Perceptual image hashing based on three-dimensional global features and image energy

XIAORAN YUAN[1,2], **YAN ZHAO**[1,2]
[1]Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin, 541004, China
[2]College of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai, 200090, China

Corresponding author: Yan Zhao (yanzhao79@ hotmail.com)

This work was supported by the National Natural Science Foundation of China under Grant 61802250, the Capacity Building Project of Some Local Colleges of Shanghai Science and Technology Commission under Grant 20020500700 and Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security MIMS18-04.

ABSTRACT In order to improve classification performance and operating efficiency of the hash algorithm, this paper proposes a novel hash algorithm that combines three-dimensional global features and local energy features. During the stage of three-dimensional features extraction, the image is firstly compressed by SVD decomposition to form a secondary image. Then the statistical features of the secondary image at the three-dimensional visual angle are extracted as the global features. Finally, the global feature hash is generated by using the relationship between the statistical features of the image layers from different three-dimensional visual angles. In the energy feature extraction stage, the luminance image is divided into blocks, and then the energy value of each image sub-block is obtained. The multi-directional energy change features are taken as the local features of the image. Subsequent experimental results prove the effectiveness of the proposed algorithm. The algorithm not only has good robustness to the conventional content-preserving operations, but also achieves a good balance between discrimination capability and robustness. In addition, compared with several state-of-the-art schemes, this algorithm has the best ROC curve, the shortest running time and the best local tamper detection ability.

INDEX TERMS Image hashing, image energy, three-dimensional global features, tampered detection

I. INTRODUCTION

DUE to the interconnected network environment and the rapid development of free image editing software, the editing and dissemination of digital images have become very easy, which inevitably includes malicious editing and illegal transmission, such as copying and editing the original image for commercial profit; Spreading maliciously tampered images to damage the reputation of organizations or individuals, so image authentication and image retrieval become more and more important. Image hashing is a method of converting human visual perception of images into short characters for representation. Short characters do not change with the specific data representation of the image, and the required storage space is small, therefore, image hashing has been widely used in image retrieval and content authentication. The design principle of image hashing is mainly to be robust against unintentional distortion caused by content-preserving operations and geometric distortions, to be sensitive to malicious tampering to image content, and have a certain degree of security. The performance of the image hashing algorithm depends largely on the method of extracting image features, so the hash algorithm is divided into the following four categories according to the extraction method of image features.

A. BASED ON INVARIANT FEATURE TRANSFORMATION

The hash algorithm based on invariant feature transformation mainly uses the frequency coefficients of the image in the transform domain to be robust to one or more attack operations. Ouyang *et al.* [1] utilized the amplitude correlation of the low-frequency quaternion discrete Fourier transform (QDFT) coefficients of the secondary image obtained by polar coordinate transformation to construct hash sequence. This algorithm can better resist rotation attacks. In scheme, Qin *et al.* [2] performed Weber local binary patterns (W-LBP) operation on the low-frequency sub-blocks obtained by discrete wavelet transform (DWT) transformation to extract local texture features, and utilized discrete cosine transform (DCT) transform for the color angle matrix to extract the color features. The texture and color features are combined to generate hash sequence. Tang *et al.* [3] utilized the phase spectrum of Fourier transform (PFT) visual model for the luminance Y component to generate visual saliency maps, extracted low-frequency characteristics from the visual saliency maps transformed by the dual-tree complex wavelet transform. The algorithm has good rotation robustness. Qin *et al.* [4] performed DCT transformation on the image blocks containing rich edge information, and the coefficient features and position information were processed by principal component analysis (PCA) dimensionality reduction to generate hash. Vadlamudi *et al.* [5] performed two-dimensional DWT decomposition on overlapping blocks containing feature points, and used the row average of DWT approximation coefficients to generate hash sequence. In scheme [6], Tang *et al.* applied discrete Fourier transform (DFT) to each row of the image processed by log-polar transformation (LPT), and used the amplitude of the DFT coefficient to construct a rotation-resistant feature matrix, and finally generated hash sequence by multidimensional scale decomposition (MDS). Lei *et al.* [7] first calculated the invariant moment of the image in the Radon transform domain, and then constructed the hash using the magnitude of the DFT coefficient of the invariant moment. Experiments show that the algorithm can effectively resist most non-malicious attacks. Ou *et al.* [8] performed one-dimensional DCT transform on multiple random direction projections, and finally sorted the low-frequency coefficients of each projection to generate image hash. In scheme [9], Liu *et al.* combined the distance relationship between low-frequency DWT coefficients of image blocks and the distance relationship between invariant moments in the spatial domain to generate hash. Sajjad *et al.* in scheme [10] utilized the main DCT coefficients of the rich information image block and the position information of the rich and sparse blocks of the edge information to construct hash. This algorithm implements real-time authentication in smart industrial applications.

B. BASED ON DIMENSION REDUCTION

Data dimensionality reduction can effectively reduce the redundancy of the extracted features, facilitate the generate of hash sequences of moderate length, and reduce the time complexity of the algorithm. In scheme [11], Tang *et al.* applied non-negative matrix factorization(NMF) decomposition on the secondary image obtained by the ring segmentation, and utilized the coefficient matrix of each image ring to construct a compact hash sequence. Tang *et al.* [12] first randomly selected mean image blocks to construct secondary image, then performed local linear embedding (LLE) processing on the secondary image, and utilized its embedded vector variance to design hash sequence. The robust performance and discrimination capability of the algorithm achieve an ideal compromise. In scheme [13], Tang *et al.* performed two dimensional DCT transformation on each small block of the color vector angle matrix to obtain low-frequency coefficient matrix, using the variance of the low-dimensional embedding vector as hash sequence. In scheme [14], Davarzani *et al.* constructed secondary image blocks by Singular Value Decomposition (SVD) decomposition of the image blocks, and then regarded the symbol information and amplitude information of the local discrepancies of the image sub-blocks as image features. This algorithm can effectively resist various types of noise attacks. Tang *et al.* [6] combined LPT and DFT to obtain the anti-rotation image feature matrix, MDS was performed on the feature matrix to form a compact and distinctive hash string. In scheme [15], Tang *et al.* constructed a tensor on the mean matrix of the brightness image and then used Tucker decomposition (TD) to generate hash sequences. In scheme [16], Liu *et al.* first performed low-rank representation (LRR) operation on the image to obtain a robust low-rank feature matrix, then performed DWT on the feature matrix. Finally, the hashing is generated by compressed sensing. This algorithm can achieve the recovery of tampered images under the premise of good robust capability.

C. BASED ON LOCAL FEATURE POINTS

Qin *et al.* [17] extracted the texture features of the image through dual-cross pattern (DCP) coding, and took the position information of the image block containing rich information as the structural feature. Finally, the two features are combined to generate image hashing. Shen *et al.* [18] extracted the local color change information from the color opponent component of the image, and applied quadtree decomposition to the image intensity component to extract the structural feature. The algorithm can locate the tampered area under the premise of good robustness. Qin *et al.* [19] combined edge detection and selective sampling to extract the location information and main DCT coefficients of rich edge information blocks, which were compressed by PCA to generate hash. Wang *et al.* [20] used the adaptive Harris operator to extract feature points from the low-frequency sub-band. Qin *et al.* [21] combined visual features based on color vector angles and prominent structural features based on image rings and image blocks to construct image hashing.

D. BASED ON STATISTICAL CHARACTERISTICS

Many algorithms based on statistical features are constantly mentioned. Such algorithms usually have good robustness in geometric attacks such as noise blurring and compression distortion. Tang *et al.* performed equal-area loop operation on the image in scheme [22], and then used the distance between the four statistical features of each image ring to construct hash sequences. This algorithm can effectively resist rotation attacks, but the discrimination capability needs to be improved. In scheme [23], Srivastava *et al.* performed DCT transformation on Radon coefficients in different directions, and extracted the statistical value of the feature vector as hash sequences. Tang *et al.* [24] utilized DWT to compress the histogram of the image ring to generate image hash. In scheme [25], Ouyang *et al.* used the amplitude coefficient information of quaternion Zernike moments (QZMs) of the image to generate hash characters. Huang *et al.* [26] extracted the statistical features of the texture image such as contrast, correlation, gradient, and homogeneity as the global features

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/ACCESS.2021.3069045, IEEE Access

of the image, and combined them with the DCT transform to construct the image hash. Hosny *et al.* [27] extracted Gaussian-Hermite moments and variance of grayscale images as image features. Wang *et al.* [28] designed hash sequences by combining the Watson visual model, Zernike moments and DCT coefficients. The hash algorithm can detect content changes and content forgeries caused by malicious attacks, and has good perceived robustness. Zhao *et al.* [29] used the Zernike moments of the luminance and chroma components as the global features, and the location information and texture information of the salient regions as the local features. The algorithm not only can detect the tampering of the image, but also locate the tampered area. Tang *et al.* [30] combined the color vector angle with the edge information obtained by the Canny operator, and used its statistical features to construct the image hash.

