Research Article

Image Hashing Algorithm Based on Mirror Flipping and a Three-Dimensional Space Angle

Yan Zhao and **Shuai Liu**

College of Electronics & Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China

Correspondence should be addressed to Yan Zhao; yanzhao79@hotmail.com

Received 29 April 2021; Revised 23 July 2021; Accepted 9 September 2021; Published 28 September 2021

Academic Editor: Xu Zheng

Copyright © 2021 Yan Zhao and Shuai Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to make full use of image flipping information to get comprehensive image features and improve the distinguishing performance of hash algorithm, this paper proposes a new image hashing algorithm based on mirror flipping and a three-dimensional space angle. Firstly, the original image is preprocessed and then combined with mirror flipping image to obtain the new luminance component and opposite color components. Then, we combine new luminance component with the different sizes of structural elements to construct morphological features. The new opposite color components are used to construct a three-dimensional space. The angle between vectors formed by the pixels in the three-dimensional space is computed to construct the space angle features. Finally, the morphological features and space angle features are combined and disturbed to form the final hash sequence. Simulation results show that the algorithm has good security and tamper image recognition accuracy. Compared with some existing algorithms, it has better image classification performance and shorter computation time.

1. Introduction

With the rapid development of Internet technology, multimedia information security is becoming more and more important in people's life. Image hashing is an important part of multimedia information security. Nowadays, individuals have the ability to easily and rapidly process the images on computers, mobile phones, and other digital devices. In addition, the modern image editing tools allow the individuals to temper the content of images. These changes are very hard to be distinguished by the naked eye. The analysis for detecting the tampered parts of the image requires expertise and time. Thus, there is a need to devise methods that allow the concerned individuals to assess the image authenticity and detect the changes. The image hashing technology is an important and effective method to solve the aforementioned problem.

Image hash refers to a unidirectional mapping of an image to a sequence of numbers. These numbers can be binary or decimal. The image hashing should be robustness, discrimination, and key secure. The robustness means that the hash sequence of the original image after processing should preserve the digital content and should be the sim-

ilar to the hash sequence of the original unprocessed image. The discrimination means that the hash sequences of different images should be significantly different when subject to same hash mapping; i.e., the probability of collision of hash sequences for different images should be very small. The key security refers to the failure of an attacker to obtain the correct hash sequence without knowing the correct key. These features enable the image hashing to be applied to image retrieval, image classification, and image tamper detection. The previous hashing algorithms use two-dimensional plane features or global features of a three-dimensional space. All of them have good performance. But it is noticeable that the algorithms based on a three-dimensional space may ignore some three-dimensional local features. And image mirror flipping has more useful image information. Therefore, image classification performance of existing image hash algorithms can be improved. In this paper, we combine the image flipping of the image component with the local features of the three-dimensional space to construct the hash sequence. This method makes full improvement of the classification performance of hashing algorithm.

2. Related Work

There are different methods proposed in recent literature for performing image hashing. These methods are majorly divided into two categories, namely, spatial domain methods and transform domain methods.

2.1. Spatial Domain Methods. In [1], Tang et al. proposed an image hashing algorithm based on local color features. The proposed technique extracts are featured from HSI space and YCbCr space, in order to obtain image hashing. The algorithm proposed in [1] is robust to most of the image processing operations; however, the algorithm performs poorly for the rotated images. The authors in [2] presented an algorithm that is based on opposite colors and quadtree decomposition of image hashing algorithm. The proposed algorithm effectively localizes the tampers in the image. The authors in [3] proposed an image hashing scheme that combines color vector angle (CVA) and Canny operator. The features obtained from the CVA and the edges are combined to compute the hash sequence. This algorithm performs efficiently against image rotation, brightness contrast adjustment, and other attacks. However, the discrimination of this technique is not ideal. Qin et al. in [4] proposed an image hashing technique based on hybrid structure of the color image perception. The proposed algorithm performs image preprocessing by performing image normalization, Gaussian low-pass filtering, and singular value decomposition (SVD). After preprocessing, the algorithm performs edge detection and computes the CVA. The edge information and the CVA are used to obtain the hash sequence. It is noticeable that as compared to the aforementioned CVA-based technique, this algorithm showcases better robustness and discrimination. However, this method is less robust to the changes in brightness. Tang et al. proposed an image hashing technique based on ring segmentation and invariant vector distance [5]. This technique extracts statistical features from the image segmentation ring to construct the image hashing, as the features extracted from within the image segmentation ring are invariant to rotation. The robustness of this algorithm can be further improved. In [6], Tang et al. proposed an image hashing technique that makes use of visual attention model and the invariant moments. In this method, the visual significance model and invariant moments are used to construct the hash sequence. This technique performs effectively against the conventional attacks, such as brightness and contrast, but performs poorly against image rotation and cropping attacks. In [7], Zhao et al. proposed an image hashing algorithm based on Zernike moments and significant regions. The method proposed in [7] performs authentication effectively and can be used for tampering and positioning. In [8], Davarzani et al. proposed a hashing scheme based on the central local symmetric binary mode (CS-LBP) and SVD. This method is robust against brightness and JPEG compression. In addition, this technique has the capability to locate tampering. However, this algorithm performs poorly against rotation.

