International Journal of Web Services Research
Volume 19 • Issue 1

Strong Robustness Watermarking Algorithm Based on Lifting Wavelet Transform and Hessenberg Decomposition

Fan Li, Sichuan University, China
Lin Gao, Chengdu University of Information Technology, China*
Junfeng Wang, Sichuan University, China
Ruixia Yan, Southwest University of Science and Technology, China

ABSTRACT

Watermark imperceptibility and robustness in the present watermarking algorithm based on discrete wavelet transform (DWT) could be weakened due to data truncation. To solve this problem, a strong robustness watermarking algorithm based on the lifting wavelet transform is proposed. First, the color channels of the original image are separated, and the selected channels are processed through lifting wavelet transform to obtain low-frequency information. The information is then split into blocks, with Hesseneberg decomposition performed on each block. Arnold algorithm is used to scramble the watermark image, and the scrambled watermark is transformed into a binary sequence that is then embedded into the maximum element of Hessenberg decomposed matrix by quantization modulation. The experimental results exhibit a good robustness of this new algorithm in defending against a wide variety of conventional attacks.

KEYWORDS

Digital Image Watermarking, Fruit Fly Optimization Algorithm, Hessenberg Decomposition, Lifting Wavelet Transform

1. INTRODUCTION

As a new means of digital image copyright protection, image digital watermarking technology is increasingly connected with various facets of society and plays a decisive role in medical images, commercial publicity, industrial production, information security, etc. The embedding domain of watermarks is generally divided into three kinds: spatial domain, frequency domain, and compression domain. The watermark embedding algorithm based on the spatial domain (Wu et al., 2021) features a low time complexity and a large watermark capacity due to its simplicity and capability to avoid changing the original image. However, as a conventional semi-fragile watermarking algorithm, it only provides adequate resistance against image compression attacks but is less robust against noise disturbances. Frequency-domain-based watermark embedding algorithms (Wu et al., 2021), such as discrete cosine transform, wavelet transform, contourlet transform, shear wave transform, vector

DOI: 10.4018/IJWSR.314948 *Corresponding Author

transform, and Hadamard transform, require specific alteration of the original image before embedding the watermark, and the watermark can be embedded by modifying, replacing, or exchanging the frequency band coefficients of quality images. In general, the low-frequency section contains important contour information of the image, while the high-frequency section contains redundant details of the image. If robustness is emphasized, it would be suitable to embed the watermark into low-frequency information, while if invisibility and embedding capacity are regarded as more important, it would be suitable to embed the watermark information into the high-frequency section.

Various suggestions have been proposed to improve the invisibility and robustness of after marking algorithms. Roy and Pal (2019) proposed a hybrid robust image watermarking scheme based on discrete wavelet transform (DWT) and singular value decomposition (SVD), DWTSVD. The Y-component of the image is decomposed into non-overlapping blocks, and the watermark is embedded into the singular values of these image blocks after DWT transform and SVD decomposition. This approach yields better robustness against geometric transformation attacks, enhancement attacks, and combination attacks. Naik et al. (2018) came up with a key matrix, integer wavelet, binary watermark embedding method based on logistic mapping to increase operational efficiency and enhance the security of watermark images. Makbol et al. (2017) presented an SVD-based image watermarking scheme to effectively overcome false positive probability (FPP) by using integer wavelet transform (IWT) and obtaining additional secret keys from the watermark. Thakkar and Srivastava (2021) disclosed a DWT-SVD watermarking algorithm with enhanced robustness and invisibility by using particle swarm optimization (PSO) for self-adaptation. Soleymani et al. (2019) proposed a fast discrete curvelet transform blind watermarking method (FDCuT) by adopting Arnold transform to preprocess the host image and using a strong gain factor to increase the security and imperceptibility of watermark embedment. In addition, two weakly correlated pseudo-noise strings used as symbols for each 0 or 1 bit of the watermark are highly resistant to common image watermarking attacks, such as noise, compression, and image quality enhancing. Nazari and Mehrabian (2021) introduced a secure blind watermarking algorithm based on integer wavelet transform (IWT), least significant bit (LSB), and chaotic sequence (iterative learning control, ILC) to increase the capacity and security of watermark embedment. Agarwal et al. (2015) combined watermarking and biometrics to improve owner identification/verification techniques based on discrete wavelet transform (DWT), and proposed and compared four blind invisible watermarking approaches based on face images and wavelets. The four methods offer respective advantages for different types of attacks. Khare and Srivastava (2021) suggested a new image watermarking technique that yields high robustness against most conventional attacks by incorporating DWT, homomorphic transform (HT), and SVD (DWTHTSVD). The algorithm employed in this article (Abdulrahman & Ozturk, 2019; Budiman et al., 2017; Li et al., 2021; Veni & Meyyappan, 2017; Wang et al., 2020) makes watermarks far more robust by combining discrete cosine transform (DCT) and DWT, giving full play to the strengths of both with advantages over either one individually. In this article (Dhar et al., 2020; Su et al., 2017), Hessenberg decomposition is combined with nonsubsampled contourlet transform (NSCT) and non-nonsubsampled shearlet transform (NSST), which are more superior in resisting attacks such as rotation and shear. Vaidya et al. (2019) used game theory to design a robust watermarking scheme, which has good robustness against various signal processing attacks. Hsu and Tu (2020) used the sine function-based embedding rule to embed multiple independent watermarks into different image blocks, which has excellent robustness against clipping attacks. Meng et al. (2021) proposed an adaptive reversible watermarking algorithm in the integer wavelet transform (IWT) domain. It can be embedded adaptively according to the size of the watermark and achieve completely lossless blind extraction. In order to improve the ability to resist geometric rotation attacks and solve the problem of high computational complexity of existing watermarking algorithms, Yang et al. (2022) proposed a robust zero-watermarking algorithm in spatial domain, which is robust to geometric rotation attacks and common image processing attacks, and has low time complexity. Bose and Maity (2022) suggested a *sparse* watermark image embedding scheme on the singular values of the pre-selected wavelet

sub-band coefficients of the digital images. Dictionary learning (DL) method is used to make the watermark image sparse. Then a watermark decoder is designed based on the theory of compressed sensing (CS) in the framework of alternating direction method of multiplier (ADMM).