Although the above-mentioned hash algorithms have their own advantages, there are some hash algorithms whose classification performance and operation efficiency cannot be balanced; the robustness and discrimination capability of some hash algorithms cannot achieve the performance trade-off. In order to solve the above problems, we propose a hash algorithm based on the global features from three-dimensional visual angle in different directions and energy features. Our contributions mainly include the following:

(1) The algorithm innovatively uses the statistical features of the image in different three-dimensional visual angles as the global features of the image, and the relationship between the statistical features of the image layers from different visual angles as the final image hash.

(2) The image energy adopted in this article is almost never mentioned in the hash algorithms, but it has good robustness to the conventional content-preserving operation, which has been proved in subsequent section II-C. In this paper, the energy matrix of the image block is used to construct the energy matrix, and the multi-directional change feature of the energy matrix is used to construct the energy local feature hash. On the basis of excellent robustness of image energy, the discrimination capability of the algorithm is improved.

(3) The hash sequence in this paper is compact, only 162 bits. It has good robustness to the conventional geometric distortion. And its classification performance, operating efficiency, and local tampering detection are superior to five state-of-the-art schemes. It also has a good detection result in image copy detection.

The framework of this paper is mainly divided into the following parts: the second part is the specific steps of feature extraction and hash generation; the third part is a series of experiments and experimental analysis; the fourth part is a summary and prospects for future work.

II. PROPOSED IMAGE HASHING SCHEME

The main content of the hash algorithm in this paper is shown in Fig. 1, which includes four parts: preprocessing, global features extraction under different three-dimensional visual angles, local features extraction of multi-directional energy changes of image blocks and hash encryption.

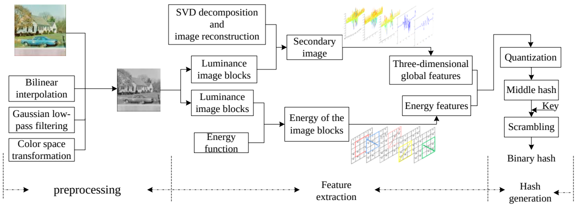


FIGURE 1. Flowchart of the proposed image hashing.

A. PREPROCESSING

First, the resolution of the input image I_0 is uniformly adjusted by bilinear interpolation to $M \times M$, which improves the robustness against image scaling attacks of the algorithm. In addition, input images of any size have the same hash length, which facilitates subsequent performance analysis. Then, Gaussian low-pass filtering is performed on the size-normalized image. This operation can reduce the impact of noise, compression and other minor operations on the image [17]. Finally, the preprocessed image is converted to $\mathrm{YC}_b\mathrm{C}_r$ color space, and its luminance component is taken for subsequent feature extraction.

B. GLOBAL FEATURES EXTRACTION OF 3D PERSPECTIVE

Before extracting features from image, the luminance component Y is first divided into non-overlapping blocks, the block size is $n \times n$, and an image block matrix B is formed.

$$B = \begin{bmatrix} B_{1,1} & B_{1,2} & \cdots & B_{1,M/n} \\ B_{2,1} & B_{2,2} & \cdots & B_{2,M/n} \\ \vdots & \vdots & \vdots & \vdots \\ B_{M/n,1} & B_{M/n,2} & \cdots & B_{M/n,M/n} \end{bmatrix} \tag{1}$$

where, $B_{i,j}$ is the image block located in the i-th row and the j-th column.

On the one hand, in order to reduce storage requirements and improve algorithm efficiency, on the other hand, in order to further improve the robustness of the algorithm to noise, each image block $B_{i,j}$ is further divided into four non-overlapping sub-blocks with size of $(n/2) \times (n/2)$, and then perform SVD decomposition on the image sub-block $b_{i,j}^{(k)}$ (k=1,2,3,4) according to (3).

$$B_{i,j} = \begin{bmatrix} b_{i,j}^{(1)} & b_{i,j}^{(2)} \\ b_{i,j}^{(3)} & b_{i,j}^{(4)} \end{bmatrix} \tag{2}$$

$$b_{i,j}^{(k)} = U_{i,j}^{(k)} S_{i,j}^{(k)} V_{i,j}^{(k)} \tag{3}$$

where, $b_{i,j}^{(k)}$ is the k-th image sub-block in image block $B_{i,j}$, $U_{i,j}^{(k)}$ and $V_{i,j}^{(k)}$ are the unit orthogonal matrices of the image sub-block $b_{i,j}^{(k)}$ after SVD decomposition, $S_{i,j}^{(k)}$ is a diagonal matrix containing the square roots of eigenvalues from $U_{i,j}^{(k)}$ or $V_{i,j}^{(k)}$ in descending order.

The first singular vector $u_{i,j}^{(k)}$ of $U_{i,j}^{(k)}$ and the first singular vector $v_{i,j}^{(k)}$ of $V_{i,j}^{(k)}$ are arranged and combined according to (4) to form a secondary image block $p_{i,j}$, with size of $(n/2) \times 8$. All the secondary image blocks are rearranged according to (5) to form the secondary image P with size of $(M/2) \times (8M/n)$.

$$p_{i,j} = \left[u_{i,j}^{(1)}, u_{i,j}^{(2)}, u_{i,j}^{(3)}, u_{i,j}^{(4)}, v_{i,j}^{(1)}, v_{i,j}^{(2)}, v_{i,j}^{(3)}, v_{i,j}^{(4)}\right] \tag{4}$$

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,8M/n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,8M/n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{M/2,1} & p_{M/2,2} & \cdots & p_{M/2,8M/n} \end{bmatrix} \tag{5}$$

Taking the horizontal resolution of the secondary image P as the x axis and the vertical resolution as the y axis, and the pixel value of the coordinate (x, y) as the z axis. Through the above operations, we can get the three-dimensional visual angle of the secondary image shown in Fig. 2. Observing Fig. 2, through the x-axis viewing angle and the y-axis viewing angle respectively, we can get completely different visual effects, as shown in Fig. 3. The following separately extracts the statistical characteristics of image from different three-dimensional visual angles to construct hash sequences.

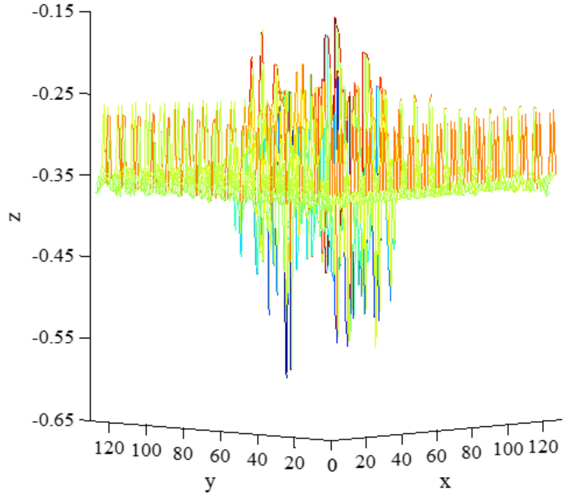
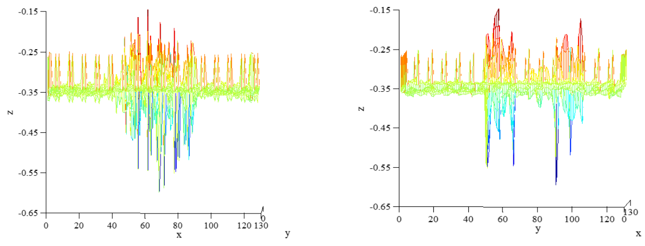


FIGURE 2. Secondary image with three-dimensional visual angle.