2.2. Transform Domain Methods. In [9], Tang et al. proposed a robust image hashing based on CVA. This algorithm first extracts the CVA from the input image and then computes the pixel histogram from within the maximum inscribed circle of the image. The extracted information is then combined with the discrete cosine transform (DCT) for generating the image hashing. This algorithm is robust and secure; however, the different color may generate same CVA which affects the performance. In [10], Qin et al. proposed an image hashing technique that uses significant structural features. This technique performs efficient edge detection. On the basis of edge information, some nonoverlapping image blocks containing rich information are adopted. Finally, DCT is used to obtain the final hash sequence. The algorithm is secure and robust, but the scheme only uses some boundary information, and some local information may be omitted. This may lead to performance degradation. In [11], Tang et al. proposed an image hashing algorithm based on DCT and local linear embedding (LLE). This method first extracts the CVA of the image and then extracts the feature matrix on the basis of DCT. The hash sequence is computed by combining the LLE. This scheme is robust to various transformations but is not effective against rotations at large angles. In [12], Tang et al. proposed an image hashing technique that uses multidimensional scaling (MDS) analysis. In this algorithm, log-polar transform (LPT) and discrete Fourier transform (DFT) are successively applied to the image for extracting the rotation invariant feature matrix. The final hash sequence is obtained using the MDS. This algorithm is robust to rotations, but it is inefficient.

Moreover, there are some image hashing algorithms, such as algorithms based on tensor decomposition (TD) [13], methods based on Laplacian Pyramids [14], methods based on two-dimensional principal component analysis (PCA) [15], techniques based on the structure and gradient hashing algorithm [16, 17], and techniques based on the cool and warm hue [18].

All the above hashing algorithms have good performance, but none of them combine the image flipping, two-dimensional plane features, and three-dimensional space local features of image components to construct the hash sequence. In this work, we propose an image hashing algorithm based on the mirror flipping and a three-dimensional space angle. The proposed algorithms significantly consider the two-dimensional global features of luminance and opposite colors of three-dimensional space local features. The algorithm combines mirror flipping for the image components in a three-dimensional space of each small build specific three-dimensional space coordinates and extracts the features of a three-dimensional space angle. The proposed algorithm shows robustness and discrimination. In addition, the proposed algorithm shows the key security. The contributions of this work are as follows.

- (1) We lead the idea of image mirroring flipping into image hashing; hash sequence is constructed by combining the image component after the mirror flipping with the image component; the image after mirroring flipping can provide more useful

information, which is something that was never mentioned in the previous hashing algorithm. The experimental results show that the introduction of mirror flipping can improve the image classification and tamper detection performance

- (2) This algorithm is different from the previous hashing algorithm based on the three-dimensional space of image component [16, 17], which is mainly based on the global feature of a single image component. In this paper, the image component and the image component after mirror flipping are combined. Then, three-dimensional coordinates for each block are built to extract the local features of the image components in a three-dimensional space. Compared with previous algorithms based on a three-dimensional space, this algorithm can get more image information and extract more detailed image features in a three-dimensional space so as to improve the performance of the algorithm

Simulation results reveal that the proposed algorithm has better discriminability and better ROC curve as compared to other algorithms presented in literature. The average computation time of the proposed hashing algorithm is shorter.

3. Image Hashing Algorithm Based on Mirror Flipping and a Space Angle

3.1. Hashing Algorithm Framework. The flowchart of the proposed image hashing algorithm is presented in Figure 1. The major components of the proposed algorithm comprise image preprocessing, image mirror flipping and superposition of each component, extraction of morphological features, extraction of three-dimensional space angle features, and generation of hash sequence. In the following subsections, we discuss each component of the proposed algorithm in detail.

3.2. Image Preprocessing. At first, the input image I is resized to $N \times N$ by applying bilinear interpolation. This allows us to generate the hash sequence with same length for each input image. Then, we smooth the image by applying Gaussian low-pass filter. This is done in order to reduce the influence of noise. The resultant image I_0 is obtained for the corresponding input image I .

3.3. Image Mirror Flipping. In this work, we proposed a technique called mirror flipping. The mirror flipping refers to the process of flipping the left and right sides of the preprocessed image along the vertical axis. The image I_0 after mirror flipping is named I_1 . Figure 2(a) is the original image, and Figure 2(b) is the mirror flipping image.

Then, we compute the RG and BY colors [19, 20] by using the following equations.

$$RG = T_R(x, y) - T_G(x, y), \quad (1)$$

$$BY = T_B(x, y) - T_Y(x, y), \quad (2)$$

where $T_R(x, y)$, $T_G(x, y)$, $T_B(x, y)$, and $T_Y(x, y)$ are defined as equations [3–6].

$$T_R(x, y) = f_r(x, y) - \frac{f_g(x, y) + f_b(x, y)}{2}, \quad (3)$$

$$T_G(x, y) = f_g(x, y) - \frac{f_r(x, y) + f_b(x, y)}{2}, \quad (4)$$

$$T_B(x, y) = f_b(x, y) - \frac{f_r(x, y) + f_g(x, y)}{2}, \quad (5)$$

$$T_Y(x, y) = -T_B(x, y) - \frac{|f_r(x, y) - f_g(x, y)|}{2}, \quad (6)$$

where $f_r(x, y)$, $f_g(x, y)$, and $f_b(x, y)$ represent the red, green, and blue channels of RGB color space, respectively.

