property. Most of the contents of an image may be represented using only a few low-frequency components in the transform domain (in this paper,  $n$  represents the number of low-frequency DCT coefficients from each  $k$ -th block). From these low-frequency coefficients, using inverse DCT (IDCT), the approximate image contents may be reconstructed (in this manuscript is known as an image map, which has been used for

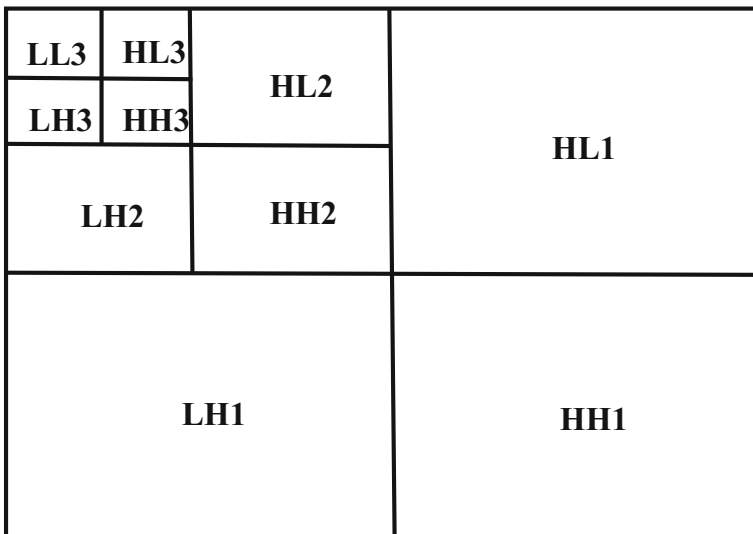


Fig. 2 Sketch map, LWT decomposition of an image. **H** and **L** indicate high and low, respectively. Number represents level

tampered area localization, discussed in sub-Section 3.3). The  $n$  low-frequency DCT coefficients from each  $k$ -th block are selected and concatenated to generate a short image hash, discussed in the following subsection.

### 2.4 Generation of an image hash

Let  $n$  low-frequency DCT coefficients are selected from each  $k^{th}$  ( $1 \leq k \leq \delta$ ) matrices,  $\mathbf{B}_k(u, v)$ , using zigzag ordering, as shown in Appendix 2. It has been observed that low-frequency components are represented in the front side of the zigzag sequence, while high-frequency coefficients are in the later parts. Here, we select the first  $n$  coefficients in the zigzag sequence to form a vector  $\mathbf{h}^k$  sized  $1 \times n$ . Now, concatenate  $n$  low-frequency DCT coefficients from  $k^{th}$  blocks yields image hash,  $\mathbf{h}$ , of  $n \times \delta = r$  digits as follows.

$$\mathbf{h} = [h^1(1), h^1(2), \dots, h^1(n), h^2(1), h^2(2), \dots, h^2(n), \dots, h^\delta(1), h^\delta(2), \dots, h^\delta(n)] \quad (2)$$

The length of the final image hash is  $r$  digits. Also, the details of the hash generation, in implementation form, are shown in algorithm 1.

**Algorithm 1:** Pseudocode for hash generation.

---

**Input:** Source image,  $\mathbf{I}$   
**Output:** An image hash  $\mathbf{h}$

- 1: An arbitrary size source image,  $\mathbf{I}$ , is mapped to size  $p \times p$ , i.e.,  $\mathbf{I}'$ .
- 2:  $\mathbf{I}'$  is converted into CIE  $L^*a^*b^*$ .
- 3:  $L^*$  component is decomposed using LWT up to 3<sup>rd</sup> level, as shown in Fig. 2.
- 4: LL3 sub-band, let it be  $f(x, y)$  is divided into  $k^{th}$  ( $1 \leq k \leq \delta$ ) non-overlapping blocks,  $\mathbf{M}_k(x, y)$ .
- 5: Find DCT of  $\mathbf{M}_k(x, y)$  as  $\mathbf{B}_k(u, v) = \mathbf{T}(u, v) \times \mathbf{M}_k(x, y) \times \mathbf{T}'(u, v)$  yields  
 $\mathbf{B}(u, v) = \begin{bmatrix} \mathbf{B}_1(u, v) & \mathbf{B}_2(u, v) \\ \mathbf{B}_3(u, v) & \mathbf{B}_4(u, v) \end{bmatrix}$  (For example, DCT of  $f(x, y)$ , if  $f(x, y)$  is divided into four non-overlapping regions,  $\mathbf{M}_k(x, y)$ , i.e.,  $\delta = 4$ )
- 6: Now,  $n$  low-frequency DCT coefficients are selected from  $\mathbf{B}_k(u, v)$  ( $1 \leq k \leq \delta$ ) using zigzag ordering.
- 7: Concatenate  $n$  low frequency DCT coefficients from  $k^{th}$  blocks yields image hash,  $\mathbf{h}$ , of  $n \times \delta = r$  digits, as follows  
 $\mathbf{h} = [h^1(1), h^1(2), \dots, h^1(n), h^2(1), h^2(2), \dots, h^2(n), \dots, h^\delta(1), h^\delta(2), \dots, h^\delta(n)]$

---

### 2.5 Metric of performance

The metric of performance comparison is the L2 norm. Let,  $\mathbf{h}$  and  $\mathbf{h}'$  are the hash for transmitted and received images, respectively. The L2 norm (or Hash Distance) is given by:

$$\text{Hash Distance } (d) = \sqrt{\sum_{i=1}^r |h(i) - h'(i)|^2} \quad (3)$$

where,  $h(i)$  and  $h'(i)$  show  $i^{th}$  elements of  $\mathbf{h}$  and  $\mathbf{h}'$ , respectively. When the hash distance is less than a threshold  $\tau_1$  ( $d < \tau_1$ ), the image pair is considered to be “similar image pairs”, if  $\tau_1 < d < \tau_2$  then “tampered image pairs”, otherwise “different content image pairs”. FPR (False Positive Rate) and TPR (True Positive Rate) are two other performance comparison metrics, based on hash distances, discussed as follows.

$$FPR = \zeta_1 / \eta_1 \quad (4)$$

$$TPR = \zeta_2 / \eta_2 \quad (5)$$

where  $\zeta_1$  reveals the total number of the different content image pairs considered similar ones, and  $\zeta_2$  reveals the total number of similar content image pairs considered similar ones.  $\eta_1$  and  $\eta_2$  represent visually different and similar content image pairs, respectively. For a better image authentication approach, TPR and FPR should be high and low, respectively. The receiver operating characteristics (ROC) is drawn using TPR in the ordinate and FPR in abscissa, which is used to compare the overall performance of different image authentication methods.

