This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

Efficient Hashing Method Using 2D-2D PCA for Image Copy Detection

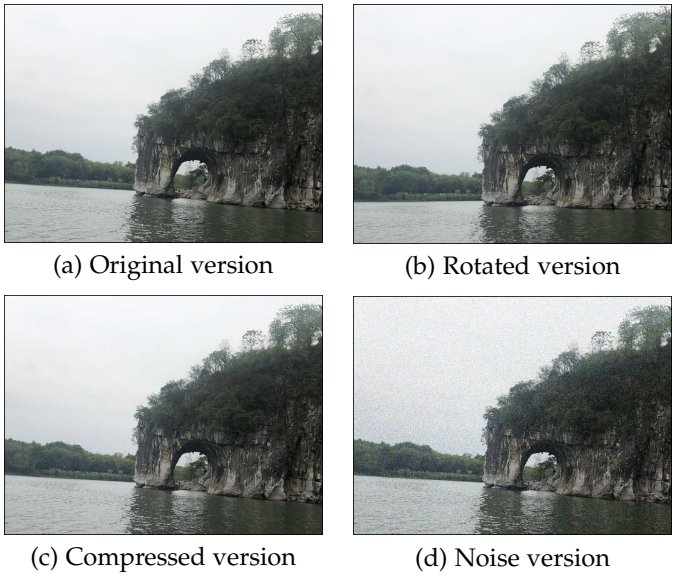
Xiaoping Liang, Zhenjun Tang, Ziqing Huang, Xianquan Zhang, and Shichao Zhang, *Senior Member, IEEE*

Abstract—Image copy detection is an important technology of copyright protection. This paper proposes an efficient hashing method for image copy detection using 2D-2D (two-directional two-dimensional) PCA (Principal Component Analysis). The key is the discovery of the translation invariance of 2D-2D PCA. With the property of translation invariance, a novel model of extracting rotation-invariant low-dimensional features is designed by combining PCT (Polar Coordinate Transformation) and 2D-2D PCA. The PCT can convert an input rotated image to a translation matrix. Since the 2D-2D PCA is invariant to translation, the low-dimensional features learned from the translation matrix are rotation-invariant. Moreover, vector distances of low-dimensional features are stable to common digital operations and thus hash construction with the vector distances is of robustness and compactness. Three open image datasets are exploited to conduct various experiments for validating efficiencies of the proposed method. The results demonstrate that the proposed method is much better than some representative hashing methods in the performances of classification and copy detection.

Index Terms—Image hashing, 2D-2D PCA, translation invariance, rotation-invariant feature, image copy detection

1 Introduction

With the wide applications of multimedia technology [1], [2], many images are transmitted and shared via social networks, e.g., Facebook, Flicker, YouTube and Snapchat. Some people would like to download images, process them and republish the processed images by social networks. As a consequence, many copies of an image may be stored in cyberspace. Therefore, it is a challenge to detect image copies from massive images. Fig. 1 illustrates some image copies of an image, where (a) is an image with an imperfect horizontal plane, (b) is a corrected version after rotation, (c) is a compressed version after JPEG compression with factor equaling 50, and (d) is a noise version attacked by AWGN (Additive White Gaussian Noise) with variance equaling 0.01. Given a query image, e.g., Fig. 1 (a), efficient techniques for copy detection are expected to successfully find its image copies, such as Fig. 1 (b), (c) and (d). In fact, image copy detection [3], [4] has become an important technology of copyright protection for finding illegal copies with the aid of watermarking technology. This paper investigates a new and efficient hashing method for image copy detection.



(a) Original version (b) Rotated version

(c) Compressed version (d) Noise version

Fig. 1: Some image copies of an original image

Image hashing [5], [6], [7], [8] can compress an input image of any size to a short sequence known as the hash. It is a useful technology of image representation for improving efficiencies of data processing of massive images. In practice, image hashing has been applied to numerous fields [9], [10], [11], such as copy detection, screen quality assessment, tampering detection and social hot event detection. Generally, image hashing needs to meet two essential performance indicators [12], [13], [14], namely, robustness and discrimination. Suppose that $\mathbf{I}$ represents an image, $\mathbf{I}_{\mathrm{s}}$ represents a similar image of $\mathbf{I}$, and $\mathbf{I}_{\mathrm{d}}$ represents a different image from $\mathbf{I}$. Let $H(\cdot)$ be the hash function for a digital image, $P_{\mathrm{r}}(\cdot)$ represent the probability of occurrence of an event, and $D(\cdot)$ represent the distance function for calculating the similarity of two hashes. Thus, robustness and discrimination can be formally defined as follows.

(1) *Robustness.* Robustness means that, for visually similar input images, hashing method ought to encode them into similar or the same hash sequences. Specifically, the probability that the hashes of $\mathbf{I}$ and $\mathbf{I}_{\mathrm{s}}$ have the same or similar values should be close to 1. The formula is below.

$$P_{\mathrm{r}}\left(D\left(H(\mathbf{I}), H\left(\mathbf{I}_{\mathrm{s}}\right)\right) < T\right) \geq 1-\varepsilon \tag{1}$$

- *X. Liang, Z. Tang, X. Zhang and S. Zhang are with the Guangxi Key Lab of Multi-Source Information Mining & Security, Guangxi Normal University, Guilin 541004, China. (Corresponding authors: Zhenjun Tang, Shichao Zhang) E-mail: tangzj230@163.com, zhangsc@gxnu.edu.cn.*
- *Z. Huang is with School of Artificial Intelligence, Henan University, Kaifeng 475004, China.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

where T represents a given threshold, and ε is a very small positive constant number close to zero.

(2) *Discrimination.* Discrimination is also called anti-collision [11], [13]. It means that, for images with different contents, hashing method ought to compress them into different hash sequences. Specifically, the probability that the hashes of $\mathbf{I}$ and $\mathbf{I}_{\mathrm{d}}$ have different values should be close to 1. Its formula is below.

$$P_{\mathrm{r}}\left(D\left(H(\mathbf{I}), H\left(\mathbf{I}_{\mathrm{d}}\right)\right) \geq T\right) \geq 1-\varepsilon \tag{2}$$

Generally, there is a mutual constraint between these two indicators. An improvement in one indicator results in a reduction in the other. Currently, most hashing methods' classifications between robustness and discrimination are not satisfied yet, and thus they are inefficient in application to copy detection. To address this issue, we propose an efficient image hashing method using 2D-2D (two-directional two-dimensional) PCA (Principal Component Analysis) for copy detection. Compared with the reported hashing methods, the prime contributions are presented here.

(1) A new property of 2D-2D PCA called translation invariance is discovered. It is found that if all training images are shifted along the column direction with the same size, the projection feature matrix of 2D-2D PCA is invariant. In other words, the obtained feature matrix after column shifting is the same with that before column shifting. Theoretical proof of the translation invariance of 2D-2D PCA is given. As 2D-2D PCA is a popular technique of data dimensionality reduction, the property of translation invariance is helpful to other image applications.

(2) A novel model of extracting rotation-invariant low-dimensional features is designed by combining the PCT (Polar Coordinate Transformation) and the 2D-2D PCA. The PCT can convert an input rotated image to a translation matrix. Since the 2D-2D PCA is invariant to translation, it can extract rotation-invariant low-dimensional features from the translation matrix. As far as known, the 2D-2D PCA is first used in image hashing and will be helpful to hashing research.

(3) Low-dimensional features are viewed as vectors and their vector distances are taken to construct hash. As common digital operations slightly disturb low-dimensional features in the 2D-2D PCA domain, their vector distances are stable to common operations. Therefore, hash construction with vector distance can guarantee compactness and robustness.

(4) Performances of the proposed hashing method are tested using three open image datasets. Comparisons show that the classification performance of the proposed hashing method outperforms those of some well-known hashing methods. Copy detection results illustrate that the proposed hashing method is much better than some representative hashing methods in terms of the P-R (Precision-Recall) curve.

The rest parts of this paper are organized as follows. Hashing methods closely related to our work are introduced in Section 2. Principle and detailed procedures of the proposed hashing method are explained in Section 3. Various experiments are discussed in Section 4. Conclusions of our work are summarized in Section 5.

2 Related Work

In general, an efficient hashing method should achieve a satisfactory balance between robustness and discrimination. In the literature, many scholars have tried their best to design some meaningful hashing methods for pursuing high classification performance. According to the robustness against image rotation, these image hashing methods can be grouped into the following three categories.

(1)*Hashing methods sensitive to rotation*. Generally, these methods do not take into account the robustness of rotation operation. For instance, Kang et al. [15] utilized compressive sensing to build an image hash. This method is sensitive to rotation. Wang et al. [16] proposed to calculate a secure hash for authentication using the techniques of DWT (Discrete Wavelet Transform) and DCT (Discrete Cosine Transform). Their method is robust to gamma correction, and JPEG compression. Huang et al. [17] applied the RW (Random Walk) technique to design a secure hashing scheme. Vadlamudi et al. [18] utilized SIFT (Scale-Invariant Feature Transform) and DWT features to build a hashing method. This method has good robustness against noise, scaling and other operations, while its discrimination is not good enough. Qin, Sun and Chang [19] jointly utilized SVD (Singular Value Decomposition), CVA (Color Vector Angle) and edge detection to design a new hashing method for color images. Their method simultaneously improves robustness and discrimination, but one of its disadvantages is the high time complexity. In another work [20], a hashing scheme with LBP (Local Binary Pattern) and color features is designed. The time complexity of this scheme is reduced, but the discrimination requires to be improved. Sajjad et al. [21] proposed a novel hashing method by Canny edge detection and DCT coefficients. This method has a good application in the field of smart industrial environment, but its rotation robustness should be improved. Iram et al. [22] used Gaussian pyramid and random noise to construct a hashing method. This method has good security. Recently, Hamid et al. [23] exploited the difference of Laplacian pyramids to design a hashing method. This method has good identification performance for small area tampering. However, the effect of rotation is not considered in this method. Shen and Zhao [24] utilized color opponent component and quadtree structure features to devise a hashing method. Shen and Zhao's method has a good localization capability on the tampered image. Liu et al. [25] introduced a novel hashing method with QLRBP (Quaternionic Local Ranking Binary Pattern) and mean pooling. This method is also sensitive to the operation of rotation.