The image I_0 is obtained after preprocessing original image and used to extract the opposite color components RG_0 and BY_0 . I_1 is I_0 after mirror flipping; I_1 is used to extract the opposite color components RG_1 and BY_1 . We add RG_0 and RG_1 to compute the opposite color component RG_2 . Similarly, add BY_0 and BY_1 to compute the opposite color component BY_2 . Finally, we transform the images I_0 and I_1 from RGB color space to YCbCr color space. The luminance Y_0 and Y_1 are extracted from I_0 and I_1 . We add Y_0 and Y_1 to compute the luminance component Y_2 .

3.4. Morphological Features. We perform adaptive histogram equalization and luminance adjusted on the luminance component Y_2 to obtain Y_3 . Y_3 is used to perform the open operations, i.e., the construction of element specified as “diamond” and which has a distance “ d ” from the origin. When “ $d=1$ ”, the operation on Y_3 results in Y_4 . When “ $d=2$ ”, the operation performed on Y_3 results in Y_5 . Y_4 subtracts Y_5 obtains Y_6 . In this process, it is called the parameter “ $d=1, 2$ ”.

Finally, we divide the luminance component Y_6 into $M \times M$ nonoverlapping blocks of equal size. We compute the standard deviation $S_{i,j}$ of the pixels contained in each block, get the matrix S . The dimensions of matrix S are $M \times M$. Each row of the matrix is connected to the hash sequence $H_1(n)$, where $1 \leq n \leq M$.

3.5. Three-Dimensional Space Angle Features. Let (x, y) represent the pixel location and z represent the value of the pixel at location (x, y) . Then, considering the location variables, i.e., (x, y) , and the value variable, i.e., z , we can represent the image in a three-dimensional space. The RG_2 in a three-dimensional space are presented in Figure 3(a).

In this work, we divide the images RG_2 and BY_2 into $M \times M$ nonoverlapping blocks of equal size. In order to construct these images with a three-dimensional space, we consider the number of small block's rows i of the image along the x -axis, the number of small block's columns j of the image along the y -axis, and the average value of the pixels of images RG_2 and BY_2 in a small block $L_1(i, j)$ and $L_2(i, j)$ along the z -axis. Then, the coordinates of RG_2 and BY_2 images' small block in the i -th row and j -th column (i, j) in a three-dimensional space are $(i, j, L_1(i, j))$ and $(i, j$

FIGURE 1: The flowchart of the proposed image hashing algorithms.

FIGURE 2: Image mirroring flipping.

, $L_2(i, j)$). Figure 3(b) presents the coordinates of RG_2 's blocks of the image in a three-dimensional space.

The three-dimensional space angle features extracted from RG_2 are shown in Figure 4. The three-dimensional space angle features of the image are extracted from small blocks. These small blocks comprise the central pixel at location (i, j) and eight adjacent pixels. First, we estimate the coordinate of the block with position $(i, j - 1)$ and $(i, j + 1)$ represented by $a_{i,j}$ and $e_{i,j}$, respectively. Then, we calculate the vector $A_{i,j}$ by using these two coordinates, i.e., $A_{i,j} = e_{i,j} - a_{i,j}$. In a similar fashion, we estimate the coordinates of the little block with positions $(i + 1, j)$ and $(i - 1, j)$ repre-

sented by $b_{i,j}$ and $f_{i,j}$, respectively. The vector $B_{i,j}$ is calculated by using these two coordinates, i.e., $B_{i,j} = f_{i,j} - b_{i,j}$. Finally, we use the vector $A_{i,j}$ and the vector $B_{i,j}$ to calculate $\cos \alpha_{i,j}$ in a three-dimensional space.

$$\cos \alpha_{i,j} = \frac{A_{i,j} B_{i,j}}{|A_{i,j}| |B_{i,j}|}. \quad (7)$$

The matrix E with $(M - 2) \times (M - 2)$ is composed of $\cos \alpha_{i,j}$, where $2 \leq i \leq M - 1$ and $2 \leq j \leq M - 1$.

FIGURE 3: (a) The RG_2 images in a three-dimensional space. (b) The coordinates of RG_2 's blocks of the image in a three-dimensional space.

FIGURE 4: The three-dimensional space angle features extracted.

Similarly, we estimate the coordinates of the little block at $(i+1, j-1)$ and $(i-1, j+1)$ represented as $c_{i,j}$ and $g_{i,j}$, respectively. The vector $C_{i,j}$ is computed as $C_{i,j} = g_{i,j} - c_{i,j}$.