### 3 Proposed image authentication system

The proposed image authentication system is shown in Fig. 3. The process of image authentication is discussed in detail in the following subsections.

#### 3.1 Blind geometric correction approach

Let us have the received image and hash. For image authentication and tampered area localization, the effect of the geometric transformations is eliminated in the received image. In the literature, methods [3, 14, 20, 21, 35] are geometric correction approaches. The methods [3, 20, 21, 35] are affected by tampering as well as the length of a hash is too large. In [14], it needs additional information from the transmitter side. In this paper, we have proposed a blind geometric correction method that does not require any information from the transmitter side, which discussed as follows.

Let a received image is processed through combined geometric correction such as RST, shown in Fig. 4a. First, the rotation angle is estimated as follows.

$$\theta = \arctan(\Delta Y / \Delta X), \quad \theta' = \arctan(\Delta Y' / \Delta X') \quad (6)$$

where,  $\Delta Y = Y_b - Y_r$ ,  $\Delta X = X_r - X_b$ ,  $\Delta Y' = Y_l - Y_t$ , and  $\Delta X' = X_t - X_l$ .  $(X_r, Y_r)$ ,  $(X_l, Y_l)$ ,  $(X_t, Y_t)$ , and  $(X_b, Y_b)$  are indexes of non-zero-pixel values of the rightmost, leftmost, top, and bottom points, respectively. If  $\theta \cong \theta'$  and  $X_t > X_b$ , then the image is considered to be rotated anticlockwise (i.e., angle of rotation is positive,  $\theta_p = \theta'$ ), as shown in Fig. 4a, otherwise clockwise (i.e., angle of rotation is negative,  $\theta_n = \theta'$ ), as shown in Fig. 4b. Then, the rotated image is anti-rotated to either  $\theta_p$  or  $\theta_n$  to restore the image, shown in Fig. 4c. Finally, the region

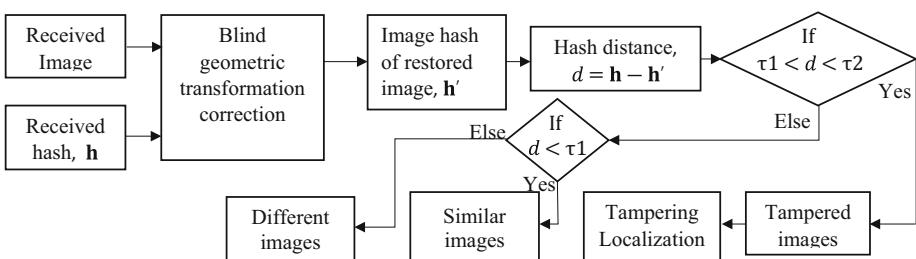


Fig. 3 Proposed image authentication system

of interest is extracted, as shown in Fig. 4d. The proposed geometric correction is limited in the range  $(-45^\circ + 45^\circ)$ . The Pseudocode for the blind geometric transformation correction algorithm is shown in algorithm 2.

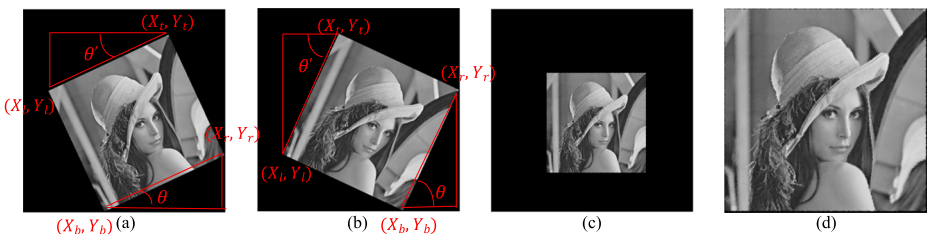
**Algorithm 2:** Pseudocode for the blind geometric transformation correction algorithm by exploiting the inherent characteristics of the geometric transform

```

Input: F (the received image gone through the geometric transformation)
Output: Restored image (corrected image after eliminating geometric transformation effect), F2''
1: Compute the indexes of the rightmost, leftmost and the top and the bottom points, i.e.  $(X_r, Y_r)$ ,  $(X_l, Y_l)$ ,  $(X_t, Y_t)$ , and  $(X_b, Y_b)$ , respectively from F.
2: Compute  $\theta = \arctan(\Delta Y / \Delta X)$ ,  $\theta' = \arctan(\Delta Y' / \Delta X')$  where,  $\Delta Y = Y_b - Y_r$ ,  $\Delta X = X_r - X_b$ ,  $\Delta Y' = Y_l - Y_t$ , and  $\Delta X' = X_t - X_l$ .
3: if  $\theta \cong \theta'$  do
4:   if  $X_t > X_b$  then
5:      $\theta_p = \theta$  // image is rotated in anticlockwise direction
6:   else
7:      $\theta_n = (90 - \theta)$  // image is rotated clockwise direction
8:   end if
9: Rotate the image by either  $-\theta_p$  or  $-\theta_n$  to get the rotation corrected image F2'
10: else
    There is no rotation
11: end if
12: Compute bounding box over F2' as  $(X_{min}, Y_{min})$  and (width, height)
13: Crop F2' where rows go from  $X_{min}$  to  $X_{min} + \text{width}$  and columns go from  $Y_{min}$  to  $Y_{min} + \text{height}$ , resize the cropped image to  $p \times p$ , let it F2''
14: Return, restored image F2''.
    
```

### 3.2 Image authentication

After eliminating the geometric transformation effect in the received image (if any), the image hash,  $\mathbf{h}'$  is generated as discussed in Section 2. Now, estimate the hash distance ( $d$ ) between  $\mathbf{h}$  and  $\mathbf{h}'$ . If  $d$  lies between threshold  $\tau_1$  and  $\tau_2$  (both the thresholds are selected based on an experiment in Section 4), then the received image has been manipulated by fraudulent during the transmission, i.e., tampered version of transmitted one. Else if  $d$  is less than  $\tau_1$ , the received image is considered a similar one, otherwise a different one. From the tampered image, the tampering area has been localized, as discussed in the following subsection.



**Fig. 4** Correction in geometric transformation (a) Composite RST, rotated in anticlockwise direction,  $[25^\circ, 0.5, [100 100]$  (b) Composite RST, rotated in clockwise direction,  $[-25^\circ, 0.5, [100 100]$  (c) Anti-rotated image (d) Restored image

### 3.3 Tampering localization

The received image has been passed through blind geometric correction. The image hash  $\mathbf{h}'$  has also been obtained, as shown in Fig. 5. Now, the image maps are generated for both the received hash,  $\mathbf{h}$ , and generated hash,  $\mathbf{h}'$ , as follows.

