Perceptual Image Hashing With Texture and Invariant Vector Distance for Copy Detection

Ziqing Huang and Shiguang Liu, *Member, IEEE*

***Abstract*—Content-based image copy detection has become one of the important technologies in copyright protection, where two major processes, content-based feature extraction and matching are included. However, it is certainly true that enough storage space is required to establish feature database for matching, which greatly increases time and storage consumption, as well as lacks flexibility. Fortunately, perceptual image hashing is a good strategy to address these problems, in which content-based features are extracted and further encoded to hash codes. On the one hand, content-based features provide and ensure higher copy detection accuracy, while on the other hand, hash codes instead of feature database reduce storage space and improve time efficiency. Meanwhile, a better balance between robustness and discrimination is one of the most objectives of image hashing, which is conducive to its application in multimedia management and security. Consequently, we present an effective image hashing method for copy detection. Specifically, to obtain perceptual robustness against to copy attacks, we extract the global statistical characteristics in gray-level co-occurrence matrix (GLCM) to reveal texture changes. Then, to make up the discrimination limitation, we leverage the local dominant DCT coefficients from the first row/column in each sub-image to calculate vector distance. Finally, two kinds of complementary information (global feature via texture and local feature via vector distance) are simultaneously preserved to generate hash codes. Various experiments performed on benchmark database indicate that our proposed perceptual image hashing provides higher detection accuracy and better balance between robustness and discrimination than the state-of-the-art algorithms.**

***Index Terms*—Copy detection, perceptual hashing, robustness, discrimination, invariant vector distance.**

I. Introduction

With the rapid technological development of multimedia and network communication, digital content (such as images, videos, sounds) has gradually become one of the main carriers of information dissemination. However, digital content is easy to be modified and redistributed, for instance, digital images are always suffered from copy attacks for meeting a variety of needs. Specifically, the copy versions of an original image are not only defined as the exact duplicates, but also the legal distorted versions generated by various copy attacks in most cases, e.g., rotation, scaling, noise, compression, contrast/brightness adjustment. The original image and its copy versions are understood to share the same digital copyright. In this case, the copyright protection of copy images is particularly important.

Generally, watermarking and content-based copy detection are two common approaches for protecting digital content copyright. In watermarking scheme, it embeds the copyright information into an original image, which plays a certain protective role but irreversibly destroying the image information [1]. It is even unreliable because watermarks would be removed by some post-processing methods. Subsequently, content-based copy detection [2], [3] has become one of the important technologies in copyright protection field, in which legal copies are correctly detected by matching the extracted content-based features. The major advantages of content-based copy detection are that it does not need additional embedded watermarks, and content-based features can better represent the intrinsic characteristics of an image for improving the detection accuracy. However, it is very necessary that enough storage space is required to establish feature database for matching, which greatly increases time and storage consumption, as well as lacks flexibility in practice.

Furthermore, it is important to note that a large number of near-duplicate similar images captured by different conditions, i.e., positions, illuminations, and viewpoints, are widely distributed compared with the copy versions of an original image. Although near-duplicate similar images could be more visually similar to an original image than the copy versions generated by strong copy attacks, the near-duplicate similar image and an original image are determined to have different digital copyright [4]–[6]. Fig. 1 is an example of copy image and near-duplicate similar image, where Fig. 1(a) is an original image, Fig. 1(b) is a legal copy image generated from Fig. 1(a) by brightness adjustment, as well as Fig. 1(c) and (d) are near-duplicate similar images captured by different conditions. Indeed, they are near-duplicate similar with Fig. 1(a) rather than the legal copy version and do not share the same digital copyright. This situation undoubtedly increases the difficulty for content-based copy detection.

Perceptual image hashing is a good strategy to cope with the above problems, in which content-based features are extracted

Manuscript received March 18, 2019; revised April 16, 2020; accepted May 20, 2020. Date of publication June 3, 2020; date of current version May 26, 2021. This work was supported by the Natural Science Foundation of China under Grants 61672375 and 61170118. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Benoit Huet. *(Corresponding author: Shiguang Liu.)*

Ziqing Huang is with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: skyhuangzq@163.com).

Shiguang Liu is with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and also with the Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin 300350, China (e-mail: lsg@tju.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2020.2999188

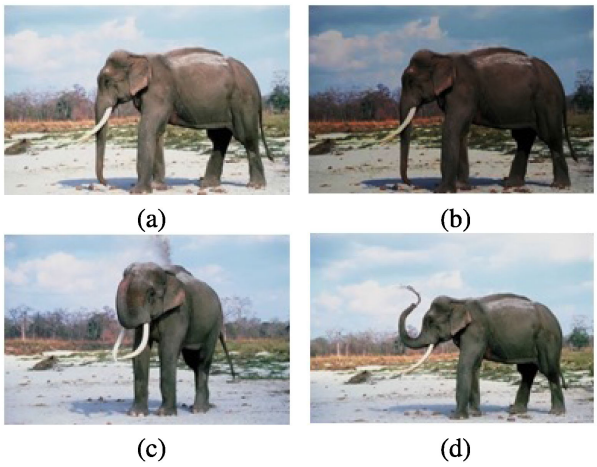

Fig. 1. Examples of copy image and near-duplicate similar images. (a) is an original image, (b) is a legal copy image generated from (a) by brightness adjustment, as well as (c) and (d) are near-duplicate similar images captured by different conditions with (a).

and further encoded to hash codes [7]–[9]. On the one hand, content-based features provide and ensure higher copy detection accuracy, while on the other hand, hash codes instead of feature database greatly reduce storage space and improve time efficiency. With this fact in mind, we focus on the study of content-based copy detection via perceptual image hashing to protect digital content copyright, and this paper is an extension work of an earlier image hashing method[1] presented at the International Conference on ACM Multimedia. There are three main improvements on the earlier version as follows:

1) To well increase the security of the proposed algorithm, a pseudo-random secret stream generated with a secret key is deployed for constructing a final hash, which will be validated in experiment to illustrate that the improved proposed hashing is key sensitivity and sufficient security.
2) To elucidate HSI color space is the best choice to extract image feature for generating hash codes, other two typical color spaces are leveraged to design the hashing algorithm, i.e., CIE Lab color space based hashing, and YCbCr color space based hashing. Experimental results also prove that hashing performance in HSI color space is optimal.
3) To demonstrate that the proposed method is robust to various copy attacks, one typical image database is added and utilized to examine its robustness against to copy attacks with sets of experiments. Various experimental results will indicate that our improved image hashing method exhibits superior perceptual robustness and discrimination over existing popular methods, as well as reaches better detection accuracy in image copy detection.

The remainder of this paper is organized as follows. The related work is reviewed in Section II. The proposed perceptual image hashing for copy detection is presented in Section III. Section IV illustrates experimental results towards image hashing performance between robustness and discrimination and the detection accuracy in copy detection system, respectively. Conclusions are finally given in Section V.

[1]Robustness and Discrimination Oriented Hashing Combining Texture and Invariant Vector Distance. In Proceedings of ACM International Conference on Multimedia 2018. [Online]. Available: https://doi.org/10.1145/3240508.3240690

II. Related Work

In this section, we will concisely review some previous works related to our method, mainly from two aspects: the content-based copy detection and the content-based perceptual image hashing.

A. Content-Based Copy Detection

The crucial issue of content-based copy detection method is to extract the intrinsic descriptive features as the identification of an image. According to the extracted descriptive features, image copy detection methods can be roughly divided into two categories [10]: global-feature methods and local-feature methods. Specifically, global-feature methods usually retrieve copy versions with color features, histogram features, texture features, structure features, and shape features, which is fast with high global discrimination. However, it fails to achieve good robustness against some geometric attacks and local modification. For example, Kim [2] extracted AC coefficients in discrete cosine transform (DCT) domain to calculate distances between original image and copy image, and then optimal threshold was selected to detect copy versions. Li *et al.* [11] considered the contribution of Gabor texture descriptor to geometric robustness, and took the statistical moment in Gabor texture with different orientations and scales to describe an image, which is effective to most copy attacks except for large-angle rotations.

Different from global-feature methods, local-feature methods are not only robust to conventional copy attacks, such as noise, compression, contrast/brightness adjustment, blurring, but also robust to geometric attacks [6], [12]–[17]. Among many methods, scale invariant feature transform (SIFT) features and its extensions, e.g., principal component analysis on SIFT [15], speeded-up robust feature [18], multi-scale SIFT [6], had been widely favoured. For example, Kang *et al.* [12] accomplished copy detection with sparse representations and reconstruction errors in secure SIFT domain. In another work, Yan and Sukthankar [15] used principal component analysis to decrease the dimension and increase the matching precision of the SIFT. However, we need to understand that the local-feature methods could achieve better robustness to wide-range attacks at the expense of time complexity. More importantly, lower global discrimination is an inevitable disadvantage due to the features extracted from small blocks.

As introduced above, it is a fact that the global-features are sensitive to geometric attacks, especially scaling and rotation, while the local-features fail to remain global context information result in poor discrimination. Therefore, one good idea is to combine global and local features for providing complementary information to each other so as to achieve better detection accuracy. Zhou *et al.* [4] took into account the global context descriptor for verification of local SIFT feature matches, and

reduced false matches as much as possible to improve the accuracy. Nevertheless, the huge feature database still largely limits the usability of such methods.