Most of the existing DWT methods rely on Fourier transform, and discrete wavelet operation calculates continuous type data while the image is stored in the form of discrete data in the computer, consequently resulting in some accuracy loss during watermark embedment and affecting the robustness of the embedded watermark. Lifting wavelet transform (LWT) is an improvement of DWT with speedy computation and does not rely on Fourier transform, guaranteeing image accuracy needed to improve watermark embedding robustness and invisibility. In this article, the authors propose a lifting wavelet watermarking algorithm with stronger robustness based on satisfying fundamental requirements in invisibility and watermark capacity. This method optimizes image distortion caused by DWT data truncation and enhances the fidelity of images containing watermarks. Embedding the watermark into the Hessenberg decomposition's (HD) maximum element increases watermark resistance against external disturbances. The FOA adaptive search for optimal embedding strength balances the imperceptibility and robustness of the watermark. The experiments indicate that this method based on lifting wavelet transform and Hessenberg decomposition (LWTHD), in contrast to the present DWT watermarking algorithm, is more robust in terms of resistance against a wide range of attacks such as rotation, filtering, noise, and cropping, and has low computational complexity, with good practical application value.

2. RELATED WORK

2.1 Lifting Wavelet Transform (LWT)

DWT decomposes the image into four frequency bands, namely LL, LH, HH and HL, wherein L denotes low-pass filter and H high-pass filter. The low-frequency sub-band LL concentrates most of the image energy, and is thus called approximation subgraph. The medium-frequency sub-band LH represents details of the original image in horizontal and vertical directions, and the high-frequency sub-band HH represents details of the original image in the diagonal direction, known as detail subgraph. The multi-resolution decomposition feature of wavelet transform allows it to possess an excellent spatial direction selectivity, and is similar to the human visual system (HVS) which is more sensitive to low-frequency information and less sensitive to texture changes. In this article, wavelet decomposition is first performed on the carrier image. Figure 1(a) is the original host image Lenna; the upper left, upper right, lower left, and lower right of Figure 1(b) represent the low-frequency sub-band, horizontal high-frequency sub-band, vertical high-frequency sub-band, and diagonal high-frequency sub-band of Lenna respectively after the first-level wavelet decomposition.

Figure 1. Wavelet Decomposition of Image Lenna: (a) Original Host Image Lenna, (b) Four Sub-Bands After the First-Level Wavelet Transform of Lenna

Lifted wavelet transform (LWT) is a special form of discrete wavelet transform (DWT). LWT possesses both the advantages of DWT and properties that DWT lacks. On the one hand, the conventional Fourier transform-dependent DWT algorithm is computationally slower than the LWT algorithm, which does not rely on Fourier transform, so LWT is more suitable for embedding and extracting watermarks. On the other hand, discrete wavelet outputs floating points, resulting in rounding errors in coefficient quantification. Quality of the image when reconstructed is related to the way the boundary is treated, and the grayscale values of the image express kernel storage in the form of integers (Su, 2015). In contrast, LWT can directly map the data into integers and avoid rounding errors. In this paper, the QR code needs to be embedded in the low frequency sub-band of LWT. Compared with the traditional DWT-based watermarking algorithm, it can not only ensure the invisibility of the watermark, but also improve the robustness of the watermark and the accuracy of watermark detection. The time complexity of LWT is $O\left(N \cdot \lg\left(N\right)\right)$, and the computational complexity is linearly related to the signal length, which is less time-consuming and can meet real-time computing requirements. Through splitting, prediction, and uploading, LWT realizes the transformation of a series of digital signals as follows:

Splitting: The original signal S_i is split into two mutually disjoint subsets, i.e, odd subset S_{i-1} and even subset d_{i-1}, which can be expressed as:

$$F\left(S_i\right) = \left(S_{i-1}, d_{i-1}\right) \tag{1}$$

Prediction: Usually, the two sets above are predictable of each other. There is no possibility of an errorless deduction of d_{i-1} from S_{i-1}, but since $P\left(S_{i-1}\right)$ is possibly very close to d_{i-1}, the original d_{i-1} can be replaced by the difference between d_{i-1} and $P\left(S_{i-1}\right)$. In doing so, the resulting d_{i-1} will contain less information than the original d_{i-1}, as in the equation below:

$$d_{i-1} = d_{i-1} - P\left(S_{i-1}\right) \tag{2}$$

where P denotes the prediction operator, which needs to consider the original signal's characteristics and reflects the interrelationship between the data.

Update: An update is required for the subset S_{i-1} to keep certain local characteristics. Finding a better subset so that it maintains the same properties as a certain scalar property of the original graph can be expressed as:

$$S_{i-1} = S_{i-1} + U\left(d_{i-1}\right) \tag{3}$$

where U is the update operator. The odd subset S_{i-1} becomes a high-frequency component and the even subset d_{i-1} becomes a low-frequency component after IWT transformation. The high frequency component contains more detailed information, while the low frequency component contains more contour information that can be further decomposed.

2.2 Hessenberg Decomposition

Definition: Matrix $H = \left(h_{x,y}\right)_{n\times n}\left(n \in \mathbb{R}\right)$. If the elements of H satisfy $h_{x,y} = 0(y > x+1)$, then H is an upper Hessenberg matrix in the form of Equation (4):

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \cdots & h_{n,n} \end{bmatrix} \tag{4}$$

Theorem: Let $A = \left(a_{x,y}\right)_{n\times n}\left(n \in \mathbb{R}\right)$, then there is an orthogonal matrix $P_1, P_2, \cdots, P_{n-2}$ such that A is transformed into an upper Hessenberg matrix by orthogonal similarity, that is, $P_{n-2}\cdots P_2 P_1\ A\ P_1 P_2 \cdots P_{n-2} = H$ holds.

Equation (5) is the Hessenberg decomposition formula for an $n \times n$ matrix where P is a unit orthogonal matrix and H is an upper triangular Hessenberg matrix:

$$\left[P, H\right] = hess\left(A\right) \tag{5}$$

Research shows (Han et al., 2017) that in the *H* matrix, $h\left(2,2\right)$ concentrates most of the energy information of the image with good numerical stability against external interference. Therefore, the algorithm embeds the watermark information into this element to improve the robustness of the watermark. Since Hessenberg decomposition is an important step of QR decomposition, its time complexity is less than $O\left(2/3n^3\right)$, which is lower than the traditional watermarking algorithm based on singular value decomposition or Schur decomposition.

2.3 Fruit Fly Optimization Algorithm (FOA)

In watermark embedding, the choice of embedding strength directly affects the performance of the watermark. If embedding strength is too high, the invisibility of the watermark is poor, and it will cause more damage to the host image. If embedding strength is too low, the watermark is not robust enough to withstand attacks. In this article, we use the fruit fly optimization algorithm (FOA) to find the optimal embedding strength factor to strike a balance between invisibility and robustness of the watermark. Compared with traditional particle swarm algorithm, ant colony algorithm, etc., FOA algorithm has the advantages of simple and easy implementation, low time complexity and high optimization accuracy, so it is easier to solve practical problems.