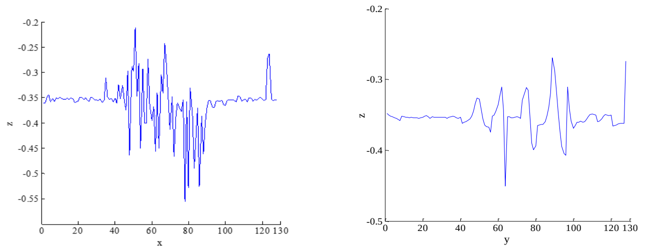
At the x-axis perspective, the secondary image P is layered according to the y-axis resolution, and is divided into $M/2$ layers, of which the i-th image layer is shown in Fig. 4. Calculate the statistical characteristics of each layer separately: mean, variance and kurtosis, and then form the mean matrix m_x, variance matrix v_x and kurtosis matrix s_x with size of $1 \times M/2$. The three matrices jointly form a statistical feature matrix T_x under the x-axis viewing angle, with size of $3 \times M/2$.



(a)

(b)

FIGURE 3. Secondary image at different visual angles. (a) Secondary image at x-axis visual angle. (b) Secondary image at y-axis visual angle.



(a)

(b)

FIGURE 4. Image layer at the different viewing angles. (a) An Image layer at the x-axis viewing angle. (b) An Image layer at the y-axis viewing angle.

$$m_x = \left[m_1, m_2, m_3, \cdots, m_{M/2-1}, m_{M/2}\right] \tag{6}$$

$$v_x = \left[v_1, v_2, v_3, \cdots, v_{M/2-1}, v_{M/2}\right] \tag{7}$$

$$s_x = \left[s_1, s_2, s_3, \cdots, s_{M/2-1}, s_{M/2}\right] \tag{8}$$

$$T_x = \left[m_x, v_x, s_x\right] \tag{9}$$

Similarly, construct the mean matrix m_y, the variance matrix v_y, the kurtosis matrix s_y, and the statistical feature matrix T_y from the y-axis viewing angle.

The matrix T_x is subjected to row standardization according to (10) to obtain the matrix F.

$$F_{i,j} = \frac{T_{i,j} - u_i}{\sigma_i} \tag{10}$$

where, $T_{i,j}$ is the i-th row and j-th column of the matrix T_x, u_i is the mean of the vector in the i-th row, and σ_i is the standard deviation of the vector in the i-th row.

Similarly, the matrix T_y is subjected to the above normalization operation to obtain the matrix Q.

Calculate the Euclidean distance of each column of matrix F and matrix Q according to (11), and obtain an invariant feature matrix h of size $1 \times M/2$.

$$h(j) = \sqrt{\sum_{i=1}^{3} \left|F_{i,j} - Q_{i,j}\right|^2} \tag{11}$$

where, $F_{i,j}$ and $Q_{i,j}$ are the i-th row and j-th column of the matrix F and Q, $h(j)$ is the j-th element of the matrix h.

By operating the invariant feature matrix h according to (12), we can obtain the binary sequence H_S with size of $M/2-1$.

$$H_S(j)=\begin{cases}1, & h(j+1)>h(j)\\ 0, & otherwise\end{cases} \tag{12}$$

where, $h(j)$ is the j-th element of matrix h, $H_S(j)$ is the j-th element of sequence H_S.

C. FEATURE EXTRACTION OF LOCAL ENERGY

For the luminance image Y of size $M\times M$, the energy $E(Y)$ can be expressed as (13) [31]:

$$E(Y)=\sum_{i=1}^{M}\sum_{j=1}^{M}y_{ij}^2=trace(Y^TY) \tag{13}$$

where, $trace(\cdot)$ represents the trace of the matrix, and y_{ij} represents the pixel value of the luminance image Y.

When the image Y is disturbed by a small amount of W during transmission, its energy will not change significantly, the proof process is shown in (14). Because the conventional content-preserving operation on the image will only have a slight impact on the pixel values of the image, so it can be considered that the image energy has good robustness for the conventional content-preserving operation.

$$\begin{aligned}\Delta E&=|E(Y+W)-E(Y)|\\&=\left|E(W)+2trace(Y^TW)\right|\\&\le E(W)+2\sqrt{E(Y)\times E(W)}\end{aligned} \tag{14}$$

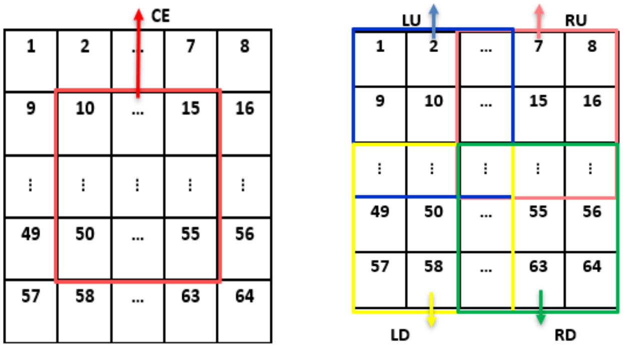
where, ΔE is the amount of energy change caused by small disturbances in the image.

When extracting the energy features of the image, firstly, the luminance Y component is divided into non-overlapping blocks, the block size is $a\times a$, and the energy value of each image sub-block is obtained in proper order to form the energy matrix N_1. The reasons for choosing to extract the energy of each image block are mainly the following three aspects: firstly, the energy of the image block has good robust to the content-preserving operation; secondly, different images may have the same image energy, but it is difficult for different images to have exactly the same image block energy; finally, the image block processing can improve the algorithm's robustness to subtle operations.

$$N_1=\begin{bmatrix}n_{1,1} & n_{1,2} & \cdots & n_{1,M/a}\\ n_{2,1} & n_{2,2} & \cdots & n_{2,M/a}\\ \vdots & \vdots & \vdots & \vdots\\ n_{M/a,1} & n_{M/a,2} & \cdots & n_{M/a,M/a}\end{bmatrix} \tag{15}$$

where, $n_{i,j}$ is the energy value of the image sub-block located in the i-th row and j-th column.

Perform matrix operations on the energy matrix N_1 in four directions as shown in Fig. 5. The upper left energy change matrix N_{lu}, the upper right energy change matrix N_{ru}, the lower left energy change matrix N_{ld}, and the lower right energy change matrix N_{rd} are obtained by sequentially subtracting the matrices LU, RU, LD and RD from the center matrix CE. In order to obtain a concise feature matrix, the above four matrices are processed according to (16), and then the energy change matrix N_v is obtained.



(a) (b)

FIGURE 5. Energy changes. (a) Energy value of pixels in the central area. (b) Energy value of pixels in the corner area.

$$N_v=N_{lu}\times N_{ru}\times N_{ld}\times N_{rd} \tag{16}$$

In order to ensure the operating efficiency of the algorithm and reduce the storage space redundancy, the energy change matrix N_v is expanded into a matrix N by rows, and quantized into a binary sequence H_N according to (17).

$$H_N(i)=\begin{cases}1, & N(i+1)>N(i)\\ 0, & otherwise\end{cases} \tag{17}$$

where, $N(i)$ and $H_N(i)$ are the i-th elements of the matrices N and H_N, respectively.

D. HASH GENERATION

The three-dimensional global statistical feature H_S and the energy local feature H_N are combined to obtain the intermediate hash sequence H_m=[H_S, H_N]. The lengths of the binary sequences H_S and H_N are $M/2-1$ bits and $(M/a-2)^2-1$ bits respectively, so the hash length $L=M/2+(M/a-2)^2-2$ bits.

In order to ensure the security of the algorithm, we rearrange the columns of H_m through the pseudo-random number sequence S generated by the randperm $(\cdot)$ function in MATLAB to obtain the final hash sequence H, as shown in (18).

$$H(i)=H_m(S[i]) \tag{18}$$

where, $S[i]$ represents the i-th number in the pseudo-random number sequence S , $H(i)$ is the i-th element of sequence H.