(2)*Hashing methods moderate to rotation*. These methods take rotation into account, but they can only resist rotation within 20°. For instance, Monga and Evans [26] proposed a hashing method with an end-stopped wavelet. Monga and Evans's method achieves robustness against the rotation of 5°. Wu, Zhou and Niu [27] proposed a print-scan resistant hashing method using RT (Radon Transform), DWT and DFT (Discrete Fourier Transform). Their method can resist small angle rotation within 3°. Motivated by the performance of RT, Ou and Rhee [28] designed a hashing method based on DCT and RT. Ou and Rhee's method can resist rotation up to 5°. Zhao et al. [29] calculated a hash by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

combining the ZMs (Zernike Moments) and the position and texture features of salient region. Their method can be used for image authentication and can resist the small angle rotation of 5°. Davarzani et al. [30] combined SVD and LBP to find image hash. Davarzani's method has promising robustness, but the discrimination is not good enough. Li and Wang [31] used LRSD (Low-Rank Sparse Decomposition), CS (Compressed Sampling) and random projection to generate a hash. Li and Wang's method has good robustness and can resist the rotation angle of 10°. Qin et al. [32] exploited saliency structure and dual-cross pattern encoding to make up a hashing method. Qin's method is resilient to rotation of 5°. Tang et al. [33] proposed to use Canny edge detection and DWT features to generate an image hash. Tang's method shows good results in the application of image quality assessment, but it can only resist rotation within 5°. Yang et al. [34] used low-rank decomposition and LBP to generate a hash. Yang's method can only resist rotation of 3°. Liu and Huang [35] combined Hu moments in spatial domain and DCT coefficients in frequency domain to design a perceptual hashing method. Liu and Huang's method is robust to rotation of 10°. In another work, a global feature called GLCM (Gray-Level Co-occurrence Matrix) and local features with DCT coefficients are both used to come up with a novel hashing method in [36]. This hashing method is also robust to rotation of 10° and can be applied in copy detection. Recently, Ouyang, Zhang and Wen [37] introduced quaternion Gyrator transform to generate a hash. Their method demonstrates good performance in image authentication. Zhao and Yuan [38] utilized color structure and luminance gradient features to generate a hash. Zhao and Yuan's method is robust to rotation of 8°, but their discrimination needs to be improved. Recently, Tang et al. [39] utilized QSVD (Quaternion SVD) to learn compact hash of color image from PCS (Perceptual Color Space). As all color components are used in the quaternion domain, the discrimination of Tang's method is improved. In another work, Liang et al. [40] computed salient map with a visual attention model and extracted robust hash from the salient map using 2D PCA. Liang's method is robust against common operations except for large rotation.

(3)*Hashing methods robust to rotation*. These methods can resist the rotation with angle more than 20°. An early hashing method was given by Swaminathan, Mao and Wu [41], in which FMT (Fourier–Mellin Transform) and randomization method are used to generate a secure hash. The rotation robustness of Swaminathan's method is provided by FMT. Chen et al. [42] proposed a hashing method based on the Tchebichef moments. Chen's method can be used for authentication. Wang et al. [43] used Watson's visual model and SIFT features to come up with a hashing method. Wang's method can resist the rotation of 25° and finds good application in detecting tampered images. Karsh et al. [44] combined the spectral residual model, DWT and SVD to generate a hash. The discrimination of Karsh's method needs to be improved. Nie et al. [45] exploited the relationship between local feature points and unsupervised quantization techniques to calculate a hash. Nie's method shows good rotation robustness. To fully use color information, Tang et al. [46] selected CVA histogram as feature and compressed it with DCT for generating a short hash. Tang et al. [47] proposed a technique called RP (Ring Partition) and used RE (Ring Entropy) to produce image hash. In another work, Tang et al. [48] calculated a hash by MDS (Multi-Dimensional Scaling) and LPT (Log-Polar Transform). This method has promising robustness, but its discrimination is not desirable yet. Based on the RP technique, Alice et al. [49] used Haralick features to devise a hashing method for authentication. Similarly, Khelaifi and He [50] used RP and fractal image coding to design a hashing method. These RP-based methods can resist rotation of large degrees, but their discrimination needs to be improved. Recently, Biswas et al. [51] utilized DNS (Dominant Neighborhood Structure) and DCT to extract textural energy map for hash construction. Biswas's method can be used for recognizing Tor domains, but the time complexity is too high. Abdullahi et al. [52] combined FMT and fractal coding to devise a novel hashing method. Abdullahi's method is stable to rotation and has low time complexity. Singh et al. [53] utilized KAZE feature and SVD technique to produce image hash. Singh's method is also robust to rotation and illustrates a good application in tampering detection.

The rotation robustness and the main techniques of some representative hashing methods are presented in Table 1, where the first column is the reported methods, the second column is the rotation robustness, and the third column is the main techniques of each method. The results of Table 1 show that some methods are sensitive to rotation. Some methods are robust to rotation, but their discrimination performances should be greatly improved. In summary, most hashing methods do not make a satisfactory classification between discrimination and rotation robustness, and thus carry out inefficient performance in copy detection. To address this issue, this paper proposes an efficient hashing method using 2D-2D PCA for image copy detection.

TABLE 1: Robustness and techniques of some methods

Method	Rotation robustness	Main techniques
[18]	Sensitive	SIFT + DWT
[19]	Sensitive	SVD + CVA
[20]	Sensitive	LBP + DCT + DWT
[21]	Sensitive	Canny edge + DCT
[25]	Sensitive	QLRBP + mean pooling
[27]	Within 5°	RT + DWT + DFT
[28]	Within 5°	RT + DCT
[29]	Within 5°	ZMs + salient map
[30]	Within 5°	SVD + LBP
[31]	Within 10°	LRSD + CS
[33]	Within 5°	Canny edge + DWT
[35]	Within 10°	Hu moments + DCT
[36]	Within 10°	GLCM + DCT
[39]	Within 5°	QSVD + PCS
[40]	Within 5°	2D PCA + salient map
[43]	arbitrarily angle	Visual model + SIFT
[46]	arbitrarily angle	CVA + DCT
[47]	arbitrarily angle	RP + entropy
[48]	arbitrarily angle	MDS + LPT
[51]	arbitrarily angle	DCT + DNS
[52]	arbitrarily angle	FMT + fractal coding
[53]	arbitrarily angle	KAZE feature + SVD

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

3 Proposed Hashing Method

This section consists of four parts. Two main techniques, i.e., PCT and 2D-2D PCA, of the proposed hashing method are firstly explained in Section 3.1 and Section 3.2, respectively. Detailed procedures of the proposed hashing method are described in Section 3.3. Finally, the recommended similarity for evaluating hashes of the proposed method is introduced in Section 3.4. The below Sections explain the above-mentioned techniques in detail.

3.1 PCT

The PCT can convert an image from Cartesian coordinates to polar coordinates. It has been applied to numerous image applications, such as watermarking [54] and retrieval [55]. Details of the PCT are explained below.

Given an $M_0 \times M_0$ image $\mathbf{I}_0$, the PCT can map it into an $M_1 \times M_2$ image $\mathbf{I}_1$ in the polar coordinates, where $M_1=M_0/2$ and $M_2=360$. Let $I_1(r,\beta)$ be the element of $\mathbf{I}_1$ in the r-th row and β-th column, where $r = 1, 2, \cdots, M_0/2$, $\beta = 1, 2, \cdots, 360$. Thus, $I_1(r,\beta)$ can be calculated by the below formula.

$$I_1(r,\beta) = I_0(y,x) \tag{3}$$

where the coordinates (x,y) can be determined by the following equations.

$$x = \max(\text{floor}(M_0/2) + \text{round}(r\cos(\beta\frac{\pi}{180}), 1)) \tag{4}$$

$$y = \max(\text{floor}(M_0/2) + \text{round}(r\sin(\beta\frac{\pi}{180})), 1) \tag{5}$$

where $\text{round}(\cdot)$ is the rounding operation, $\text{floor}(\cdot)$ is the round down function. Note that the image $\mathbf{I}_1$ is generated by substituting all combination values of r and β into the equations (4) and (5). To understand the above calculations, the relation between polar coordinates and Cartesian coordinates is shown in Fig. 2.

