

Article

A Robust Color Image Watermarking Algorithm Based on APDCBT and SSVD

Xiaoyan Yu , Chengyou Wang * and Xiao Zhou 

School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China; xyy@mail.sdu.edu.cn (X.Y.); zhouxiao@sdu.edu.cn (X.Z.)

* Correspondence: wangchengyou@sdu.edu.cn; Tel.: +86-631-568-8338

Received: 1 August 2019; Accepted: 11 September 2019; Published: 2 October 2019



Abstract: With the wide application of color images, watermarking for the copyright protection of color images has become a research hotspot. In this paper, a robust color image watermarking algorithm based on all phase discrete cosine biorthogonal transform (APDCBT) and shuffled singular value decomposition (SSVD) is proposed. The host image is transformed by the 8×8 APDCBT to obtain the direct current (DC) coefficient matrix, and then, the singular value decomposition (SVD) is performed on the DC matrix to embed the watermark. The SSVD and Fibonacci transform are mainly used at the watermark preprocessing stage to improve the security and robustness of the algorithm. The watermarks are color images, and a color quick response (QR) code with error correction mechanism is introduced to be a watermark to further improve the robustness. The watermark embedding and extraction processes are symmetrical. The experimental results show that the algorithm can effectively resist common image processing attacks, such as JPEG compression, Gaussian noise, salt and pepper noise, average filter, median filter, Gaussian filter, sharpening, scaling attacks, and a certain degree of rotation attacks. Compared with the color image watermarking algorithms considered in this paper, the proposed algorithm has better performance in robustness and imperceptibility.

Keywords: color image watermarking; copyright protection; robustness; all phase discrete cosine biorthogonal transform (APDCBT); shuffled singular value decomposition (SSVD); Fibonacci transform

1. Introduction

With the rapid development of the information age, digital images, as important information carriers, have been widely spread on the network. Images are easy to copy and spread in large quantities, which brings convenience to people, but also makes piracy and infringement more and more serious. In recent years, endless piracy and infringement incidents have emerged, seriously affecting the legitimate rights and interests of copyright owners and disrupting the normal market order and social public order. Improving the level of copyright protection and information security for digital images has become a research hotspot. Techniques for copyright protection include cryptography [1], steganography [2], digital fingerprinting [3], and digital watermarking [4]. Among them, digital watermarking is the most commonly used copyright protection technology for digital products due to its good concealment, robustness, and stability. Until now, digital watermarking has been applied to the copyright protection of texts [5], images [6–8], audios [9], videos [10–12], and so on, especially images.

Since the advent of color displays, black-and-white displays have been rapidly replaced. With the rise of this technology, color images have replaced black-and-white images and gray images. As a result, the watermarking technology for copyright protection of color images has gradually developed. Depending on the difference of the embedding locations, image watermarking can be divided into

two categories: watermarking in the spatial domain [13–15] and watermarking in the transform domain [16–28]. Chou and Wu [13] embedded the color watermark image into the color host images uniformly by modifying the quantization indices of color pixels. This algorithm cannot effectively resist various attacks. Su et al. [14] calculated the approximate maximum eigenvalue of the Schur decomposition in the spatial domain to embed the watermark without true Schur decomposition. Su et al. [15] embedded the watermark into the direct current (DC) coefficient of two-dimensional (2D) discrete Fourier transform (DFT) obtained in the spatial domain. These two algorithms have both the advantages of the spatial domain and frequency domain.

Watermarking algorithms in the spatial domain are easy to implement but have poor robustness, while watermarking algorithms in the transform domain are robust and have wide application range. Barni et al. [16] embedded the watermark by changing all discrete cosine transform (DCT) coefficients of each color channel. A threshold was used in this algorithm to minimize the difference of extracted watermark information. Liu [17] extended the perceptual model used in gray images to the color images, and estimated the noise attack threshold of each wavelet coefficient in the luminance and chrominance of the color image to meet the requirements of imperceptibility and robustness. Patvardhan et al. [18] took the quick response (QR) code as the carrier of watermark information, and combined discrete wavelet transform (DWT) and singular value decomposition (SVD) to embed the watermark into the luminance component of host images. Roy and Pal [19] proposed a color multiple watermarking method based on DCT and repetition code. Two watermarks were embedded into green and blue components, respectively. The computational complexity of this algorithm is high. Based on the local invariant significant bit-plane histogram, Niu et al. [20] presented a new robust color image watermarking algorithm, which embedded the watermark into the affine invariant local feature regions. Its limitation is that it has lower watermark capacity. In addition, it cannot be used effectively in real-time applications. Liu et al. [21] proposed a robust hybrid color image watermarking algorithm by fusing a local KAZE feature-based watermarking scheme with a conventional watermarking scheme based on integer wavelet transform (IWT). Its watermark capacity is limited. Chang et al. [22] converted the color host image into the YIQ color space, and embedded two binary watermarks into the DCT domain of the quadrature chrominance component. The imperceptibility of this algorithm is relatively poor. Lakrissi et al. [23] proposed a novel dynamic color image watermarking algorithm combining the DWT with SVD. The watermark was randomly embedded into LL sub-bands based on the human visual system (HVS) to choose an adaptive scaling factor. Li et al. [24] proposed a novel color image watermarking scheme based on the quaternion Hadamard transform (QHT) and Schur decomposition. Rosales-Roldan et al. [25] used SVD, DWT, and DCT to transform the luminance component of host images and embedded the watermark using the quantization index modulation (QIM). Roy and Pal [26] embedded the watermark into the luminance component of host images based on DWT and SVD. Vaidya and Mouli [27] used multiple decompositions, including DWT, contourlet transform (CT), Schur decomposition, and SVD, to embed the watermark into the luminance component of the host image. Mohammad and Gholamhossein [28] adopted the teaching-learning-based optimization (TLBO) method to automatically determine the embedding parameters and suitable locations for inserting the watermark. Laur et al. [29] proposed a non-blind color image watermarking scheme based on DWT, chirp Z-transform (CZT), QR decomposition, SVD, and entropy, which embedded the watermark into low entropy parts of all three RGB components. Cedillo-Hernandez et al. [30] proposed a robust-encoded color image watermarking algorithm using the image normalization procedure and DCT transform. To improve the security of the algorithm, a convolutional encoder was used to encode the watermark. All of these algorithms used gray or binary images as the watermark.