B. Content-Based Perceptual Image Hashing

In content-based perceptual image hashing scheme, the key steps are feature extraction and quantization. Similarly, features involved will be global, local, or a combination of both. But, we start from the main techniques used in the current hashing methods to roughly divide, i.e., matrix theory [19]–[23], manifold learning [24]–[27], and frequency domain coefficients [28]–[35].

Essentially, matrix theory focuses more on the global feature extraction of the image. That is, an image is treated as a global matrix, and eigenvalue or eigenvectors are extracted for generating hash codes through matrix decomposition methods, such as singular value decomposition (SVD), nonnegative matrix factorization (NMF), and QR factorization. Kozat *et al.* [19] first introduced SVD to extract eigenvectors for generating hash codes in 2004. Although the robustness and discrimination of this method is not very good, it does not affect that matrix decomposition is a good attempt. Subsequently, NMF instead of SVD achieves a series of meaningful results. For example, Tang *et al.* [21] constructed secondary image by ring partition, conducted NMF on secondary image to extract robust features, and quantized robust features to hash sequence, which presented good robustness to rotation, scaling. Meanwhile, in order to address the nonnegative constraint of NMF, Chen *et al.* [23] further developed semi-NMF for image hashing. Altogether, this type of approach has good detection precision for various copy attacks, but the discrimination is less satisfactory.

Manifold learning methods usually consider principal component analysis (PCA), linear discriminant analysis (LDA), multidimensional scaling (MDS), locally linear embedding (LLE), or locality preserving projection (LPP), to study the data distribution and preserve local similarity structures from high dimensional space. For example, Kang *et al.* [24] combined two classical manifold learning approaches, LDA and LPP, to remain the global label information and local structure information for achieving good retrieval accuracy. Zhu *et al.* [27] learned hash function in PCA domain, and took the probability of all data points to preserve the local neighbourhood. This type of approach always focuses on the local structural and ignores the effect of global features, which makes it hard to reach a good robustness and discrimination to copy attacks.

Compared with the above two kinds of approaches, the most advantage of frequency domain hashing is that it can freely separate different frequency coefficients to represent image features, and analyze the distribution, statistical moments or histogram shape to construct hash codes by discrete cosine transform (DCT), discrete wavelet transform (DWT), and discrete Fourier transform (DFT) or their corresponding extensions.

For example, in wavelet domain, low-frequency coefficients represent the global structure, while high-frequency coefficients reveal local contours. The extraction of different coefficients is actually a combination of global and local to achieve a better balance. Venkatesanet *et al.* [28] proposed the definition of perceptual hashing and used randomized signal processing strategies for a non-reversible compression of images into random binary strings in DWT domain. This method has good robustness to compression, noise, but it is sensitive to gamma correction, contrast adjustment and image rotation. Lin and Chang [30] designed an image authentication system with robust hashing, which was based on the invariant relation of DCT coefficients at the same position in each block. This method can distinguish JPEG compression from malicious attacks well. Lv and Wang [34] studied the application of fast Johnson-Lindenstrauss transform (FJLT) in hashing system. But overall, methods for better balance and detection accuracy still need to be continuously studied.

After the above discussion and summary, in this paper, we conduct some research works on perceptual image hashing for copy detection from two perspectives. One is to simultaneously preserve two kinds of complementary global features and local features to reflect the intrinsic attributes of an image. The other is to explore the invariant vector distance of dominant coefficients with the help of the distribution characteristics in DCT domain. Consequently, our research is feasible to aim at achieving better balance between robustness and discrimination, as well as higher copy detection accuracy with less time and storage consumption.

III. The Proposed Method

The framework of our proposed image hashing for copy detection is presented in Fig. 2. It includes three components: content-based feature extraction, hash generation, and similarity calculation, corresponding to different colors for illustration. Given a query image, some joint operations are performed for stable feature extraction, including bilinear interpolation, Gaussian filtering and color space conversion, which are very fundamental techniques for image preprocessing and always used by many hashing algorithms [8], [21], [25], [30]. Then, global texture features by GLCM and local distance features in DCT domain are separately extracted, which will be described in Section III-A. Furthermore, the content-based features are quantized and converted into hash codes according to the predetermined regulation in Section III-B. Finally, by comparing the similarity of a pair of hash codes between a query image and test image in database, the copy versions of a query image are detected, as presented in Section III-C.

A. Content-Based Feature Extraction

To alleviate effects of content-based digital attacks on images, three operations are jointly exploited to produce a normalized image for feature extraction in preprocessing. The original image is resized to a standard size with $B \times B$ by bilinear interpolation so as to ensure that different images have the same hash length, as well as achieve the primary correction for those geometric distortions between pixels. Then, to decrease the influence of slight noise and interpolation errors on image quality, we utilize Gaussian low-pass filtering to smooth the standard image. The convolution mask of Gaussian low-pass filtering can be defined

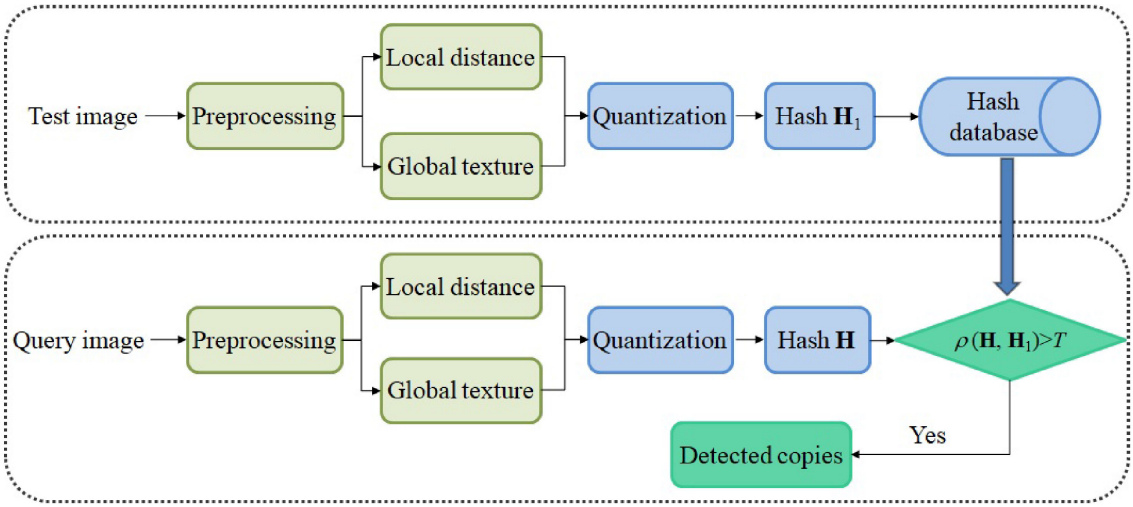

Fig. 2. An overview of the proposed image hashing for copy detection. It includes three components: content-based feature extraction, hash generation, and similarity calculation, corresponding to different colors for illustration.

as follows:

$$T_G = \frac{T^{(1)}(i,j)}{\sum_i \sum_j T^{(1)}(i,j)} \tag{1}$$

where $T^{(1)}$ is calculated by the following equation:

$$T^{(1)}(i,j) = e^{\frac{-(i^2+j^2)}{2\sigma^2}} \tag{2}$$

and σ is a given standard deviation in Gaussian distribution. i and j represent the index of the elements in the convolution mask. For example, if the mask size is 3×3, $-1 \leq i \leq 1$ and $-1 \leq j \leq 1$.

Furthermore, due to high correlation among three color components in RGB color space, the stability and robustness of feature extracted from RGB components are easily affected. Meanwhile, it is known that the HSI color space [36] is based on human visual system, which can be described by a conical space model. This model is quite complicated, but it can clearly show the changes in hue, saturation, and intensity. Hue and saturation are commonly referred to as chroma, and human vision is much more sensitive to intensity than to chroma. In order to facilitate color processing and recognition, HSI color space is often taken into account, which is more in line with human visual perception than RGB color space. So, we convert an image from RGB color system to HSI color system, and the detailed transformation rule is as follows:

$$H = \begin{cases} \theta & if\ B \leq G \\ 360 - \theta & if\ B > G \end{cases} \tag{3}$$

with

$$\theta = \cos^{-1}\left\{ \frac{[(R-G)+(R-B)]}{2[(R-G)^2+(R-B)(G-B)]^{1/2}} \right\} \tag{4}$$

The saturation component is given by:

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)] \tag{5}$$

in which R, G, and B are red, green, and blue components of each image pixel, respectively. Finally, the intensity component is calculated by:

$$I = \frac{1}{3}(R+G+B) \tag{6}$$

In our method, we choose intensity component in HSI color space to extract robust feature for generating hash codes. This is because of the following considerations. As we know in theory, HSI color space is quite closer to human visual perception. On the other hand, we compare the balance performance between robustness and discrimination in different color spaces, as proved in Section IV-F. The results show that HSI color space has the best performance. Finally, after the image preprocessing, two types of features, i.e., global texture feature and local vector distance feature, will be extracted from the normalized intensity image, and the description is elaborated as follows.