First, dis-concatenate  $\mathbf{h}'$  into  $k$ -th ( $1 \leq k \leq \delta$ ) number of arrays, i.e.,  $z^k(i)$  where  $i = 1, 2, \dots, n$ , each consisting  $n$  low-frequency DCT coefficients. Next, padding 49 zeroes on  $z^k(i)$  yields  $c^k(j)$  where ( $j = 1, 2, \dots, n + 49$ ). Apply inverse zigzag coding (shown in Appendices 2) on  $c_k$ , which generates  $m \times m$  size  $k$ -th  $B'_k(u, v)$  sub-matrices. This process extracts the original sequence of  $n$  high energy frequency coefficients for each  $k$ -th sub-matrices. After that, find the inverse discrete cosine transform (IDCT) of  $k$ -th sub-matrices  $B'_k(u, v)$  as follows

$$f'_k(x, y) = \mathbf{T}'(u, v) \times \mathbf{B}'_k(u, v) \times \mathbf{T}(u, v); 1 \leq k \leq \delta \tag{7}$$

where  $\mathbf{T}$  is DCT matrix as discussed in Appendix 1 and  $\mathbf{T}'$  is the transpose of  $\mathbf{T}$ . Rearrange  $k$ -th sub-matrices  $f'_k(x, y)$  from left to right, top to bottom non-overlapping pixel blocks yields an image map  $f'(x, y)$  of size  $p/m \times p/m$ . Similarly, find an image map  $f''(x, y)$  from the received image hash,  $\mathbf{h}$ . The two image maps are subtracted and normalized. Next, it has been converted into a binary image, multiplied with the restored received image to detect the tampered regions. The details in the implementation form are shown in algorithm 3.

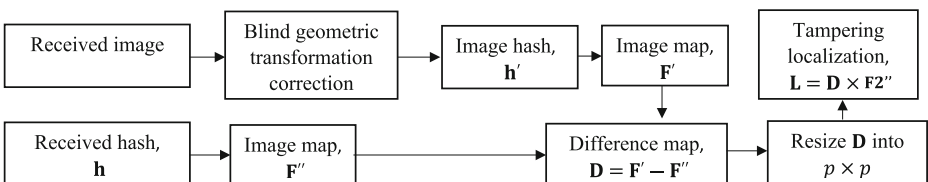
**Algorithm 3:** Pseudocode for tampering localization

---

**Input:** Restored image,  $\mathbf{F2}''$ , and received hash,  $\mathbf{h}$ .  
**Output:** Image showing tampered areas,  $\mathbf{L}$

- 1: Generate an image hash,  $\mathbf{h}'$ , for the restored image,  $\mathbf{F2}''$ .
- 2: Dis-concatenate hash  $\mathbf{h}'$ ,  $\mathbf{h}' = [z^1(1), z^1(2), \dots, z^1(n), z^2(1), z^2(2), \dots, z^2(n), \dots, z^\delta(1), z^\delta(2), \dots, z^\delta(n)]$
- 3: Padding 49 zeroes on  $z^k(i)$  yields  $c^k(j) = [z^k(i), 0_1, 0_2, \dots, 0_{49}]$ .
- 4: Apply inverse zigzag coding on  $c^k(j)$  to generate  $m \times m$  size  $k$ -th  $B'_k(u, v)$  sub-matrices.
- 5: Find IDCT of  $B'_k(u, v)$  as  $f'_k(x, y) = \mathbf{T}'(u, v) \times \mathbf{B}'_k(u, v) \times \mathbf{T}(u, v)$  yields  $f'(x, y) = \begin{bmatrix} f'_1(x, y) & f'_2(x, y) \\ f'_3(x, y) & f'_\delta(x, y) \end{bmatrix}$  or  $\mathbf{F}'$  is called an image map.
- 6: Generate an image map,  $\mathbf{F}''$ , from the received hash,  $\mathbf{h}$ , using a similar approach as from step 2 to step 5.
- 7: Find the difference  $\mathbf{D} = \mathbf{F}' - \mathbf{F}''$  and normalized.
- 8: Resize  $\mathbf{D}$  into  $p \times p$  and converted into a binary image.
- 9: Tampering Localization,  $\mathbf{L} = \mathbf{D} \times \mathbf{F2}''$

---



**Fig. 5** Flow chart of the proposed tampering localization method



## 4 Analysis of experimental results

In this section, the proposed method has been evaluated using a large number of image pairs, and the selected optimal value of the model parameters are as follows:  $p \times p = 128 \times 128$ ,  $m \times m = 8 \times 8$ ,  $\delta = 4$ ,  $n = 15$ ,  $r = 60$ ,  $\tau_1 = 0.88$ , and  $\tau_1 = 27$ . The proposed model has been analyzed in three categories: the performance of blind geometric transformation correction, the performance of hashing technique for robustness and discriminative capability, and detection of tampered images and their localization, shown in Sections 4.1, 4.2, and 4.3, respectively.

### 4.1 Performance of blind geometric transformation correction

To examine the efficacy of the proposed blind geometric transformation correction method, we have selected 200 similar images and 400 tampered images from Ground Truth Database [10] and CASIA V2.0 database [5], respectively. These selected images are gone through the geometric transformation (i.e., composite RST) with the parameters shown in Table 1. The geometric transformation parameters, i.e., rotation angles for clockwise and anticlockwise, are estimated using the proposed method. The estimation error (mean and standard deviation of the error), is shown in Tables 2 and 3. It can be observed that for both similar and tampered images, the mean and standard deviation of estimation error is too low. It can also be seen that the mean and standard deviation estimation error is invariant to rotation angles. In the case of tampered images, the estimation error is highly affected in the existing methods [3, 20, 21, 35]. The geometric distortions are eliminated due to the very low estimation error, as discussed in Section 3.1. However, the proposed blind geometric transformation correction approach is limited for the rotation angle from  $-45$  to  $+45$  degrees.