With the development of color image watermarking algorithms, color images have gradually been implemented as watermarks [31–33]. Su et al. [31] performed two-level DCT on the color host image and embedded the color watermark into DC coefficients and alternating current (AC) coefficients. Su and Chen [32] proposed an improved dual color image watermarking scheme based on the Schur decomposition, using the features obtained by the Schur decomposition to embed or extract

the watermark images. Pandey et al. [33] embedded the singular values of the watermark into the luminance component of host images using variable scaling factors. This algorithm has high payload capacity, while it is also a non-blind watermarking algorithm. Jia et al. [34] proposed a color image watermarking algorithm based on DWT and QR decomposition, using the features obtained by QR decomposition to embed the watermark. Taking color images as watermarks poses more challenges to the watermark capacity and the imperceptibility of the algorithm.

In this paper, a robust color image watermarking algorithm based on all phase discrete cosine biorthogonal transform (APDCBT) and shuffled singular value decomposition (SSVD) is proposed. The better low-frequency energy accumulation characteristics of the APDCBT and the stability of SSVD are combined to ensure the robustness of the algorithm. Firstly, the host image is divided into nonoverlapping 8×8 blocks, and the APDCBT is performed to obtain the DC coefficient matrix. Then, SVD is performed on the DC matrix to embed the watermark. The SSVD is mainly used at the watermark preprocessing stage to improve the security and robustness of the algorithm.

The main contributions of the proposed algorithm are as follows. (1) The APDCBT transform is introduced into the color image watermarking algorithm, and its better low-frequency energy accumulation characteristic is used to find the watermark embedding locations. (2) Fibonacci transform with larger key space is used to improve the security of the algorithm. (3) The watermarks for embedding are color images, not gray or binary images. Additionally, the color QR code with error correction mechanism is introduced to be a watermark, which can further improve the robustness of the algorithm. (4) The proposed algorithm has high robustness to common image processing attacks, such as JPEG compression, Gaussian noise, salt and pepper noise, average filter, median filter, Gaussian filter, sharpening, and scaling attacks.

The rest of this paper is organized as follows. Preliminary concepts are provided in Section 2, including the concepts and characteristics of APDCBT, SSVD, Fibonacci transform, and QR code. Section 3 details the proposed algorithm, including the preprocessing, embedding, and extraction of the watermark. The experimental results and discussions are presented in Section 4. Finally, the conclusions and future work are discussed in Section 5.

2. Preliminary Concepts

To improve the robustness of the algorithm, the advantages of the APDCBT and SSVD are combined to determine the appropriate locations for watermark embedding. To improve the security of the algorithm, the Fibonacci transform is adopted to scramble the watermark image. In addition, the QR code is introduced to further improve the robustness of the algorithm. This section mainly introduces the concepts and characteristics of these three transforms and the QR code.

2.1. APDCBT

The APDCBT was proposed to solve the serious block artifacts caused by DCT at low bit rates in image compression [35]. Compared with DCT, the APDCBT has better low-frequency energy accumulation and high-frequency energy attenuation characteristics. Due to these characteristics, the APDCBT has been applied in many fields, such as image compression and digital watermarking [36]. The transform matrix of APDCBT can be expressed as Equation (1):

$$B(i, j) = \begin{cases} \frac{N-j}{N^2}, & i = 0, 1, \dots, N-1, j = 0 \\ \frac{1}{N^2} \left[(N-i) \cos \frac{ij\pi}{N} - \csc \frac{j\pi}{N} \sin \frac{ij\pi}{N} \right], & i = 0, 1, \dots, N-1, j = 1, 2, \dots, N-1 \end{cases} \quad (1)$$

where N is the size of the image block X . Performing the APDCBT on X can be presented as Equation (2):

$$A = BXB^T, \quad (2)$$

where A is the transform coefficient matrix and B is the transform matrix of APDCBT.

2.2. SSVD

The SSVD [37] was developed on the basis of the standard SVD. Different from the SVD, SSVD adds a scrambling process, which can be expressed as $Y = \hat{S}\{X\}$, where X is the input image, Y is the scrambled image, and \hat{S} is the shuffled operator. In other words, SSVD scrambles the input image first, and then performs standard SVD on the scrambled image Y . The scrambling method used in the proposed algorithm is the Fibonacci transform, which will be introduced in Section 2.3.

SVD is a type of matrix decomposition that can be applied to any matrix. By performing SVD on a complex matrix, three smaller and simpler submatrices can be obtained, which can describe the important characteristics of the complex matrix. For a matrix Y , its SVD can be expressed as Equation (3):

$$Y = USV^T, \quad (3)$$

where U and V are orthogonal matrices and S is a diagonal matrix. The matrix S contains the singular values of matrix Y , which can represent the main information of matrix Y and has high stability. Due to this characteristic, SVD is widely used in the watermarking field.

2.3. Fibonacci Transform

Matrix transform is one of the most commonly used image scrambling methods. For an image F of size $N \times N$, its Fibonacci transform [11] can be shown as Equation (4):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod(N), \quad (4)$$

where (x, y) is the coordinate of image F and (x', y') is the coordinate after the transform. The Fibonacci transform is periodic, and its transform periods for images with different sizes are shown in Table 1. When the images are the same size, the period of Fibonacci transform is twice as long as an Arnold transform [38]. Therefore, the Fibonacci transform can provide more key space and improve the security of watermarking algorithms.

Table 1. The periods of the Fibonacci transform for images with different sizes.

Size of Image	4 × 4	8 × 8	16 × 16	32 × 32	64 × 64	128 × 128	256 × 256
Period	6	12	24	48	96	192	384

2.4. QR Code

The QR code is a two-dimensional bar code that can be read quickly. Compared with the traditional bar code, the QR code can store more data information. Due to this characteristic, QR codes have been widely used in the information identification of intelligent devices. The size of QR codes can range from 21×21 to 177×177 , mainly including data area, alignment pattern, timing pattern, finder pattern, quit zone, dark module, and separator. Since QR codes have four levels of error correction capability and its maximum fault tolerance can reach 30%, it can still be read correctly even if it is damaged. In addition, QR codes also have strong bending resistance.

Due to the error correction algorithm of the QR code, it can ensure the security of the information to be hidden. In watermarking algorithms, embedding the hidden information into the QR code and using the generated QR code as the watermark can improve the robustness of the algorithm. Even if the extracted QR code image is destroyed, the hidden information can still be decoded.