1) Global Texture Feature: Texture is a robust global visual feature that reflects the homogeneity in an image, which is the permutation property of surface structures with slow or periodic changes on the surface of an image. Specifically, texture is represented through a set of pixels with a certain gray distribution and spatial structure, as well as gray-level co-occurrence matrix (GLCM) [37] is a common approach to describe texture by studying the spatial correlation of pixels. The calculation of GLCM involves two aspects of pixel distance and orientation, which demonstrates the comprehensive information in orientation, interval, and amplitude of variation through counting how often pairs of pixels with specific values and spatial relationship occur in an image.

In addition, 14 parameters in GLCM are defined to analyze texture, and only 4 parameters among them are irrelevant that can give a higher classification accuracy with less time cost. Therefore, we determine to describe the global texture feature by these 4 parameters, i.e., contrast, correlation, energy, and homogeneity. Contrast reflects the intensity change between a

pixel and its neighbour over the whole image. Correlation is a similarity measure between a pixel and its neighbour in a row or column direction, and the range is between -1 and 1. Energy is the quadratic sum of each element in GLCM, which represents the stability of gray level change of image texture, and reflects the uniformity of gray level distribution and texture roughness. Homogeneity evaluates the closeness between the distribution of elements and the GLCM diagonal, as well as indicates local changes in image texture.

Here, for ease of understanding, let $P(x, y|d, \theta)$ denote the number of times that the pixel with value x occurs at d distance and θ orientation to a pixel with value y. When the values of d and θ are determined, $P(x, y|d, \theta)$ is simplified by $P(x, y)$, and the 4 parameters are calculated.

$$contrast = \sum_x \sum_y |x - y|^2 p(x, y) \tag{7}$$

$$correlation = \sum_x \sum_y \frac{(x - u_x)(y - u_y)p(x, y)}{\sigma_x \sigma_y} \tag{8}$$

$$energy = \sum_x \sum_y p(x, y)^2 \tag{9}$$

$$homogeneity = \sum_x \sum_y \frac{1}{1 + |x - y|^2} p(x, y) \tag{10}$$

And through the experiment comparison in a famous method [38], $d = 1$ and $\theta = 0°, 45°, 90°, 135°$ are suitable for keeping a trade-off between accuracy and complexity. Therefore, we also exploit these parameters to compute image texture information for the intensity component. In each orientation, four features are calculated so as to generate global texture features $\mathbf{T}$ with the length of 16.

2) Local Distance Feature: It is worth noting that texture is a global robust feature extracted from the surface structure of an image, but it can not fully reflect the essential attributes of an image. On the contrary, DCT is a lossless orthogonal transformation under the condition of minimum mean square deviation. It has the characteristics of removing information redundancy, fast computing speed and high precision. More importantly, 2D-DCT is capable of concentrating the energy of an image into different frequency, and its low-frequency coefficients are mainly concentrated in the upper left corner, reflecting the structural information of the image. And the human visual system also indicates that it is more sensitive to the structure information, that is, the low-frequency coefficients can affect the visual perception of an image. Therefore, we are committed to extracting the intrinsic features of an image in DCT domain. In addition, those low-frequency DCT coefficients in the first row/column can fully indicate the changes of pixels in each local sub-image. The corresponding bases in the first row/column are the most basic elements, and DCT coefficients at other locations can be calculated by weighting the sequences of these two bases. Thus, selecting those DCT coefficients in the first row/column to construct vector features could guarantee the characteristics of low-frequency coefficients.

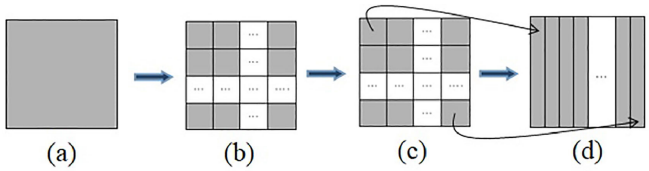

Fig. 3. The process of generating feature matrix. From (a) to (b), an intensity component is divided into non-overlapping sub-images, each sub-image is performed on 2D-DCT from (b) to (c), and low-frequency coefficients in the first row/column are extracted to construct feature matrix from (c) to (d).

Specifically, the intensity component is divided into non-overlapping sub-images sized $b \times b$. The number of sub-images is $N = (B/b)^2$, where B represent the size of normalized image. We assume that $\mathbf{B}_i$ is the i-th sub-image indexed from left to right and from top to bottom $(1 \leq i \leq N)$, and $B_i(j, k)$ indicates the pixel in the $(j + 1)$-th row and $(k + 1)$-th column of $\mathbf{B}_i$. In detail, the 2D-DCT is applied to sub-image $\mathbf{B}_i$, and the low-frequency coefficients in the first row/column are extracted to construct vector feature. The coefficients in first row $C_i(0, v)$ and first column $C_i(u, 0)$ are given with the following equations.

$$C_i(0, v) = \frac{\sqrt{2}}{b} \sum_{j=0}^{b-1} \sum_{k=0}^{b-1} B_i(j, k) \cos\left[\frac{(2\,k + 1)v\pi}{2b}\right] \tag{11}$$

$$C_i(u, 0) = \frac{\sqrt{2}}{b} \sum_{j=0}^{b-1} \sum_{k=0}^{b-1} B_i(j, k) \cos\left[\frac{(2j + 1)u\pi}{2b}\right] \tag{12}$$

Since the coefficients at the end of $C_i(0, v)$ and $C_i(u, 0)$ are vulnerable to compression and noise, we only select the stable low-frequency coefficients from the second element to the $(n + 1)$-th element to construct a vector $\mathbf{Q}_i = [C_i(0, 1), C_i(0, 2), \cdots, C_i(0, n), C_i(1, 0), C_i(2, 0), \cdots, C_i(n, 0)]^T$, where $n = b/2$. So, a feature matrix based on DCT low-frequency is obtained and denoted as $\mathbf{Q}$ with the size of $b \times N$, as well as the process of generating feature matrix is shown in Fig. 3.

$$\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_N] \tag{13}$$

Next, data normalization [26] is applied to every row of $\mathbf{Q}$, and $\mathbf{U}$ is adopted to denote the normalization result of $\mathbf{Q}$. Meanwhile, a reference vector $\mathbf{U}_0 = [u_0(1), u_0(2), \ldots, u_0(b)]^T$ is utilized for calculating vector distance, where $u_0(l)$ $(1 \leq l \leq b)$ is the average value calculated from every row of $\mathbf{U}$. Let $\mathbf{U}_i = [u_i(1), u_i(2), \ldots, u_i(b)]^T$ be the i-th column of $\mathbf{U}$. The Euclidean distance $d(i)$ between $\mathbf{U}_i$ and $\mathbf{U}_0$ is calculated by:

$$d(i) = \sqrt{\sum_{l=1}^{b} (u_i(l) - u_0(l))^2} \tag{14}$$

As a result, the vector distance between DCT low-frequency coefficients is obtained to improve the robustness and discrimination, shown as $\mathbf{D} = [d(1), d(2), \ldots, d(N)]$, which will be proved in experiment that the vector distance is basically the same under content-preserving copy attacks.

B. Hash Generation

Two kinds of complementary features, i.e., global robust texture feature in GLCM, and local distance feature in low-frequency DCT coefficients, have been extracted from the intensity component, respectively. Thus, our hash codes can be generated with the texture feature $\mathbf{T}$ and the vector distance feature $\mathbf{D}$, described as $\mathbf{F} = [\mathbf{TD}]$, where the number of elements in $\mathbf{F}$ is (16+N). We assume that the standard image size in bilinear interpolation is 512×512, and the sub-image is 64×64 for indicating $N = (512/64)^2 = 64$. Obviously, as has been explained, the number of elements in $\mathbf{F}$ is 80.

Next, in order to increase the security of our method, a pseudo-random secret stream generated with a secret key is deployed on $\mathbf{F}$ for constructing the security hash. Specifically, we first take a secret key as the seed of a random generator to produce a pseudo-random secret stream with the length of 80. Then, the elements in this pseudo-random secret stream are sorted, and the vector $\mathbf{Z}$ is used to recode the original position for the sorted numbers. Finally, every element in final hash $\mathbf{H}$ is generated by the following equation.

$$h(q) = f(z(q)) \tag{15}$$

where $h(q)$ and $z(q)$ are the q-th elements in $\mathbf{H}$ and $\mathbf{Z}$ ($1 \leq q \leq 80$), respectively. Obviously, the hash length determines the number of vector permutations, that is, $80! = 7.16 \times 10^{118}$, which means that it is almost impossible to conduct successful guess of our image hash codes without the knowledge of correct secret key. Meanwhile, secret key sensitivity analysis will be validated in experiments to illustrate that the proposed hashing is key sensitivity and sufficient security.