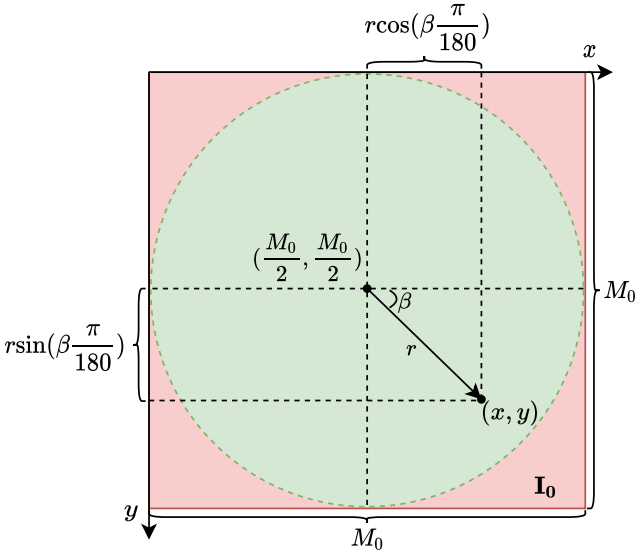
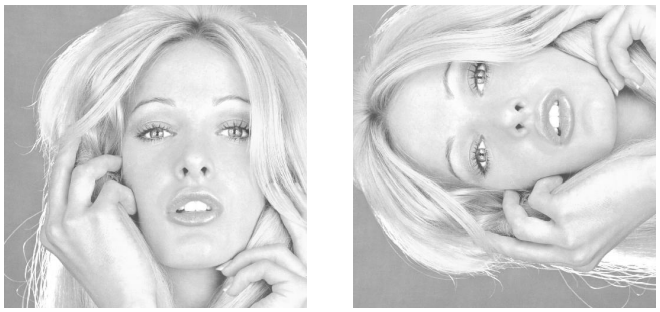


Fig. 2: Relation between polar coordinates and Cartesian coordinates

It is found that image rotation in the Cartesian coordinates corresponds to the translation in the polar coordinates. To understand this property, Fig. 3 presents visual examples of PCT, where (a) is an original image, (b) is the rotated image of (a) with 90 degrees, (c) and (d) are the PCT results of (a) and (b), respectively. Comparing (c) with (d), (d) is a translated version of (c) by circularly shifting with 90 along the column direction. This illustrates that the PCT can convert a rotated image into a translated image.

Note that the PCT plays an important role in the model of rotation-invariant feature extraction. In this model, the PCT converts an input rotated image to a translation matrix and the 2D-2D PCA learns rotation-invariant features from the translation matrix using the property of translation invariance. Details of this model are explained in Section 3.3.2.



(a) Original image (b) Rotated image

(c) PCT result of (a) (d) PCT result of (b)

Fig. 3: Visual examples of PCT

3.2 2D-2D PCA

2D-2D PCA [56] is an improvement of the classical PCA [57], [58], [59] for data dimensionality reduction. Compared with PCA, 2D-2D PCA can directly and quickly calculate the covariance matrix from the image matrix without transforming the image matrix into a vector. At present, 2D-2D PCA has been applied to face recognition [60], pattern recognition [61], target recognition [62], etc. In this paper, we investigate the use of 2D-2D PCA in hashing method. Specifically, we find a new property of 2D-2D PCA called translation invariance and then exploit it to design an efficient hashing method for copy detection. Section 3.2.1 introduces the principle of 2D-2D PCA and Section 3.2.2 proves the new property of 2D-2D PCA.

3.2.1 *Principle of 2D-2D PCA*

In general, the calculation of 2D-2D PCA can be divided into three steps, including column dimension reduction, row dimension reduction and projection feature matrix calculation. Details of these steps are introduced below.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

(1) *Column dimension reduction*

Assume that $\mathbf{B}_j$ $(1 \leq j \leq N)$ is the j-th training image with size $m \times n$, where N is the number of training images. Let $\mathbf{G}_1$ be an $n \times n$ non-negative covariance matrix of the training images. Thus, $\mathbf{G}_1$ is generated as follows.

$$\mathbf{G}_1 = \frac{1}{N}\sum_{j=1}^{N}\left(\mathbf{B}_j - \overline{\mathbf{B}}\right)^{\mathrm{T}}\left(\mathbf{B}_j - \overline{\mathbf{B}}\right) \tag{6}$$

where $\overline{\mathbf{B}}$ is the average result of the training images which can be computed below.

$$\overline{\mathbf{B}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{B}_j \tag{7}$$

Next, the eigenvalues and eigenvectors of $\mathbf{G}_1$ are calculated. The eigenvalues are arranged in descending order. Therefore, the matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d]$ can be determined by using the eigenvectors $\mathbf{u}_i (i = 1, 2, \ldots, d)$ of the first d maximum eigenvalues of $\mathbf{G}_1$.

For an image $\mathbf{B}_j$, the projected eigenvectors are obtained below.

$$\mathbf{F}_j = \mathbf{B}_j\mathbf{U} \tag{8}$$

where $\mathbf{F}_j$ is an $m \times d$ matrix composed of d eigenvectors.

(2) *Row dimension reduction*

After column dimension reduction, the $m \times n$ image is reduced to $m \times d$, that is, only the column size is reduced and the row size remains unchanged. With the new training samples $\mathbf{F}_j$ $(j = 1, 2, \ldots, N)$, the covariance matrix $\mathbf{G}_2$ can be calculated below.

$$\mathbf{G}_2 = \frac{1}{N}\sum_{j=1}^{N}\left(\mathbf{F}_j - \overline{\mathbf{F}}\right)\left(\mathbf{F}_j - \overline{\mathbf{F}}\right)^{\mathrm{T}} \tag{9}$$

where $\overline{\mathbf{F}}$ is computed as follows.

$$\overline{\mathbf{F}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{F}_j \tag{10}$$

Similarly, the eigenvalues and eigenvectors of $\mathbf{G}_2$ are calculated and the eigenvalues are arranged in descending order. Then, the matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ can be generated by using the eigenvectors $\mathbf{v}_i (i = 1, 2, \ldots, k)$ corresponding to the first k maximum eigenvalues of $\mathbf{G}_2$.

(3) *Projection feature matrix calculation*

With the matrices $\mathbf{U}$ and $\mathbf{V}$, the final projection matrix for the training image $\mathbf{B}_j$ is determined as follows.

$$\mathbf{Y}_j = \mathbf{V}^{\mathrm{T}}\mathbf{F}_j = \mathbf{V}^{\mathrm{T}}\mathbf{B}_j\mathbf{U} \tag{11}$$

where $\mathbf{Y}_j$ is the $k \times d$ matrix, which is also called the principal components (vectors) of $\mathbf{B}_j$. More detailed explanations about the 2D-2D PCA are in [56].

3.2.2 New property of translation invariance

After the investigation of 2D-2D PCA in hashing method, we find a new property of translation invariance. Specifically, if all training images are translated along the column direction with the same size, the projection matrix of the translated training image is the same with that of the original training image. As the 2D-2D PCA has become a useful technique of image processing, the new property is helpful to other image applications. The property of 2D-2D PCA can be theoretically proved as follows.

Theorem 1. Suppose that $\mathbf{B}_j^{'}$ is the translated matrix of $\mathbf{B}_j$ along the column direction $(1 \leq j \leq N)$. Let $\mathbf{Y}_j^{'}$ and $\mathbf{Y}_j$ be the projection feature matrices of $\mathbf{B}_j^{'}$ and $\mathbf{B}_j$, respectively. Thus, $\mathbf{Y}_j^{'}$=$\mathbf{Y}_j$.

Proof. According to the theory of linear algebra [63], the translated matrix $\mathbf{B}_j^{'}$ can be represented as follows.

$$\mathbf{B}_j^{'} = \mathbf{B}_j\mathbf{P} \tag{12}$$

where $\mathbf{P}$ is a matrix of order n, which consists of a set of the elementary matrix.

Let $\mathbf{G}_1^{'}$ be the covariance matrix after translation. Thus, according to the formula (6), $\mathbf{G}_1^{'}$ can be written as below.

$$\mathbf{G}_1^{'} = \frac{1}{N}\sum_{j=1}^{N}\left(\mathbf{B}_j^{'} - \overline{\mathbf{B}}^{'}\right)^{\mathrm{T}}\left(\mathbf{B}_j^{'} - \overline{\mathbf{B}}^{'}\right) \tag{13}$$

where $\overline{\mathbf{B}}^{'}$ is the translated mean matrix. According to the formula (7), $\overline{\mathbf{B}}^{'}$ is calculated below.

$$\overline{\mathbf{B}}^{'} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{B}_j^{'} \tag{14}$$

According to the formula (12), the formula (14) can be written as follows.

$$\overline{\mathbf{B}}^{'} = \frac{1}{N}\sum_{j=1}^{N}(\mathbf{B}_j\mathbf{P}) = (\frac{1}{N}\sum_{j=1}^{N}\mathbf{B}_j)\mathbf{P} = \overline{\mathbf{B}}\,\mathbf{P} \tag{15}$$

Substitute the formula (12) and the formula (15) into the formula (13) and thus obtain the below formula.

$$\begin{aligned}
\mathbf{G}_1^{'} &= \frac{1}{N}\sum_{j=1}^{N}\left(\mathbf{B}_j\mathbf{P} - \overline{\mathbf{B}}\mathbf{P}\right)^{\mathrm{T}}\left(\mathbf{B}_j\mathbf{P} - \overline{\mathbf{B}}\mathbf{P}\right) \\
&= \frac{1}{N}\sum_{j=1}^{N}\left[\left(\mathbf{B}_j - \overline{\mathbf{B}}\right)\mathbf{P}\right]^{\mathrm{T}}\left[\left(\mathbf{B}_j - \overline{\mathbf{B}}\right)\mathbf{P}\right] \\
&= \frac{1}{N}\sum_{j=1}^{N}\left[\mathbf{P}^{\mathrm{T}}\left(\mathbf{B}_j - \overline{\mathbf{B}}\right)^{\mathrm{T}}\left(\mathbf{B}_j - \overline{\mathbf{B}}\right)\mathbf{P}\right] \\
&= \mathbf{P}^{\mathrm{T}}\frac{1}{N}\sum_{j=1}^{N}\left[\left(\mathbf{B}_j - \overline{\mathbf{B}}\right)^{\mathrm{T}}\left(\mathbf{B}_j - \overline{\mathbf{B}}\right)\right]\mathbf{P} \\
&= \mathbf{P}^{\mathrm{T}}\mathbf{G}_1\mathbf{P}
\end{aligned} \tag{16}$$

Since $\mathbf{P}$ is an orthogonal matrix, the following formula can be deduced.

$$\mathbf{P}^{\mathrm{T}} = \mathbf{P}^{-1} \tag{17}$$

Next, substitute the formula (17) into the formula (16) and thus $\mathbf{G}_1^{'}$ can be written as follows.