3. Proposed Color Image Watermarking Algorithm

In this section, the specific steps of the proposed color image watermarking algorithm are explained. The algorithm consists of three parts: watermark preprocessing, watermark embedding, and watermark

extraction. The APDCBT is introduced into the color image watermarking algorithm, and its better low-frequency energy accumulation characteristic is used to find the watermark embedding locations, which is the main contribution of the proposed algorithm. In addition, the Fibonacci transform and the QR code are used to improve the security and the robustness of the proposed algorithm, which are also the contributions of the proposed algorithm. The frameworks of watermark embedding and watermark extraction are shown in Figure 1, and the specific steps are explained in the following subsections.

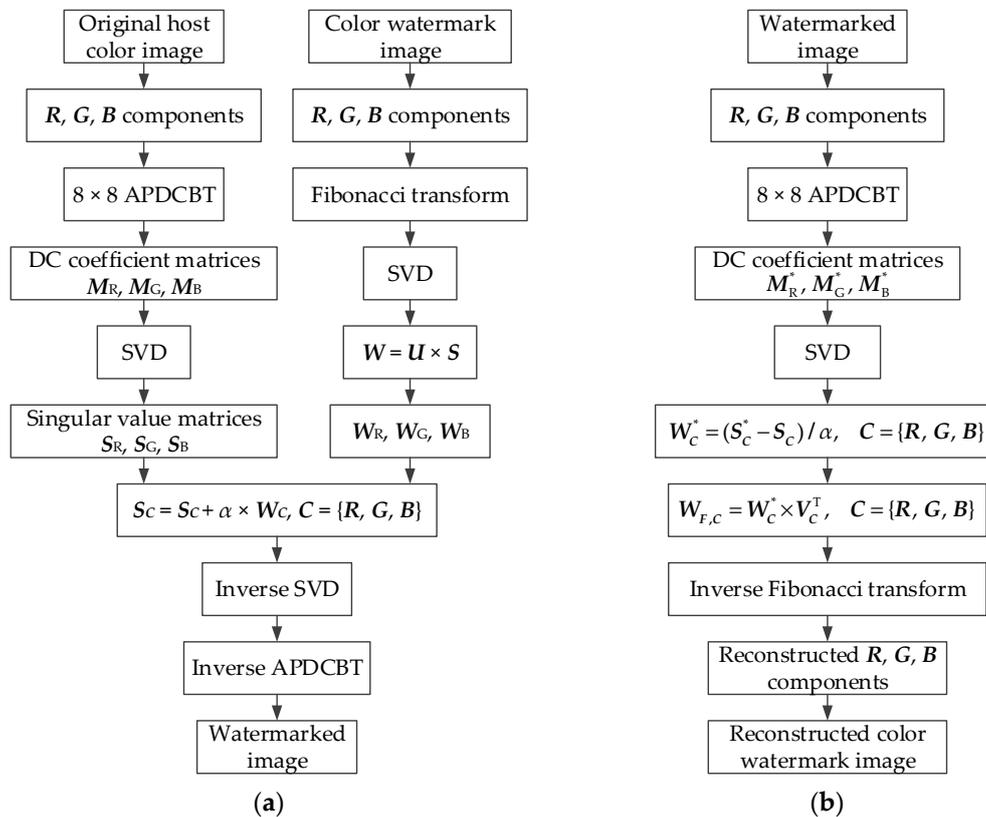


Figure 1. The frameworks of the proposed color image watermarking algorithm: (a) Watermark embedding; (b) Watermark extraction. APDCBT: all phase discrete cosine biorthogonal transform; SVD: singular value decomposition; DC: direct current.

3.1. Watermark Preprocessing

The watermark preprocessing is performed to improve the security of the watermarking algorithm. SSVD is used to preprocess the watermark image. Here, the Fibonacci transform is used to scramble the color watermark image, and the scrambling process is shown in Figure 2.

The specific watermark preprocessing process is presented below.

Step 1: Divide the color watermark image W of size 64×64 into R_W , G_W , and B_W components.

Step 2: Apply the Fibonacci transform to scramble the three color components.

Step 3: Perform SVD on the scrambled components, as shown in Equation (5):

$$C_s = U_{C_s} S_{C_s} V_{C_s}^T, \quad C = \{R, G, B\}, \quad (5)$$

where C_s are the three scrambled color components, U_{C_s} and V_{C_s} are orthogonal matrices for C_s , and S_{C_s} are singular value matrices for C_s .

Step 4: Generate embedded watermark using Equation (6):

$$W_C = U_{C_s} S_{C_s}, \quad C = \{R, G, B\}, \quad (6)$$

where W_C , namely W_R , W_G , and W_B , are the generated watermarks for embedding.

To more comprehensively illustrate the watermark preprocessing process, its pseudocode is shown in Algorithm 1.

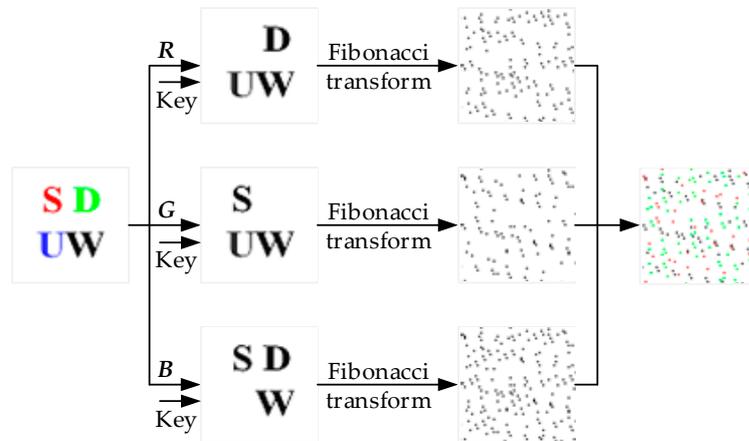


Figure 2. The scrambling process of a color watermark image using the Fibonacci transform.