### 4.2 Performance of hashing for robustness and discriminative capability

In this section, the proposed model is used to segregate the received image as “perceptually similar image pairs”, “tampered image pairs”, or “different image pairs”. The experiment has been carried out on 3,948 “perceptually similar image pairs”, generated as shown in Table 1. Here, 42 source

**Table 1** Content preserving operations specifications

Software	Manipulations	Parameters	Parameter Values
StirMark	JPEG contraction	Quality factor	30,40,50,60,70,80, 90, 100
StirMark	Scaling	Ratio	0.5,0.75,0.9,1.1,1.5,2.0
StirMark	Rotation	Rotation angle in degree	$\pm 5, \pm 15, \pm 30, \pm 45, \pm 90$
MATLAB	Gamma rectification	$\Gamma$	0.75,0.9,1.1,1.5
MATLAB	3×3 Gaussian low pass filtering	Standard deviation	0.3,0.4,0.5,0.6, 0.7 ,0.8, 0.9 ,1
MATLAB	Salt and pepper noise	Density	0.001 0.01 (step size 0.001)
MATLAB	Speckle noise	Variance	0.001 0.01 (step size 0.001)
MATLAB	Translation	[x-pixels, y-pixels]	10 to 100 (Both axis)
MATLAB	Composite RST	Degree of rotation, Ratio, and [x-pixels, y-pixels]	# 1-(5°0.5,[10,10]),#2 – (10°,0.75,[20,20]), # 3-(15°,0.9,[30,30]),#4 – (30°,1.1,[40,40]), # 5-(45°,1.5,[50,50]),#6 – (87°,2.0,[60,60]). Made 36 combinations from above six parameters

**Table 2** Performance of the proposed blind geometric transformation correction, the error of the rotation angle estimation in case of clockwise rotation

Rotation angle (degree)		-4	-9	-14	-29	-35	-44
Similar and tampered images	Mean	0.1418	0.0903	0.0664	0.0526	0.0704	0.0797
	Standard deviation	0.0498	0.0263	0.0372	0.0399	0.0234	0.0507

images are selected from the USC-SIPI database [32], such as 37 and 5 from the “Aerial” and “Miscellaneous” categories, respectively, the size varies from 512 x 512 to 2250 x 2250. Also, 19,900 “different image pairs” are generated from different combinations of 200 source images. Where, the source images are as follows: 75 from Ground Truth Database [10] of size 756 x 504, 75 using Nikon D3200, of size from 3008 x 2000 to 4512 x 3000; and 50 from the Internet, of size from 256 x 256 to 1024 x 768. 400 “tampered image pairs” are selected from the CASIA V2.0 database [5], where size varies from 240 x 160 to 900 x 600.

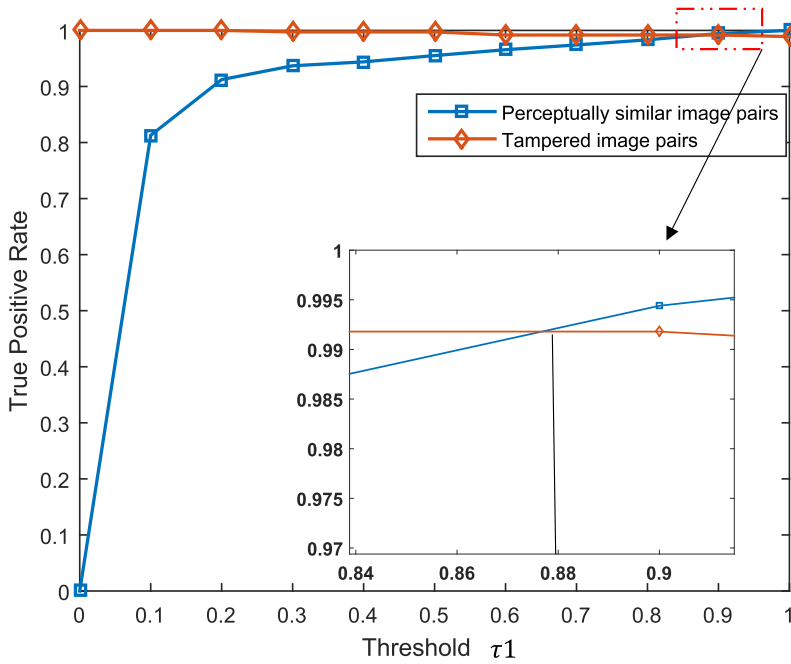
The hash distances are calculated for the above three categories of image pairs and estimated the true positive rate via varying the threshold, drawn in Fig. 6. Figure 6a shows the true positive rate for “perceptually similar image pairs” and “tampered image pairs”. Here, the overlapping curve shown in the red rectangle in the right upper corner is expanded and presented in the right bottom corner to view the details. It can be observed that an optimal threshold  $\tau_1 = 0.88$  may be selected to segregate between “perceptually similar image pairs” and “tampered image pairs”. Similarly, Fig. 6b shows the true positive rate for “tampered image pairs” and “different image pairs”. The overlapping in the center portion is enlarged and presented at the bottom to view details. It can be seen that  $\tau_2 = 27$  may be selected to differentiate between “tampered image pairs” and “different image pairs”. Both thresholds are determined based on a trade-off between robustness and discriminations. It can be observed from Fig. 6 that the proposed method has better robustness (TPR is high at the optimal threshold), as well as most of the “tampered image pairs” and “different image pairs”, are truly detected.

### 4.3 Detection of tampered images and its localization

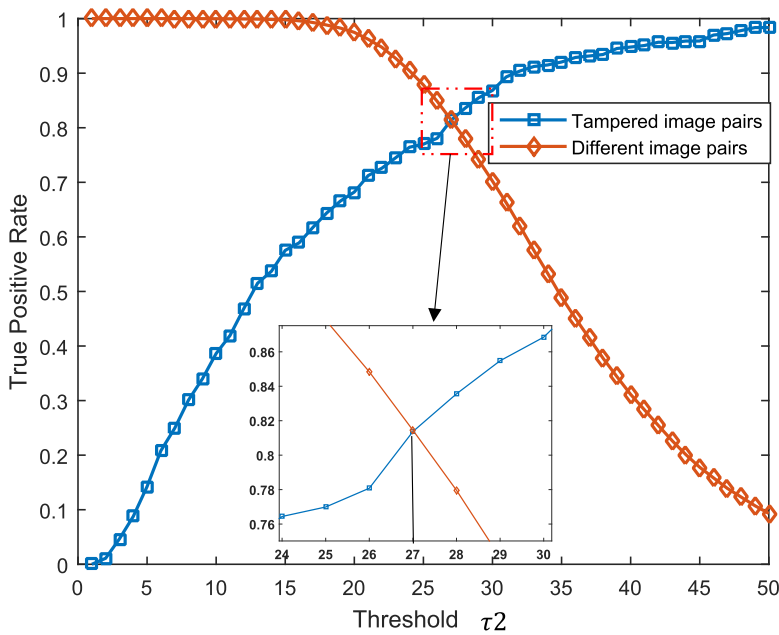
This section discussed the experiment on 400 tampered image pairs taken from the CASIA V2.0 database [5], which varies from 240 x 160 to 900 x 600. The hash distances of tampered image pairs are shown in Fig. 7, where the red and green line shows  $\tau_1 = 0.88$  and  $\tau_2 = 27$ , respectively. It can be seen that most of the tampered images are detected by the proposed method. In case the tampering area is large, the hash distances are larger than  $\tau_2 = 27$ , shown above the green line. The large area tampering may be considered different image pairs. For the detected tampered images, the tampering area can be localized using the proposed method, as shown in Table 4. Here, the first and second row shows original and tampered pairs,