Fruit flies have a better sense of smell and vision than other species and can sensitively perceive various odors floating in the air and fly toward food sources. The FOA algorithm is a simulation of the foraging behavior of fruit flies, which is simple to implement and can quickly find the global optimum. The basic procedure of the algorithm is as follows:

1. The locations of the fruit fly population are randomly initialized. Seeking the direction and distance of foraging.

2. Since there is no way to know the location of food source from the start, the distance between the point and the origin is estimated first, and then the gustation concentration determination value is set to S. In this article, this value is set as the embedding strength factor of the watermark.
3. The value S is substituted into the gustation concentration determination function to find the gustation concentration of individual fruit fly locations. In this experiment, the gustation concentration determination function is set as Equation (6):

$$O_2 = \frac{10}{PSNR} + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(W\left(i,j\right)\times W'\left(i,j\right)\right)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(W'\left(i,j\right)\times W'\left(i,j\right)\right)} \tag{6}$$

where $PSNR$ is peak signal-to-noise ratio $\left(PSNR\right)$, which is used to measure the peak error between the original host image and the image embedded with a watermark. The larger the value, the less noise contained in the carrier, the less visual distortion of the image, the more invisible the embedded watermark. $W\left(I,j\right)$ and $W'\left(i,j\right)$ represent the pixels of the original image and watermark-embedded image, respectively; N is the image size, and $\left(i,j\right)$ represents image pixel position coordinates.

4. Fruit flies with the highest gustation concentration in the population are identified by finding the maximum value and recording the coordinates of the location.
5. The fruit fly population uses visual perception to find optimization in that direction.
6. Repeat steps (2)-(4) to determine whether the gustation concentration is better than that obtained from the previous iterations. If so, perform (5) until the global optimal concentration is found.

3. WATERMARKING DESIGN

3.1 Watermark Image Pre-Processing

In digital watermarking, it is not secure to only encrypt a watermark with an encryption algorithm, because watermark information can be extracted once an attacker breaks the encryption algorithm. Therefore, the watermark image needs to be scrambled first to make it look disorganized, thus augmenting the security of the information being concealed. Usually, digital image scrambling methods can be classified as linear transformations, geometric transformations, and affine transformations. The most widely used algorithms are Arnold, Fibonacci, Gray, Hilbert curve, magic square, E-curve, affine, and orthogonal Latin square transforms. All of these scramling algorithms are reversible, i.e., it is possible to revert to the original image after several inverse transformations. In this article, Arnold scrambling algorithm is employed. The Arnold transform (Kumar & Singh, 2021), also known as cat face transform, is mapping from a regular position to a random position. For an image of size $M\times M$, the Arnold transform and the inverse transform are performed as (7) and (8), respectively:

$$\begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}1 & 1\\1 & 2\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}\left(\mathrm{mod}\, N\right) \quad x,y\in\left\{0,1,\cdots,N-1\right\} \tag{7}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \end{bmatrix} \left(\mathrm{mod}\, N\right) \tag{8}$$

where $\left(x, y\right)^T$ is the pixel of the original image, $\left(x', y'\right)^T$ is the pixel after mapping, N is a positive integer that represents the order of the image matrix, and mod is complementation.

3.2 Watermark Embedding Process

Let the matrix size of the host image I be $M \times M$ and the matrix size of the watermark image be $N \times N$, then decompose the host image into three-primary-color channels. The study demonstrates (Kumar & Singh, 2021) that embedding the watermark into the red channel component can ensure good robustness. In this article, the red component image is selected for lifting wavelet transform, whereby the transformed low-frequency sub-band LL is chunked without overlapping, and the subblock is Hessenberg-decomposed to obtain the maximum elements of the upper triangular H matrix. Since the maximum elements of the Hessenberg matrix are almost always in the second row and the second column, or the third row and the second column of this matrix (Su, 2016), this algorithm embeds watermark sequences into the second row and second column elements of the H matrix. The watermark embedding procedure is as follows:

1. Put the watermark image W_i under Arnold transform based on the secret key K_i to get a scrambled watermark W_i.
2. Transform the scrambled watermark W_i into 0,1 sequence.
3. Separate the host image I into three-primary color channels, and take out the red channel component image.
4. Perform LWT on the red channel, and take out the transformed low-frequency component sub-band LL.
5. Perform 4×4 non-overlapping chunking on sub-band LL. Use the secret key $Ks_i\left(i = 1, 2, \ldots\right)$-based pseudo-random sequence to select the watermark embedding block and mark it as $A_i\left(i = 1, 2, \ldots\right)$ to enhance watermark security.
6. Hessenberg decompose the selected block Ai to get the H matrix as in Equation (9):

$$\left[Q, H\right] = hess\left(A\right) \tag{9}$$

7. Find the optimal embedding strength factor T through FOA.
8. Extract element h'_{22} in the second row and second column of the H matrix, and according to the embedding strength factor T obtained by FOA mentioned in 1.4, embed the 0,1 sequence into the H matrix while updating h_{22} to h'_{22}, as in Equation (10):

$$\begin{cases} d = \mathrm{mod}\left(h_{22}, T\right) \\ w = 1, d < 0.25T, h'_{22} = h_{22} - d - 0.25T \\ w = 1, d \geq 0.25T, h'_{22} = h_{22} - d + 0.75T \\ w = 0, d \geq 0.75T, h'_{22} = h_{22} - d + 1.25T \\ w = 0, d < 0.75T, h'_{22} = h_{22} - d + 0.25T \end{cases} \tag{10}$$

9. Perform Hessenberg inverse transformation using the Q matrix and the H matrix with embedded watermark information as in Equation (11):

$$B' = QH'Q^T \tag{11}$$

where H' is the H matrix after embedding the watermark, and Q^T is the transposed matrix of the orthogonal matrix Q.

10. Merge the non-overlapping 4×4 blocks obtained.
11. Perform inverse LWT transformation on the combined low-frequency component image to obtain the r-channel component image.
12. Merge R, G, B channel component images to finally obtain the color image I_w with watermark.
13. The flow chart of watermark embedding is illustrated in Figure 2.