Algorithm 1. Watermark Preprocessing Algorithm

Variable Declaration:

SDUW: color watermark image

W : read the watermark image

R_W , G_W , and B_W : three components of the color watermark image

R_s , G_s , and B_s : three scrambled components using Fibonacci transform

U_{R_s} , V_{R_s} , U_{G_s} , V_{G_s} , U_{B_s} , and V_{B_s} : orthogonal matrices for R_s , G_s , and B_s

S_{R_s} , S_{G_s} , and S_{B_s} : singular value matrices for R_s , G_s , and B_s

W_R , W_G , and W_B : three watermark components for embedding

Watermark Preprocessing Procedure:

Start Procedure:

1: Read the color watermark image

$W \leftarrow \text{SDUW.png}$ (color watermark image of size 64×64);

$R_W \leftarrow W(:, :, 1)$; $G_W \leftarrow W(:, :, 2)$; $B_W \leftarrow W(:, :, 3)$;

2: Scramble the three color components of the watermark image

$R_s \leftarrow \text{Fibonacci transform}(R_W)$; $G_s \leftarrow \text{Fibonacci transform}(G_W)$; $B_s \leftarrow \text{Fibonacci transform}(B_W)$;

3: Perform SVD on each scrambled color components

$[U_{R_s}, S_{R_s}, V_{R_s}] \leftarrow \text{SVD}(R_s)$; $[U_{G_s}, S_{G_s}, V_{G_s}] \leftarrow \text{SVD}(G_s)$; $[U_{B_s}, S_{B_s}, V_{B_s}] \leftarrow \text{SVD}(B_s)$;

4: Obtain watermark for embedding

$W_R = U_{R_s} \times S_{R_s}$; $W_G = U_{G_s} \times S_{G_s}$; $W_B = U_{B_s} \times S_{B_s}$;

End Procedure

3.2. Watermark Embedding

The watermark embedding process is shown in Figure 1a, and the detailed steps are given below.

Step 1: Divide the color host image I of size 512×512 into R , G , and B components.

Step 2: Divide each component into nonoverlapping 8×8 blocks, and apply the APDCBT to each block to obtain the DC coefficient, and then the DC coefficient matrices M_C can be obtained, where $C = \{R, G, B\}$.

Step 3: Perform SVD on M_C to obtain singular value matrices S_C , where $C = \{R, G, B\}$.

Step 4: Use Algorithm 1 to preprocess the color watermark image and obtain W_C , where $C = \{R, G, B\}$.

Step 5: Embed W_C into S_C according to Equation (7):

$$S_{CW} = S_C + \alpha \times W_C, S_{CW} = U_{CW} S_{CW1} V_{CW}^T, C = \{R, G, B\}, \quad (7)$$

where α is the embedding strength of the watermark, and S_{CW} are three singular value matrices with the embedded watermark. U_{CW} and V_{CW} are orthogonal matrices for S_{CW} , and S_{CW1} are the singular value matrices for S_{CW} . Save U_{CW} , V_{CW} , and S_C for watermark extraction process.

Step 6: Apply the inverse SVD to S_{CW1} , then the watermarked M_C^* can be obtained.

Step 7: After performing the inverse APDCBT, three watermarked color components can be obtained. Use the concatenate (cat) operator to stack them, and then the watermarked color image I^* can be obtained.

To more comprehensively illustrate the watermark embedding process, its pseudocode is shown in Algorithm 2.

Algorithm 2. Watermark Embedding Algorithm

Variable Declaration:

Lena: color host image
 I: read the color host image
 R, G, and B: three color components of the host image
 M_C: three DC coefficient matrices obtained by APDCBT, where C = {R, G, B}
 U_C and V_C: orthogonal matrices for M_C
 S_C: singular value matrices for M_C
 W_C: three preprocessed color components of the watermark image, where C = {R, G, B}
 α: scaling factor
 S_{CW}: three singular value matrices with the embedded watermark
 U_{CW} and V_{CW}: the orthogonal matrices for S_{CW}
 S_{CW1}: the singular value matrices for S_{CW}
 M_C^{*}: watermarked DC coefficient matrices
 C^{*}: three watermarked color components, including R^{*}, G^{*}, and B^{*}
 I^{*}: watermarked image

Watermark Embedding Procedure:

Start Procedure:

- 1: **Read the color host image**
 $I \leftarrow \text{Lena.ppm}$ (color host image of size 512×512);
 $R \leftarrow I(:, :, 1)$; $G \leftarrow I(:, :, 2)$; $B \leftarrow I(:, :, 3)$;
- 2: **Perform 8×8 APDCBT to three color components**
 $M_C \leftarrow \text{APDCBT}(C) // C = \{R, G, B\}$
- 3: **Perform SVD on M_C**
 $[U_C, S_C, V_C] \leftarrow \text{SVD}(M_C)$
- 4: **Watermark Embedding**
 $S_{CW} = S_C + \alpha \times W_C$
 $[U_{CW}, S_{CW1}, V_{CW}] \leftarrow \text{SVD}(M_C)$
 $M_C^* \leftarrow U_C S_{CW1} V_C^T$
- 5: **Perform the inverse APDCBT**
 $C^* \leftarrow \text{inverse APDCBT}(M_C^*) // C^* = \{R^*, G^*, B^*\}$
 $I^* \leftarrow \text{cat}(3, R^*, G^*, B^*)$

End Procedure

3.3. Watermark Extraction

The watermark extraction process is shown in Figure 1b, and the detailed steps are given below.

Step 1: Divide the suspicious color image I' into R' , G' , and B' components.

Step 2: Divide each component into nonoverlapping 8×8 blocks, and apply the APDCBT to each block to obtain the DC coefficient matrices M'_C , where $C = \{R', G', B'\}$.

Step 3: Perform SVD on M'_C to obtain singular value matrices S'_{CW1} , which can be presented as Equation (8):

$$M'_C = U'_C S'_{CW1} (V'_C)^T, C = \{R', G', B'\}, \quad (8)$$

where U'_C and V'_C are orthogonal matrices for M'_C .

Step 4: The embedded watermark W'_C can be obtained according to Equation (9):

$$S'_C = U_{CW} S'_{CW1} (V_{CW})^T, W'_C = (S'_C - S_C) / \alpha, C = \{R', G', B'\}, \quad (9)$$

Step 5: Obtain the three color components C'_W using Equation (10):

$$C'_W = W'_C (V_{Cs})^T, C = \{R', G', B'\}, \quad (10)$$

Step 6: Apply inverse Fibonacci transform on C'_W to obtain the reconstructed color components C'_{WR} .

Step 7: Reconstruct the final watermark image W' by using the concatenate operator to stack the three color components C'_{WR} , that is, R'_{WR} , G'_{WR} , and B'_{WR} .

To more comprehensively illustrate the watermark extraction process, its pseudocode is shown in Algorithm 3.