E. DISTANCE MEASURE

In this paper, the hash sequence H_1 of the original image and the hash sequence H_2 of the image to be tested are obtained by the proposed algorithm, the difference between the two sequences is measured by the normalized Hamming distance $D(H_1,\,H_2)$. When $D(H_1,\,H_2)>T$, it is considered that the test image has been malicious tampering or is different from the original image; when $D(H_1, H_2) \leq T$, the test image and the original image are similar image pairs, and the threshold T is obtained through subsequent experiments.

$$D(H_1, H_2) = \frac{1}{L}\sum_{i=1}^{L}|H_1(i) - H_2(i)| \tag{19}$$

where, $H_1(i)$ and $H_2(i)$ are the i-th elements of the hash sequences H_1 and H_2, and L is the total length of the hash sequence of the proposed algorithm.

III. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, firstly, we conduct robust experiments and discrimination experiments to test whether the proposed algorithm can meet the requirements of the basic properties of image hashing. Then we analyze the effect of parameters on the performance of the proposed algorithm. The proposed algorithm is compared with several state-of-the-art schemes in various aspects. Finally, copy detection experiment and local tampering detection experiment are carried out. All the experiments were simulated by MATLAB R2014a platform, and the computer was configured with Intel (R) Core (TM) i5-7200U CPU @2.50GHz 2.7GHz and 8.00GB RAM.

A. PARAMETER SETTINGS

The specific settings of the proposed hash algorithm are as follows: the size of normalized image and the standard deviation of the Gaussian low-pass filter in the image preprocessing section are 256 and 1, respectively. As for global feature extraction at different three-dimensional visual angles, the image block size is 16 × 16. In the feature extraction part of local energy change, the image block size is 32 × 32. That is, $M = 256$, $\sigma = 1$, $n = 16$ and $a = 32$. According to the above parameter settings, the hash length of this article is $L = M/2 + (M/a - 2)^2 - 2 = 162$ bits.

B. PERCEPTUAL ROBUSTNESS

The experiments in this section mainly reflect the robust performance of the hash algorithm between the input image and the similar images generated by the content preservation operation. In this paper, 20 color images are selected as the robust experimental samples, some standard images are shown in Fig. 6. Firstly, the 20 sample images are subjected to 12 kinds of attack operations shown in Table 1, respectively, and a total of 1380 similar images are generated; Then use the proposed algorithm to obtain hash sequences of sample images and similar images; Finally, the hash distance between each sample image and its similar images are calculated according to formula (19). Table 2 shows the statistics of hash distance (minimum, maximum, average and standard deviation) between 20 sample images and their similar images under different attack types. In Table 2, except for mean filtering and rotation attacks, for the remaining 10 attack operations, the minimum hash distance between the sample image and its similar versions is 0, and the maximum distance does not exceed 0.1; Except for rotation attacks, the mean and standard deviation of the hash distance for other attack types are both less than 0.1; Therefore, this algorithm can effectively resist other conventional content-preserving operations except for rotation attacks.

FIGURE 6. Five standard images for robustness test.

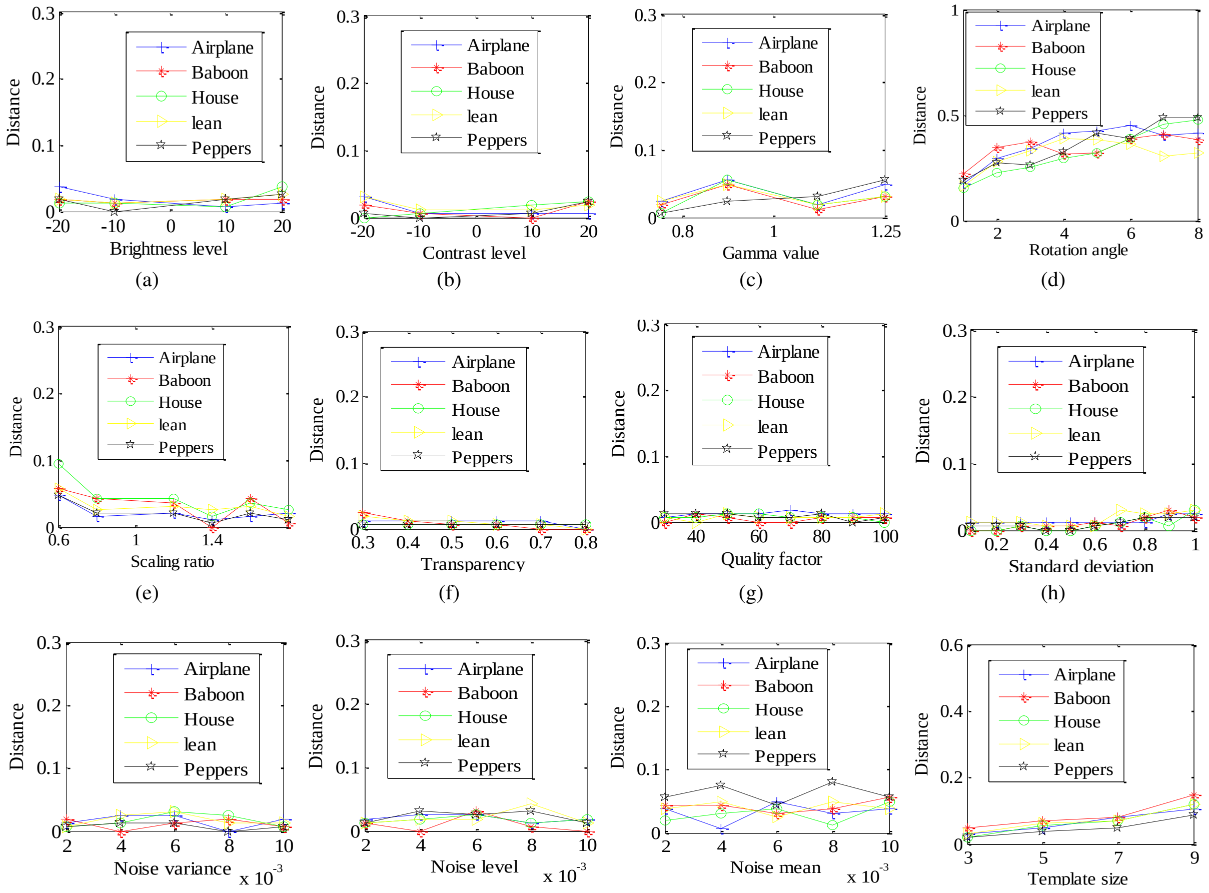
TABLE 1. Operations and parameter settings.

Operation	Parameter	Parameter values
Rotation	Angle	1, 2, · · · , 7, 8
Brightness adjustment	Level	-20, -10, 10, 20
Contrast adjustment	Level	-20, -10, 10, 20
Gamma correction	Gamma	0.75, 0.9, 1.1, 1.25
3×3 Gaussian low-pass filtering	Standard deviation	0.1, 0.2, · · · , 0.9, 1
JPEG compression	Quality	30, 40, · · · , 90, 100
Watermark embedding	Transparency	0.3, 0.4, · · · , 0.7, 0.8
Mean filter	Neighborhood	3×3, 5×5, 7×7, 9×9
Speckle noise	Noise variance	0.002, 0.004, · · · , 0.01
Salt and Pepper noise	Noise level	0.002, 0.004, · · · , 0.01
Gaussian noise	Noise mean	0.002, 0.004, · · · , 0.01
Scaling	Ratio	0.6, 0.8, 1.2, 1.4, 1.6, 1.8

Fig. 7 are the graphs of the robust experimental results of 5 standard images (Airplane, Baboon, House, Lena and Peppers) and their similar images under various types of content-preserving operations, which is convenient for intuitively displaying the robust performance of the algorithm. The horizontal coordinate of the sub-picture is the corresponding conventional image processing parameter setting, and the vertical coordinate is the Hamming distance between the standard image and its corresponding conventional processing images. As shown in Fig. 7, besides the rotation attack, the distance curve of the same attack operation with different parameter settings has small fluctuation range and gentle change, which further illustrates that the propose algorithm has good robustness to multiple image attacks. For the rotation attack, the distance increases with the increase of the rotation angle, so the algorithm in this paper cannot effectively resist the large angle rotation.