3.3 Watermarking Extraction Process

A blind watermarking algorithm that does not require the participation of the original data during watermark extraction is proposed in this article. Before watermark extraction, it is necessary to use Scale Invariant Feature Transform (SIFT)to geometrically correct the watermarked host image, so that the image to be extracted and the original host image maintain the same geometric shape. The scale-invariant feature transformation was proposed by Lowe (2004). A characteristic of SIFT point is that the value will remain even if the image suffers from rotation, scaling, rotation, translation, and brightness-change. The watermark extraction steps are described below:

1. Using SIFT to perform geometric correction on image I_w (watermarked image).
2. Separate RGB channels for the image I_w.
3. Take out the red channel image and perform LWT to get the transformed low-frequency sub-band LL.
4. Perform a 4×4 non-overlapping chunking process on sub-band LL, and use the pseudo-random sequence selection waters based on the secret key $Ks_i \left(i = 1,2,\ldots\right)$ to choose a watermark embedding block $A_i \left(i = 1,2,3\ldots\right)$.
5. Perform Hessenberg decomposition on the selected block A_i to obtain the H matrix.
6. Obtain the second row and second column element h'_{22} of the H-matrix, and extract the watermark information from h_{22} containing watermark information via Equation (12):

$$w' = \begin{cases} 1, if \bmod\left(\hat{h}_{22}, T\right) > 0.5T \\ 0, if \bmod\left(\hat{h}_{22}, T\right) \le 0.5T \end{cases} \tag{12}$$

where T is the embedding strength factor derived from FOA. h'_{22} is the second row and second column element of the H matrix after embedding the watermark, and w' is the extracted watermark sequence.

7. Through Arnold scrambling of the extracted watermark sequence, finally obtain the extracted watermarked w'.

Figure 2. Watermark Embedding Flow Chart

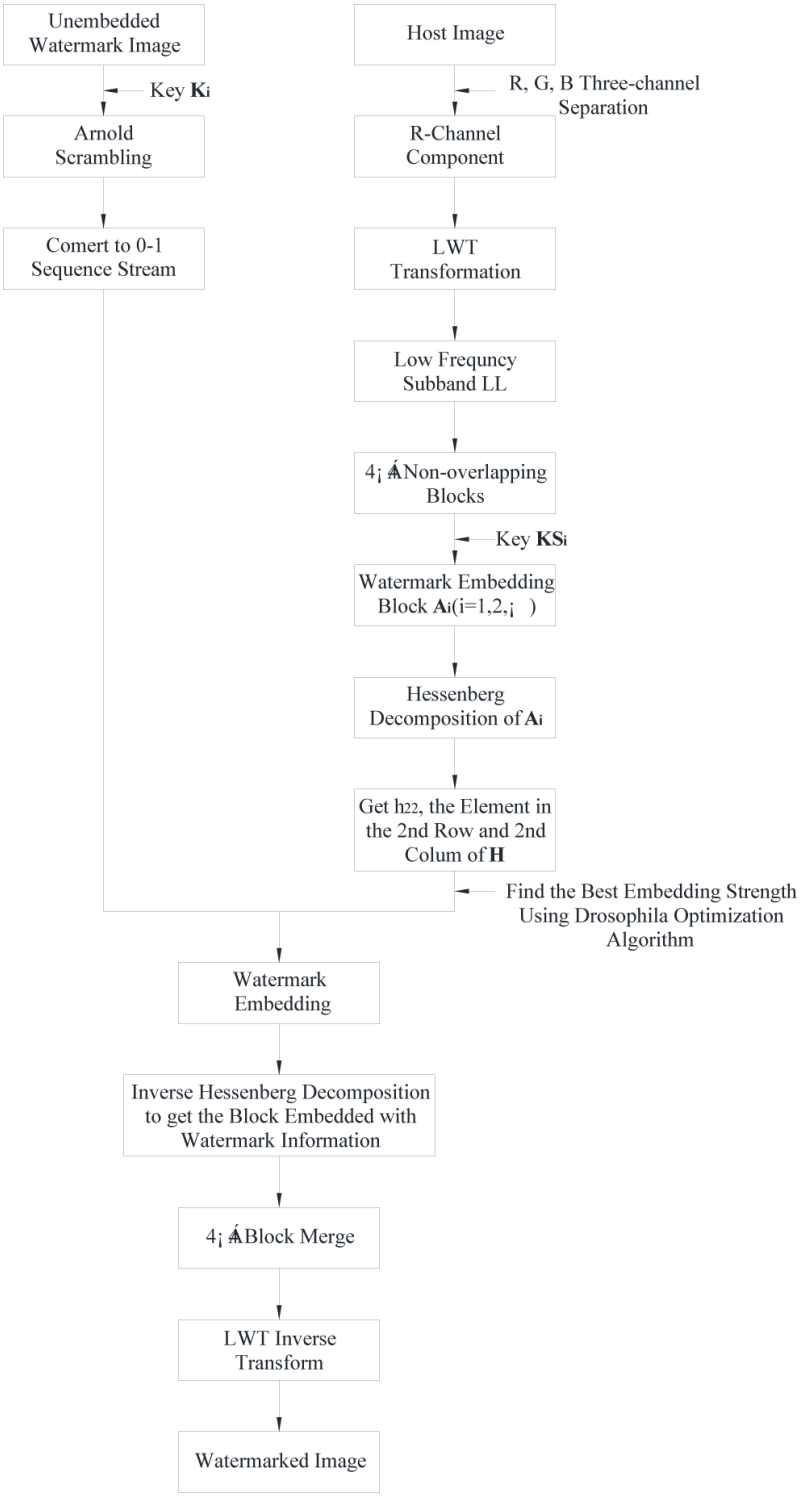


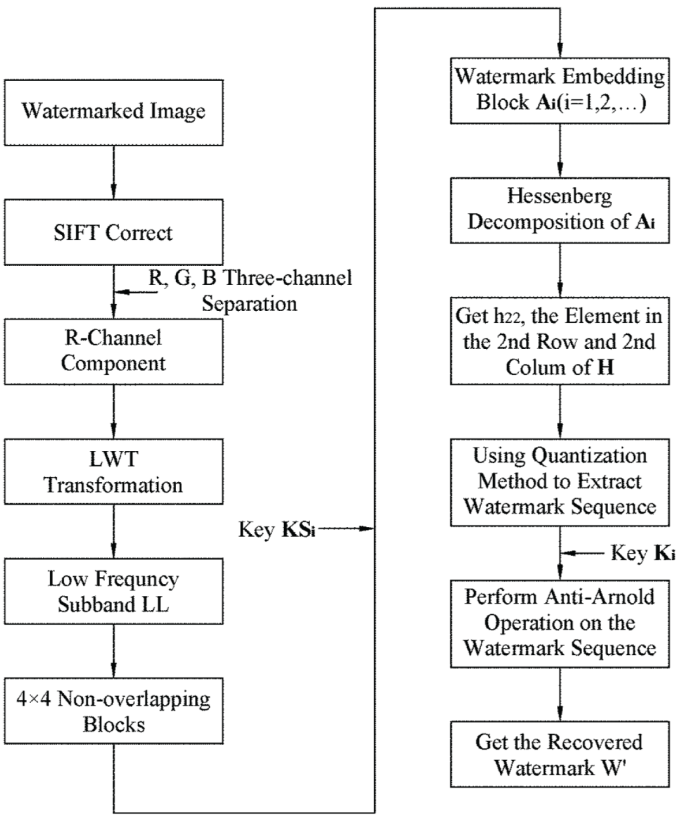
Figure 3 gives the watermark extraction process.