Algorithm 3. Watermark Extraction Algorithm

Variable Declaration:

- I' : the suspicious color image
- $R', G',$ and B' : three color components of I'
- M'_C : three DC coefficient matrices obtained by APDCBT, where $C = \{R', G', B'\}$
- U'_C and V'_C : orthogonal matrices for M'_C
- S'_{CW1} : singular value matrices for M'_C
- S_C : saved singular value matrices for M_C of host image
- U_{CW} and V_{CW} : saved orthogonal matrices for S_{CW} in watermark embedding process
- S'_C : watermarked singular value matrices
- α : scaling factor
- W'_C : extracted watermark
- C'_W : three reconstructed color components of watermark image
- C'_{WR} : three inverse scrambled color components of watermark image
- W' : reconstructed watermark image

Watermark Embedding Procedure:

Start Procedure:

- 1: **Read the suspicious color image**
 $I' \leftarrow$ Suspicious image.ppm
 $R' \leftarrow I'(:, :, 1); G' \leftarrow I'(:, :, 2); B' \leftarrow I'(:, :, 3);$
- 2: **Perform 8×8 APDCBT to three color components**
 $M'_C \leftarrow$ APDCBT(C) // $C = \{R', G', B'\}$
- 3: **Perform SVD on M'_C**
 $[U'_C, S'_{CW1}, V'_C] \leftarrow$ SVD(M'_C)
- 4: **Watermark Extraction**
 $S'_C \leftarrow U_{CW} S'_{CW1} (V_{CW})^T // C = \{R', G', B'\}$
 $W'_C \leftarrow (S'_C - S_C) / \alpha$
 $C'_W \leftarrow W'_C (V_{Cs})^T$
- 5: **Apply inverse Fibonacci transform on C'_W**
 $C'_{WR} \leftarrow$ inverse Fibonacci transform (C'_W) // $C = \{R', G', B'\}$
- 6: **Reconstruct the final watermark image**
 $W' \leftarrow$ cat(3, $R'_{WR}, G'_{WR}, B'_{WR}$)

End Procedure

4. Experimental Results and Analysis

In this section, to evaluate the performance of the proposed algorithm, several experiments are performed using MATLAB R2014a on an Intel Core i5-4590 3.30 GHz CPU. All 24-bit color images of size 512×512 in the CVG-UGR image database [39] are prepared as host images. In this paper, six color images, including “Lena”, “Baboon”, “Airplane”, “Sailboat”, “House”, and “Peppers”, are selected to illustrate the experimental results, which are shown in Figure 3. In addition, three 24-bit color images of size 64×64 are used as the watermarks, which are shown in Figure 4. Among them, Figure 4a is the school badge of Shandong University; Figure 4b is the English abbreviation of Shandong University (Weihai) drawn with Microsoft Visio 2010; and Figure 4c is a color QR code containing a personal website [40] generated by a QR code generator. The imperceptibility and robustness of the proposed algorithm are tested and compared with algorithms [18,31,34].

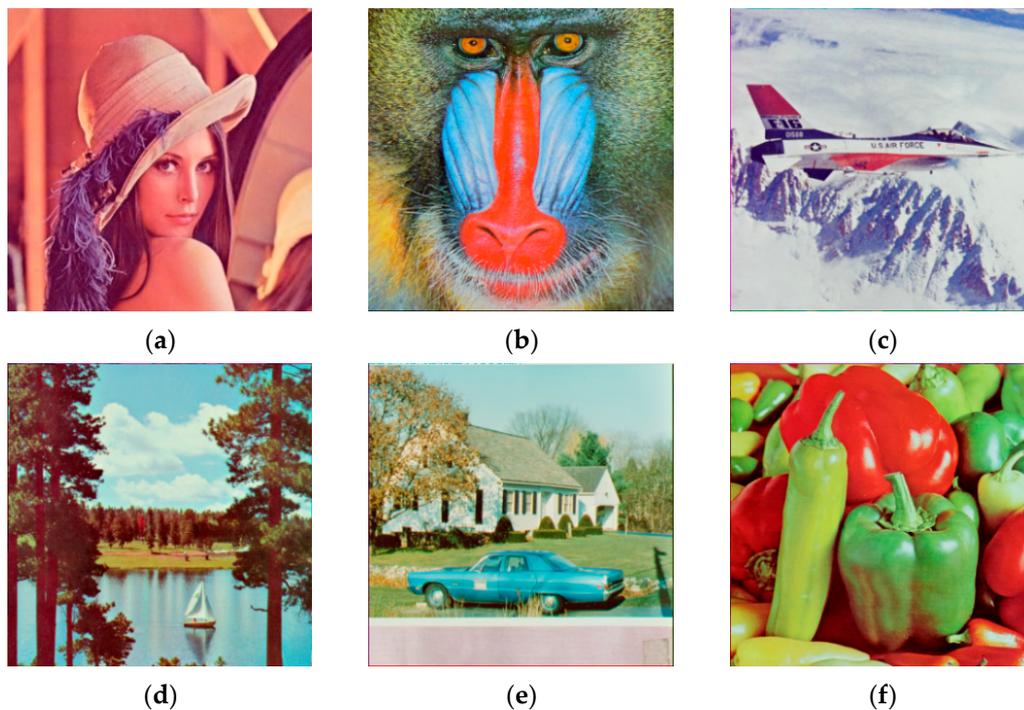


Figure 3. The original host images: (a) Lena; (b) Baboon; (c) Airplane; (d) Sailboat; (e) House; (f) Peppers.

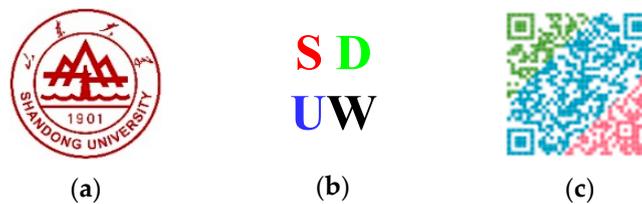


Figure 4. The color watermark images: (a) Logo; (b) SDUW; (c) QR code.