**Table 3** Performance of the proposed blind geometric transformation correction, the error of the rotation angle estimation in case of anticlockwise rotation

Rotation angle (degree)		4	9	14	29	35	44
Similar and tampered images	Mean	0.0568	0.1151	0.0128	0.0735	0.0447	0.0330
	Standard deviation	0.0133	0.0555	0.0007	0.0586	0.0265	0.0265

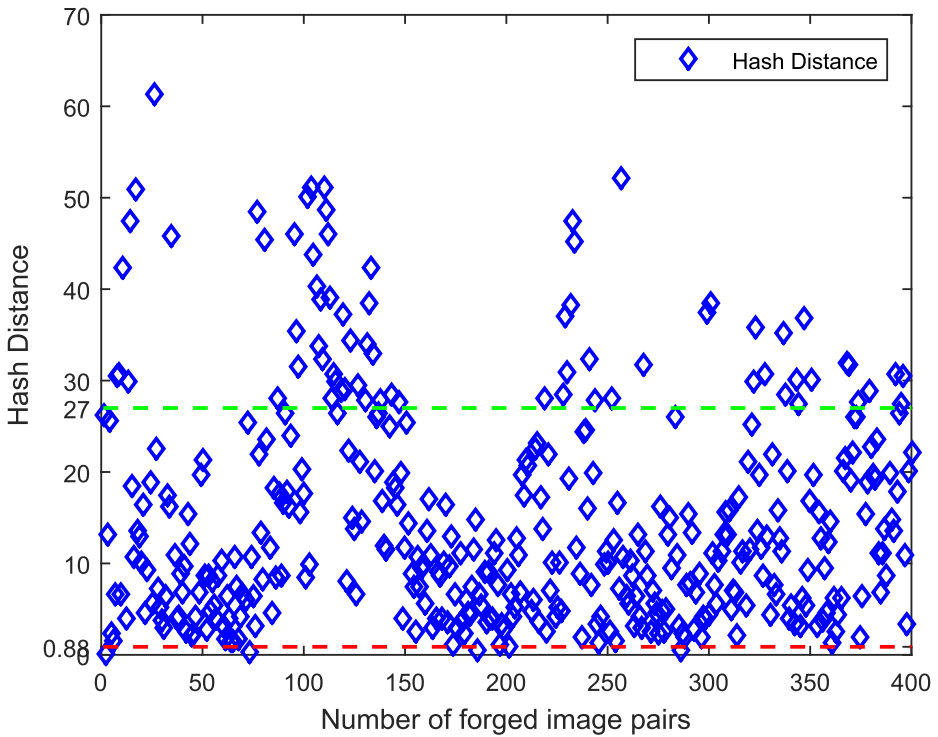


(a)



(b)

**Fig. 6** True positive rate (performance of image authentication) with the varying threshold for (a) perceptually similar and tampered image pairs and (b) tampered and different image pairs



**Fig. 7** Hash distance distribution of tampered image pairs, where red and green line shows  $\tau_1 = 0.88$  and  $\tau_2 = 27$ , respectively

respectively. The tampering is localized in the third row. It can be observed that the tampered areas are indeed detected in the second and third columns, respectively. But, in the fourth column, some small regions which are not tampered, along with tampered ones, are detected as tampered areas, which is a limitation of the proposed method. Whereas, if some portions of the

**Table 4** Localization of tampered regions



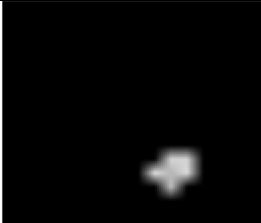


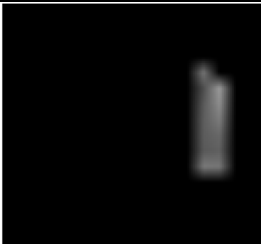
	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
Original						
Tampered						
Localization						

tampered regions are similar to the original image, that portion has not been detected, as shown in the fifth and sixth, respectively. However, most of the tampered objects are detected using the proposed method. Due to the page limitation, few samples are presented. Besides, the proposed method can detect tampering, even if the composite RST and tampering occur simultaneously, as shown in Table 5. Here, the first column shows the original image, and the second one reflects tampering and the composite RST transformation. In the third column, the tampered regions (i.e., some objects) are identified using the proposed method.

### 5 Performance comparison with state-of-the-art methods

The proposed method is compared with state-of-the-art techniques such as Radon Transform, and Discrete Fourier Transform (RT-DFT) based hashing [15], Binary multi-view based hashing [9], SIFT based hashing [19], Zernike Moments (ZM) based hashing [38], and Ring invariant vector distance (Ring-IVD) based hashing [31]. Tables 6, 7, and 8 demonstrate the benefits and drawbacks of comparative approaches. The values of parameters required for the implementation of the compared methods have been reserved from corresponding papers. But, the input images are resized to  $128 \times 128$ , and the threshold values are selected based on our image database, discussed in Section 4. The Euclidean distance that produces TPR and FPR has only been chosen as a performance metric for fair comparison among all the methods. A ROC curve consists of several coordinate points (TPR, FPR), with the x-axis being FPR and the y-axis being TPR. If compared algorithms have the same FPR in the ROC curve, high TPR methods are better than low TPRs. Likewise, if two algorithms with the same TPR, the low FPR approach outperforms the high FPR method. All compared methods are evaluated with the same database, discussed in Section 4.