In this article, $PSNR$ (peak signal-to-noise ratio) and NC (normalized correlation) coefficient are used to measure the invisibility of the embedded watermark and the robustness of the extracted watermark. $PSNR$ is one of the most common image quality evaluation metrics and an error comparison pixel by pixel. If the value is greater than 40 dB, it indicates excellent image quality (the watermark-embedded image is very close to the original host image). If $PSNR$ value is 30-40 dB, it means that image quality is relatively good (with a little distortion but acceptable). A $PSNR$ value between 20 and 30dB suggests a poor image quality (distortion visible to the naked eye). If $PSNR$ is smaller than 20, image quality is unacceptable. In fact, when $PSNR$ is greater than 35, it is already difficult for the naked eye to discern differences in the image. The $PSNR$ calculation formula is shown in Equation (13):

$$PSNR = 10 \cdot \lg\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \lg\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{13}$$

where MAX_I denotes the maximum value of the pixel color, which is 255 if each sampling point is expressed in an 8-bit binary. MSE is the mean square error, which is used to measure the extent of differences between two images of the same size; the more significant the differences between the pixels of the two images, the larger the MSE . The MSE value of two

Figure 3. Watermark Extraction Flow Chart



identical images is 0. NC coefficient (Xiao et al., 2020) is the main indicator of image similarity. The comparison of two images in terms of correlation between corresponding pixels reveals that the higher the NC value of two images is, the closer the NC value is to 1. NC value is derived from Equation (14):

$$NC\left(x_1, x_2\right) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} x_1\left(i,j\right)x_2\left(i,j\right)}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} x_1^2\left(i,j\right)}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} x_2^1\left(i,j\right)}} \tag{14}$$

4. PERFORMANCE EVALUATION

Based on different image features, the standard images on USC-SIPI were selected for testing. The size of the test images was 512×512. In this paper, the authors only choose Lenna, baboon, and peppers to demonstrate. As shown in Figure 4 (a), (b), and (c), and the binary image in Figure 4 (d) is selected as a 64×64 watermark image. The three images containing watermarks are respectively subjected to the following ten attacks: (a) cropping, (b) scaling, (c) brightness adjustment, (d) occlusion, (e) salt-and-pepper noise, (f) JPEG compression, (g) Gaussian noise, (h) line drawing, (i) median filter, and (j) rotation. $PSNR$ is used to measure the invisibility of the embedded watermark, the value of which is positively correlated with invisibility of the watermark embedded. NC coefficient is adopted to measure the robustness of the extracted watermark, the value of which approaching to 1 indicates that the image has good robustness. According to the FOA algorithm, the optimal embedding strengths of the three images Lenna, baboon, and peppers are 50.0, 49.75, and 49.45, respectively.

4.1 Imperceptibility Evaluation

In this experiment, the invisibility of watermark embedding is measured by $PSNR$. The watermark is embedded into the images Lenna, baboon, and peppers via the embedding strength factors 50.0, 49.75, and 49.45, respectively, calculated according to FOA. The host images $PSNR$ before and after watermark embedment are shown in Table 1. As can be seen from the table, the $PSNR$ values

Figure 4. Host Images and Watermark Image: (a) Lenna, (b) Baboon, (c) Peppers, (d) Watermark

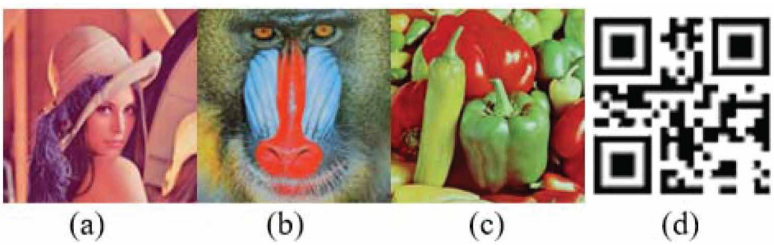


Table 1. PSNR Values of Different Images Before and After Embedding the Watermark

Image	Lenna (50.0)	Baboon (49.75)	Peppers (49.45)
PSNR	39.7041	40.8225	40.0511

of the three images are greater than 38.0, indicating that the perceptible errors before and after watermark embedment are outside the range distinguishable by the naked eye. Judging from the visual effect, there is no significant difference between the watermarked images and the original host images; in other words, invisibility is good.

4.2 Robustness Evaluation

4.2.1 Rotation Attack

Among image attacks, rotation attack is a very common geometric attack method that uses rotation transformation to change the spatial position of image pixel. In this article, for watermarked images subjected to rotation attack, SIFT is harnessed firstly for geometric correction, and then the watermark extracted. The watermark values extracted from Lenna, peppers, and baboon after being attacked by rotations in varying degrees are given in Table 2. In the algorithm proposed in this article, the NC value of the extracted watermark is almost always above 0.75 when attacked from any angle, especially 90-degree rotation whereby the NC value reaches 1. This is because when subjected to a 90-degree attack, there is no loss of pixel values, so the 90-degree rotation attack has no effect on the extraction of the watermark. Similarly, when the image is rotated 180 degrees, 270 degrees, and 360 degrees, the extracted NC value is still 1.

4.2.2 Geometry Attacks

The images Lenna, baboon, and peppers are attacked as such: Cropping 1/16, 1/8, 1/4, 1/2, and 3/4 of the images; Occluding 3 blocks, 6 blocks, and 9 blocks randomly; shrinking 0.5 times, reducing 0.5 times, 0.7 times, and 0.9 times; and magnifying 1.2 times and 1.5 times. The cropped, masked, and scaled images and the extracted watermark images are shown in Figure 5 (a)-(l). Table 3 presents

Table 2. NC Values for Different Images Resisting Rotation Attacks From Different Angles

Degree of rotation	Lenna	Peppers	Baboon
Left 1°	0.995186	0.974749	0.958172
Right 1°	0.976025	0.973500	0.958853
Left 2°	0.964074	0.963408	0.946178
Right 2°	0.967411	0.962328	0.946990
Left 3°	0.955180	0.953359	0.937881
Right 3°	0.957028	0.952753	0.937498
Left 5°	0.943413	0.934380	0.917511
Right 5°	0.950870	0.993372	0.916861
Left 10°	0.900129	0.892389	0.873360
Right 20°	0.845457	0.834568	0.811753
Left 30°	0.811482	0.800573	0.777533
Right 40°	0.800165	0.785076	0.761458
Left 50°	0.776142	0.785251	0.762614
Right 60°	0.810315	0.800549	0.772185
Left 70°	0.840000	0.834119	0.811563
Right 80°	0.901859	0.891256	0.842975
Left 90°	1.000000	1.000000	1.000000