In the same way, we estimate the coordinates of the little block at $(i+1, j+1)$ and $(i-1, j-1)$ represented as $d_{i,j}$ and $h_{i,j}$, respectively. The vector $D_{i,j}$ is calculated as $D_{i,j} = h_{i,j} - d_{i,j}$. We use vector $C_{i,j}$ and vector $D_{i,j}$ to compute $\cos \beta_{i,j}$ in a three-dimensional space as follows:

$$\cos \beta_{i,j} = \frac{C_{i,j} D_{i,j}}{|C_{i,j}| |D_{i,j}|}. \quad (8)$$

The matrix F with $(M-2) \times (M-2)$ is composed of $\cos \beta_{i,j}$, where $2 \leq i \leq M-1$ and $2 \leq j \leq M-1$.

Finally, we add the two matrices E and F to obtain the resultant matrix J . The dimensions of matrix J are $(M-2) \times (M-2)$. Each column is successively connected to a vector $J(n)$. And in the same way, a three-dimensional space angle eigenvector $J_1(n)$ is extracted in BY_2 . We add $J(n)$ and $J_1(n)$ to obtain $J_2(n)$ and multiply every number in $J_2(n) \times 10$ to get the hash sequence $H_2(n)$, where $1 \leq n \leq (M-2)^2$.

3.6. Hash Generation. The morphological feature sequence H_1 and the three-dimensional space angle feature sequence H_2 are combined to form the intermediate hash sequence $H_m = [H_1 H_2]$. It is evident from Sections 3.4 and 3.5 that the resultant hash of the image obtained using the hash algorithm presented in this work has a length equal to $L_H = 2M^2 - 4M + 4$ decimal numbers. The intermediate hash sequence H_m is perturbed by the key to obtain the final hash sequence H .

4. Experimental Results and Analysis

The parameters that are used in this study and for performing simulations include 3×3 Gaussian low-pass kernel with normalized size $N = 256$ and standard deviation 1, $d = "1, 2"$. The numbers of image subblocks are $M \times M = 16 \times 16 = 256$. Thus, the hash length is $L_H = 2M^2 - 4M + 4 = 452$ decimal numbers. All experiments were implemented with MATLAB 2018b on a computer, which the CPU is Inter Core i5-8300H with 8 GB memory and Window 10 operating system.

FIGURE 5: Color image in robustness experiment.

TABLE 1: Attack setting in robustness experiment.

Operation	Parameter	Parameter values
JPEG compression	Quality	30, 40,...,100
Brightness adjustment	Level	-20, -10, 10, 20
Contrast adjustment	Level	-20, -10, 10, 20
Salt and pepper	Noise level	0.002, 0.004,...,0.01
Image resizing	Ratio	0.6, 0.8, 1.2, 1.4, 1.6, 1.8
3×3 Gaussian filter	Standard deviation	0.1, 0.2,...,1
Rotation	Angle	1, 2, 3,...,7, 8
Watermark embedding	Transparency	0.3–0.8
Gamma correction	Gamma	0.75, 0.9, 1.1, 1.25
Speckle	Noise variance	0.002, 0.004,...,0.01

4.1. Robustness Experiment. In order to evaluate the robustness of the image hashing algorithm proposed in this work, we use 20 images as test samples. Figure 5 represents one of the sample images. We also generate similar images for each test according to the type of attack as presented (Table 1); there are 59 attack samples for each image as test samples of similar images. We use the proposed algorithm to compute the hash sequence of the original image. Similarly, we also compute the hash sequence of a closely similar image. Then, we calculate the Euclidean distance between the two resultants. The Euclidean distance charts are drawn according to attack category. The simulation results for assessing robustness of the proposed algorithm are presented in Figure 6. It is evident from Figure 6 that the JPEG compression, brightness adjustment, contrast adjustment, salt and pepper noise, image scaling, watermark embedding, gamma correction, speckle, and Gaussian filtering have a small fluctuation range in the curve. The Euclidean distance in all cases is less than 165. In the simulations performed to test the algorithms robustness against the rotation attack, we observe that the distance of the rotation attack within 1° of

image rotation is small. The distance of the rotation attack gradually increases with the increase in the angle. This is because the proposed algorithm blocks the image, thus making it more sensitive to the rotation attack. In Table 2, we present the statistical results of Euclidean distance for 20 images under different attacks. The presented cases include JPEG compression, brightness adjustment, contrast manipulation, salt and pepper noise, image scaling, watermark embedding, rotation, gamma correction, speckle, and Gaussian filtering. In Table 3, we give the average mean square error (MSE) and peak signal-to-noise ratio (PSNR) values between the original images of five test images and the images generated after the corresponding maximum parameter attack in each attack. For all these attacks, except rotation, the maximum Euclidean distance is less than 165 and the mean and variance are less than 88. The maximum and average Euclidean distance for rotation attack is 226 and 318. The results show that the maximum value of the Euclidean distance is 165 except the rotation attack. This indicates that the proposed algorithm is robust against all types of attacks except large angle rotation attacks, such as JPEG

FIGURE 6: Robustness experiment.

compression, brightness adjustment, contrast adjustment, and other attacks.