**Table 5** Localization of tampered regions, even if tampering and composite RST occur simultaneously

Original	Tampered with RST	Localization of tampered regions
		
		

**Table 6** Comparison of robustness against digital operations

Algorithms	TPR at an optimal threshold					
	Lei et al. [15]	Du et al. [9]	Lv et al. [19]	Zhao et al.[38]	Tang et al. [31]	Proposed method
Operations						
3 × 3 Gaussian low-pass filtering	1	1	0.9821	1	1	1
Salt & pepper noise	1	0.95	0.7286	1	1	1
Speckle noise	1	0.91	0.6738	1	1	1
JPEG compression	0.9524	1	0.9048	1	1	0.9524
Rotation	1	0.56	0.0119	0.4130	0.9955	1
Scaling	0.8770	1	0.7143	1	1	0.8770
Translation	1	0.4309	0.0024	0.1714	0.1024	1
RST	0.8929	0.3272	0.0040	0.0317	0.0040	0.8929

The proposed method has been compared with existing techniques via four phases: Firstly, the overall robustness and discriminative capability (considering only different image pairs). Secondly, individual robustness against digital operations. Next, sensitiveness towards content-changing operations. Finally, some more performance parameters such as length of hash, time of computation in MATLAB, etc.

The overall robustness and discriminative capability performance are represented using the receiver operating characteristic (ROC) curve, as shown in Fig. 8. Here, the curve near the upper left portion (defined within the red rectangle) has been zoomed and kept in the right lower part to view details. Each compared method estimates TPR and FPR via varying thresholds and generates the ROC curve, as shown in Fig. 8. TPR and FPR indicate the performance of robustness and discriminative capability, respectively. A ROC curve followed towards the upper left corner is a better technique. It can be observed that the proposed image authentication system ROC curve is followed in the leftmost corner, hence better than the compared ones, for robustness and discriminative capability. The methods [9, 15, 31] successively followed the proposed method.

The performances of compared methods against digital operations are shown in Table 6. A better approach should have high TPR (i.e., close to one) and low FPR (i.e., close to zero). It can be observed that the performance of the proposed method for robustness is better than the existing methods, particularly in the case of geometric transformations such as rotation, translation, and composite RST, etc. The techniques [31, 38] are robust against many digital manipulations, but sensitive to translation and composite RST. Whereas the method [15] robustness is better in translation, but susceptible to scaling and compression. The robustness of the method [9] against geometric operations followed [15].

The sensitivity of the compared methods towards content-changing operations at an optimal threshold is shown in Table 7. It can be observed that the proposed method sensitiveness is

**Table 7** Performance comparison of image hashing methods for content-changing operations

Algorithms	FPR at an optimal threshold					
	Lei et al. [15]	Du et al. [9]	Lv et al. [19]	Zhao et al.[38]	Tang et al. [31]	Proposed method
Operations						
Tampering	0.1625	0.1023	0.4042	0.1062	0.1800	0.0843

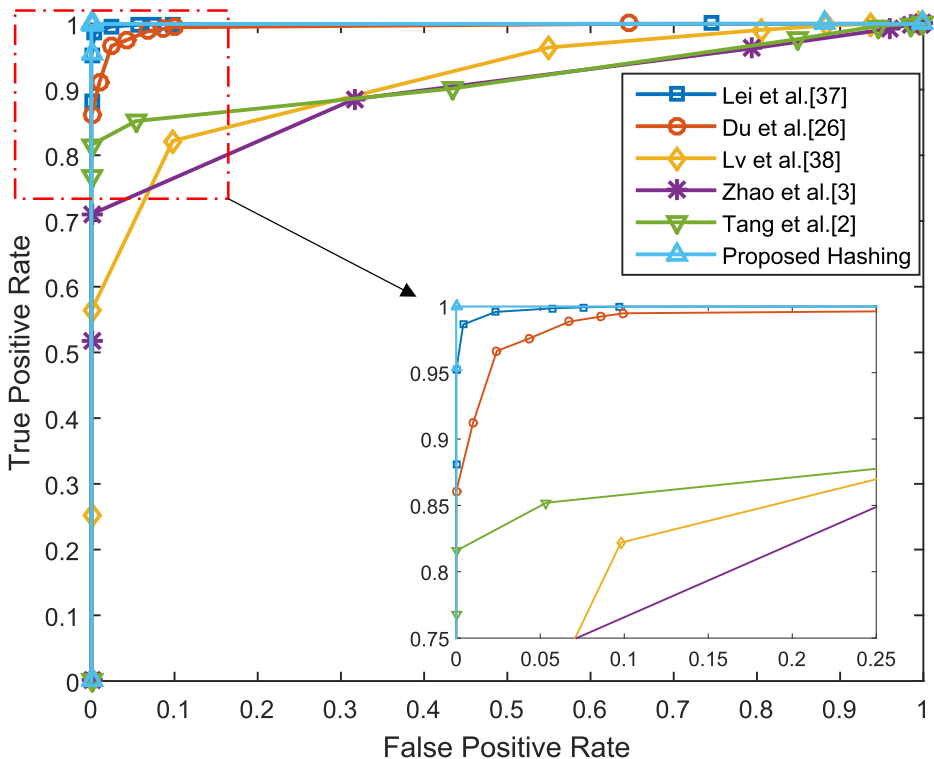
**Table 8** Performance comparison for discrimination with different existing techniques

Performances of algorithms	Lei et al. [15]	Du et al. [9]	LV et al. [19]	Zhao et al. [38]	Tang et al. [31]	Proposed method
1. Forgery detection in the case of copy- move/splicing	Yes	No	Yes	Yes	No	Yes
2. Robust against arbitrary rotation	No	No	No	No	Yes*	Yes
3. Robust against composite RST	No	No	No	No	No	Yes
4. Tampered area localization along with composite RST	No	No	No	No	No	Yes
5. The length of a hash	15 digits	63 digits	20 digits	70 digits	40 digits	60 digits
6. TPR at an optimal threshold	0.9871	0.9623	0.8023	0.7234	0.8236	0.9901
7. FPR at an optimal threshold	0.0054	0.0076	0.0453	0.0617	0.0015	$2.512 \times 10^{-4}$
8. Average time (sec)	0.7016	0.6043	0.6651	0.5642	0.0749	0.0178

Yes\* means the method is robust, but there is information loss

higher (false acceptance is lower, i.e.,  $FPR = 0.0843$ ) than that of compared methods. The methods [9, 15, 31, 38] successively followed the proposed method. It has been experimentally observed that in the case tampered color is similar to the original one; then the proposed method may not detect that part. Hence, there is a small misclassification. However, FPR is the lowest among the compared methods due to better image map construction.

The performance compared with some more parameters is shown in Table 8. It can be observed that the proposed method  $TPR = 0.9901$ , which is higher than the compared ones.