$$\mathbf{G}_1^{'} = \mathbf{P}^{-1}\mathbf{G}_1\mathbf{P} \tag{18}$$

According to the definition of similarity matrices, from the equation (18), it can be concluded that $\mathbf{G}_1^{'}$ is similar to $\mathbf{G}_1$, namely $\mathbf{G}_1 \sim \mathbf{G}_1^{'}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

Let λ_1 and $\lambda_1^{'}$ be the eigenvalues of $\mathbf{G}_1$ and $\mathbf{G}_1^{'}$, and $\mathbf{U}$ and $\mathbf{U}^{'}$ be the eigenvectors of $\mathbf{G}_1$ and $\mathbf{G}_1^{'}$, respectively. Thus, the below formulas are obtained.

$$\mathbf{G}_1\mathbf{U} = \lambda_1\mathbf{U} \tag{19}$$

$$\mathbf{G}_1^{'}\mathbf{U}^{'} = \lambda_1^{'}\mathbf{U}^{'} \tag{20}$$

According to formula (18), the formula (20) can be written as follows.

$$\mathbf{P}^{-1}\mathbf{G}_1\mathbf{P}\mathbf{U}^{'} = \lambda_1^{'}\mathbf{U}^{'} \tag{21}$$

Left multiply $\mathbf{P}$ on both sides of the formula (21) as below.

$$\mathbf{P}\mathbf{P}^{-1}\mathbf{G}_1\mathbf{P}\mathbf{U}^{'} = \mathbf{P}\lambda_1^{'}\mathbf{U}^{'} \tag{22}$$

Thus, the formula (22) can be written as follows.

$$\mathbf{G}_1\mathbf{P}\mathbf{U}^{'} = \mathbf{P}\lambda_1^{'}\mathbf{U}^{'} \tag{23}$$

According to the property that the eigenvalues of similar matrices $\mathbf{G}_1^{'}$ and $\mathbf{G}_1$ are invariant. The below formula is available.

$$\lambda_1 = \lambda_1^{'} \tag{24}$$

Therefore, the formula (23) can be written as follows.

$$\mathbf{G}_1\mathbf{P}\mathbf{U}^{'} = \mathbf{P}\lambda_1\mathbf{U}^{'} = \lambda_1\mathbf{P}\mathbf{U}^{'} \tag{25}$$

Comparing the formula (25) ($\mathbf{P}\mathbf{U}^{'}$ is viewed as a whole) with the formula (19), the below formula can be deduced.

$$\mathbf{P}\mathbf{U}^{'} = \mathbf{U} \tag{26}$$

Thus, the following formula can be obtained by left multiplying the matrix $\mathbf{P}^{-1}$ on both sides of the formula (26) as follows.

$$\mathbf{U}^{'} = \mathbf{P}^{-1}\mathbf{U} \tag{27}$$

According to the formula (8), the projection of the translated eigenvector $\mathbf{F}_j{}^{'}$ is as follows.

$$\mathbf{F}_j{}^{'} = \mathbf{B}_j^{'}\mathbf{U}^{'} \tag{28}$$

Substitute the formula (12) and the formula (27) into the formula (28) and then the formula (28) can be written as follows.

$$\mathbf{F}_j^{'} = \mathbf{B}_j\mathbf{P}\mathbf{P}^{-1}\mathbf{U} = \mathbf{B}_j\mathbf{U} = \mathbf{F}_j \tag{29}$$

It is clear that $\overline{\mathbf{F}}^{'} = \overline{\mathbf{F}}$ according to the formula (10) and the formula (29). Consequently, $\mathbf{G}_2^{'}$ can be represented as follows.

$$\begin{aligned}\mathbf{G}_2^{'} &= \frac{1}{N}\sum_{j=1}^{N}\left(\mathbf{F}_j^{'} - \overline{\mathbf{F}}^{'}\right)^{\mathrm{T}}\left(\mathbf{F}_j^{'} - \overline{\mathbf{F}}^{'}\right) \\ &= \frac{1}{N}\sum_{j=1}^{N}\left(\mathbf{F}_j - \overline{\mathbf{F}}\right)^{\mathrm{T}}\left(\mathbf{F}_j - \overline{\mathbf{F}}\right) \\ &= \mathbf{G}_2\end{aligned} \tag{30}$$

Let $\mathbf{V}$ and $\mathbf{V}^{'}$ be the eigenvectors of $\mathbf{G}_2$ and $\mathbf{G}_2^{'}$, respectively. Thus, $\mathbf{V}^{'} = \mathbf{V}$ because $\mathbf{G}_2^{'} = \mathbf{G}_2$. According to the formula (11), the projection matrix of the translated matrix $\mathbf{B}_j^{'}$ is as follows.

$$\mathbf{Y}_j{}^{'} = \mathbf{V}^{'\mathrm{T}}\mathbf{F}_j{}^{'} = \mathbf{V}^{\mathrm{T}}\mathbf{F}_j = \mathbf{Y}_j \tag{31}$$

Theoretical proof is finished. The above proof can be briefly summarized as follows. After translation, the covariance matrix $\mathbf{G}_1^{'}$ is a similar matrix of its original covariance matrix $\mathbf{G}_1$. With the property of similar matrices, we have $\mathbf{F}_j^{'} = \mathbf{F}_j$ and thus obtain $\mathbf{G}_2^{'} = \mathbf{G}_2$. As $\mathbf{G}_2^{'}$ stays the same, its eigenvector matrix $\mathbf{V}^{'}$ also remains unchanged. Therefore, $\mathbf{Y}_j^{'}$ is the same with $\mathbf{Y}_j$. This illustrates that the 2D-2D PCA has the property of translation invariance, i.e., the projection matrix of the translated training image is the same with that of the original training image.

To make an easy understanding of the property of the 2D-2D PCA, visual examples of covariance matrices are presented in Fig. 4, where (a) is the benchmark image Lena, (b) is the covariance matrix $\mathbf{G}_1$ of (a), (c) is the covariance matrix $\mathbf{G}_2$ of (a), (d) is the translated version of (a) by cyclically shifting 60 columns from right to left, (e) is the covariance matrix $\mathbf{G}_1$ of (d), and (f) is the covariance matrix $\mathbf{G}_2$ of (d). Compared with (b), (e) is shifted by 60 units along the diagonal direction. For (c) and (f), they are exactly the same.

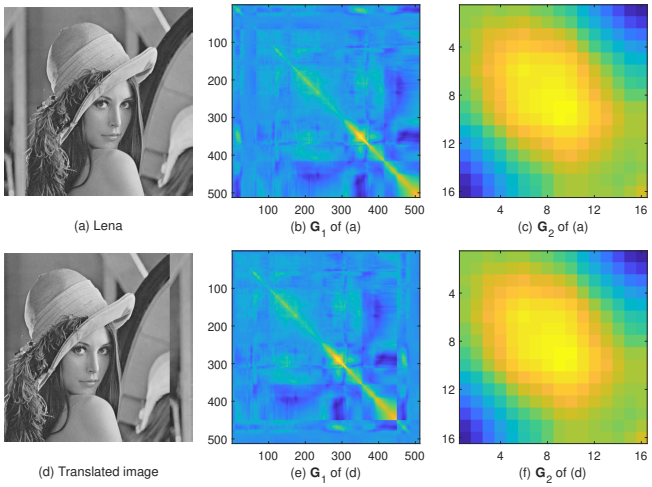


Fig. 4: Visual examples of covariance matrices

3.3 Detailed procedures of the proposed method

The proposed hashing method contains three procedures. Firstly, preprocessing is used to prepare a normalized image for stable feature extraction. Secondly, rotation-invariant features are extracted by combining the PCT and the 2D-2D PCA. Finally, low-dimensional features are treated as vectors and their distances are used to construct hash. Details of these procedures and pseudo-code of the proposed method are explained in the below sections.

3.3.1 Preprocessing

The preprocessing mainly includes three manipulations. Firstly, the input image is resized with the size of $M_0 \times M_0$. The main purpose is to generate hashes with the same lengths. Secondly, Gaussian low-pass filtering is used to mitigate the effects of noise and compression. Finally, the intensity component in the HSI color space is chosen to represent the image. Note that, in contrast to other color spaces, the choice of the HSI space makes preferable classification

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

performance. Section 4.6 will verify the color space selection through experiments.

3.3.2 Rotation-invariant feature extraction

Firstly, the PCT is exploited to convert the preprocessed image sized $M_0 \times M_0$ to the transformed image sized $M_1 \times M_2$ in the polar coordinates. Note that image rotation in the Cartesian coordinates corresponds to the translation in the polar coordinates. In other words, compared with the transformed image of an original image, the transformed image of its rotated image is translated along the column direction.

Secondly, non-overlapping block partition is conducted on the transformed image, where the block size is $m \times n$. Thus, the block number is $N = (M_1 \times M_2)/(m \times n)$. Here, $n = M_2$ is selected to preserve translation relation in blocks. Consequently, $N = M_1/m$.

Finally, the 2D-2D PCA is exploited to learn rotation-invariant low-dimensional features from image blocks. Here, the 2D-2D PCA is applied to all blocks $\mathbf{B}_j$ ($1 \leq j \leq N$). It is worth noting that the image blocks are the training images of the 2D-2D PCA. Next, $\mathbf{Y}_j$ of $\mathbf{B}_j$ is used to make up a vector $\mathbf{r}_j$ with the size of $L \times 1$, in which $L = k \times d$. Therefore, a feature matrix $\mathbf{R}$ can be obtained as follows.

$$\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N] \tag{32}$$

Note that the feature matrix $\mathbf{R}$ is rotation-invariant. The reasons are as follows. The PCT can convert an input rotated image to a translation matrix. During image blocking, translation relation is preserved in image blocks due to $n = M_2$. In addition, the 2D-2D PCA is invariant to translation. This property makes rotation-invariant features.