4.1. Evaluation Indexes

In general, peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [41] are used to evaluate the imperceptibility of watermarking algorithms, while the normalized correlation coefficient (NCC) is used to evaluate the robustness of watermarking algorithms [31]. The definition of PSNR is shown as Equation (11):

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}} \text{ (dB)}, \quad (11)$$

where the mean square error (MSE) can be defined as Equation (12):

$$\text{MSE} = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N [f(m, n) - f_w(m, n)]^2, \quad (12)$$

where f and f_w are the original host and watermarked images of size $M \times N$, respectively. In this paper, the PSNR of color images is calculated by averaging the PSNRs of R, G, and B components (PSNR_R, PSNR_G, and PSNR_B), which can be defined as Equation (13):

$$\text{PSNR}_c = \frac{1}{3}(\text{PSNR}_R + \text{PSNR}_G + \text{PSNR}_B), \quad (13)$$

where PSNR_c is the PSNR value of color images. The definition of SSIM is shown as Equation (14):

$$\text{SSIM}(f, f_w) = \frac{(2\mu_f\mu_{f_w} + C_1)(2\sigma_{ff_w} + C_2)}{(\mu_f^2 + \mu_{f_w}^2 + C_1)(\sigma_f^2 + \sigma_{f_w}^2 + C_2)}, \quad (14)$$

where f and f_w are the original host and watermarked images, respectively. μ_f and μ_{f_w} represent the mean values of f and f_w , respectively; σ_f and σ_{f_w} are the variances of f and f_w , respectively; σ_{ff_w} is the covariance between f and f_w ; and C_1 and C_2 are two constants to maintain the stability. Usually, the larger the PSNR and SSIM are, the better the imperceptibility is.

The definition of NCC is shown as Equation (15):

$$\text{NCC} = \frac{\sum_{p=1}^3 \sum_{i=1}^M \sum_{j=1}^N [W(i, j, p) \times W'(i, j, p)]}{\sqrt{\sum_{p=1}^3 \sum_{i=1}^M \sum_{j=1}^N [W(i, j, p)]^2} \sqrt{\sum_{p=1}^3 \sum_{i=1}^M \sum_{j=1}^N [W'(i, j, p)]^2}}, \quad (15)$$

where W and W' are the original color watermark image and extracted watermark image of size $M \times N$, respectively. The NCC values range from 0 to 1, and the larger the NCC is, the stronger the robustness is.

4.2. Imperceptibility

Imperceptibility means that the watermark embedded into the image cannot be perceived by the human eyes, that is, the embedded watermark cannot affect the visual quality of the image. The embedding strength of the watermark plays an important role in the performance of a watermarking algorithm. The selection of it must take the imperceptibility and robustness of the watermark into account. Here, taking “Lena” as the host image, the changes of PSNR, SSIM, and NCC with the embedding strength α are shown in Figure 5.

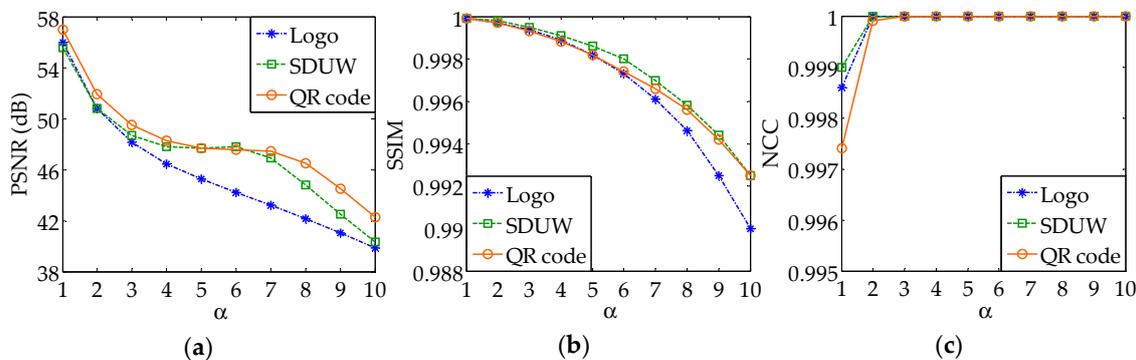


Figure 5. The changes of peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and normalized correlation coefficient (NCC) with the embedding strength α : (a) PSNR; (b) SSIM; (c) NCC.

To show the imperceptibility of the algorithm more comprehensively, taking “Baboon”, “Airplane”, “Sailboat”, “House”, and “Peppers” as objects, the change of the imperceptibility with the embedding strength is tested, and the test results are shown in Table 2.

Table 2. The change of the proposed algorithm’s imperceptibility with the embedding strength α .

Host Images	α	Logo			SDUW			QR Code		
		PSNR (dB)	SSIM	NCC	PSNR (dB)	SSIM	NCC	PSNR (dB)	SSIM	NCC
Baboon	1	55.90	1.0000	0.9992	55.49	1.0000	0.9993	56.88	1.0000	0.9980
	3	47.88	0.9998	1.0000	48.25	0.9998	1.0000	49.20	0.9998	1.0000
	5	44.84	0.9992	1.0000	46.53	0.9995	1.0000	46.78	0.9995	1.0000
	7	42.91	0.9982	1.0000	45.97	0.9988	1.0000	45.80	0.9989	1.0000
Airplane	1	55.69	0.9997	0.9979	55.20	0.9998	0.9986	56.63	0.9998	0.9965
	3	47.17	0.9979	1.0000	47.15	0.9984	1.0000	48.32	0.9984	1.0000
	5	43.63	0.9942	1.0000	44.34	0.9955	1.0000	45.15	0.9960	1.0000
	7	41.43	0.9877	1.0000	43.11	0.9914	1.0000	43.40	0.9922	1.0000
Sailboat	1	55.98	0.9999	0.9995	55.46	0.9999	0.9998	56.85	0.9999	0.9992
	3	48.09	0.9993	1.0000	48.18	0.9994	1.0000	49.09	0.9995	1.0000
	5	45.43	0.9979	1.0000	46.52	0.9986	1.0000	46.62	0.9985	1.0000
	7	44.10	0.9960	1.0000	46.56	0.9973	1.0000	45.78	0.9970	1.0000
House	1	55.78	0.9997	0.9991	55.29	0.9998	0.9995	56.71	0.9998	0.9975
	3	47.49	0.9972	1.0000	47.57	0.9984	1.0000	48.60	0.9985	1.0000
	5	44.25	0.9934	1.0000	45.22	0.9963	1.0000	45.73	0.9965	1.0000
	7	42.33	0.9879	1.0000	44.45	0.9932	1.0000	44.36	0.9936	1.0000
Peppers	1	56.13	0.9826	0.9984	55.79	0.9844	0.9990	57.16	0.9843	0.9977
	3	48.80	0.9729	1.0000	49.70	0.9793	1.0000	50.36	0.9775	1.0000
	5	46.34	0.9685	1.0000	48.89	0.9808	1.0000	49.07	0.9777	1.0000
	7	44.13	0.9662	1.0000	46.06	0.9824	1.0000	46.21	0.9806	1.0000

As can be seen from Figure 5 and Table 2, under the same conditions, the imperceptibility of the watermark using a QR code is better than the other two types of watermarks, which shows that QR code can be used as the carrier of watermark information to achieve the watermark embedding. When the embedding strength α equals 5, most PSNR values are larger than 44 dB and most SSIM values are larger than 0.99, while all the NCC values equal 1. Therefore, the embedding strength in the proposed algorithm is set to 5. Taking “Lena” and “Baboon” as objects, the subjective effects of the watermarked images and the extracted watermark images when without performing any attacks on host images are shown in Table 3.