C. DISCRIMINATION CAPABILITY

The discrimination experiments can effectively reflect the classification performance of the algorithm. The experimental dataset consists of 1000 different images, of which 700 images are from the University of Washington Ground Truth database [32], and 300 images are taken from the VOC2007 database [33]. Any two of the 1000 images are different image pairs, and the total number of different image pairs is $C_{1000}^2 = 499500$. Perform 11 kinds of content-preserving operations on the above 1000 different images. The specific attack types and parameter settings are shown in Table 3. The total number of similar image pairs is $C_{23}^2 \times 1000 = 253000$.



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

(e) (f) (g) (h)

(i) (j) (k) (l)

FIGURE 7. Performance of perceptual robustness. (a) Brightness adjustment. (b) Contrast adjustment. (c) Gamma correction. (d) Rotation. (e) Scaling. (f) Watermark embedding. (g) JPEG compression. (h) Gaussian low-pass filtering. (i) Speckle noise. (j) Salt and pepper noise. (k) Gaussian noise. (l) Mean filter.

TABLE 2. Statistic values of hash distances.

Operation	Min	Max	Mean	Std
Scaling	0	0.0930	0.0357	0.0156
Rotation	0.1481	0.4877	0.3581	0.0842
Mean filter	0.0123	0.2284	0.0793	0.0475
Speckle noise	0	0.0432	0.0116	0.0091
Gaussian noise	0	0.0968	0.0429	0.0191
Gamma correction	0	0.0802	0.0322	0.0212
JPEG compression	0	0.0494	0.0080	0.0074
Contrast adjustment	0	0.0741	0.0152	0.0132
Salt and Pepper noise	0	0.0556	0.0159	0.0124
Brightness adjustment	0	0.0741	0.0197	0.0154
Watermark embedding	0	0.0494	0.0109	0.0097
3×3 Gaussian low-pass filtering	0	0.0988	0.0242	0.0200

The distance distribution between similar image pairs and different image pairs can be intuitively seen through Fig. 8, where the red curve is the distance distribution between similar image pairs and the blue curve is the distance distribution between different image pairs. The abscissa of the red curve is between 0 $\sim$ 0.2461, the abscissa of the blue curve is in the range of 0.2222 $\sim$ 0.6728, the distance between the two curves overlapping is 0.2222 $\sim$ 0.2461, because the overlap distance is short, the number of overlaps is small, therefore, an appropriate threshold can be selected to effectively distinguish between different images and similar images.

When the selected threshold is too small, similar image pairs are easily misjudged as different image pairs, resulting in a large error detection rate P_E [34]; when the selected threshold is large, different image pairs are easily mistaken for similar image pairs, resulting in a large collision rate P_C [34], that is, the collision rate and the error detection rate are mutually suppressed. Therefore, the threshold should be selected when the collision rate and the error detection rate are small, so that the robustness and discrimination capability of the algorithm reach a good trade-off. The formulas of collision rate and error detection rate are shown in (20). The collision rate P_C and the error detection rate P_E under specific thresholds are shown in Table 4, and T = 0.24 is chosen in this paper.

$$P_C = \frac{N_C}{N_D},$$
$$P_E = \frac{N_E}{N_S} \tag{20}$$

where, N_C is the total number of different image pairs

TABLE 3. Operations and parameter settings.

Operation	Parameter	Parameter values
JPEG compression	Quality	40, 80
Contrast adjustment	Level	-20, 20
Brightness adjustment	Level	-20, 20
Scaling	Ratio	0.8, 1.6
Watermark embedding	Transparency	0.3, 0.8
3×3 Gaussian low-pass filtering	Standard deviation	0.2, 0.6
Gamma correction	Gamma	0.75, 1.25
Mean filter	Neighborhood	3×3, 5×5
Speckle noise	Noise variance	0.002, 0.006
Gaussian noise	Noise mean	0.002, 0.006
Salt and Pepper noise	Noise level	0.002, 0.006

TABLE 4. Collision probability and error detection probability with different threshold values.

Threshold T	P_C	P_E
0.22	0	1.976×10^{-5}
0.23	2.002×10^{-6}	1.976×10^{-5}
0.235	2.002×10^{-6}	7.905×10^{-6}
0.24	2.002×10^{-6}	3.953×10^{-6}
0.25	4.004×10^{-6}	0

misjudged as similar image pairs, N_E is the total number of similar image pairs misjudged as different image pairs, and N_D and N_S are the total number of different image pairs and similar image pairs, respectively.

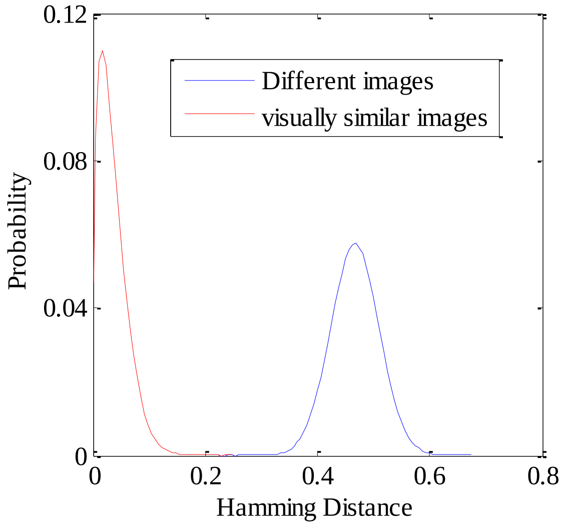


FIGURE 8. Hash distance distribution of different images and similar images.

D. IMPACT OF IMPORTANT PARAMETERS ON ALGORITHM PERFORMANCE

In the process of extracting energy change features, because the brightness image is processed by non-overlapping block, so the size of the image block will affect the performance of the proposed hash algorithm. Therefore, other experimental parameters are unchanged, the performances are compared with different a, that is $a = 8$, $a = 16$, and $a = 32$.

We analyze the impact of image block size on algorithm classification performance by plotting by plotting (Receiver Operating Characteristics) ROC curves [35] at different values of a. This experiment still uses 499500 different image pairs and 253000 similar image pairs mentioned in section III-C, and the horizontal and vertical coordinates of the ROC curve can be obtained by (21).

$$P_{FPR} = \frac{N_F}{N_D},$$
$$P_{TPR} = \frac{N_T}{N_S} \tag{21}$$

where, N_F is the total number of image pairs misjudged as similar image pairs, N_T is the total number of similar image pairs correctly judged, and N_D and N_S are the total number of different image pairs and similar image pairs, respectively.

Fig. 9 shows the ROC curves at different values of a. First of all, it can be seen intuitively that when $a = 32$, $a = 16$, and $a = 8$, the ROC curve is all close to the upper left corner, indicating that the classification performance of the algorithm is good. Secondly, when $a = 32$, the ROC curve of the algorithm is closest to the upper left corner, so it can be considered that the algorithm achieves the best classification performance when $a = 32$. In addition, the hash length and average hash generation time when $a = 32$, $a = 16$ and $a = 8$ are summarized in Table 5. It is not difficult to find that when $a = 32$, the hash length is the shortest and the time required is the least, which is more in line with the low storage and high efficiency requirements of the hash algorithm. In summary, we set $a = 32$.

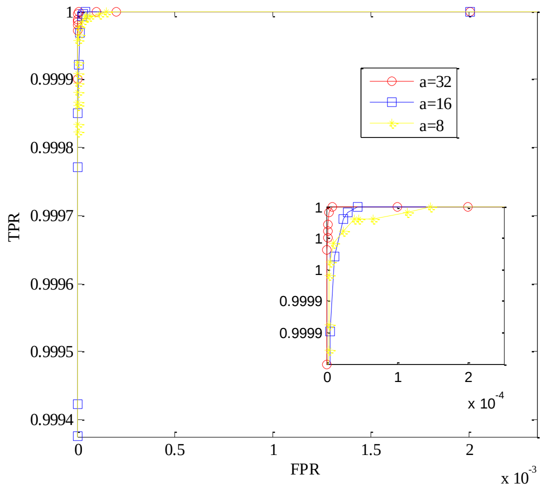


FIGURE 9. ROC curves with different a values.