3.3.3 Vector distance calculation

To get a compact hash, each column of the feature matrix $\mathbf{R}$ is viewed as a vector and the distances of these feature vectors are used to construct hash. The vector distance is selected in this work. This is based on the observation that common digital attacks slightly disturb low-dimensional features in the 2D-2D PCA domain and thus their vector distances are stable to digital operations. Therefore, hash construction based on vector distances can ensure robustness and compactness. Details of vector distance calculation are explained as follows.

Firstly, a reference vector $\mathbf{r}_0$ is calculated as below.

$$\mathbf{r}_0 = [r_0(1), r_0(2), \ldots, r_0(L)]^{\mathrm{T}} \tag{33}$$

where $r_0(i)$ is the i-th element of $\mathbf{r}_0$ computed below.

$$r_0(i) = \frac{1}{N} \sum_{j=1}^{N} r_j(i) \tag{34}$$

in which $r_j(i)$ is the i-th element of the vector $\mathbf{r}_j$.

Secondly, the L_2 norm is utilized to calculate the distance between $\mathbf{r}_0$ and $\mathbf{r}_j$.

$$q_j = \sqrt{\sum_{i=1}^{L} [r_0(i) - r_j(i)]^2} \tag{35}$$

Thirdly, the distance q_j is quantized below.

$$h(j) = \mathrm{round}(q_j + 0.5) \tag{36}$$

In the end, the hash sequence $\mathbf{h}$ is obtained as follows.

$$\mathbf{h} = [h(1), h(2), \ldots, h(N)] \tag{37}$$

Hence, the hash length of the proposed method is N integers.

3.3.4 Pseudo-code of the proposed method

The proposed hashing method consists of preprocessing, rotation-invariant feature extraction with PCT and 2D-2D PCA, and vector distance calculation. To make an easy understanding of the procedures, the pseudocode of the proposed method is demonstrated in **Algorithm 1**.

Algorithm 1 Proposed hashing method

Input: An input image, parameters: M_0, m, n, k, d.
Output: A hash $\mathbf{h}$.

1: An input image is fixed to $M_0 \times M_0$ size.
2: The resized image is filtered by Gaussian low-pass filtering.
3: The intensity component of the filtered image in the HSI color space is utilized to represent the image.
4: The PCT is used to convert image from Cartesian coordinates to polar coordinates.
5: Divide the transformed image in the polar coordinates into N blocks sized $m \times n$.
6: Apply 2D-2D PCA to all blocks using the parameters k and d, and obtain the feature matrix $\mathbf{R}$ sized L×N.
7: **for** j=1 to j=N **do**
8: Calculate the distance q_j between $\mathbf{r}_0$ and $\mathbf{r}_j$ by the Eq. (35).
9: Calculate $h(j) = \mathrm{round}(q_j + 0.5)$.
10: **end for**
11: The hash is determined by $\mathbf{h} = [h(1), h(2), \ldots, h(N)]$.
12: **return** $\mathbf{h}$.

3.4 Hash similarity evaluation

The Pearson correlation coefficient [48] is adopted to analyze the similarity between two hashes. The value of the correlation coefficient is within the range [-1, 1]. Generally, the bigger the value is, the more similar the images of the input hashes are. Practically, a threshold can be used to determine whether two images are similar or not.

4 Experimental Results

Our used parameters are set below. The input image is fixed to the size of 512×512, the block size is 8×360, the Gaussian low-pass filtering is performed by the 3×3 convolution mask with a standard deviation of 1, and the dimension of feature selection is two. Therefore, the image size after the PCT is 256×360. In other words, our parameter settings are M_0=512, M_1=256, M_2=360, m=8, n=360, k=2, and d=2. Hence, $N = (M_1 \times M_2)/(m \times n) = 32$. Moreover, MATLAB R2018b is used to implement the proposed method, and the used PC is equipped with the Intel i7 dual-core 8700 CPU with 3.2 GHz and the random-access memory with 8.0 GB.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

4.1 Classification performance

The Kodak dataset [64] is used to verify the robustness of the proposed hashing method. There are 24 color images in the Kodak dataset, and the sizes of these images are 768×512 and 512×768. Fig. 5 lists some images of this database. StirMark, Photoshop and MATLAB are utilized to perform different robust attacks on the test images. These images are attacked by 10 operations [48], including CA (Contrast Adjustment), IS (Image Scaling), WE (Watermark Embedding), JPEG compression, GC (Gamma Correction), SPN (Salt and Pepper Noise), BA (Brightness Adjustment), GLF (Gaussian Low-pass Filtering), SN (Speckle Noise), and the CO (Combinational Operation) of image rotation and cropping. The operations and their parameter settings are shown in Table 2. Consequently, 24×80=1920 pairs of similar images are produced, and there are 1920+24=1944 images in the robustness test. An image database called VOC2012 [65] is used to conduct the discrimination test. The VOC2012 includes 17125 color images. The width sizes of these images are from 142 to 500 and their height sizes are from 71 to 500. Fig. 6 lists some sample images of the VOC2012 database. Similarity values between the hash of each image and those hashes of the other 17124 images are computed. Therefore, the total number of similarity values can be determined by the combinatorial number $C(17125, 2) = 17125(17125 - 1)/2 = 146624250$.

TABLE 2: Operations and their parameter settings

Operation	Parameter setting	Num
CA	magnitude $\in \{\pm 10, \pm 20\}$	4
IS	ratio$\in \{2, 1.5, 1.1, 0.9, 0.75, 0.5\}$	6
WE	strength:[10, 100], step:10	10
JPEG	factor:[30, 100], step:10	8
GC	$\gamma \in \{1.25, 1.1, 0.9, 0.75\}$	4
SPN	density:[0.001, 0.01], step:0.001	10
BA	magnitude $\in \{\pm 10, \pm 20\}$	4
GLF	standard deviation:[0.3, 1.0], step:0.1	8
SN	variance:[0.001, 0.01], step:0.001	10
CO	angle $\in \{\pm 1°, \pm 2°, \pm 5°, \pm 10°, \pm 15°, \pm 30°, \pm 45°, \pm 90°\}$	16
Total		80

Fig. 5: Some images of Kodak database

The ROC (Receiver Operating Characteristics) graph [66] is used for analysis. The y-axis and the x-axis of the graph are defined by true positive rate (R_1) and false positive rate (R_2) as follows.

$$R_1 = \frac{\#\text{ similar images correctly recognized}}{\#\text{ similar images}} \tag{38}$$

$$R_2 = \frac{\#\text{ different images wrongly classified}}{\#\text{ different images}} \tag{39}$$

Clearly, R_1 indicates robustness and R_2 shows discrimination. In addition, the AUC (Area Under the Curve) is calculated. The scope of AUC is from 0 to 1. Generally, a bigger AUC value means a better classification performance between discrimination and robustness.

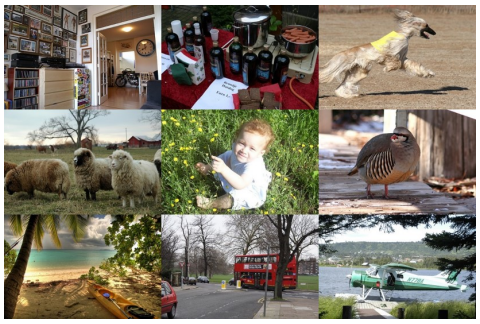


Fig. 6: Some images of VOC2012

To test the advantage of the proposed hashing method, some famous hashing methods are compared, including RW hashing method [17], QSVD hashing method [39], 2D PCA hashing method [40], MDS hashing method [48], GLCM-DCT hashing method [36] and RE hashing method [47]. The compared methods have been recently published in prestigious conferences or journals, and some of them are also dimension reduction-based methods, such as QSVD hashing method [39], 2D PCA hashing method [40] and MDS hashing method [48]. For these compared methods, their parameters and measures of hash similarity are configured by using their reported settings, and all input images are adjusted to 512×512 before calculating the hash.

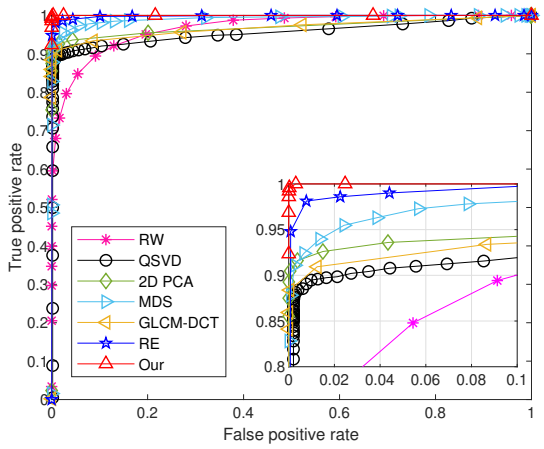


Fig. 7: Curves of different hashing methods

The curves of different methods are presented in Fig. 7. To better compare the local details of these curves, their parts located at the upper-left corner are magnified in the lower-right corner. The curve of the proposed hashing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

method is much nearer to the upper-left corner than those of other methods. Therefore, the classification performance of the proposed hashing method is better than those of other methods. For further analysis, the AUC values of the proposed hashing method and the compared methods are also calculated. The AUCs of RW hashing method, QSVD hashing method, 2D PCA hashing method, MDS hashing method, GLCM-DCT hashing method, RE hashing method and the proposed hashing method are 0.96716, 0.95499, 0.97012, 0.99211, 0.96821, 0.99800 and 0.99999, respectively. The AUC of the proposed hashing method is greater than those of the compared methods. This illustrates that the proposed hashing method is superior to the compared methods in classification performance. The proposed hashing method demonstrates advanced classification performance. This is mainly attributed to the rotation-invariant feature extraction by combining PCT and 2D-2D PCA. With the PCT, an input rotated image is converted to a translation matrix. Since the 2D-2D PCA is invariant to translation, the learned low-dimensional features from the translation matrix are rotation-invariant. Moreover, the use of 2D-2D PCA can make low-dimensional features discriminative. As common digital operations slightly disturb low-dimensional features in the 2D-2D PCA domain, their vector distances are stable to digital operations.