Figure 5. Different Images Subjected to Different Extents of Cropping, Occluding, Scaling Attacks, and Extracted Watermarks: (a) Lenna Cropped by 1/6 and the Extracted Watermark, (b) Lenna Cropped by 1/8 and the Extracted Watermark, (c) Baboon Cropped by 1/4 and the Extracted Watermark, (d) Baboon Cropped by 1/2 and the Extracted Watermark, (e) Peppers Cropped by 1/2 and the Extracted Watermark, (f) Peppers Cropped by 3/4 and the Extracted Watermark, (g) Lenna Occluded by Three Blocks and The Extracted Watermark, (h) Baboon Occluded by Six Blocks and the Extracted Watermark, (i) Peppers Occluded by 9 blocks and the Extracted Watermark, (j) Lenna Reduced by 0.5 Times and the Extracted Watermark, (k) Baboon Reduced by 0.9 Times and the Extracted Watermark, (l) Peppers Amplified by 1.5 Times and the Extracted Watermark

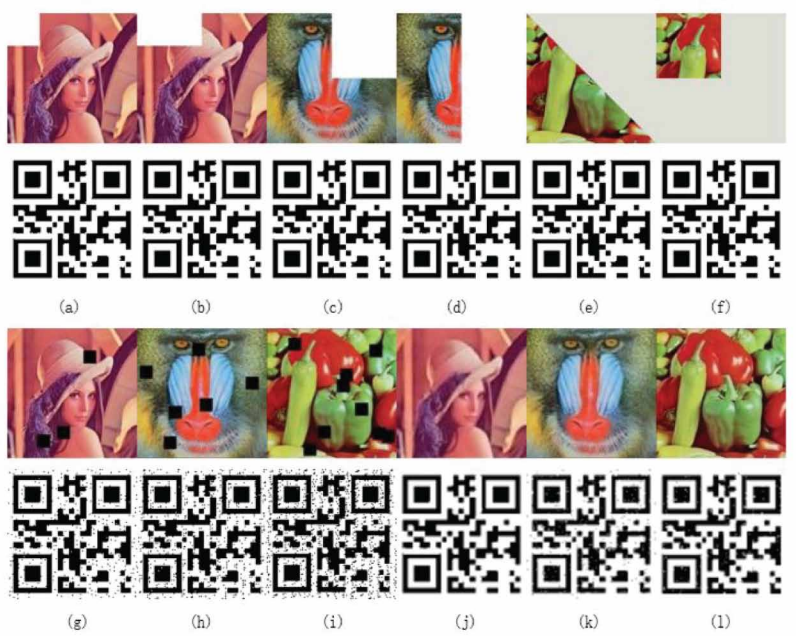


Table 3. Cropping, Occlusion, and Scaling Attacks in Varying Degrees, and NC Values of Extracted Watermarks

Type of attacks		Lenna	Peppers	Baboon
Cropping	1/16	1.000000	1.000000	1.000000
	3/4	1.000000	1.000000	1.000000
Occlusion	3 blocks	0.960988	0.961406	0.958034
	6 blocks	0.936324	0.924216	0.927550
	9 blocks	0.891973	0.896347	0.893706
Scaling	0.5	0.999976	0.999678	0.993482
	0.7	0.981545	0.988092	0.922169
	0.9	0.974410	0.987759	0.962208
	1.2	0.961753	0.979621	0.957367
	1.5	0.999952	0.979887	0.962660

the NC values of the extracted watermark. The experimental results demonstrate that the watermark NC value extracted by this algorithm is still 1 after different degrees of cropping attacks. This is because in this article, in order to improve the images' anti-cropping robustness, the four regions of a host image, namely the top, bottom, left, and right, are embedded with complete independent

watermarks, improving the anti-cropping performance of the watermark by 75%. So, an uncorrupted watermark region can always be found to protect the watermark, regardless of cropping attack degrees. The NC value of the extracted watermark is still greater than 0.89 in the case of as many as nine occlusion blocks, indicating that the proposed algorithm is robust against occlusion attacks. After any degree of scaling attacks, the NC value of the watermark mostly stays above 0.96, and the NC value of the extracted watermark is even as high as 0.9999 after a 0.5 times reduction. Either 0.5 times reduction or restoration is presented in an integer form, and the accuracy loss of the image is very small, almost negligible (0), so damage inflicted upon the watermark is minimal.

4.2.3 Other Conventional Attacks

The NC values of watermarks extracted after other conventional attacks on Lenna, peppers, and baboon ranging from Gaussian noise to salt-and-pepper noise, median filter, and JPEG compression are detailed in Table 4, from which it can be seen that the values of extracted watermarks for the three images attacked by Gaussian noises 0.005, 0.01, 0.02, and 0.05 mostly remain above 0.8. For the three images attacked by 0.005, 0.01, 0.02, and 0.05 salt-and-pepper noises, most of the extracted watermark's NC values are higher than 0.9, indicating that the algorithm proposed in this article is robust against Gaussian noise and salt-and-pepper noise attacks. For the median filter attacks of 3×3 and 5×5, the NC values of watermarks all stayed above 0.85, suggesting that the algorithm is quite robust in resisting median filter attacks. The NC values of extracted watermarks are all greater than 0.95 after JPEG compression attacks because this algorithm embeds the watermark into the LWT low-frequency sub-band of the host image, which contains more contour information. Since JPEG compression compresses the high-frequency sub-band information of the image that contains more detailed information, and has little effect on the low-frequency sub-band, therefore this algorithm is more robust against JPEG compression.

4.3 Contrast Experiment

The method proposed in this paper is further compared with literature on DWTSVD (Roy & Pal, 2019), FDCuT (Soleymani et al., 2019), and DWT (Agarwal et al., 2015), DWTHTSVD (Khare & Srivastava, 2020), based on JPEG compression attack, median filter attack, Gaussian filter, Gaussian noise, salt-and-pepper noise, scaling attack, and cropping attack. From Table 5, it can be observed that the algorithm LWTHD outperforms method DWTSVD, FDCuT,

Table 4. Other Conventional Attacks in Varying Degrees, and NC Values of Extracted Watermarks

Other conventional attacks		Lenna	Peppers	Baboon
Gaussian noise	0.005	0.965254	0.969726	0.977410
	0.01	0.935336	0.934180	0.945430
	0.02	0.907900	0.847415	0.867483
	0.05	0.865774	0.736317	0.776954
Salt-pepper noise	0.005	0.999743	0.997159	0.987490
	0.01	0.995580	0.992217	0.976962
	0.02	0.983901	0.977389	0.940738
	0.05	0.904126	0.913324	0.879096
Median filter	3×3	0.954220	0.969225	0.964177
	5×5	0.850900	0.878150	0.870182
JPEG compression	10	0.964130	0.994069	0.984137
	30	0.9748636	0.995640	0.989011