Finally, since the decimal requires many bits for storage, encrypted hash elements are quantized to integers through the rounding operation. For easy description, we still utilize $\mathbf{H}$ to represent the final hash codes, and the image hash $\mathbf{H}$ is available as follows.

$$\mathbf{H} = [h(1), h(2), \ldots, h(80)] \tag{16}$$

C. Similarity Calculation

In fact, copy images are detected by comparing the similarity between a pair of hash codes. In other words, given a query image, it will be first encoded into a fixed-length hash sequence $\mathbf{H}$, and then the similarity between the hash $\mathbf{H}$ and the hashes in index database is calculated according to the following equation. If the similarity between them is more than the threshold T, an image corresponding to the hash sequence in index database is considered to be a copy version of the query, otherwise it is a non-copy version. Altogether, the procedures are illustrated in Algorithm 1.

$$\rho(\mathbf{H}, \mathbf{H}_1) = \frac{\sum_q^{80} [h(q) - u][h_1(q) - u_1]}{\sqrt{\sum_q^{80} [h(q) - u]^2 \sum_q^{80} [h_1(q) - u_1]^2} + \Delta s} \tag{17}$$

in which $h(q)$ and $h_1(q)$ are the q-th elements in $\mathbf{H}$ and $\mathbf{H}_1$, and u and u_1 are the means of $\mathbf{H}$ and $\mathbf{H}_1$, respectively. Δs is a small constant to avoid zero denominator. The larger of $\rho(\mathbf{H}, \mathbf{H}_1)$, the higher precision that correctly detects the copy images, and vice versa.

Algorithm 1: The Detection of Copy Images Through Perceptual Hashing

Input: A query image $\mathbf{I}$;
b: the size of each sub-image in $\mathbf{I}$
Output: The detected copy images
1: Conduct preprocessing to generate $\mathbf{X} \in \mathbf{R}^{B \times B}$;
2: Extract the global robust texture feature $\mathbf{T}$ in GLCM;
3: Calculate the local vector distance feature $\mathbf{D}$;
- Perform the non-overlapping sub-image on $\mathbf{X}$
- Generate feature matrix through DCT coefficients
- Calculate the local distance between vectors

4: Conduct quantization to get the final hash codes $\mathbf{H}$ with features $\mathbf{T}$ and $\mathbf{D}$;
5: Detect the copy images by Eq. (17)
if $\rho(\mathbf{H}, \mathbf{H}_1) > Threshold$
return the corresponding copy image
else
return non-copy response

IV. Experiments and Discussions

A. Experimental Settings and Datasets

In our proposed method, an image is resized to 512×512 and the size of sub-image is 64×64, which makes the total number of sub-images is 64. Then, we select the low-frequency DCT coefficients from the second element to the 33-th element in the first row/ column of every sub-image to construct feature matrix, i.e., $B = 512, b = 64, N = (B/b)^2 = 64, n = b/2 = 32$, so the hash length in integer form is 80. In addition, Eq. (17), i.e., correlation coefficient, is utilized to calculate the similarity between two hash codes, and a higher correlation coefficient indicates that the corresponding images have more similar visual content.

All the experiments are conducted with MATLAB 2016a, running in a desktop PC with Intel (R) Core (TM) i7-7700 CPU @3.6 GHz and 8.0 GB RAM. The comprehensive experiments are performed on three datasets, i.e., USC-SIPI database [39], Copydays Database [40], and UCID [41], which have been widely used in image hashing and copy detection. Moreover, many state-of-the-art methods [4], [10], [21] reported in famous journals are also experimented on these databases, which encourages us to use them to verify whether the algorithm is effective. The experimental results on these three databases are unified through ROC curves for overall performance evaluation.

USC-SIPI database: Four standard benchmark images sized 512×512 are selected from USC-SIPI database to verify the rationality of our proposed hashing algorithm, and they are Airplane, Baboon, House, and Lena. Specifically, we use some digital attack tools, such as, Matlab, Photoshop, and StirMark, to generate the content-preserving visually similar copy images. The adopted content-preserving operations are shown in Table I, and clearly a total of 74 different content-preserving operations

TABLE I
CONTENT-PRESERVING OPERATIONS AND PARAMETERS

Operation	Parameter value	Num
JPEG compression	30, 40,...,100	8
Watermarking	10, 20,...,100	10
Speckle noise	0.001, 0.002,...,0.01	10
Salt and Pepper noise	0.001, 0.002,...,0.01	10
Brightness adjustment	± 10, ± 20	4
Contrast adjustment	± 10, ± 20	4
Gamma correction	0.7, 0.9, 1.1, 1.2	4
Gaussian filtering	0.3, 0.4,...,1.0	8
Image scaling	0.5,0.75,0.9,1.1,1.5,2.0	6
Rotation, cropping, rescaling	± 1, ± 2, ± 3, ± 4, ± 5	10
Total number		74

are performed. Specially, for the operation of rotation, cropping and re-scaling listed in Table I, each original image is firstly rotated, the rotated version is then cropped to remove those padded pixels introduced by rotation, and the cropped version is finally rescaled to the same size as the original image. In addition, for the parameters of this operation, "+" indicates the direction of anticlockwise rotation, and "−" represents the direction of clockwise rotation.

Copydays Database: It consists of 157 images with various resolutions from 1200×1600 to 3008×2000. The content-preserving operations shown in Table I are taken to get the visually similar images, i.e., $157 \times 74 = 11618$ pairs of visually similar images, which are used to analyze the robustness.

UCID database: It is taken to validate the discrimination of our hashing method, which provides 1338 different color images with the resolution 512×384 or 384×512. In other words, $1338 \times (1338 - 1)/2 = 894453$ pairs of visually different images are obtained for discrimination analysis.

B. Performance Analysis of Perceptual Robustness

It may be interested to know how much the proposed method contributes to perceptual robustness. To answer this question, we carry out experiments on the USC-SIPI database, Copydays database and its visually similar images. Specifically, we first extract hash codes of four standard benchmark images in USC-SIPI database and their 296 similar versions, evaluate their similarity with correlation coefficient. In theory, the values of correlation coefficient are supposed to be close to 1 between the standard benchmark images and their visually similar versions.

As has been explained, the correlation coefficient results under different content-preserving operations are presented in Fig. 4, where the x-axis indicates the specific parameter of digital operations and the y-axis represents the corresponding correlation coefficient. From Fig. 4(a)∼(i), it can be clearly observed that the correlation coefficients of our method are almost all close to 1, which indicates our hashing could better detect visually similar images, in other words, we have achieved quite better robustness against to copy attacks. Besides, Fig. 4(j) is the result of rotation operations, in which the correlation coefficients are relatively smaller than other operations. The main

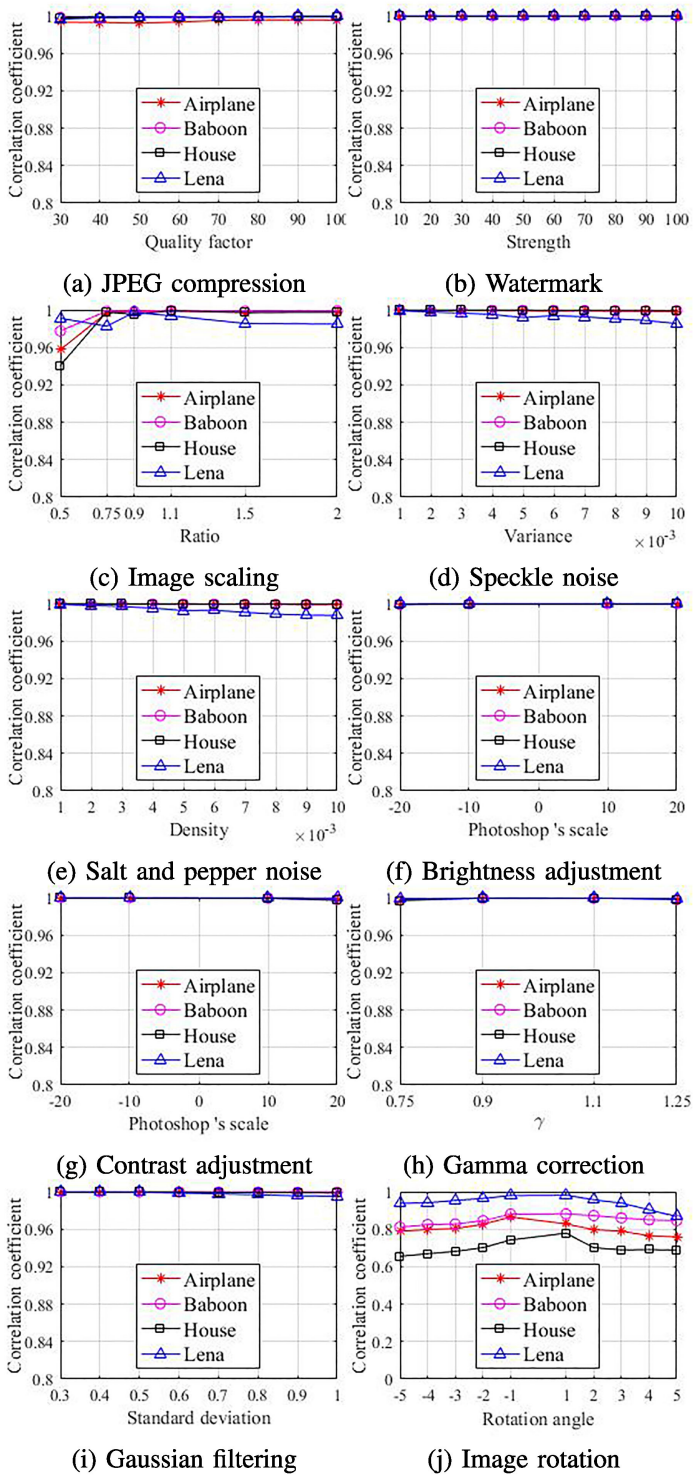

Fig. 4. Correlation coefficient results under specific digital operations based on four standard benchmark images.

reason for this phenomenon is that those images used to calculate Fig. 4(j) actually undergo three kinds of manipulations, i.e., rotation, cropping and rescaling. The correlation coefficients of multiple operations are inevitably smaller than those of single operations. In spite of this, our hashing method is still robust to most digital operations under the appropriate threshold.