4.2 Copy detection performance

To illustrate our superiority in copy detection, comparisons are also conducted. To do so, Wang's database [67] is used. Fig. 8 demonstrates some images of Wang's database. Wang's database has 10 categories and each category contains 100 images. Hence, the total number of images in the database is 1000. In the experiment, one image is randomly selected from each category, some operations are used to attack the selected image of each category, and 24 image copies of every selected image are then generated. The used operations include: SN, GLF, CA, SPN, JPEG compression, BA, IS, AWGN, TI (Text Insertion), GM (Global Mosaic), LI (Logo Insertion), GC, and the IR (Image Rotation). These operations are implemented by MATLAB and their parameters for producing image copies are listed in Table 3. As there are 10 categories, the number of total image copies reaches 240. Hence, the test database has 1240 images. Note that the IR operation will expand the sizes of the attacked images. To make equal sizes of a query image and its image copies, every selected image is resized to a square image whose width is set to the maximum value of its width and its height before implementing these operations. And after these operations, the inscribed square of the inscribed circle of the square image is selected as the image copy. Similarly, every selected image is processed with the same manipulations to produce the actual query image. The well-known tool called P-R curve [36] is selected to theoretically analyze copy detection performance. In the P-R curve, the y-axis is the Precision and the x-axis is the Recall. They are calculated as follows:

$$\text{Precision} = \frac{\#\text{ image copies correctly detected}}{\#\text{ all images returned}} \quad (40)$$

$$\text{Recall} = \frac{\#\text{ image copies correctly detected}}{\#\text{ all image copies}} \quad (41)$$

Fig. 8: Some sample images of Wang's database

TABLE 3: Parameters of the operations for image copies

Operation	Parameter	Num
SN	variance: 0.02	1
GLF	standard deviation: 0.3	1
CA	magnitude: 20	1
SPN	density: 0.02	1
JPEG	factor $\in \{30, 50, 80\}$	3
BA	magnitude: 20	1
IS	ratio: $\in \{0.5, 0.75\}$	2
AWGN	variance: 0.01	1
TI	text: love in 2021	1
GM	block size: 5×5, 10×10	2
LI	logo size: 66×70, weight: 0.2	1
GC	γ: 0.75	1
IR	angle $\in \{1°, 5°, 8°, 10°, 15°, 30°, 45°, 90°\}$	8
Total		24

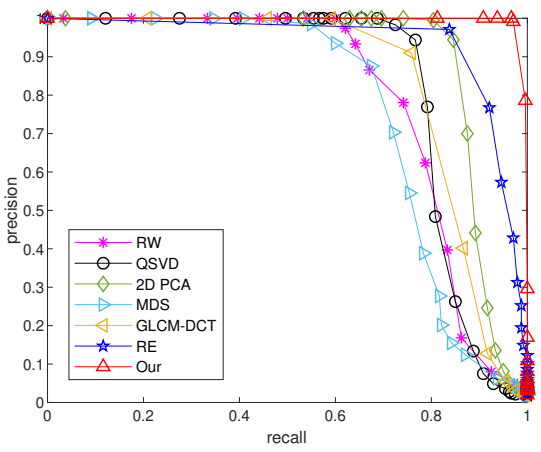


Fig. 9: P-R curves of different hashing methods

A P-R curve is formed by some points with the coordinates (Recall, Precision). In general, the P-R curve near the upper-right corner is better than the curve far away from the upper-right corner. Fig. 9 demonstrates the P-R curves of different hashing methods. Compared with the P-R curve of other methods, our P-R curve is much nearer to the upper-right corner. Hence, the proposed method is better than the compared hashing methods in copy detection performance. The PRAUC (P-R AUC) is utilized for quantitative analysis. The scope of PRAUC is from 0 to 1. Generally, a bigger PRAUC means a better copy detection performance. The results indicate that the PRAUCs of RW hashing method, QSVD hashing method, 2D PCA hashing method, MDS

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

hashing method, GLCM-DCT hashing method, RE hashing method and the proposed hashing method are 0.76737, 0.82526, 0.85605, 0.76031, 0.84006, 0.93436 and 0.99529, respectively. The PRAUC of the proposed hashing method is much greater than those of other methods. This also proves that the proposed hashing method achieves better copy detection performance than the compared methods.

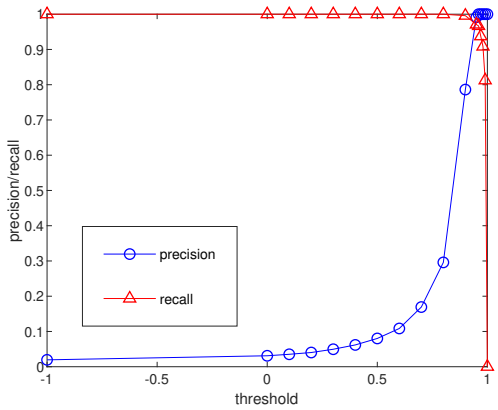


Fig. 10: Our precision curve and recall curve under different thresholds

The definitions of precision and recall reveal that a hashing method will have good performance in copy detection when its precision and recall are both high. Fig. 10 shows the precision curve and recall curve of the proposed hashing method under different thresholds, where the horizontal axis is the used threshold and the vertical axis is the precision/recall curve. In general, when the precision curve and the recall curve intersect, the corresponding threshold is selected as the optimal threshold, which provides the same value of precision and recall. From Fig. 10, it can be found that the optimal threshold of the proposed hashing method is 0.94552, and the corresponding values of precision and recall are both 0.97308. This means that 97.30% of image copies are correctly detected under the optimal threshold.

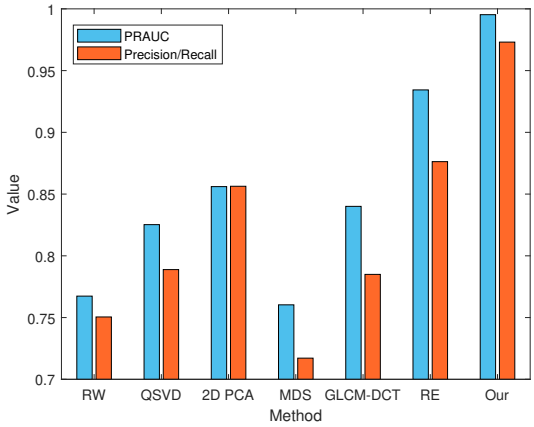


Fig. 11: Quantitative comparison of copy detection

Similarly, the optimal thresholds of the compared methods are calculated. With the optimal thresholds, their precision/recall values are also computed. The results show that the precision/recall values of the RW hashing method, QSVD hashing method, 2D PCA hashing method, MDS hashing method, GLCM-DCT hashing method, and RE hashing method are 0.75049, 0.78885, 0.85633, 0.71714, 0.78497, and 0.87627, respectively. Our precision/recall value under the optimal threshold is much greater than those of the compared methods. Fig. 11 presents the comparison results of these methods in copy detection, where the x-axis is the hashing method and the y-axis is the PRAUC and precision/recall.

4.3 Complexity performance

Complexity performance includes time complexity and storage complexity. Here, the time complexity is measured by the computational time of generating a hash using the evaluated hashing method. Specifically, the computational time is the average time of computing a hash in the discrimination test. The results indicate that the computational time of the proposed hashing method is 0.03423 seconds and the computational time of RW hashing method, QSVD hashing method, 2D PCA hashing method, MDS hashing method, GLCM-DCT hashing method and RE hashing method is 0.05875, 0.37287, 0.06126, 0.42853, 0.10248 and 0.01546 seconds, respectively. Obviously, the proposed hashing method is quicker than all compared methods, except the RE hashing method. Compared with some dimension reduction-based methods, such as the QSVD hashing method, 2D PCA hashing method and MDS hashing method, the proposed hashing method reaches a much faster speed due to the low computational cost of 2D-2D PCA.

Storage complexity is measured by the storage cost of a hash. To analyze the required bits of a hash, 17125 different images in VOC2012 are utilized. As every hash of the proposed hashing method has 32 integers, there are $17125 \times 32 = 548000$ integers in total. It is found that the maximum integer is 9978 and the minimum integer is 0. As $9978 < 16384{=}2^{14}$, 14 bits can represent a hash element. Therefore, the bit number of a hash of the proposed hashing method is 14×32 = 448. For reference, RW hashing method, QSVD hashing method, 2D PCA hashing method, MDS hashing method, GLCM-DCT hashing method and RE hashing method require 144 bits, 640 bits, 384 bits, 720 bits, 720 bits and 64 float digits to store a hash, respectively. Since a float digit needs 32 bits at least according to the IEEE standard, the bit length of a hash of the RE hashing method is 64×32 = 2048. It can be seen that the storage complexity of the proposed hashing method is higher than those of the RW hashing method and 2D PCA hashing method, but it is lower than those of the QSVD hashing method, MDS hashing method, GLCM-DCT hashing method and RE hashing method. The complexity summary is presented in Table 4.