Table 5. Watermark NC Values Extracted via Conventional Attacks Contrast Experiment

Type of attacks		DWTSVD	DWTHTSVD	DWT	FDCuT	LWTHD
JPEG compression	10	/	0.9995	/	0.7900	**0.9641**
	15	0.9525	/	/	/	**0.9668**
	20	/	/	/	**1.0000**	0.9703
	30		**0.9999**	/	**/**	0.9749
	50	0.9831	**0.9999**	/	**/**	0.9790
	60		**0.9998**	/	**/**	0.9859
	80	0.9959	**1.0000**	0.991	/	0.9924
	95	0.9968	/	0.9981	/	**0.9987**
Median filter	3×3	0.9796	0.9994	/	**1.0000**	0.9542
	5×5	0.9741	0.9973	/	**1.0000**	0.8509
Gaussian filter	3×3	0.9611	0.9999	**0.9985**	/	0.9740
	5×5	0.9621	0.9999	**0.9897**	/	0.8977
Gaussian noise	0.00001	/	0.9985	0.9984	/	**1.0000**
	0.0001	/	/	0.998	/	**1.0000**
	0.001		**0.9975**	/	/	0.9726
	0.01	**0.9368**	**/**	/	/	0.9353
	0.02	/	/	/	/	**0.9079**
	0.03	**0.9342**	/	/	/	0.8903
	0.05	0.9294	/	/	**0.9700**	0.8658
	0.1	0.9278	/	/	**0.9000**	0.7964
Salt-pepper noise	0.001		**0.9998**			0.9989
	0.01	0.9715	0.9980	**1.0000**	/	0.9956
	0.02	/	/	/	/	**0.9839**
	0.03	**0.9593**	/	/	/	0.9425
	0.05	0.9583	/	0.9396	**1.0000**	0.9041
	0.95	/	/	0.0274	/	/
	0.1	0.9461	/	/	**1.0000**	0.8503
Scaling	0.1	/	/	0.0268	/	**1.0000**
	0.25	0.9605	/	/	/	**1.0000**
	0.5	0.9655	/	/	**1.0000**	**1.0000**
	0.9	/	/	0.8987	/	**0.9744**
	1.5	**/**	**/**	**/**	**1.0000**	**1.0000**
	2	0.9659	0.9999	/	**/**	**1.0000**
Cropping	1/10	/	/	0.9435	**1.0000**	**1.0000**
	1/8	0.9864	/	/	/	**1.0000**
	1/4	0.9609	/	/	0.9500	**1.0000**
	1/2	0.9479	/	/	/	**1.0000**
	3/5	/	/	0.6284	/	**1.0000**

DWT, and DWTHTSVD in terms of withstanding JPEG compression due to the fact that it embeds the watermark into the low frequency domain of LWT, while compression attack has little effect on the watermark because it compresses the high frequency domain. The algorithm is robust enough to resist Gaussian noise and salt-and-pepper noise attacks. The watermark NC value is above 0.79 when Gaussian noise and the salt-and-pepper noise reach 0.1. This is because the second row and second column element of the Hessenberg decomposed H matrix contains the maximum energy of the whole matrix and has the stable numerical characteristic conducive to resisting external disturbances. Obviously, the NC values of this algorithm perform better against scaling and cropping attacks than those covered in method DWTSVD, FDCuT, DWT, and DWTHTSVD, with all cases even reaching 1.0. This outstanding performance is attributed to the fact that this algorithm uses the SIFT operator to make geometric adjustments after scaling attacks to restore the image to the original size. The scaling factors are integer multiples after 0.5, 0.1, and 1.5 times of scaling, which do not affect the image's pixel accuracy, and so the NC value stays as high as 1.0. When scaling factor is 0.9, the result would be pixel accuracy loss, but even then, the NC value is still as high as 0.97. Moreover, the algorithm LWTHD divides the host image into four separate parts to embed the watermark, which significantly enhances watermark anti-cropping performance, and the NC values of the watermark extracted remain at 1.0 even after various degrees of cropping. Overall, the minimum and maximum NC values of this algorithm are 0.79 and 1, respectively, indicating that the algorithm is robust against various types of conventional attacks.

Regarding filter and noise attacks, the NC value of the extracted watermark tends to decrease slightly as attack intensity increases, but the NC value is still greater than 0.79 on the whole, suggesting that the algorithm LWTHD is a robust watermark extraction method.

The watermarking algorithm based on DWTSVD, DWTHTSVD, and DWT embeds the watermark in the DWT domain, which is easy to cause a large loss of precision, and has high time complexity and large storage space, which is inconvenient for hardware implementation. This algorithm embeds the watermark in the IWT domain, which improves the accuracy of watermark extraction, reduces the time complexity and saves storage space. The watermarking algorithm based on FDCuT is only suitable for grayscale images, but this algorithm can be used for color images. Compared with the traditional watermarking algorithm, this algorithm has the following advantages:

1. The embedding and extraction process meets the requirements of blind watermarking.
2. Invisibility and robustness meet practical application requirements.
3. The time complexity is low, which can meet real-time watermark embedding and extraction.

5. CONCLUSION

Here, the authors propose a robust watermarking algorithm based on lifting wavelet transform (LWT) and Hessenberg decomposition (LWTHD), which has important implications for information security and copyright protection. Embedding the watermark information into the maximum Hessenberg decomposition element of the low-frequency sub-band after performing LWT can effectively avoid external interferences. Embedding the watermark image into the four separate areas on the top, bottom, left, and right of the host image can effectively withstand cropping attacks. Distortion correction using the SIFT algorithm during watermark extraction can effectively resist geometric attacks. The fruit fly optimization algorithm (FOA) can adaptively find embedding strength to achieve an optimal balance of invisibility and robustness. The experiments show that the algorithm LWTHD exhibits high robustness, especially in

resisting attacks such as rotation, cropping, salt-and-pepper attack, JPEG compression, and salt-and-pepper noise. The screen shot attack is not considered in this paper, which will be the next research content.

COMPETING INTERESTS

The authors of this publication declare there are no competing interests.

FUNDING AGENCY

This research was supported by Sichuan Science and Technology Program [2020YFS0316].

REFERENCES

Abdulrahman, A. K., & Ozturk, S. (2019). A novel hybrid DCT and DWT based robust watermarking algorithm for color images. *Multimedia Tools and Applications*, *78*(12), 17027–17049. doi:10.1007/s11042-018-7085-z

Agarwal, H., & Husain, F. (2021). Development of payload capacity enhanced robust video watermarking scheme based on symmetry of circle using lifting wavelet transform and SURF. *Journal of Information Security and Applications*, *59*, 102846.