4.2. Discrimination Experiment. We download 1000 images from Ground Truth database [21] and VOC2007 database

[22] for performing discrimination analysis of the proposed algorithm. The hash sequences of 1000 images are computed by using the algorithm proposed in this work. Then, we calculate the Euclidean distance between these 1000 images hash sequences. We generate a total of $C_{1000}^2 = 499,500$

TABLE 2: Euclidean distance in different attacks.

Operation	Min	Max	Mean	SD
JPEG	5.1312	75.6609	32.2408	19.9104
Brightness	31.5577	138.0417	61.4337	27.3783
Contrast	10.3443	91.6494	37.1287	22.6655
Salt and pepper	22.0341	89.7511	47.0000	17.1471
Image resizing	15.2572	86.3003	38.1603	18.6133
Gaussian filter	22.4072	54.2933	37.8813	9.3474
Rotation	66.5561	317.5768	225.6172	71.9825
Watermark	2.4377	33.4658	11.6641	9.6917
Gamma	45.7193	164.7039	87.4441	32.2778
Speckle	17.8794	87.5365	48.7098	18.4361

TABLE 3: Mean value of MSE and PSNR.

Operation	Parameter values	MSE	PSNR
JPEG compression	100	2.7567	49.7483
Brightness adjustment	20	260.3211	28.8134
Contrast adjustment	20	11.3999	42.6260
Salt and pepper	0.01	109.7289	32.5156
Image resizing	1.8	38.2634	38.6182
3 × 3 Gaussian filter	1	83.9163	35.0653
Rotation	8	3805.8	17.1840
Watermark embedding	0.8	2.4291	49.6675
Gamma correction	1.25	265.2298	28.7490
Speckle	0.01	120.7298	32.2695

different image pairs. The similar image pairs are generated according to Table 4.

Each downloaded image is used to generate 20 similar image pairs. The Euclidean distance between the similar generated images and the original image is calculated. There are $(21 \times 20)/2 \times 1000 = 210000$ pairs of similar images. The distribution of Euclidean distances for similar and different images is presented in Figure 7 for different Euclidean distances and ordinate for the different frequencies of Euclidean distance; find the minimum Euclidean distance between different images by a calculation value of 274.06, similar image on the maximum Euclidean distance of 344.96, and similar and different images of Euclidean distance intersection between 274.06 and 344.96. In order to obtain the optimal value for distinguishing similar image pairs from different image pairs in this work, we use collision rate and error detection rate to perform analysis. The collision rate and error detection rate are computed as follows:

$$P_C = \frac{N_C}{N_D}, \quad (9)$$

$$P_E = \frac{N_E}{N_S}, \quad (10)$$

TABLE 4: Operations and parameter setting in discrimination experiment.

Operation	Parameter	Parameter values	
JPEG compression	Quality factor	40	80
Brightness adjustment	Level	20	20
Contrast adjustment	Level	20	20
Salt and pepper	Noise level	0.002	0.006
Image resizing	Ratio	0.8	1.6
3 × 3 Gaussian filter	Standard deviation	0.2	0.6
Rotation	Angle	1	2
Watermark embedding	Transparency	0.3	0.8
Gamma correction	Gamma	0.75	1.25
Speckle	Noise variance	0.002	0.006

where N_C and N_D represent the number of different image pairs detected as similar image pairs and the total number of different image pairs, respectively. Similarly, N_E and N_S represent the number of similar image pairs detected as different image pairs and the total number of similar image pairs, respectively. P_C and P_E represent the collision rate and error detection rate, respectively.

The computation results are presented in Table 5. The results presented in Table 5 reveal that the threshold value is 322, and at this point, the algorithm has a low collision rate and a low error detection rate, with good discrimination.

4.3. Key-Dependent Security. For any secure hashing algorithm, each different key results into a different hashing sequence; i.e., the probability of sequence collision is minimum. Figure 8 illustrates the key security of the hash algorithm in this work. The x -axis represents the index of 1000 wrong keys, and the y -axis represents the Euclidean distance. It is evident from the figure that the minimum value of Euclidean distance and the average value of all error keys are 345.2521 and 376.8121, respectively. It is noticeable that both of these quantities are significantly greater than the threshold value of 322 as discussed above. The threshold is presented in the red line in Figure 8. This means that it is difficult for an attacker to generate the correct hash sequence without the correct key. Therefore, the hash algorithm proposed in this work satisfies the security requirements.

4.4. Performance Comparison of Different Blocks, Different Mirror Flipping, and Different Components. In order to analyze the influence of the number of blocks on the performance of the algorithm, we use receiver operating characteristics (ROC) curve to compare the performance, which mainly includes robustness and discrimination. We set a series of different thresholds to obtain a series of false positive rate (P_{FPR}) and the true positive rate (P_{TPR}) in order to draw the ROC curve. The simulation results are presented in Figure 9. The abscissa represents the P_{FPR} and ordinate represents the P_{TPR} . The P_{FPR} and P_{TPR} are computed as follows:

FIGURE 7: The hash distance distribution of similar image pairs and different image pairs.