E. PERFORMANCE COMPARISON

In this section, the algorithm of this paper is compared with five state-of-the-art schemes for performance comparison experiments, i.e., Davarzani *et al.'s* scheme [14], Tang *et al.'s* scheme [15], Shen *et al.'s* scheme [18], Huang *et al.'s*

TABLE 5. Average time and hash length of different a values.

a	Length of hash	Average time
8	1026 bits	0.0387s
16	322 bits	0.0312s
32	162 bits	0.0287s

scheme [26], Tang *et al.'s* scheme [30]. The performance of the above algorithms is mainly measured by three aspects: classification performance, storage requirements, and hash generation efficiency. In order to ensure the fairness of the experimental results, we abide by the following three rules during the experiment: do not change the original parameter settings of the comparison algorithm, all algorithms use the same data set, and all experiments are completed on the same computer.

1) Comparison of classification performance

In the classification performance comparison experiment, the ROC curve is also used as the theoretical analysis tool. The data set in this experiment is consistent with section III-C, with 499500 different image pairs and 253000 similar image pairs. It can be seen intuitively from Fig. 10 that the ROC curve of the proposed algorithm is closest to the upper left corner of the square area. Since the upper left corner area represents that when P_{FPR} has a smaller value, P_{TPR} has a larger value, so this algorithm has the best classification performance. In fact, when $P_{FPR} = 0$, the P_{TPR} of the proposed algorithm and schemes [14], [15], [18], [26] and [30] are 0.99997, 0.9985, 0.865, 0.9997, 0.8823 and 0.0861, respectively. When $P_{TPR} \approx 1$, the P_{FPR} of the proposed algorithm and schemes [14], [15], [18], [26] and [30] are 8.008×10^{-6}, 0.1121, 0.0830, 6.126×10^{-5}, 0.0963 and 0.3501 in order. That is, under the same conditions, the proposed algorithm has the largest P_{TPR} and the smallest P_{FPR} compared with other algorithms.

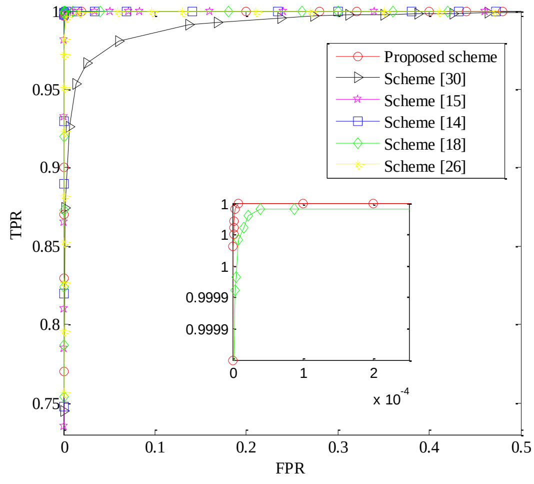


FIGURE 10. ROC curves of proposed scheme and the schemes [14], [15], [18], [26], [30].

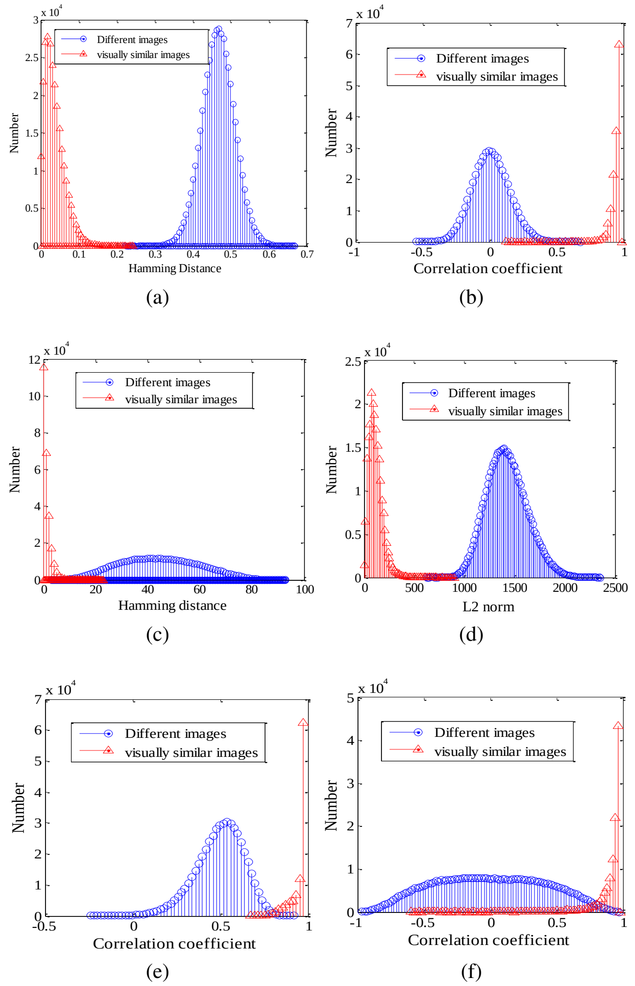


FIGURE 11. The distributed of different images and visually similar images of different schemes. (a) Our scheme. (b) Scheme [14]. (c) Scheme [15]. (d) Scheme [18]. (e) Scheme [26].(f) Scheme [30].

In order to further show the classification performance of the hash algorithm, we have drawn the distance distribution maps of various algorithms. As shown in Fig. 11, the subgraphs (a) ∼ (f) in Fig. 11 correspond to the algorithm of this paper and the schemes [14], [15], [18], [26] and [30]. The red part in the distance distribution diagram represents similar image pairs, and the blue part represents different image pairs. It can be seen intuitively from these distance distribution graphs that the overlapping area of the red part and the blue part of the proposed algorithm is the smallest, indicating that the classification performance of this algorithm is the best.

2) Comparison of computational complexity

When the number of test images is huge, the storage requirements of hash sequences and the efficiency of hash generation are particularly important, so shorter hash generation time and shorter hash length should be the basic requirements for the proposed algorithm. In the comparison experiment of hash generation time between different hash algorithms, under the premise that the external parameters such as the experimental parameter settings, computer configuration, and experimental database are same, we recorded the total time of producing hashes of 1000 different images, and then divide the total time by 1000 to get the average generation time. The average time to generate 1000 different image hash sequences of proposed algorithm and schemes [14], [15], [18], [26] and [30] are 0.0287s, 0.2213s, 0.3021s, 0.0617s, 0.1376s and 2.9783s respectively, so the proposed algorithm takes the shortest time to generate hash. The hash length of the proposed algorithm is 162 bits, and the hash lengths of literatures [14], [15], [18], [26] and [30] are 64 decimal digits, 96 bits, 452 decimal digits, 720 bits and 400 bits. Since a decimal number requires at least 4 binary numbers, the hash length of the proposed algorithm is only slightly longer than that of scheme [15], but its classification performance and hash generation efficiency are significantly better than scheme [15].

According to the performance comparison results of the above six hash algorithms, it can be seen that the proposed algorithm not only has the best classification performance but also requires the shortest hash generation time. The specific comparison results of the six algorithms are summarized in Table 6.

F. APPLICATION OF IMAGE COPY DETECTION

By choosing an appropriate threshold, the algorithm in this paper can effectively detect the copied images. The experimental data set contains 3600 test images, of which 1000 are different images downloaded from the network, 100 randomly selected from the above different images as query images, 13 content retention operations are performed on each query image to generate 2600 copies images, specific attack types and corresponding parameter settings are shown in Table 7. The copy detection capability of the algorithm is described by the recall rate R and precision rate P [36] under different thresholds. The definition of the recall rate and precision rate is defined as (22), the specific results are shown in Table 8. When the threshold is 0.27, the algorithm can detect all the copied images, but the accuracy rate needs to be improved; when the threshold is 0.31, the precision can also achieve good level.

$$\begin{aligned} P &= \frac{N_p}{N_q}, \\ R &= \frac{N_p}{N_a} \end{aligned} \tag{22}$$

where, N_p is the number of copy images in the query result that correctly match the query images, N_q is the number of all copied images included in the query result, and N_a is the number of all copied images in the test image set.