Agarwal, H., Raman, B., & Venkat, I. (2015). Blind reliable invisible watermarking method in wavelet domain for face image watermark. *Multimedia Tools and Applications*, *74*(17), 6897–6935. doi:10.1007/s11042-014-1934-1

Bose, A., & Maity, S. P. (2022). Secure sparse watermarking on DWT-SVD for digital images. *Journal of Information Security and Applications*, *68*, 103255. doi:10.1016/j.jisa.2022.103255

Budiman, G., Novamizanti, L., & Iwut, I. (2017). Genetics algorithm optimization of DWT-DCT based image Watermarking. *Journal of Physics: Conference Series*, *795*(1), 012039. doi:10.1088/1742-6596/795/1/012039

Dhar, P. K., Chowdhury, A. H., & Koshiba, T. (2020). Blind audio watermarking based on parametric slant-Hadamard transform and Hessenberg decomposition. *Symmetry*, *12*(3), 333. doi:10.3390/sym12030333

Han, S., Wang, R., & Jia, G. (2017). Colorblind image watermarking algorithm based on Hessenberg decomposition. *Computer Engineering and Design*, *38*(12), 3354–3360.

Hsu, C. S., & Tu, S. F. (2020). Enhancing the robustness of image watermarking against cropping attacks with dual watermarks. *Multimedia Tools and Applications*, *79*(17), 11297–11323.

Khare, P., & Srivastava, V. K. (2021). A reliable and secure image watermarking algorithm using homomorphic transform in DWT domain. *Multidimensional Systems and Signal Processing*, *32*(1), 131–160.

Kumar, L., & Singh, K. U. (2021). A secure image watermarking scheme based on DWT, SVD and Arnold transform. *IOP Conference Series. Materials Science and Engineering*, *1099*(1), 012076.

Li, Z., Zhang, H., Liu, X., Wang, C., & Wang, X. (2021). Blind and safety-enhanced dual watermarking algorithm with chaotic system encryption based on RHFM and DWT-DCT. *Digital Signal Processing*, *115*, 103062.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Lusson, F., Bailey, K., Leeney, M., & Curran, K. (2013). A novel approach to digital watermarking, exploiting colour spaces. *Signal Processing*, *93*(5), 1268–1294.

Makbol, N. M., Khoo, B. E., Rassem, T. H., & Loukhaoukha, K. (2017). A new reliable optimized image watermarking scheme based on the integer wavelet transform and singular value decomposition for copyright protection. *Information Sciences*, *417*, 381–400.

Meng, L., Liu, L., Tian, G., & Wang, X. (2021). An adaptive reversible watermarking in IWT domain. *Multimedia Tools and Applications*, *80*(1), 711–735.

Naik, K., Trivedy, S., & Pal, A. K. (2018). An IWT based blind and robust image watermarking scheme using secret key matrix. *Multimedia Tools and Applications*, *77*(11), 13721–13752.

Nazari, M., & Mehrabian, M. (2021). A novel chaotic IWT-LSB blind watermarking approach with flexible capacity for secure transmission of authenticated medical images. *Multimedia Tools and Applications*, *80*(2), 1–41.

Roy, S., & Pal, A. K. (2019). A hybrid domain color image watermarking based on DWT–SVD. *Iranian Journal of Science and Technology. Transaction of Electrical Engineering*, *43*(2), 201–217.

Soleymani, S. H., Taherinia, A. H., & Mohajerzadeh, A. H. (2019). Waca: A new blind robust watermarking method based on Arnold cat map and amplified pseudo-noise strings with weak correlation. *Multimedia Tools and Applications*, *78*(14), 19163–19179.

Su, Q. (2015). *Digital blind watermarking technology for color images*. Tsinghua University Press.

Su, Q. (2016). Novel blind colour image watermarking technique using Hessenberg decomposition. *IET Image Processing*, *10*(11), 817–829.

Su, Q., Wang, G., Lv, G., Zhang, X., Deng, G., & Chen, B. (2017). A novel blind color image watermarking based on contourlet transform and Hessenberg decomposition. *Multimedia Tools and Applications*, *76*(6), 8781–8801.

Thakkar, F., & Srivastava, V. K. (2021). An adaptive, secure and imperceptive image watermarking using swarm intelligence, Arnold transform, SVD and DWT. *Multimedia Tools and Applications*, *80*(8), 12275–12292.

Vaidya, S. P., Mouli, P. C., & Santosh, K. C. (2019). Imperceptible watermark for a game-theoretic watermarking system. *International Journal of Machine Learning and Cybernetics*, *10*(6), 1323–1339.

Veni, M., & Meyyappan, T. (2017). DWT DCT based new image watermarking algorithm to improve the imperceptibility and robustness. In *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*. IEEE.

Wang, Y., Guan, H., Huang, Y., & Zhang, S. (2020). Text image watermarking scheme using DWT-DCT joint transform. *Computer Engineering and Design*, *41*(6), 1676–1682.

Wu, D., Zhang, J., Rong, W., Tang, Y., Zhao, J., & Qu, C. (2021). Summary of digital image water marking technology. *High-Tech Communication*, *31*(2), 148–162.

Xiao, Z., Ning, Q., Zhang, H., Tang, X., & Chen, H. (2020). Adaptive zero-watermarking algorithm based on block NMF and boost normed singular value decomposition. *Jisuanji Yingyong Yanjiu*, 4.

Yang, J., Hu, K., Wang, X., Wang, H., Liu, Q., & Mao, Y. (2022). An efficient and robust zero watermarking algorithm. *Multimedia Tools and Applications*, 1–19.

Fan Li received the master's degree from University of Electronic Science and Technology of China, China, in 2005. He is currently working with the College of Blockchain Technology, Chengdu University of Information Technology, China. His research interests include information security and image processing.

Lin Gao received the Ph.D. degree in Computer Application from Sichuan University, China, in 2010. He is currently working with the College of Blockchain Technology, Chengdu University of Information Technology, China. His research interests include image processing and artificial intelligence.

JunFeng Wang received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, in 2004. From 2004 to 2006, he held a postdoctoral position with the Institute of Software, Chinese Academy of Sciences. Since 2006, he has been with the College of Computer Science and the School of Aeronautics and Astronautics, Sichuan University, as a Professor. His recent research interests include network and information security, spatial information networks, and data mining.

Yan Ruixia, female, from Baoji, Shaanxi Province, studied for a master's degree at the School of Computer Science, Southwest University of Science and Technology, with a research direction of image processing.