TABLE 5: Threshold selection.

Threshold	Collision rate	Error detection rate
274	0	5.16×10^{-3}
286	4.00×10^{-6}	2.45×10^{-3}
301	3.40×10^{-5}	9.52×10^{-4}
313	1.30×10^{-4}	3.00×10^{-4}
322	3.64×10^{-4}	5.24×10^{-5}
334	1.20×10^{-3}	9.52×10^{-6}
345	3.21×10^{-3}	0

$$P_{\text{FPR}} = \frac{n_1}{N_1}, \quad (11)$$

$$P_{\text{TPR}} = \frac{n_2}{N_2}, \quad (12)$$

where n_1 represents the number of different image pairs misjudged as similar image pairs, n_2 represents the number of similar image pairs correctly judged, and N_1 and N_2 denote the total number of different image pairs and similar image pairs, respectively. The horizontal axis represents the discrimination performance, while the vertical axis represents the robustness performance. The closer the ROC curve is to the upper left corner, the better is the performance of the proposed algorithm.

In the simulations, we use 1000 images for discrimination test. According to the attack settings presented in Table 4, we generate visually similar images. Each image is used to generate 20 similar images. Therefore, the total number of similar image pairs is 210,000, and the number

of different image pairs is 499,500. When the image is divided into 8×8 , 16×16 , and 32×32 blocks, the resultant hash code length of the proposed algorithm is 100 decimal numbers, 452 decimal numbers, and 1924 decimal numbers, respectively. When the image is divided into 32×32 blocks, the resultant hash sequence length is too long. Therefore, in the simulations, we only evaluate the performance of hash algorithm whose image is divided into 8×8 blocks and image flipping along the vertical axis 16×16 blocks and image flipping along the vertical axis 16×16 blocks, but the algorithm does not add the image component after mirror flipping (NAF), flipping the up and down sides of the preprocessed image along the horizontal axis (UDH), and the image component L in the lab space is used to replace the brightness component Y . As presented in Figure 9, the ROC curve of the hash algorithm divided into 16×16 blocks is closer to the upper left corner than other hash algorithm. Thus, the performance of the former is better than the latter. In this work, the hash algorithm divides the image into 16×16 blocks and image flipping along the vertical axis, which is an effective choice for the balance between robustness and discrimination.

4.5. Performance Comparison of Different Parameters. In order to analyze the influence of the parameters on the performance of the proposed algorithm, we use ROC curve to compare the performance. This major component includes robustness and discrimination. A series of different thresholds are set to obtain a series of P_{FPR} and the P_{TPR} . These computed parameters are then used to draw the ROC curve. The simulation results are presented in Figure 10. In Figure 10, the abscissa represents the P_{FPR} and the ordinate represents the P_{TPR} . The computations

FIGURE 8: The distance between hashes with the correct secret key and 1000 wrong secret keys.

are performed by the mathematical relations presented in Equations (11) and (12).

In the experiment, we use 1000 images for performing discrimination tests. According to the attack situations presented in Table 4, we generate visually similar images. Each image is used to generate 20 similar images. Therefore, the total number of similar image pairs is 210,000, and the number of different image pairs is 499,500. In this section, we discuss the influence on the performance of the algorithm when the parameter “ d ” takes different values, where “ d ” represents the distance from the origin of elements to the four angles of the diamond. In this section, we compare three pairs of parameters: $d = “1, 2”$, $d = “2, 3”$, and $d = “3, 4”$, respectively. As presented in Figure 10, the ROC curve of the hash algorithm with a parameter set $d = “1, 2”$ is closer to the upper left corner as compared to the hash algorithm with parameter sets $d = “2, 3”$ and $d = “3, 4”$. In this work, the hashing algorithm chooses to set the hyper pixel segmentation parameter $d = “1, 2”$, which is a good choice to balance between robustness and discrimination.

4.6. Performance Comparison of Different Algorithms. We compare the performance of the image hashing algorithm proposed in this work with the previous works presented in literature, such as CVA-Canny algorithm [3], the Ring algorithm [5], TD algorithm [13], LP algorithm [14], and color structure algorithm [17]. The parameters of the comparison are equivalent to the parameter values selected in the respective published papers. As shown in Table 6, the hash length of the algorithm compared is 40 decimal numbers, 440 binary numbers, 96 binary numbers, 1024 decimal numbers, and 354 binary numbers, respectively. The hash code length of the proposed hash algorithm is 452 decimal numbers.

In this simulation, the hash algorithm is used to extract the hash sequences of similar image pairs and different

image pairs. The Euclidean distance between these image pairs is computed. The attack settings for similar images are also kept the same as presented in Table 4. There are 210,000 similar image pairs and 499,500 different image pairs. A series of different thresholds are set to obtain a series of P_{FPR} and the P_{TPR} to draw the ROC curve. The simulation results are presented in Figure 11. It is evident that the curve of the proposed algorithm in this work is closer to the upper left corner than the ROC curves of other algorithms. This indicated that the classification ability of the proposed algorithm is better as compared to previous algorithms presented in literature.