G. APPLICATION OF IMAGE TAMPERING DETECTION

When the image is partially tampered, the hash distance between the tampered image and the original image should be greater than the distance between similar image pairs and smaller than the distance between different image pairs. The total number of different image pairs in this experiment is 499500, and the total number of similar image pairs is 253000, which are the data sets used in section III-C. The tampered image set contains 15000 original images and 15000 tampered images. The original images are taken from the VOC2012 database [37], and 20% of the original image area is added to each original image to form tampered images. Fig. 12 shows the distance distribution between similar image pairs, original images and tampered images, and different image pairs. The red curve is the distance distribution between similar image pairs, ranging from 0 to 0.247; the blue curve is the distance distribution between the original images and the tampered images, and the endpoint values are 0.0123 and 0.401; the green curve is the distance distribution between different image pairs, ranging from 0.222 to 0.673. It can be seen intuitively from Fig. 12 that the blue curve is between the red curve and the green curve, the horizontal coordinate of the intersection point T_1 of the blue curve and the red curve is 0.0710, and the horizontal coordinate of the intersection point T_2 of the blue curve and the green curve is 0.3364. When the distance between the test image and the original image is less than T_1, the test image and the original image are considered to be similar image pairs; when it is greater than T_2, the test image and the original image are different from each other; when the distance is between T_1 and T_2 , The detect image is regarded as partial tampering image. When the threshold value is T_1, the probability of the proposed algorithm correctly identifying similar image pairs is 93.29%; when the threshold value is T_2, the probability of the algorithm correctly identifying different image pairs is 99.89%; the probability of the algorithm correctly identifying local tampered images is 94.17%.

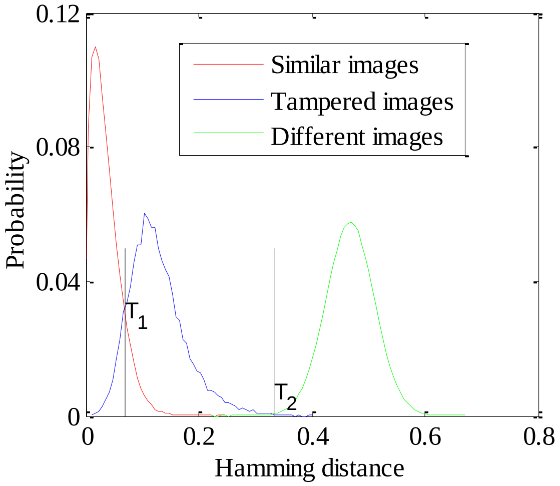


FIGURE 12. Hash distance distribution.

Regarding the effect of a value on the tampering detection results, we obtained the results shown in Table 9 through

TABLE 6. Overall performance comparison of hashing schemes.

Scheme	Capability of classification	Length of hash	Average time	Optimal TPR when FPR =0	Optimal FPR when TPR =1
Scheme [14]	Moderate	64 decimal digits	0.2213 s	0.9985	0.1121
Scheme [15]	Moderate	96 bits	0.3021 s	0.865	0.0830
Scheme [18]	Good	452 decimal digits	0.0617 s	0.9997	6.126×10^{-5}
Scheme [26]	Moderate	720 bits	0.1376 s	0.8823	0.0963
Scheme [30]	Poor	400 bits	2.9783 s	0.0861	0.3501
Our scheme	Best	162 bits	0.0287 s	0.99997	8.008×10^{-6}

TABLE 7. Operations and parameter settings.

Operation	Parameter	Parameter values
Mosaic	Square size	6, 10
Plus subtitles	Font size	10, 20
JPEG compression	Quality	40, 80
Contrast adjustment	Level	-20, 20
Brightness adjustment	Level	-20, 20
Scaling	Ratio	0.8, 1.6
Watermark embedding	Transparency	0.3, 0.8
3×3 Gaussian low-pass filtering	Standard deviation	0.2, 0.6
Gamma correction	Gamma	0.75, 1.25
Mean filter	Neighborhood	3×3, 5×5
Speckle noise	Noise variance	0.002, 0.006
Gaussian noise	Noise mean	0.002, 0.006
Salt and Pepper noise	Noise level	0.002, 0.006

TABLE 8. The recall rate and precision rate with different thresholds.

Threshold	Recall rate	Precision rate
0.27	100%	94.54%
0.29	99.89%	95.19%
0.31	97.67%	96.30%
0.33	90.79%	96.85%

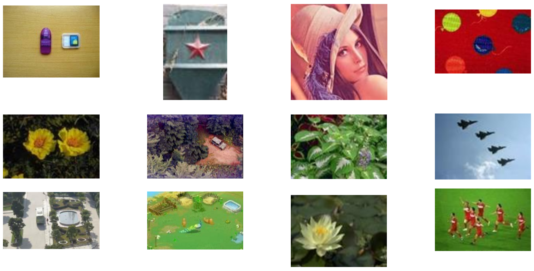


FIGURE 13. Original images.

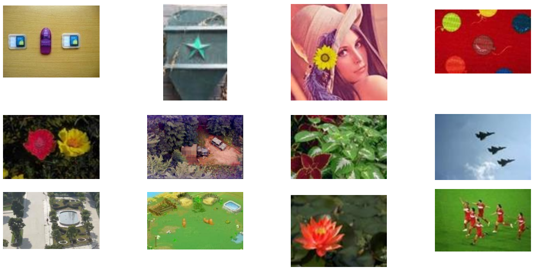


FIGURE 14. Tampered images.

experiments. It can be found from Table 9 that as the side length of the image block decreases and the number of image blocks increases, the algorithm can detect more detailed changes of the image, which is conducive to the tamper detection of the proposed algorithm, and also illustrates the effectiveness of the algorithm in this paper to detect tampered images.

Next, the tampering detection capabilities of the proposed algorithm ($a = 32$) and the comparison algorithms will be further explained through some tampering examples. The original image and the corresponding partial tampering image are shown in Fig. 13 and Fig. 14, where the type of tampering includes local color tampering and local content tampering (including the deletion of objects and the addition of objects).In Fig. 13, from left to right and from top to bottom are (a_1) to (l_1). In Fig. 14, from left to right and from top to bottom are (a_2) to (l_2). Since Schemes [15], [26] and [30] in the comparison algorithm do not mention the algorithm's ability to tamper detection in the original article, the detection result may be unsatisfactory, but all algorithms use the same data sets. In addition, the hash distance measurement standards of this article and schemes [14], [15], [18], [26] and [30] are normalized hamming distance, correlation coefficient, hamming distance, L_2 norm, correlation coefficient, and correlation coefficient, respectively. From Table 10, we can find that the distance between the original image and the tampered image for the proposed algorithm is all between T_1 and T_2, and other schemes cannot completely detect the tampered image, therefore, the proposed algorithm has a certain detection ability for tampered images.

IV. CONCLUSIONS

This algorithm uses the three-dimensional statistical features from different visual angles as the image global features, and the energy variation features in the four directions as the local features. Finally, the three-dimensional global features and the multi-directional local variation features are combined and scrambled to obtain the final hash sequences. It can be seen that the algorithm achieves a good trade-off between discrimination capability and robustness from the collision rate and error detection rate under different thresholds. Compared with five state-of-the-art schemes, the proposed algorithm has the advantages of the best ROC curve, compact hash sequence, the most excellent operating efficiency and the best tamper detection performance. However, the algorithm in this paper also has some shortcomings, such as the inability to effectively resist large-angle rotation attack operations, the

TABLE 9. Detection probability of different a values.

The value of a	Probability of correctly identifying similar image pairs	Probability of correctly identifying tampered image pairs	Probability of correctly identifying different image pairs
8	93.11%	99.43%	97.02%
16	91.36%	98.95%	98.74%
32	93.29%	94.17%	99.89%

TABLE 10. Hash distances of original image and tampered image pairs of different algorithms.