**Fig. 8** Performance comparison of the proposed technique with some existing techniques

Whereas the  $FPR = 2.512 \times 10^{-4}$  is the lowest among the compared methods. Hence, the proposed method has a better trade-off between robustness and discrimination. The robustness against an arbitrary rotation and composite RST is a major finding, severely limiting the state-of-the-art methods. The robustness against these geometric operations has been achieved due to the proposed blind geometric correction approach. The method [31] is robust to an arbitrary rotation, but there is much information loss. All compared methods are implemented using a desktop computer with an Intel i7 processor of 8 GB RAM having a windows 8 operating system using MATLAB 2015a. The image hash has been generated for 200 images and considers the average time. It can be seen that computational cost is the lowest among compared methods. But, the hash length is slightly larger than some of the techniques. However, the proposed method can locate the tampering region, even if tampering and composite RST occur simultaneously, which is the main focus of the proposed method.

## 6 Conclusion and future works

In this work, a blind geometric transformation correction has been proposed. An image hash has been generated based on LWT and DCT. The proposed image hashing technique is applied for content authentication and tampered area localization. An image map has been generated from the short hash. Based on the differences in image maps, the tampered areas have been localized. The main focus of this work is to keep the hash length short, maintain robustness against digital operations, and localize the area of tampering, even if the tampering and composite RST occur simultaneously. The experiment has been carried out on an extensive database, and the results demonstrated the effectiveness of the proposed method against digital operations. Besides, good discriminative capability and localize tampered regions, even if tampering and composite RST occur simultaneously. ROC curve shows that the proposed trade-off between robustness and discriminative capacity is better than some state-of-the-art techniques.

In future work, the accuracy of tampering area localization may be improved. The proposed method may be extended for video hashing.

## Appendix 1 DCT matrix

$$\mathbf{T}(u, v) (\text{sized } m \times m) = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

where  $\mathbf{T}(u, v)$  has been obtained as  $\mathbf{T}(u, v) = \begin{cases} \sqrt{1/m} & ; u = 0 \text{ and } 0 \leq v \leq m - 1 \\ \sqrt{2/m} \cos \left[ \frac{(2v+1)\pi u}{2m} \right] & ; 1 \leq u \leq m - 1 \text{ and } 0 \leq v \leq m - 1 \end{cases}$



## Appendix 2 Zigzag ordering and invers-ordering

The zigzag order is obtained as per arrow direction shown in Fig. 9, given below.

```
Zigzag order = [1 9 2 3 10 17 25 18 11 4 5 12 19 26 33 ...
                41 34 27 20 13 6 7 14 21 28 35 42 49 57 50 ...
                43 36 29 22 15 8 16 23 30 37 44 51 58 59 52 ...
                45 38 31 24 32 39 46 53 60 61 54 47 40 48 55 ...
                62 63 56 64];
```

```
Zigzag inverse order = [1 3 4 10 11 21 22 36 2 5 9 12 20 23 35 ...
                       37 6 8 13 19 24 34 38 49 7 14 18 25 33 39 ...
                       48 50 15 17 26 32 40 47 51 58 16 27 31 41 46 ...
                       52 57 59 28 30 42 45 53 56 60 63 29 43 44 54 ...
                       55 61 62 64];
```

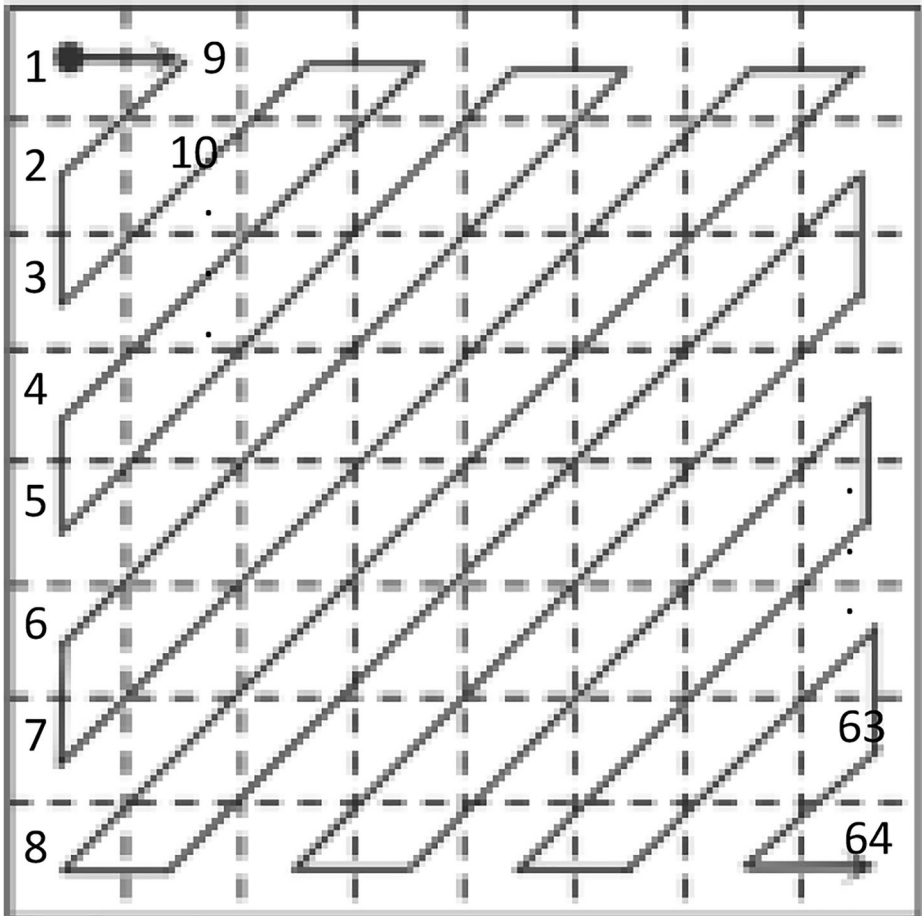


Fig. 9 Diagram of zigzag scanning

**Acknowledgements** The author would like to thank all the Ph.D. scholars of Speech and Image Processing Laboratory and National Institute of Technology Silchar, India, for offering help and vital facilities for doing this work.