4.4 Block size selection

This section analyzes the performance of selecting different block sizes on hash performance. The block sizes of 64×360, 32×360, 16×360, 8×360 and 4×360 are selected. These block sizes correspond to the block numbers of 4, 8, 16, 32 and 64, respectively. Fig. 12 shows the ROC curves under different block sizes. Furthermore, the AUCs under different block

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

TABLE 4: Complexity summary

Method	Time (s)	Bit length
RW	0.05875	144
QSVD	0.37287	640
2D PCA	0.06126	384
MDS	0.42853	720
GLCM-DCT	0.10248	720
RE	0.01546	2048
Our	0.03423	448

sizes are also calculated. The results demonstrate that the AUC values of 64×360, 32×360, 16×360, 8×360 and 4×360 are 0.97664, 0.99986, 0.99998, 0.99999 and 0.99992, respectively. It is clear that the AUC of 8×360 is greater than the AUCs of other block sizes. In addition, the AUC of the block size 64×360 is the smallest one. The reason is as follows. A larger block size means a smaller number of blocks and thus the 2D-2D PCA cannot effectively learn discriminative features from a few blocks. Therefore, the classification performance of the block size 8×360 is better than those of other block sizes.

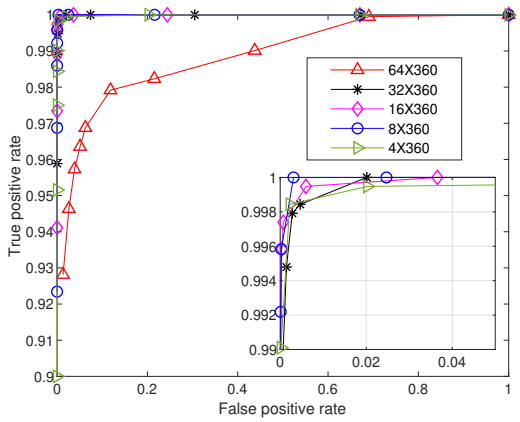


Fig. 12: Curves under different block sizes

The computational time of the proposed hashing method under different block sizes is calculated. The experiments indicate that the time of the block sizes 64×360, 32×360, 16×360, 8×360 and 4×360 is 0.02982, 0.03084, 0.03214, 0.03423 and 0.03958 seconds, respectively. The time of different block sizes is about 0.03 seconds and their difference are small. Table 5 shows the performance comparison under different block sizes. From the above results, it can be found that the proposed hashing method achieves better performances in classification and computational time when the block size is 8×360.

TABLE 5: Performance comparison under different block sizes

Block size	64×360	32×360	16×360	8×360	4×360
AUC	0.97664	0.99986	0.99998	0.99999	0.99992
Time (s)	0.02982	0.03084	0.03214	0.03423	0.03958

4.5 Dimension selection

In 2D-2D PCA, the parameters k and d correspond to the dimension selections along the row direction and column direction, respectively. For simplicity, $k = d$ is selected in the discussion. Here, the dimension values of 2, 3, 4, 5, 6 and 7 are chosen. In the experiments, only the k and d values are varied and other parameters remain unaltered. Fig. 13 is the ROC curves under different dimensions. The AUCs of the dimension values of 2, 3, 4, 5, 6 and 7 are 0.99999, 0.99993, 0.99998, 0.99994, 0.99997 and 0.99996, respectively. The AUC of dimension 2 is slightly bigger than those of other dimensions. The computational time of the dimension values of 2, 3, 4, 5, 6 and 7 is 0.03423, 0.03474, 0.03481, 0.03488, 0.03492 and 0.03556 seconds, respectively. Obviously, there is little difference in computational time. Performance comparisons of different dimensions are summarized in Table 6. From the results, it can be found that the selection of $k = d = 2$ is a good choice in terms of the whole performance.

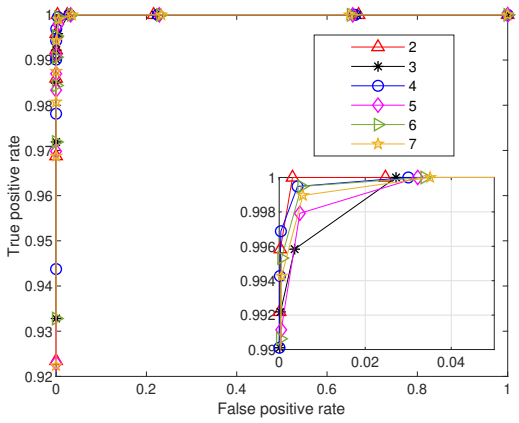


Fig. 13: Curves under different dimensions

TABLE 6: Performance comparison under different dimensions

k	2	3	4	5	6	7
AUC	0.99999	0.99993	0.99998	0.99994	0.99997	0.99996
Time (s)	0.03423	0.03474	0.03481	0.03488	0.03492	0.03556

4.6 Color space selection

To verify the superiority of the use of HSI color space, classification performances of the proposed hashing method under different spaces are also compared. The compared color spaces consist of HSI, CIE L*a*b*, YCbCr, and HSV spaces. In the comparison, only the intensity/luminance/value component is different. Specifically, the intensity component in HSI space, the luminance component in CIE L*a*b* space, the luminance component in YCbCr space and the value component in HSV space are selected. All color spaces are compared by using the ROC curves shown in Fig. 14. The calculated AUCs show that the values of CIE L*a*b* space, HSI space, YCbCr space and HSV space are 0.99964, 0.99999, 0.99951 and 0.99996, respectively. The AUC value of the HSI space is slightly larger than those of other spaces. Moreover, the computational time of CIE L*a*b* space, HSI space, YCbCr space and HSV space is 0.05286, 0.03423, 0.03292 and 0.03521 seconds, respectively. Table 7 presents performance comparison under different color spaces. In the viewpoint

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3131188, IEEE Transactions on Knowledge and Data Engineering

of the whole performance, the HSI color space is better than other spaces.

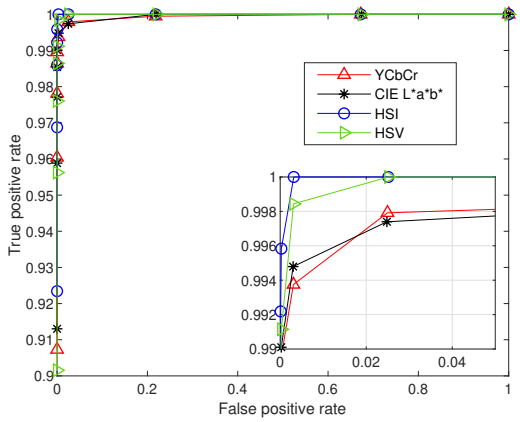


Fig. 14: Curves under different color spaces

TABLE 7: Performance comparison under different color spaces

Color space	CIE L*a*b*	HSI	YCbCr	HSV
AUC	0.99964	0.99999	0.99951	0.99996
Time (s)	0.05286	0.03423	0.03292	0.03521

5 Conclusions

This paper has proposed an efficient hashing method using 2D-2D PCA for image copy detection. The key is the discovery of the translation invariance of 2D-2D PCA which is first reported and theoretically proved in this work. With the property of translation invariance, a novel model of extracting rotation-invariant low-dimensional features is designed by combining PCT and 2D-2D PCA. The PCT can convert an input rotated image to a translation matrix. Since 2D-2D PCA is invariant to translation, the learned low-dimensional features from the translation matrix are rotation-invariant. In addition, vector distances of low-dimensional features are exploited to construct hash. As common digital operations slightly disturb low-dimensional features in the 2D-2D PCA domain, hash construction with vector distances provides good robustness and compactness of the proposed hashing method. Numerous experiments on three open image databases have been discussed to demonstrate the efficiencies of the proposed hashing method. Comparison results have demonstrated that the proposed hashing method outperforms some state-of-the-art hashing methods in the performances of classification and copy detection.

Acknowledgments

Many thanks to the anonymous referees for their helpful suggestions. This work is partially supported by the National Natural Science Foundation of China (61962008, 62062013).

References

[1] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for web image search," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4649–4661, 2012.
[2] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua, "Event driven web video summarization by tag localization and key-shot identification," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 975–985, 2012.
[3] Z. Zhou, Y. Wang, Q. M. J. Wu, C.-N. Yang, and X. Sun, "Effective and efficient global context verification for image copy detection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 48–63, 2017.
[4] C. Kim, "Content-based image copy detection," *Signal Processing: Image Communication*, vol. 18, no. 3, pp. 169–184, 2003.
[5] Z. Tang, X. Q. Zhang, and S. Zhang, "Robust perceptual image hashing based on ring partition and NMF," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 711–724, 2014.
[6] Z. Tang, L. Chen, X. Q. Zhang, and S. Zhang, "Robust image hashing with tensor decomposition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 549–560, 2019.
[7] Z. Huang and S. Liu, "Robustness and discrimination oriented hashing combining texture and invariant vector distance," in *Proceedings of the 26th ACM international conference on Multimedia (MM 2018)*, New York, NY, USA, 2018, pp. 1389–1397.
[8] J. Cheng, C. Leng, P. Li, M. Wang, and H. Lu, "Semi-supervised multi-graph hashing for scalable similarity search," *Computer Vision and Image Understanding*, vol. 124, pp. 12–21, 2014.
[9] P. J. McParlane, A. J. McMinn, and J. M. Jose, ""Picture the scene...";: visually summarising social media events," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM 2014)*, New York, NY, USA, 2014, pp. 1459–1468.
[10] Z. Tang, X. Q. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 200–214, 2016.
[11] Z. Huang and S. Liu, "Perceptual hashing with visual content understanding for reduced-reference screen content image quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2808–2823, 2021.
[12] U. Breidenbach, M. Steinebach, and H. Liu, "Privacy-enhanced robust image hashing with bloom filters," in *Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES 2020)*, New York, NY, USA, 2020, pp. 1–10.
[13] Y. Zhao, S. Wang, X. P. Zhang, and H. Yao, "Robust hashing for image authentication using zernike moments and local features," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 55–63, 2013.
[14] C. Jiang and Y. Pang, "Perceptual image hashing based on a deep convolution neural network for content authentication," *Journal of electronic imaging*, vol. 27, no. 4, Article ID. 043055, 2018.
[15] L. Kang, C. Lu, and C. Hsu, "Compressive sensing-based image hashing," in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2009)*, 2009, pp. 1285–1288.
[16] X. Wang, N. Zheng, J. Xue, and Z. Liu, "A novel image signature method for content authentication," *The Computer Journal*, vol. 55, no. 6, pp. 686–701, 2012.
[17] X. Huang, X. Liu, G. Wang, and M. Su, "A robust image hashing with enhanced randomness by using random walk on zigzag blocking," in *Proceedings of the 2016 IEE Trustcom/BigDataSE/ISPA*, 2016, pp. 14–18.
[18] L. N. Vadlamudi, R. P. V. Vaddella, and V. Devara, "Robust image hashing using SIFT feature points and DWT approximation coefficients," *ICT Express*, vol. 4, no. 3, pp. 154–159, 2018.
[19] C. Qin, M. Sun, and C. C. Chang, "Perceptual hashing for color images based on hybrid extraction of structural features," *Signal Processing*, vol. 142, pp. 194–205, 2018.
[20] C. Qin, Y. Hu, H. Yao, X. Duan, and L. Gao, "Perceptual image hashing based on weber local binary pattern and color angle representation," *IEEE Access*, vol. 7, pp. 45 460–45 471, 2019.
[21] M. Sajjad, I. U. Haq, J. Lloret, W. Ding, and K. Muhammad, "Robust image hashing based efficient authentication for smart industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6541–6550, 2019.
[22] B. Iram, A. Fawad, A. Jawad, B. Wadii, and A. Nouf, "A secure and robust image hashing scheme using Gaussian pyramids," *Entropy*, vol. 21, pp. 1–20, 2019.