From Table 3, we can see that the watermarked images have no trace of watermark embedding, and the watermarks can be completely extracted. To compare the proposed algorithm with other three robust image watermarking algorithms, the PSNR, SSIM, NCC, and the watermark capacity of them are listed in Table 4.

From Table 4, we can see that the watermarks in algorithm [18] are gray images, while algorithms [31,34] and the proposed algorithm embed color watermark images into color host images. Although the PSNR of the proposed algorithm is less than algorithm [18] and larger than algorithms [31,34], the SSIM and NCC of the proposed algorithm are both better than those of the other three algorithms.

Table 3. The subjective effects of the proposed algorithm with three color watermark images.

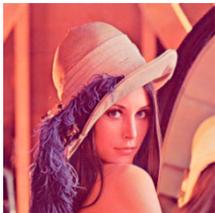
Watermarks	Logo	SDUW	QR Code
Watermarked images (PSNR (dB)/SSIM)	 45.24/0.9982	 47.66/0.9986	 47.70/0.9982
Extracted watermarks (NCC)	 1.0000	 1.0000	 1.0000
Watermarked images (PSNR (dB)/SSIM)	 44.84/0.9992	 46.53/0.9995	 46.78/0.9995
Extracted watermarks (NCC)	 1.0000	 1.0000	 1.0000

Table 4. Comparisons of the PSNR, SSIM, NCC, and the watermark capacity among different color image watermarking algorithms.

Algorithms	Patvardhan et al. [18]	Su et al. [31]	Jia et al. [34]	Proposed
Host images	$512 \times 512 \times 3$			
Watermark images	gray	color	color	color
PSNR (dB)	52.13	46.07	41.45	49.03
SSIM	0.9975	0.9913	0.9842	0.9992
NCC	0.9962	0.9999	1.0000	1.0000
Capacity	128×128	$64 \times 64 \times 3$	$32 \times 32 \times 3$	$64 \times 64 \times 3$

4.3. Robustness

Robustness means that the watermark image can still be extracted from the watermarked image after suffering various attacks. In the storage and transmission processes, it is inevitable for color images to undergo various attacks. To test the robustness of the proposed algorithm, various common image processing attacks and geometric attacks are performed on the watermarked color images, including JPEG compression, Gaussian noise, salt and pepper noise, average filter, median filter, Gaussian filter, sharpening, rotation, and scaling attacks. Taking “Lena”, “Baboon”, and “Sailboat” as objects, when “SDUW” is used as the watermark, the results of extracted watermarks and the NCC values under various attacks are shown in Table 5; when “QR code” is used as the watermark, the results of extracted watermarks and the NCC values under various attacks are shown in Table 6.

Table 5. The results of extracted watermarks (SDUW) and the NCC values under various attacks.

Attacks	Parameter	Lena	Baboon	Sailboat
JPEG	QF = 30	 0.9999	 0.9999	 0.9998
	QF = 60	 1.0000	 1.0000	 1.0000
Gaussian noise	0.01	 0.9997	 0.9998	 0.9997
	0.02	 0.9986	 0.9994	 0.9990
Salt and Pepper noise	0.01	 0.9998	 0.9999	 0.9999
	0.02	 0.9995	 0.9998	 0.9997
Average filter	5 × 5	 0.9991	 0.9988	 0.9984
Median filter	5 × 5	 0.9999	 0.9988	 0.9995
Gaussian filter	5 × 5	 1.0000	 1.0000	 1.0000
Sharpening	0.2	 0.9977	 0.9971	 0.9974
Rotation	5°	 0.9922	 0.9905	 0.9941
Scaling	0.5	 1.0000	 1.0000	 1.0000

Table 6. The results of extracted watermarks (QR code) and the NCC values under various attacks.

Attacks	Parameter	Lena	Baboon	Sailboat
JPEG	QF = 30			
		0.9997	0.9998	0.9996
JPEG	QF = 60			
		0.9999	0.9999	0.9999
Gaussian noise	0.01			
		0.9991	0.9994	0.9994
Gaussian noise	0.02			
		0.9966	0.9979	0.9978
Salt and Pepper noise	0.01			
		0.9996	0.9998	0.9999
Salt and Pepper noise	0.02			
		0.9986	0.9993	0.9991
Average filter	5 × 5			
		0.9978	0.9971	0.9963
Median filter	5 × 5			
		0.9996	0.9972	0.9988
Gaussian filter	5 × 5			
		1.0000	0.9999	0.9999
Sharpening	0.2			
		0.9955	0.9932	0.9941
Rotation	5°			
		0.9886	0.9855	0.9900
Scaling	0.5			
		1.0000	1.0000	1.0000

It can be seen from Tables 5 and 6 that the proposed algorithm has high robustness to JPEG compression, noise, filtering, sharpening, and scaling attacks. Although the proposed algorithm can resist a small degree of rotation attacks, it will fail when the rotation angle increases. In the actual image transmission process, images are usually affected by more than one attack. Most color image watermarking algorithms did not consider the robustness when multiple attacks are combined. In this paper, to more comprehensively prove the effectiveness of the proposed algorithm, taking “Lena” as the host image and “QR code” as the watermark, the robustness of the proposed algorithm is compared with algorithms [18,31,34]. In addition, hybrid attacks, which include two or three kinds of attacks, are also considered. The comparison results are shown in Table 7.

Table 7. Comparisons of the NCC values of the watermarks extracted by different robust color image watermarking algorithms.