TABLE II
STATISTICS OF CORRELATION COEFFICIENT UNDER DIFFERENT OPERATIONS

Operation	Maximum	Minimum	Mean
JPEG compression	1	0.9690	0.9983
Watermark embedding	1	0.9612	0.9983
Image scaling	1	0.8967	0.9978
Speckle noise	1	0.9455	0.9984
Salt and Pepper noise	1	0.9541	0.9988
Brightness adjustment	1	0.9603	0.9975
Contrast adjustment	1	0.9685	0.9987
Gamma correction	1	0.9514	0.9964
Gaussian filtering	1	0.9707	0.9991
Image rotation	0.9960	0.4052	0.9269

More importantly, in Fig. 4, each curve in every graph does not fluctuate greatly. For instance, it is clear that there is no obvious jump on each curve at different rotation angle in Fig. 4(j). In other words, different parameters of one digital attack will not seriously affect the values of correlation coefficient, which illustrates that the extracted feature in our proposed method is enough robust, and also proves that the design theory of our proposed method is reasonable. That is, the vector distance between low-frequency DCT coefficients is basically invariant against to content-preserving copy operations, and further global texture features provide the complementary information for local invariant distance features, which greatly benefits to improve the balance between robustness and discrimination.

In addition, in order to better validate the perceptual robustness, a large open image database named Copydays is used. Specifically, we take the content-preserving copy operations shown in Table I to obtain their corresponding visually similar versions, i.e., $157 \times 74 = 11618$ pairs of visually similar images. Then, 11618 pairs of hash codes are extracted through our proposed algorithm. Finally, for each pair of hash codes, the correlation coefficient is calculated, and totally 11618 results are gotten. Moreover, the statistical information (maximum, minimum, and mean) of these correlation coefficients under different digital operations are computed to facilitate more intuitive analysis of the robustness. The statistics of correlation coefficient under different operation are presented in Table II. We could easily conduct some meaningful results from Table II. The means of all 11618 correlation coefficients are much larger than 0.9269, and the maximums are almost close to 1 except for image rotation. We also know that when the correlation coefficient is selected as the hashing similarity measure, if the correlation coefficient is larger than the threshold T, the two images corresponding to the hash values are considered to be visually similar. This indicates that, in the 11618 similar image, the threshold $T = 0.92$, $(11618 - 532)/(11618) \times 100\% = 95.42\%$ pairs of visually similar images can be correctly detected. In addition, if there are no rotated images in the applications, the threshold $T = 0.92$, 99.98% pairs of visually similar images can be correctly identified. All these results illustrate that our hashing could better perceive visually similar copy images and achieve quite better robustness.

C. Performance Analysis of Discrimination

We evaluate the discrimination performance against to different images through an open image database named UCID. Meanwhile, Fig. 5 provides us some sample images from UCID, and it is clear that UCID contains many different images, as well as several near-duplicate images in the same scene. These near-duplicate images are considered to be different images and captured by various conditions, which may be more visually similar with an original image than some copy versions generated by strong attacks, and inevitably increases the difficulty of identifying different image. Therefore, using UCID to analyze the discrimination will make the experimental results more convincing. This is also a challenge to detect whether an algorithm can distinguish the near-duplicate image from the content-based copy images.

In detail, we first extract the hash codes corresponding to 1338 images, calculate the correlation coefficient between each pair of hash codes, and finally obtain 894453 results. Furthermore, the minimum, maximum, and mean of these correlation coefficients are computed, which are -0.3994, 0.9597, and 0.4215, respectively. The mean value is 0.4215 that is much smaller than those of similar images (the minimum mean is 0.9269 shown in Table II), which well indicates that our method can effectively recognize the different images and has good discriminative capability with satisfactory robustness.

In order to quantify the robustness and discrimination of our proposed algorithm on content-preserving copy images and different images, we introduce two criteria, true positive rate (P_{TPR}) and false positive rate (P_{FPR}). P_{TPR} represents that visual similar images are correctly identified, which is equivalent to robustness. The larger the P_{TPR}, the better the robustness. P_{FPR} indicates that different images are wrongly identified as similar images, which is equal to discrimination. The smaller P_{FPR}, the better discrimination.

$$P_{TPR}(\rho \geq T) = \frac{n_1}{N_1} \tag{18}$$

$$P_{FPR}(\rho \geq T) = \frac{n_2}{N_2} \tag{19}$$

In Eq. (18), n_1 is the number of pairs of visually similar images that are correctly detected, and N_1 means the total pairs of visually similar image (i.e., $N_1 = 11618$). In Eq. (19), n_2 is the number of pairs of different image mistakenly detected as similar image, and N_2 means the total pairs of different image (i.e., $N_2 = 894453$). T indicates the threshold that is utilized to identify images. As explained above, P_{TPR} and P_{FPR} under different thresholds are computed and presented in Table III. Obviously, for correlation coefficient as the similarity metric, when the threshold decreases, the number of similar images correctly recognized increases that means good robustness. At the same time, the number of different images misjudged as similar images increases, which makes P_{FPR} increases. But, it does not

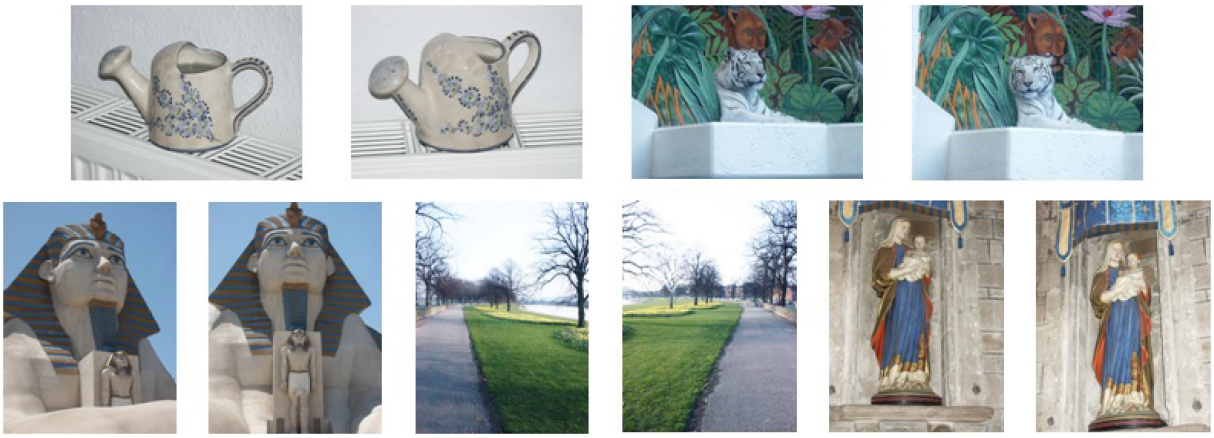

Fig. 5. Some sample images from UCID for discrimination validation. It is obvious that UCID contains many different images, as well as several near-duplicate images in the same scene.

TABLE III
P_{FPR} AND P_{TPR} UNDER DIFFERENT THRESHOLDS

Threshold	P_{FPR} (UCID)	P_{TPR} (Copydays database)
T=0.96	0	91.19%
T=0.95	2.24×10^{-6}	92.41%
T=0.94	3.35×10^{-6}	93.52%
T=0.93	4.47×10^{-6}	94.53%
T=0.92	5.59×10^{-6}	95.42%
T=0.91	8.94×10^{-6}	96.11%
T=0.90	1.34×10^{-5}	96.81%

indicate that our proposed algorithm can not achieve a good balance between robustness and discrimination. In fact, $T = 0.90$, P_{TPR} is 96.81%, and P_{FPR} is only 1.34×10^{-5}. In other words, only one image is detected by error in 10^5 images. Therefore, we can argue that our method is able to distinguish similar copy images from different images, as well as near-duplicate images.

D. Effect of the Dominant Parameters

We discuss the effect of different sub-image sizes on hashing performance through the Receiver Operating Characteristic (ROC) [42]. The ROC curve is formed by a set of points (P_{FPR}, P_{TPR}) calculated with varying threshold, and is taken to evaluate balance between robustness and discrimination. Clearly, the x-axis is considered as P_{FPR} and the y-axis is defined as the P_{TPR} in ROC diagram, and the ROC curve near the top-left corner implies a small P_{FPR} and a big P_{TPR}, i.e., better balance between robustness and discrimination than those curves far away from the top-left corner.

Specifically, only sub-image size is changed under the condition of remaining other parameters invariant, then the utilized sub-image sizes are 16×16, 32×32, 64×64, and 128×128, respectively. On the basis of 11618 similar correlation coefficients and 89443 different correlation coefficients, we design different thresholds to get the final ROC curves, as shown in Fig. 6. The four curves of different sub-image sizes are relatively close to the top-left corner, which demonstrates that our method has good generalization ability and stability to a certain extent. Moreover, the ROC curves near the top-left corner are enlarged and located at the right-bottom part in Fig. 6. It is more clear that the balance between robustness and discrimination of sub-image size 64×64 is better than those of others. Therefore, 64×64 is a moderate size for reaching better balance between perceptual robustness and discrimination.