Images	Scheme [14]	Scheme [15]	Scheme [18]	Scheme [26]	Scheme [30]	Proposed scheme
(a_1) and (a_2)	0.9550	1	742.2668	0.8948	0.9619	0.2037
(b_1) and (b_2)	0.9627	7	1137	0.9989	0.9040	0.0802
(c_1) and (c_2)	0.9441	20	761.4329	0.9778	0.6652	0.1296
(d_1) and (d_2)	0.9188	21	744.3037	0.9984	0.1574	0.2037
(e_1) and (e_2)	0.9572	9	556.6750	0.9973	-0.3646	0.1728
(f_1) and (f_2)	0.8072	6	434.5469	0.9513	0.7802	0.1605
(g_1) and (g_2)	0.7824	8	744.3037	0.9479	0.8939	0.2222
(h_1) and (h_2)	0.8893	0	1072.7	0.9564	0.6655	0.1728
(i_1) and (i_2)	0.9889	10	473.2811	0.9899	1	0.0802
(j_1) and (j_2)	0.9760	13	316.0823	0.9681	0.8984	0.1235
(k_1) and (k_2)	0.9654	9	998.6651	0.9842	0.3018	0.0988
(l_1) and (l_2)	0.9071	3	1071.9	0.9366	0.9770	0.1420
$(T_1) \sim (T_2)$	$0.5771 \sim 0.9621$	$2.8802 \sim 10.023$	$273.6 \sim 1028.2$	$0.7664 \sim 0.7949$	$0.5002 \sim 0.9233$	$0.0710 \sim 0.3364$

inability to effectively detect subtle tampering images. The next research work will focus on improving these shortcomings.

REFERENCES

[1] J. Ouyang, G. Coatrieux, and H. Shu, "Robust hashing for image authentication using quaternion discrete Fourier and log-polar transform," Digital Signal Processing, vol. 41, pp. 98–109, Jun. 2015.

[2] C. Qin, Y. Hu, H. Yao, and L. Gao, "Perceptual image hashing based on weber local binary pattern and color angle representation," IEEE Access, vol. 7, pp. 45460–45471, 2019.

[3] Z. Tang, Y. Yu, H. Zhang, M. Yu, C. Yu, and X. Zhang, "Robust image hashing via visual attention model and ring partition," Mathematical Biosciences and Engineering, vol. 16, pp. 6103–6120, 2019.

[4] C. Qin, X. Chen, J. Dong, and X. Zhang, "Perceptual image hashing with selective sampling for salient structure features," Displays, vol. 45, pp. 26–37, 2016.

[5] 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.

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

[7] Y. Lei, Y. Wang, and J. Huang, "Robust image hash in Radon transform domain for authentication," Signal Processing, vol. 26, pp. 280–288, 2011.

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

[9] S. Liu, Z.Huang, "Efficient Image Hashing with Geometric Invariant Vector Distance for Copy Detection," ACM Transactions on Multimedia Computing Communications and Applications, vol. 15, pp. 106:1–106:22, 2019.

[10] 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, pp. 6541–6550, 2019.

[11] Z. Tang, X. 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.

[12] Z. Tang, L. Ruan, C. Qin, X. Zhang, and C. Yu, "Robust image hashing with embedding vector variance of LLE," Digital Signal Process, vol. 43, pp. 17–27, 2015.

[13] Z. Tang, H. Lao, X. Zhang, and K. Liu, "Robust image hashing via DCT and LLE," Computers and Security, vol. 62, pp. 133–148, Sep. 2016.

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

[15] Z. Tang, L. Chen, and X. Zhang, "Robust Image hashing with tensor decomposition," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 3, pp. 549–560, 2019.

[16] H. Liu, D. Xiao, Y. Xiao, and Y. Zhang, "Robust image hashing with tampering recovery capability via low-rank and sparse representation," Multimedia Tools and Applications, vol. 75, no. 13, pp. 7681–7696, 2016.

[17] C. Qin, X. Chen, and X. Luo, "Perceptual image hashing bis dual-cross pattern encoding and salient structure detection," Information Sciences, vol. 423, pp. 284–302, Sep. 2017.

[18] Q. Shen, Y. Zhao, "Perceptual hashing for color image based on color opponent component and quadtree structure," Signal Processing, vol. 166, pp. 107244.1-107244.12, Jan. 2020.

[19] C. Qin, X. Chen, J. Dong, and X. Zhang, "Perceptual image hashing with selective sampling for salient structure features," Displays, vol. 45, pp. 26–37, 2016.

[20] X. Wang, J. Xue, Z. Zheng, Z. Liu, and L. Ning, "Image forensic signature for content authenticity analysis," Journal of visual communication and image representation, vol. 23, no. 5, pp. 782–797, 2016.

[21] C. Qin, M. Sun, and C. Chang, "Perceptual hashing for color images based on hybrid extraction of structural features," Signal Processing, vol. 142, pp. 194–205, Jan. 2018.

[22] Z. Tang, X. 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, Jan. 2016.

[23] M. Srivastava, J. Siddiqui, and M.A. Ali, "Robust image hashing based on statistical features for copy detection," IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering, pp. 490–495, 2016.

[24] Z. Tang, L. Huang, Y. Dai, and Y. Fan, "Robust image hashing based on multiple histograms," International Journal of Digital Content Technology and Its Applications, vol. 6, no. 23, pp. 39–47, 2012.

[25] J. Ouyang, X. Wen, J. Liu, and J. Chen, "Robust hashing based on quaternion Zernike moments for image authentication," ACM Transactions on Multimedia Computing Communications and Applications, vol. 12, no. 4, pp. 63:1–63:13, 2016.

[26] Z. Huang, S. Liu, "Robustness and discrimination oriented hashing combining texture and invariant vector distance," Proceedings of the 26th ACM International Conference on Multimedia, pp. 1389–1397, 2018.

[27] K.M. Hosny, Y.M. Khedr, W.I. Khedr, and E.R. Mohamed, "Robust image hashing using exact Gaussian-Hermite moments," Image Processing Iet, vol. 12, no. 12, pp. 2178–2185, 2018.

[28] X. Wang, X. Zhou, Q. Zhang, B. Xu, and J. Xue, "Image alignment based

perceptual image hash for content authentication," Signal Processing: Image Communicatio, vol. 80, 2019.

[29] Y. Zhao, S. Wang, X. 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] Z. Tang, L. Huang, X. Zhang, and H. Lao, "Robust image hashing based on color vector angle and Canny operator," AEU-International Journal of Electronics and Communications, vol. 70, no. 6, pp. 833–841, 2016.

[31] X. Li, "Public digital watermark algorithm based on image energy and quantization," Journal of North University of China (Natural Science Edition), vol. 28, pp. 458–461, 2007.

[32] Ground Truth Database., 2008. [Online]. Available: http://http://www.cs.washington.edu/research/imagedatabase/groundtruth. Accessed on: Oct 07, 2017.

[33] Pascal VOC 2007 Data set. [Online]. Available: https://pjreddie.com/projects/pascal-voc-dataset-mirror/. Accessed on: Oct 07, 2017.

[34] Y. Zhao, W. Wei, "Robust image hashing for image authentication and tampering detection," Application research of computers, vol. 28, no. 5, pp. 1929–1931+1939, 2011.

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

[36] H. Müller, W. Müller, and D. M. Squire, "Performance evaluation in content-based image retrieval: overview and proposals," Pattern Recognition Letters, vol. 22, no. 5, pp. 593–601, 2001.

[37] REDMON J. Pascal VOC dataset mirrir. [EB/OL]., 2012. [Online]. Available: https://pjreddie.com/pro- jects/pascal-voc-dataset-mirror/. Accessed on: Oct 07, 2017.

XIAORAN YUAN was born in 1993. She received the B.S. degree from Zheng University of Light Industry, P.R. China, in 2018. She is now pursuing the M.Eng. degree at Shanghai University of Electric Power. Her research interest is image hashing.

YAN ZHAO was born in 1979. She received the M.S. degree from Xi'an Jiaotong University, P.R. China, in 1999, and the M.Eng. degree from Shanghai Jiao Tong University, P.R. China, in 2005, and the Ph.D. degree from Shanghai University, Shanghai, P.R. China, in 2013. She is now a associate professor with the College of Electronics and Information Engineering, Shanghai University of Electric Power. Her research interests include image processing and multimedia security.