In this work, the luminance component Y is extracted and the color components RG and BY are partitioned for feature extraction. Therefore, the hash sequence length is relatively long. However, the average computation time of this algorithm is only longer than the color structure and LP algorithm, which is superior to the remaining comparison algorithms.

4.7. Tamper Detection Experiment. When a part of the image is tampered, the Euclidean distance between the tampered image and the hash sequence of the original image should be between the distance from the similar image and the distance from the different image. In order to verify the tamper detection capability of the proposed algorithm, we obtain 15,000 images from VOC2012 database [23]. Each image in the dataset is tampered to perform simulations. Figure 12 presents an example of a tampered image. The Euclidean distance distribution diagram of similar image pairs, tampered image pairs, and different image pairs is presented in Figure 13(a). The x -coordinate T_1 at the intersection of similar image pairs and tampered image pairs is 136, and the x -coordinate T_2 at the intersection point of tampered image pair and different image pair is 301. We select T_1 and T_2 as thresholds; when the distance between the detected image and the original image is less than T_1 ,

(a)

(b)

(c)

FIGURE 9: Continued.

FIGURE 9: (a) Original image. (b) Flipping along the vertical axis. (c) Flipping along the horizontal axis. (d) The ROC curves of different blocks, different mirror flipping, and different components.

FIGURE 10: ROC curves of different parameters.

TABLE 6: Comparison.

Algorithm	Our algorithm	CVA-Canny [3]	Ring [5]	TD [13]	LP [14]	Color structure [17]
Average time (s)	0.040	0.304	0.211	0.098	0.028	0.018
Hash length	452 decimal digits	40 decimal digits	440 bits	96 bits	1024 decimal digits	354 bits

FIGURE 11: ROC curves of different algorithms.

FIGURE 12: Original image and tampered image.

the target image is regarded to be similar, and when the distance between the image to be detected and the original image is greater than T_1 and less than T_2 , the image is

regarded as tampered image. When the distance between the image to be detected and the original image is greater than T_2 , the image to be detected is regarded as a different

FIGURE 13: (a) Hash distance distribution after adding mirror flipping. (b) Does not add mirror flipping's hash distance distribution.

image. The simulation results show that the proposed algorithm has 80.26%, 96.37%, and 99.99% accuracy in judging similar images, tampered images, and different images, respectively. Figure 14 shows several pairs of contrast image pairs. Similarly, Table 7 shows the Euclidean distance and relationship between several pairs of contrast image pairs.

The contrast image includes similar images, tampered images, and different images, and the relationship between the original image and the contrast image can be judged on the basis of Euclidean distance. It is evident that the relationship between the two images is accurately judged according to the Euclidean distance between the original

FIGURE 14: Original image and tampered image.

TABLE 7: Comparison of the Euclidean distance and relationship between image pairs.

Image pairs	a_1 with b_1	a_2 with b_2	a_3 with b_3	a_4 with b_4	a_5 with b_5	a_6 with b_6	a_7 with b_7
Euclidean distance	281.85	71.76	197.48	410.76	225.58	279.51	142.35
Judge relationship	Tamper	Similar	Tamper	Different	Tamper	Tamper	Tamper

TABLE 8: Recognition accuracy rate.

Algorithm	Accuracy rate of similar image	Accuracy rate of tamper image	Accuracy rate of different image
Mirror flipping	80.26%	96.37%	99.99%
Not mirror flipping	72.34%	96.60%	99.99%

image and the comparison image. This indicates that the algorithm presented in this work has good tampering detection performance.

Just to prove the advantage of mirror flipping, we present the performance of the algorithm with and without mirror flipping. The database used for this simulation is the same as above. In this work, mirror flipping is not performed in the step of “mirror flipping” in Figure 1. The luminance components extracted from two identical images superimposed together, and the opposite color components are extracted from two identical images superimposed together for performing calculation. In this way, the variables are controlled and the comparison is more reliable. The distribution diagram is shown in Figure 13(b). The x -coordinate T_1 at the intersection of similar image pairs and tampered image pairs is 111. The x -coordinate T_2 at the intersection point of tampered image pair and different image pair is 310. We select T_1 and T_2 as thresholds. When the distance between the detected image and the original image is less than T_1 , the image to be detected is judged to be similar. When the distance between the image to be detected and the original image is greater than T_1 and less than T_2 , the image to be detected is judged as tampered image. When the distance between the image to be detected and the original image is greater than T_2 , the image to be detected is judged to be different image. The simulation

results have shown that the proposed algorithm has an accuracy of 72.34%, 96.60%, and 99.99% without adding mirror flipping in the case of similar images, tampered images, and different images, respectively.

As shown in Table 8, comparing algorithm with join mirror flipping that did not join the mirror rotating algorithm in recognition tamper or different images in accuracy is similar, but the recognition similar image accuracy of 72.34% is significantly less than the join mirror flipping to identify similar image accuracy of 80.26%, so this article chooses to mirror flipping algorithm.