[23] H. Hamid, F. Ahmed, and J. Ahmad, "Robust image hashing scheme using Laplacian pyramids," *Computers and Electrical Engineering*, vol. 84, Article ID. 106648, 2020.

[24] Q. Shen and Y. Zhao, "Perceptual hashing for color image based on color opponent component and quadtree structure," *Signal Processing*, vol. 166, pp. 107 244.1–107 244.12, 2020.

[25] E. Liu, H. Yao, Y. Hu, and F. Cao, "Perceptual color image hashing based on quaternionic local ranking binary pattern," *IETE Technical Review*, vol. 38, pp. 1–14, 2020.

[26] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: Performance evaluation and tradeoffs," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3452–3465, 2006.

[27] D. Wu, X. Zhou, and X. Niu, "A novel image hash algorithm resistant to print–scan," *Signal Processing*, vol. 89, pp. 2415–2424, 2009.

[28] Y. Ou and K. H. Rhee, "A key-dependent secure image hashing scheme by using Radon transform," in *Proceedings of the 2009 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2009, pp. 595–598.

[29] Y. Zhao, S. Wang, X. P. Zhang, and H. Yao, "Robust hashing for image authentication using zernike moments and local features," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 55–63, 2013.

[30] R. Davarzani, S. Mozaffari, and K. Yaghmaie, "Perceptual image hashing using center-symmetric local binary patterns," *Multimedia Tools and Applications*, vol. 75, no. 8, pp. 4639–4667, 2016.

[31] Y. Li and P. Wang, "Robust image hashing based on low-rank and sparse decomposition," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, 2016, pp. 2154–2158.

[32] C. Qin, X. Chen, X. Luo, X. P. Zhang, and X. Sun, "Perceptual image hashing via dual-cross pattern encoding and salient structure detection," *Information Sciences*, vol. 423, pp. 284–302, 2017.

[33] Z. Tang, Z. Huang, H. Yao, X. Q. Zhang, L. Chen, and C. Yu, "Perceptual image hashing with weighted DWT features for reduced-reference image quality assessment," *The Computer Journal*, vol. 61, pp. 1695–1709, 2018.

[34] H. Yang, J. Yin, and Y. Yang, "Robust image hashing scheme based on low-rank decomposition and path integral LBP," *IEEE Access*, vol. 7, pp. 51 656–51 664, 2019.

[35] S. Liu and Z. Huang, "Efficient image hashing with geometric invariant vector distance for copy detection," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, pp. 1–22, 2019.

[36] Z. Huang and S. Liu, "Perceptual image hashing with texture and invariant vector distance for copy detection," *IEEE Transactions on Multimedia*, vol. 23, pp. 1516–1529, 2021.

[37] J. Ouyang, X. Zhang, and X. Wen, "Robust hashing based on quaternion Gyrator transform for image authentication," *IEEE Access*, vol. 8, pp. 220 585–220 594, 2020.

[38] Y. Zhao and X. Yuan, "Perceptual image hashing based on color structure and intensity gradient," *IEEE Access*, vol. 8, pp. 26 041–26 053, 2020.

[39] Z. Tang, M. Yu, H. Yao, H. Zhang, C. Yu, and X. Q. Zhang, "Robust image hashing with singular values of quaternion SVD," *The Computer Journal*, pp. 1–16, 2021.

[40] X. Liang, Z. Tang, X. Xie, J. Wu, and X. Q. Zhang, "Robust and fast image hashing with two-dimensional PCA," *Multimedia Systems*, vol. 27, pp. 389–401, 2021.

[41] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 1, pp. 215–230, 2006.

[42] Y. Chen, W. Yu, and J. Feng, "Robust image hashing using invariants of Tchebichef moments," *Optik*, vol. 125, no. 19, pp. 5582–5587, 2014.

[43] X. Wang, K. Pang, X. Zhou, Y. Zhou, L. Li, and J. Xue, "A visual model-based perceptual image hash for content authentication," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1336–1349, 2015.

[44] R. Karsh, R. Laskar, and Aditi, "Robust image hashing through DWT-SVD and spectral residual method," *EURASIP Journal on Image and Video Processing*, vol. 31, pp. 1–17, 2017.

[45] X. Nie, X. Li, Y. Chai, C. Cui, X. Xi, and Y. Yin, "Robust image fingerprinting based on feature point relationship mining," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1509–1523, 2018.

[46] Z. Tang, X. Li, X. Q. Zhang, S. Zhang, and Y. Dai, "Image hashing with color vector angle," *Neurocomputing*, vol. 308, pp. 147–158, 2018.

[47] Z. Tang, X. Q. Zhang, L. Huang, and Y. Dai, "Robust image hashing using ring-based entropies," *Signal Processing*, vol. 93, pp. 2061–2069, 2013.

[48] Z. Tang, Z. Huang, X. Q. Zhang, and H. Lao, "Robust image hashing with multidimensional scaling," *Signal Processing*, vol. 137, pp. 240–250, 2017.

[49] K. Alice, N. Ramaraj, and S. Rajagopalan, "Rotation invariant image authentication using Haralick features," *Multimedia Tools and Applications*, vol. 79, pp. 17 211–17 225, 2020.

[50] F. Khelaifi and H. He, "Perceptual image hashing based on structural fractal features of image coding and ring partition," *Multimedia Tools and Applications*, vol. 79, pp. 1–20, 2020.

[51] R. Biswas, V. González-Castro, E. Fidalgo, and E. Alegre, "Perceptual image hashing based on frequency dominant neighborhood structure applied to tor domains recognition," *Neurocomputing*, vol. 383, pp. 24–38, 2020.

[52] S. M. Abdullahi, H. Wang, and T. Li, "Fractal coding-based robust and alignment-free fingerprint image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2587–2601, 2020.

[53] S. Singh, G. Bhatnagar, and A. Singh, "A new robust reference image hashing system," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–15, 2021 (In press).

[54] A. Mohashin, P. Dhar, and T. Shimamura, "Blind image watermarking based on discrete hilbert transform and polar decomposition," in *Proceedings of the 2019 11th International Conference on Knowledge and Smart Technology (KST)*, 2019, pp. 78–81.

[55] X. Yang, X. Gao, and Q. Tian, "Polar embedding for aurora image retrieval," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3332–3344, 2015.

[56] D. Zhang and Z.-H. Zhou, "(2D)2PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1, pp. 224–231, 2005.

[57] M. Tan, Z. Hu, Y. Yan, J. Cao, D. Gong, and Q. Wu, "Learning sparse PCA with stabilized ADMM method on stiefel manifold," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 3, pp. 1078–1088, 2021.

[58] K. Allab, L. Labiod, and M. Nadif, "A semi-NMF-PCA unified framework for data clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 2–16, 2017.

[59] X. Wang, J. Liu, Y. Cheng, A. Liu, and E. Chen, "Dual hypergraph regularized PCA for biclustering of tumor gene expression data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2292–2303, 2019.

[60] K. Hancherngchai, T. Titijaroonroj, and J. Rungrattanaubol, "An individual local mean-based 2DPCA for face recognition under illumination effects," in *Proceedings of the 16th International Joint Conference on Computer Science and Software Engineering (JCSSE 2019)*, 2019, pp. 213–217.

[61] Y. Ji and H.-B. Xie, "Stationary wavelet and two-directional 2DPCA for pattern recognition of electromyographic signal," in *Proceedings of the 2017 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, 2017, pp. 83–86.

[62] J. Xu, P. Bi, X. Du, and J. Li, "Target recognition based on dynamic (2D)2PCA for UUV optical vision system," *Optik*, vol. 179, pp. 154–164, 2019.

[63] G. Strang, G. Strang, G. Strang, and G. Strang, *Introduction to linear algebra*. Wellesley-Cambridge Press Wellesley, MA, 1993.

[64] R. Franzen, *Accessed April 15 (2017)*, Kodak lossless true color image suite. [Online]. Available: http://r0k.us/graphics/kodak/

[65] M. Everingham, L. V. Gool, C. Williams, J. Winn, A. Zisserman, Y. Aytar, and A. Eslami, *Accessed April 1 (2020)*, The pascal visual object classes challenge 2012 datasets. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

[66] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[67] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.