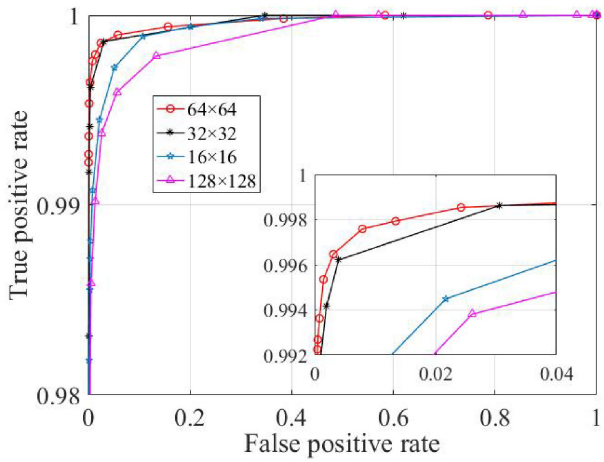

Fig. 6. ROC curves comparison under different sub-image sizes.

TABLE IV
PERFORMANCE COMPARISONS UNDER DIFFERENT SUB-IMAGE SIZES

Sub-image size	AUC	Hash length	Time (s)
16×16	0.99936	1040 digits	0.163
32×32	0.99953	272 digits	0.115
64×64	0.99970	80 digits	0.103
128×128	0.99889	32 digits	0.096

In addition, the hash length and the average running time are closely related to sub-image size, and the overall summaries are

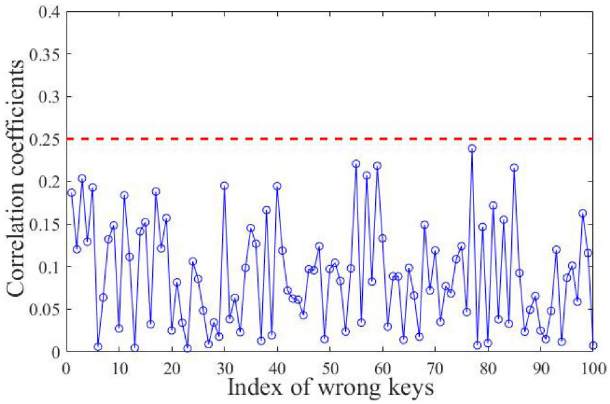

Fig. 7. Correlation coefficients between the correct key and other 100 wrong keys.

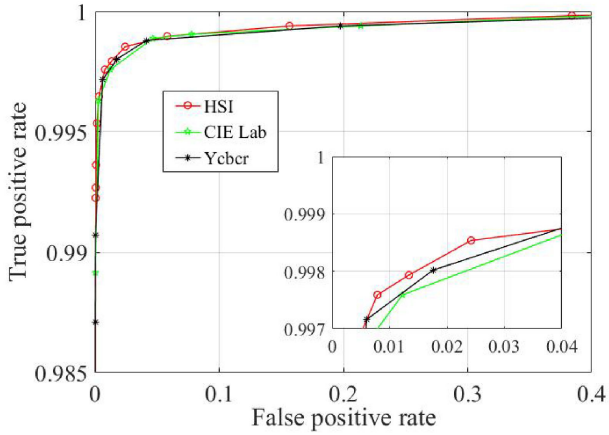

Fig. 8. ROC curves comparison among different color spaces.

presented in Table IV. In Table IV, the AUC is a quantitative index of evaluating balance performance, which is the area under the ROC curve. The range of AUC is [0, 1], and a larger AUC represents better balance between perceptual robustness and discrimination. Meanwhile, the average running time indicates the time to compute one image hash codes on the device described in Section IV-A. From Table IV, it is not difficult to find that the AUC value is consistent with the ROC result and achieves best at 64×64 sub-image size. Secondly, with the increase of sub-image, the total number of sub-image gradually reduces, i.e., the hash length gradually reduces. At last, the average running time decreases slightly with the increase of sub-image size, but almost all of them are about 0.1 seconds.

E. Secret Key Sensitivity Analysis

The secret key sensitivity indicates that the similarity between hash codes generated by the different key of one image should be small enough. That is, the proposed method has a secret key dependency to make sure sufficient security. In our proposed method, key dependency refers to the correlation coefficient between hash codes corresponding to different keys is quite small. Specifically, we take the four standard benchmark images (Airplane, Baboon, House, and Lena) as test images, and calculate the correlation coefficients between hash codes extracted with different keys of one image. For space limitation, a typical example is presented here. We first use our correct key to generate hash code of Airplane, and then select 100 wrong keys to generate 100 different hash codes, while all other parameters are kept unchanged. The correlation coefficients between the first hash and other 100 hash codes are calculated, and the results are presented in Fig. 7.

In Fig. 7, the x-axis is the index of wrong keys and y-axis is the correlation coefficient. It is clear that maximum correlation coefficient is less than 0.25. It should be noted that the minimum correlation coefficient of similar images is 0.4052 in Table II. If threshold $T = 0.25$, all similar images will be correctly detected, as well as no different images are misjudged, i.e., $P_{\mathrm{TPR}} = 1$, $P_{\mathrm{FPR}} = 0$, which achieves the optimal state. This illustrates that our proposed algorithm has a very good secret key dependency.

F. Different Color Space Analysis

In the field of digital image processing, the conversions between different color spaces have been widely used. Especially for the content-based feature extraction, different color spaces have a certain impact on algorithm performance. Therefore, to elucidate HSI color space is the best choice to extract image feature for generating hash codes in our proposed method, other two typical color spaces are leveraged to design the hashing algorithm, i.e., CIE Lab color space based hashing, and YCbCr color space based hashing.

In detail, the $\mathbf{I}$ component in HSI color space, $\mathbf{L}$ component in CIE Lab color space, and $\mathbf{Y}$ component in YCbCr color space, are extracted as the intensity image, respectively. Then, three hashing algorithms based on different color spaces are obtained while all parameters remain unchanged, which are HSI-based hashing, CIE Lab-based hashing, and YCbCr-based hashing, respectively. Finally, we compare their balance performance with respect to robustness and discrimination based on the similar copy image pairs and different image pairs, and the comparison results are evaluated through the ROC curves. The ROC curves are shown in Fig. 8. It is clear that the ROC curve of HSI color space is quite closer to the top-left corner than that of other color space. Meanwhile, the AUC score of HSI-based hashing, CIE Lab-based hashing, and YCbCr-based hashing are 0.99970, 0.99958, and 0.99965, respectively. The experimental results confirm that HSI color space is more suitable for our proposed method, and closer to people's perception for distinguishing the content-based copy images from different images.

G. Performance Comparison Among Different Methods

In this section, we evaluate our hashing performance among different methods, and the comparison method are DCT-LLE [43], CVA-Canny [45], GF-LVQ [44], MDS [26], and SVD-CSLBP [46], respectively. It is worth noting that we are not blindly looking for these algorithms for comparison. First of all, these algorithms comprehensively include the major hashing technologies at present, such as matrix decomposition [45], [46], local manifold learning [26], [43], frequency coefficients [43], [44], which is very comparable to our method because we also

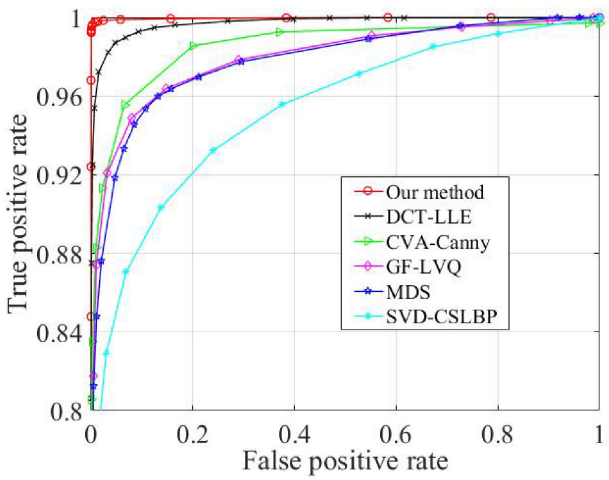

Fig. 9. ROC curves comparison among different methods.

TABLE V
PERFORMANCE COMPARISONS AMONG DIFFERENT HASHING ALGORITHMS

Performance	AUC	Hash length	Time
SVD-CSLBP	0.95268	64 floats	0.123s
MDS	0.97605	900 bits	0.221s
GF-LVQ	0.97939	120 bits	0.273s
CVA-Canny	0.98537	400 bits	0.084s
DCT-LLE	0.99712	64 bits	0.065s
Our hashing	0.99970	720 bits	0.103s

consider frequency coefficients features to design hashing. Secondly, these algorithms have paid attention to extracting the global features [26], [44], [45], and local features [46] features separately, or combining them together [43], and it is useful to verify that global and local features can provide complementary information to achieve better performance.