5. Conclusions

This work proposes an image hashing algorithm based on mirror flipping and space angle. The new luminance component and the opposite color component are obtained by mirror flipping. We extract the morphological features by the new luminance component. In addition, the space angle features are also extracted by the new opposite color component. Finally, the hash sequence is obtained by combining the morphological and space angle features. Simulation results show that the proposed algorithm is robust against brightness attack, contrast attack, and other attacks. The ROC curve shows that the algorithm presented in this work has better image classification performance than the previous algorithms proposed in literature. In addition, the proposed algorithm has good key security and good detection accuracy of tampered images.

But the algorithm in this paper has poor robustness to large angle rotation and can only detect a large range of tampering for the time being. So, our future work will focus on how to improve robustness to large angle rotations, reduce the storage space of hash sequence, and locate tampering region.

Data Availability

The image datasets used to support the findings of this study can be downloaded from the public websites whose links are provided in this paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant (61802250) and the Local College Capacity Building Project of Shanghai Municipal Science and Technology Commission (20020500700).

References

- [1] Z. Tang, X. Zhang, X. Dai, J. Yang, and T. Wu, "Robust image hash function using local color features," *AEU-International Journal of Electronics and Communications*, vol. 67, no. 8, pp. 717–722, 2013.
- [2] Q. Shen and Y. Zhao, "Perceptual hashing for color image based on color opponent component and quadtree structure," *Signal Processing*, vol. 166, p. 107244, 2020.
- [3] 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.
- [4] C. Qin, M. Sun, and C.-C. Chang, "Perceptual hashing for color images based on hybrid extraction of structural features," *Signal Processing*, vol. 142, pp. 194–205, 2018.
- [5] 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, 2016.
- [6] Z. Tang, H. Zhang, C. Pun, M. Yu, C. Yu, and X. Zhang, "Robust image hashing with visual attention model and invariant moments," *IET Image Processing*, vol. 14, no. 5, pp. 901–908, 2020.
- [7] Y. Zhao, S. Wang, X. Zhang, and H. Yao, "Robust hashing for image authentication using Zernike moments and local features," *IEEE Trans Inform Forensic Secur*, vol. 8, no. 1, pp. 55–63, 2013.
- [8] R. Davarzani, S. Mozaffari, and K. Yaghmaie, "Perceptual image hashing using center-symmetric local binary patterns," *Multimedia Tools and Applications*, vol. 75, no. 8, pp. 4639–4667, 2016.
- [9] Z. Tang, X. Li, X. Zhang, S. Zhang, and Y. Dai, "Image hashing with color vector angle," *Neurocomputing*, vol. 308, pp. 147–158, 2018.
- [10] 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.
- [11] Z. Tang, H. Lao, X. Zhang, and K. Liu, "Robust image hashing via DCT and LLE," *Computers & Security*, vol. 62, pp. 133–148, 2016.
- [12] Z. Tang, Z. Huang, X. Zhang, and H. Lao, "Robust image hashing with multidimensional scaling," *Signal Processing*, vol. 137, pp. 240–250, 2017.
- [13] Z. Tang, L. Chen, X. Zhang, and S. Zhang, "Robust image hashing with tensor decomposition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 549–560, 2019.
- [14] H. Hamid, F. Ahmed, and J. Ahmad, "Robust image hashing scheme using Laplacian pyramids," *Computers & Electrical Engineering*, vol. 84, article 106648, 2020.
- [15] X. Liang, Z. Tang, X. Xie, J. Wu, and X. Zhang, "Robust and fast image hashing with two-dimensional PCA," *Multimedia Systems*, vol. 27, no. 3, pp. 389–401, 2021.
- [16] Q. Shen, Y. Zhao, X. Zhou, and X. Yuan, "Image hashing algorithm based on structure and gradient," *Journal of Zhejiang University (Engineering Science)*, vol. 54, no. 8, pp. 1525–1533, 2020.
- [17] Y. Zhao and X. Yuan, "Perceptual image hashing based on color structure and intensity gradient," *IEEE Access*, vol. 8, pp. 26041–26053, 2020.
- [18] Y. Zhao and S. Liu, "Robust image hashing based on cool and warm hue and space angle," *Security and Communication Networks*, vol. 2021, Article ID 3803481, 13 pages, 2021.
- [19] S. Engel, X. Zhang, and B. Wandell, "Colour tuning in human visual cortex measured with functional magnetic resonance imaging," *Nature*, vol. 388, no. 6637, pp. 68–71, 1997.
- [20] C. P. Yan, C. M. Pun, and X. C. Yuan, "Quaternion-based image hashing for adaptive tampering localization," *IEEE Trans Inform Forensic Secur*, vol. 11, no. 12, pp. 2664–2677, 2016.
- [21] "Ground Truth Database," 2008, October 2017, <http://www.cs.washington.edu/research/imagedatabase/groundtruth/>.
- [22] V. O. C. Pascal, "Data set," 2007, October 2017, <https://pjreddie.com/projects/pascal-voc-dataset-mirror/>.
- [23] R. J. Pascal, "VOC dataset mirror [EB/OL]," 2012, <https://pjreddie.com/projects/pascal-voc-dataset-mirror/>.