## Declarations

**Conflicts of interest/Competing interest** There is no conflicts of interest.

## References

1. Ahmed N, Natarajan T, Rao KR (1974) Discrete cosine transform. *IEEE Trans Comput* 100(1):90–93
2. Ahmed F, Siyal MY, Abbas (2010) A secure and robust hash based scheme for image authentication. *Sig Process* 90(5):1456–1470
3. Battiato S, Farinella GM, Messina E, Puglisi G (2012) Robust image alignment for tampering detection. *IEEE Trans Inf Forensics Secur* 7(4):1105–1117
4. Brian K, Grauman K (2009) Kernelized locality-sensitive hashing for scalable image search. In: *IEEE International Conference on Computer Vision*, pp 2130–2137
5. CASIA Tampered image detection evaluation database [Online]. Available: <http://forensics.idealtest.org/>. Accessed 2010
6. Connolly C, Fliess T (1997) A study of efficiency and accuracy in the transformation from RGB to CIE Lab color space. *IEEE Trans Image Process* 6(7):1046–1048
7. Daubechies I, Sweldens W (1998) Factoring wavelet transforms into lifting steps. *J Fourier Anal Appl* 4(3): 247–269
8. Davarzani R, Mozaffari S, Yaghmaie K (2016) Perceptual image hashing using center-symmetric local binary patterns. *Multimed Tools Appl* 75(8):4639–4667
9. Du L, Chen Z, Ho AT (2020) Binary multi-view perceptual hashing for image authentication. *Multimed Tools Appl* 19:1–23
10. Ground Truth Database. <http://www.cs.washington.edu/research/imagedatabase/groundtruth/>. Accessed 8 May 2008
11. Hosny KM, Khedr YM, Khedr WI, Mohamed ER (2018) Robust color image hashing using quaternion polar complex exponential transform for image authentication. *Circuits Syst Signal Process* 37(12):5441–5462
12. Karsh RK, Laskar RH (2017) Robust image hashing through DWTSVD and spectral residual method. *EURASIP J Image Video Process* 2017(1):31
13. Karsh RK, Laskar RH, Richhariya BB (2016) Robust image hashing using ring partition-PGNMF and local features. *Springerplus* 5(1):1995
14. Karsh RK, Saikia A, Laskar RH (2018) Image authentication based on robust image hashing with geometric correction. *Multimed Tools Appl* 77(19):25409–25429
15. Lei Y, Wang Y, Huang J (2011) Robust image hash in radon transform domain for authentication. *Signal Process Image Commun* 26:280–288
16. Leng L, Zhang J (2013) Palmhash code vs. palmphasor code. *Neurocomputing* 108:1–2
17. Leng L, Li M, Teoh AB (2013) Conjugate 2DPalmHash code for secure palm-print-vein verification. In: *IEEE International Congress on Image and Signal Processing*, pp 1705–1710
18. Leng L, Zhang J, Khan MK, Chen X, Alghathbar K (2010) Dynamic weighted discrimination power analysis: a novel approach for face and palmprint recognition in DCT domain. *Int J Phys Sci* 5(17):2543–2554
19. Lv X, Wang ZV (2012) Perceptual image hashing based on shape contexts and local feature points. *IEEE Trans Inf Forensics Secur* 7(3):1081–1093
20. Lu W, Wu M (2010) Multimedia forensic hash based on visual words. In: *IEEE International Conference on Image Processing*, pp 989–992
21. Lu W, Varna AL, Wu M (2010) Forensic hash for multimedia information. In: *Proc SPIE Media Forensics and Security*, pp 75410Y
22. Mishra M, Adhikary MC (2013) Digital image tamper detection techniques: A comprehensive study. *Int J Comput Sci Bus Inf* 2(1):1–12
23. Ouyang J, Liu Y, Shu H (2017) Robust hashing for image authentication using SIFT feature and quaternion Zernike moments. *Multimed Tools Appl* 76(2):2609–2626
24. Paul M, Karsh RK, Talukdar FA (2019) Image hashing based on shape context and Speeded Up Robust Features (SURF). In: *IEEE International Conference on Automation, Computational and Technology Management*, pp 464–468

25. Pun CM, Yan CP, Yuan (2016) Image alignment-based multi region matching for object-level tampering detection. *IEEE Trans Inf Forensics Secur* 12(2):377–391
26. Reddy S, Arya U, Karsh U, Laskar RK (2020) Hash code based image authentication using rotation invariant local phase quantization. In: Elçi A, Sa P, Modi C, Olague G, Sahoo M, Bakshi S (eds) *Smart computing paradigms: New progresses and challenges. Advances in intelligent systems and computing*, vol 766. Springer, Singapore
27. Roy S, Sun Q (2007) Robust hash for detecting and localizing image tampering. In: *IEEE International Conference on Image Processing*, pp VI-117
28. Saikia A, Karsh RK, Lashkar RH (2017) Image authentication under geometric attacks via concentric square partition based image hashing. In: *TENCON 2017 IEEE Region 10 Conference*, pp 2214–2219
29. Sajjad M, Haq IU, Lloret J, Ding W, Muhammad K (2019) Robust image hashing based efficient authentication for smart industrial environment. *IEEE Trans Industr Inf* 15(12):6541–6550
30. Su Z, Yao L, Mei J, Zhou L, Li W (2020) Learning to hash for personalized image authentication. *IEEE Trans Circuits Syst Video Technol*. <https://doi.org/10.1109/TCSVT.2020.3002146>
31. Tang Z, Zhang X, Li X, Zhang S (2016) Robust image hashing with ring partition and invariant vector distance. *IEEE Trans Inf Forensics Secur* 11(1):200–214
32. USC-SIPI Image database. <http://sipi.usc.edu/database/>. Accessed Feb 2007
33. Venkatesan R, Koon SM, Jakubowski MH, Moulin P (2000) Robust image hashing. In: *IEEE International Conference on Image Processing*, pp 664–666
34. Yan CP, Pun CM (2017) Multi-scale difference map fusion for tamper localization using binary ranking hashing. *IEEE Trans Inf Forensics Secur* 12(9):2144–2158
35. Yan CP, Pun CM, Yuan XC (2016) Quaternion-based image hashing for adaptive tampering localization. *IEEE Trans Inf Forensics Secur* 11(12):2664–2677
36. Yan CP, Pun CM, Yuan XC (2016) Multi-scale image hashing using adaptive local feature extraction for robust tampering detection. *Sig Process* 121:1–16
37. Yunchao G, Lazebnik S, Gordo A, Perronnin F (2013) Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans Pattern Anal Mach Intell* 35(12):2916–2929
38. Zhao Y, Wang S, Zhang X, Yao H (2013) Robust hashing for image authentication using Zernike moments and local features. *IEEE Trans Inf Forensics Secur* 8(1):55–63

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Ram Kumar Karsh** received his B. Tech from Government Engineering College, Bilaspur, M.Tech from NIT Patna, and PhD from NIT Silchar in 2009, 2012, and 2018, respectively. He has worked as Junior Teaching-cum-Research Fellow at Birla Institute of Technology, Mesra. He is currently working as an assistant professor in NIT Silchar. His research interest includes robust image hashing and image authentication.