In the following experiments, the visually similar image pairs and different image pairs are adopted again to compare the balance performance with respect to robustness and discrimination. First, every compared algorithm is performed to generate hash codes, and the corresponding metric is utilized to calculate the similarity between a pair of hash codes. Then, a set of thresholds are chosen to compute the P_{FPR} and P_{TPR}, as well as the ROC curve is formed by a set of points (P_{FPR}, P_{TPR}). Finally, the ROC curves comparison among different methods are presented in Fig. 9. It can be seen from the results that our ROC curve is significantly close to the top-left corner those of compared hashing algorithms, which intuitively illustrates that our algorithm achieves better balance than the compared algorithms. In addition, it can be observed that DCT-LLE method is inferior to ours but better than others, which greatly proves that global and local feature could provide complementary information to each other for better performance, as well as demonstrates that design theory of our algorithm is reasonable.

Meanwhile, the average running time of every algorithm is recorded, and is implemented on the same device. Therefore, the overall comparison are concluded in Table V. In terms of AUC score, our method is significantly larger than those of other method that achieves best balance performance between robustness and discrimination. In terms of average running time, our method is 0.103 seconds, slightly slower than CVA-Canny and DCT-LLE, but it does not influence the requirements of large-scale image processing in practice. In terms of hash length, for our method, CVA-Canny, and MDS, the hash elements of 1338 images are adopted to analyze the hash length in binary form. Specifically, maximum hash elements of our method, CVA-Canny, and MDS are 300, 689, and 24, respectively. Then, the minimum number of binary bits required to represent one hash element are 9 bits, 10 bits, and 5 bits, respectively. Therefore, the hash length in binary form of our method, CVA-Canny, and MDS are 720 bits, 400 bits, and 900 bits. In fact, as to the average running time and hash length, our hashing has moderate performance.

H. Performance in Copy Detection

In this section, we analyze the precision performance of our method through the image database collected from [47]. This image database is comprised of 100 images sized 256×384 or 384×256 in each of 10 categories (Africa, Beaches, Building, Bus, Dinosaur, Elephant, Flower, Horses, Mountain, and Food). Each category consists of various near-duplicates similar images obtained by different conditions. First, we randomly select one image from each category so that get a query database. Then, each image is attacked by 16 strong copy operations, 160 copy images are available. Finally, the test image database with 1160 images are constructed. The 16 copy attacks are described as follows:

- Brightness adjustment with the parameter of 50.
- Contrast adjustment with the parameter of 50.
- Gamma correction with the parameter of 1.5.
- Gaussian filtering with the parameter of 0.6.
- Speckle noise with the parameter of 0.05.
- Salt and pepper noise with the parameter of 0.05.
- White noise with the parameter of 0.05.
- Watermarking with the parameter of 100.
- JPEG compression with the parameters of 30 and 50.
- Inserting text with "Copyright in 2019".
- Image scaling with the with the parameters of 50 and 75.
- Image rotation, cropping, rescaling with the parameters of $5°$, $8°$, and $10°$.

To evaluate the copy performance among different algorithms, we adopt the precision-recall curve (P-R curve) to illustrate the objective experimental results. Moreover, under the control of threshold, precision rate and recall rate are calculated as:

$$precision = \frac{number\ of\ correctly\ detected\ copies}{number\ of\ all\ returned\ results} \tag{20}$$

$$recall = \frac{number\ of\ correctly\ detected\ copies}{number\ of\ all\ copies} \tag{21}$$

The P-R curve can be plotted by a set of points (recall, precision), and the larger their values are, the better performance will be. In addition, three experiments are performed to evaluate (1) P-R curves among different comparison algorithms, (2) the

TABLE VI
THE TIMES OF SUCCESSFULLY DETECTED COPIES FOR 16 COPY ATTACKS AMONG COMPARED ALGORITHMS

Copy attack	CVA-Canny	MDS	GF-LVQ	SVD-CSLBP	DCT-LLE	RATR	Our
Brightness adjustment +50	10	9	10	7	10	10	10
Contrast adjustment +50	10	9	9	7	10	10	10
Gamma adjustment +1.5	9	9	6	5	10	9	10
Gaussian filtering +0.6	10	9	10	4	10	10	10
Speckle noise +0.05	6	5	9	1	9	8	10
Salt Pepper noise +0.05	3	5	8	2	7	7	10
White noise +0.05	5	7	10	2	10	10	10
Watermarking +100	9	9	9	8	10	10	10
JPEG compression +30	9	8	9	3	9	9	10
JPEG compression +50	9	9	9	4	10	9	10
Insert text	10	8	10	4	10	6	9
Image scaling +50	10	8	9	3	10	10	10
Image scaling +75	10	6	8	1	10	9	10
Rotation, cropping, rescaling +5°	1	5	0	1	4	9	8
Rotation, cropping, rescaling +8°	0	2	0	0	3	8	5
Rotation, cropping, rescaling +10°	0	1	0	0	0	5	3
Total times	111	109	116	52	132	139	145
(Total percentage)	69.38%	68.13%	72.50%	32.50%	82.50%	86.88%	90.63%
(Rotation percentage)	3.33%	26,67%	0	3.33%	23.33%	73.33%	52.33%

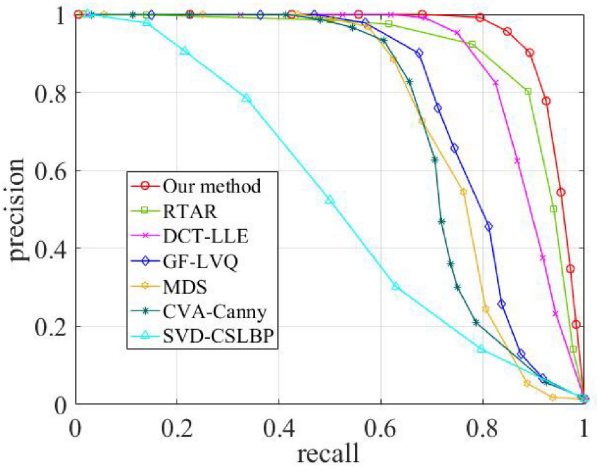

Fig. 10. P-R curves comparison among different methods for copy attacks.

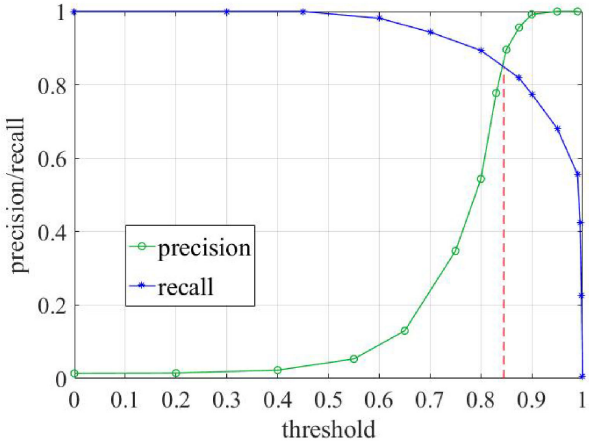

Fig. 11. The recall and precision curves with different threshold.

determination of optimal threshold value, and (3) the number of correctly detected copies.

In the first experiment, for each compared algorithm, we calculate the similarity between query image and test images with their respective metric, and further generate a set of points (recall, precision) to construct the P-R curves, as shown in Fig. 10. From Fig. 10, it is clear that our proposed method achieves the best recall and precision than that of compared algorithms. The best recall and precision indicates that our proposed method could be robust against to strong copy attacks, and well distinguish copy images from near-duplicate similar image.

In the second experiment, we analyze the recall and precision under different threshold values for our copy detection system, and the result is described in Fig. 11. When the threshold varies from 0 to 1, the recall decreases, as well as the precision increases. As a consequence, in the interval [0.8, 0.9], recall and precision are approximately equal under a certain threshold value, and then achieves the best copy performance. We determine this threshold as the optimal threshold, and the optimal threshold in our method is approximately 0.84.

In the third experiment, we observe the correctly detected copies among different methods for each attack. As we know, to count the numbers of correctly detected copies, a certain threshold is needed to judge whether an image is a copy version or not. Therefore, the optimal threshold obtained in the second experiment is available. In particular, for the compared algorithms, their corresponding optimal thresholds are generated by the same treatment.

Specifically, for each copy attack, we count the times of successfully detected copies by using these optimal thresholds, and the results are presented in Table VI. As shown in Table VI, our method presents quite better detection results in each copy attack than the compared method, a total of 160 copy images, and the successfully detected 145 copy images makes the retrieval percentage reach 90.63%. The total numbers (percentage) of CVA-Canny, MDS, GF-LVQ, SVD-CSLBP, DCT-LLE, and RATR [48] are 111 (69.38%), 109 (68.13%), 116 (72.50%), 52 (32.50%), 132 (82.50%), and 139 (86.88%), respectively. It is obvious that our method is much better than SVD-CSLBP, MDS, CVA-Canny, and GF-LVQ, while slightly superior to RATR and DCT-LLE. In addition, we also calculate the correct percentage for rotation attacks separately. Observation demonstrates that our method is second only to RATR with respective to rotation attack, and superior to other algorithms. All these experiments illustrate that our method could provide higher detection accuracy.

V. Conclusion

We have presented an effective perceptual image hashing method based on the combination of global texture feature and local invariant vector distance feature for content-based copy detection. On the one hand, content-based global and local features provided and ensured higher detection precision for a wide range of copy attacks, and on the other hand hash codes instead of feature database reduced storage space and improved time efficiency. Various experiments performed on benchmark database indicated that our proposed perceptual image hashing could provide higher copy detection precision and achieve better balance between robustness and discrimination than the state-of-the-art algorithms. In addition, the key sensitivity analysis illustrated that the introduction of secret key could greatly increase the security with a large enough key space. At the same time, experiments on color space confirmed that the HSI color space is indeed closer to human visual perception and intensity component is independent of the color information, which is beneficial to color processing and recognition.

However, as the hashing algorithm applied in content-based copy detection, the hash length of the proposed algorithm may fail to reach a satisfactory result. More efforts will be attempted to address this problem in the future.

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments.

References

[1] C. I. Podilchuk and E. J. Delp, "Digital watermarking: Algorithms and applications," *IEEE Signal Process. Mag.*, vol. 18, no. 4, pp. 33–46, Jul. 2001.
[2] C. Kim, "Content-based image copy detection," *Signal Process. Image Commun.*, vol. 18, no. 3, pp. 169–184, 2003.
[3] A. Joly, O. Buisson, and C. Frelicot, "Content-based copy retrieval using distortion-based probabilistic similarity search," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 293–306, Feb. 2007.
[4] Z. Zhou, Y. Wang, Q. Wu, C. Yang, and X. Sun, "Effective and efficient global context verification for image copy detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 1, pp. 48–63, Jan. 2017.
[5] J. Hsiao, C. Chen, L. Chien, and M. Chen, "A new approach to image copy detection based on extended feature sets," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2069–2079, Aug. 2007.
[6] H. Ling, H. Cheng, Q. Ma, F. Zou, and W. Yan, "Efficient image copy detection using multiscale fingerprints," *IEEE Multimedia*, vol. 19, no. 1, pp. 60–69, Jan. 2012.
[7] M. Schneider and S. F. Chang, "A robust content based digital signature for image authentication," in *Proc. 3rd IEEE Int. Conf. Image Process.*, 1996, pp. 227–230.
[8] F. Khelif and J. Jiang, "Perceptual image hashing based on virtual watermark detection," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 981–994, Apr. 2010.
[9] X. Lv and Z. J. Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 1081–1093, Jun. 2012.
[10] X. Nie *et al.*, "Robust image fingerprinting based on feature point relationship mining," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 6, pp. 1509–1523, Jun. 2018.
[11] Z. Li, G. Liu, H. Jiang, and X. Qian, "Image copy detection using a robust Gabor texture descriptor," in *Proc. 1st ACM Workshop Large-Scale Multimedia Retrieval Mining*, 2009, pp. 65–72.
[12] L. Kang, C. Hsu, H. Chen, and C. Lu, "Secure SIFT-based sparse representation for image copy detection and recognition," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2010, pp. 1248–1253.
[13] Z. Xu, H. Ling, F. Zou, Z. Lu, and P. Li, "A novel image copy detection scheme based on the local multi-resolution histogram descriptor," *Multimedia Tools Appl.*, vol. 52, nos. 2–3, pp. 445–463, 2011.
[14] H. Ling, L. Wang, F. Zou, and W. Yan, "Fine-search for image copy detection based on local affine-invariant descriptor and spatial dependent matching," *Multimedia Tools Appl.*, vol. 52, nos. 2–3, pp. 551–568, 2011.
[15] K. Yan and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2004, pp. 506–513.
[16] H. Ling, L. Yan, F. Zou, C. Liu, and H. Feng, "Fast image copy detection approach based on local fingerprint defined visual words," *Signal Process.*, vol. 93, no. 8, pp. 2328–2338, 2013.
[17] L. Amsaleg and P. Gros, "Content-based retrieval using local descriptors: Problems and issues from a database perspective," *Pattern Anal. Appl.*, vol. 4, no. 2–3, pp. 108–124, 2001.
[18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vision Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[19] S. Kozat, R. Venkatesan, and M. Mihcak, "Robust perceptual image hashing via matrix invariants," in *Proc. IEEE Int. Conf. Image Process.*, 2004, pp. 3443–3446.
[20] V. Monga and M. Mihcak, "Robust and secure image hashing via nonnegative matrix factorizations," *IEEE Trans. Inf. Forensics Secur.*, vol. 2, no. 3, pp. 376–390, Sep. 2007.
[21] Z. Tang, X. Zhang, and S. Zhang, "Robust perceptual image hashing based on ring partition and NMF," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 711–724, Mar. 2014.
[22] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, "A deep semi-NMF model for learning hidden representations," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1692–1700.
[23] Y. Chen, H. Zhang, X. Zhang, and R. Liu, "Regularized semi-nonnegative matrix factorization for hashing," *Comput. Vision Image Understanding*, vol. 26, no. 6, pp. 1–13, 2017.
[24] Z. Kang, H. Lu, Y. He, and S. Feng, "Locality preserving discriminative hashing," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 1089–1092.
[25] Z. Tang, L. Ruan, C. Qin, X. Zhang, and C. Yu, "Robust image hashing with embedding vector variance of LLE," *Digit. Signal Process.*, vol. 43, no. 6, pp. 17–27, 2015.
[26] Z. Tang, Z. Huang, X. Zhang, and H. Lao, "Robust image hashing with multidimensional scaling," *Signal Process.*, vol. 137, no. 6, pp. 240–250, 2017.
[27] X. Zhu *et al.*, "Graph PCA hashing for similarity search," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 2033–2044, Sep. 2017.
[28] R. Venkatesan, S. Koon, M. Jakubowski, and P. Moulin, "Robust image hashing," in *Proc. IEEE Int. Conf. Image Process.*, 2000, pp. 664–666.
[29] V. Monga and B. Evans, "Perceptual image hashing via feature points: Performance evaluation and tradeoffs," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3452–3465, Nov. 2006.
[30] C. Lin and S. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 153–168, Feb. 2001.

[31] C. Yan, C. Pun, and X. Yuan, "Quaternion-based image hashing for adaptive tampering localization," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 12, pp. 2664–267, Dec. 2016.
[32] F. Ahmed, M. Y. Siyal, and V. U. Abbas, "A secure and robust hash based scheme for image authentication," *Signal Process.*, vol. 90, no. 5, pp. 1456–1470, 2010.
[33] S. Xiang, "Histogram-based perceptual image hashing in the DWT domain," in *Proc. Int. Conf. Multimedia Inf. Netw. Secur.*, 2010, pp. 653–657.
[34] X. Lv and Z. J. Wang, "An extended image hashing concept: Content-based fingerprinting using FJLT," *EURASIP J. Inf. Secur.*, vol. 2009, no. 1, pp. 1–16, 2009.
[35] Z. Tang, F. Yang, L. Huang, and X. Zhang, "Robust image hashing with dominant DCT coefficients," *Optik-Int. J. Light Electron. Opt.*, vol. 125, no. 18, pp. 5102–5107, 2014.
[36] R. C. Gonzalez and R. E. Woods, *Dig. Image Processing.*, 3rd Ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2007.
[37] A. Rampun, H. Strange, and R. Zwiggelaar, "Texture segmentation using different orientations of GLCM features," in *Proc. Int. Conf. Comput. Vision* 2013, pp. 1–8.
[38] R. Haraclick, "Texture features for image classification," *IEEE Trans. Syst., Man Cybern.*, vol. 3, no. 6, pp. 610–621, Nov. 1973.
[39] USC-SIPI Image Database, [Online]. Available: http://sipi.usc.edu/database/. Accessed on: 2018.
[40] H. Jegou *et al.*, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704-1716, Sep. 2012.
[41] G. Schaefer, "UCID: An uncompressed color image database," in *Proc. Storage Retrieval Methods Appl. Multimedia*, 2003, pp. 472–480.
[42] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
[43] Z. Tang, H. Lao, X. Zhang, and K. Liu, "Robust image hashing via DCT and LLE," *Comput. Secur.*, vol. 62, no. 8, pp. 133–148, 2016.
[44] Y. Li, Z. Lu, C. Zhu, and X. Niu, "Robust image hashing based on random Gabor filtering and dithered lattice vector quantization," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1963–1980, Apr. 2012.
[45] Z. Tang, L. Huang, X. Zhang, and H. Lao, "Robust image hashing based on color vector angle and Canny operator," *AEU-Int. J. Electron. Commun.*, vol. 70, no. 6, pp. 833–841, 2016.
[46] R. Davarzani, S. Mozaffari, and K. Yaghmaie, "Perceptual image hashing using center-symmetric local binary patterns," *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4639–4667, 2016.
[47] Z. J Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 9, pp. 947–963, Sep. 2001.
[48] Z. Zhou, C. Yang, B. Chen, and X. Sun, "Effective and efficient image copy detection with resistance to arbitrary rotation," *IEICE Trans. Inf. Syst.*, vol. E99-D, no. 6, pp. 1531–1540, Jun. 2016.

Ziqing Huang received the M.Eng. degree from Guangxi Normal University, Guilin, China, in 2017. She is now pursuing the Ph.D. degree with Computer Science and Technology, Tianjin University, Tianjin, China. Her current research interests include image processing and multimedia security.

Shiguang Liu (Member, IEEE) received the Ph.D. degree from the State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou, China. He is currently a Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His research interests include image/video processing, computer graphics, visualization, and virtual reality.