Attacks	Patvardhan et al. [18]	Su et al. [31]	Jia et al. [34]	Proposed
Gaussian noise (0.02)	0.8104	0.8300	0.8848	0.9966
Salt and Pepper noise (0.02)	0.9335	0.9024	0.9676	0.9986
Average filter (3 × 3)	0.9910	0.8662	0.9485	0.9996
Median filter (3 × 3)	0.9958	0.8697	0.9622	0.9999
Gaussian filter (3 × 3)	0.9960	0.9449	0.9602	1.0000
JPEG (30)	0.9953	0.8437	0.8831	0.9997
Sharpening (0.2)	0.7964	0.8406	0.8943	0.9955
Rotation (5°)	0.9940	0.9666	0.8709	0.9886
Scaling (2)	0.9961	0.9901	0.9997	1.0000
Gaussian noise (0.02) + Average filter (3 × 3)	0.9951	0.8406	0.8724	0.9964
Salt and Pepper noise (0.05) + Median filter (5 × 5)	0.9937	0.8453	0.9202	0.9995
Gaussian noise (0.02) + JPEG (30)	0.8157	0.8394	0.8785	0.9963
Salt and Pepper noise (0.05) + JPEG (50)	0.8402	0.8432	0.8923	0.9934
Average filter (3 × 3) + JPEG (30)	0.9910	0.8413	0.8801	0.9995
Median filter (3 × 3) + JPEG (70)	0.9950	0.8405	0.9062	0.9999
Gaussian noise (0.02) + Scaling (0.5)	0.9907	0.8389	0.8821	0.9967
Salt and Pepper noise (0.05) + Scaling (2)	0.9639	0.8580	0.9668	0.9938
Average filter (5 × 5) + Scaling (0.5)	0.9883	0.8366	0.9002	0.9977
Median filter (5 × 5) + Scaling (2)	0.9917	0.8826	0.9260	0.9996
JPEG (30) + Scaling (0.5)	0.9895	0.8424	0.8822	0.9998
JPEG (50) + Sharpening (0.2)	0.9394	0.8462	0.8801	0.9969
JPEG (70) + Rotation (5°)	0.9964	0.8326	0.8677	0.9885
Scaling (2) + Sharpening (0.2)	0.9275	0.8306	0.9000	0.9959
Gaussian noise (0.02) + Median filter (3 × 3) + Sharpening (0.2)	0.8190	0.8320	0.8777	0.9925
Salt and Pepper noise (0.05) + Average filter (3 × 3) + Scaling (0.5)	0.9889	0.8306	0.8849	0.9936
Average filter (5 × 5) + Scaling (0.5) + JPEG (30)	0.9883	0.8402	0.8771	0.9980
Scaling (2) + Rotation (2°) + Gaussian filter (3 × 3)	0.9957	0.8339	0.9496	0.9960

As can be seen from Table 7, for JPEG compression, noise, filtering, sharpening, scaling, and most hybrid attacks, the NCC values of the proposed algorithm are higher than the other three algorithms, which means that the proposed algorithm is more robust than the other three algorithms. However, when the watermarked image is attacked by rotation attacks, the robustness of the proposed algorithm is slightly weaker than algorithm [18]. In general, SVD-based watermarking algorithms have stronger robustness than DCT-based watermarking algorithms, and the comparison results in Table 7 are no exception. When suffering noise attacks, sharpening attacks, and some hybrid attacks, the robustness of the proposed algorithm is much better than the other three algorithms.

4.4. Real-Time Analysis

To compare the real-time performance of the four algorithms, we take “Lena”, “Baboon”, “Airplane”, “Sailboat”, “House”, and “Peppers” as objects to test the average execution time for watermark embedding and extraction. The test results are given in Table 8, including embedding time, extraction time, and total time. All values in Table 8 are obtained by averaging the execution time of six objects, and the execution time of each object is obtained by averaging 10 test results.

Table 8. Comparisons of the embedding time, extraction time, and total time among different algorithms (second).

Algorithms	Embedding Time	Extraction Time	Total Time
Patvardhan et al. [18]	1.4668	0.6477	2.1145
Su et al. [31]	5.0233	2.2093	7.2326
Jia et al. [34]	1.9622	0.4170	2.3792
Proposed	1.1560	0.3880	1.5440

From Table 8, we can see that the embedding time, extraction time, and total time of the proposed algorithm are all shorter than those of the other three algorithms. The execution time of reference [31] is the longest, and the execution time of reference [18] is similar to that of reference [34]. As for reference [31], its watermark preprocessing and embedding processes prolong the total execution time. In conclusion, the real-time performance of the proposed algorithm is better than the other three algorithms.

5. Conclusions

In this paper, a double color image watermarking algorithm is proposed. The main contributions are as follows. (1) The APDCBT transform is introduced into the color image watermarking algorithm. The better low-frequency energy accumulation characteristics of the APDCBT and the stability of SSVD are combined to ensure the robustness of the algorithm. (2) The Fibonacci transform with larger key space is used to improve the security of the algorithm. (3) The watermarks for embedding are color images, not gray or binary images. Additionally, the color QR code with error correction mechanism is introduced to be a watermark, which can further improve the robustness of the algorithm. The experimental results show that the algorithm can effectively resist common image processing attacks, such as JPEG compression, Gaussian noise, salt and pepper noise, average filter, median filter, Gaussian filter, sharpening, and scaling attacks, while its robustness to rotation attacks is relatively weak. In the future, we will consider how to improve the resistance of the algorithm to high-intensity rotation attacks.

Author Contributions: X.Y. and C.W. conceived the algorithm and designed the experiments; X.Y. performed the experiments; X.Y. and X.Z. analyzed the results; X.Y. drafted the manuscript; C.W. and X.Z. revised the manuscript. All authors read and approved the final manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 61702303), the Shandong Provincial Natural Science Foundation, China (No. ZR2017MF020), and the Science and Technology Development Plan Project of Weihai Municipality (No. 2018DXGJ07).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Zhang, X.Q.; Wang, X.S. Digital image encryption algorithm based on elliptic curve public cryptosystem. *IEEE Access* **2018**, *6*, 70025–70034. [[CrossRef](#)]
- Denemark, T.; Boroumand, M.; Fridrich, J. Steganalysis features for content-adaptive JPEG steganography. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 1736–1746. [[CrossRef](#)]
- Li, Y.N.; Wang, J.R. Robust content fingerprinting algorithm based on invariant and hierarchical generative model. *Digit. Signal Process.* **2019**, *85*, 41–53. [[CrossRef](#)]
- Schynedel, R.G.V.; Tirkel, A.Z.; Osborne, C.F. A digital watermark. In Proceedings of the 1st IEEE International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994.
- Kamaruddin, N.S.; Kamsin, A.; Por, L.Y.; Rahman, H. A review of text watermarking: Theory, methods and applications. *IEEE Access* **2018**, *6*, 8011–8028. [[CrossRef](#)]
- Wang, C.Y.; Zhang, Y.P.; Zhou, X. Robust image watermarking algorithm based on ASIFT against geometric attacks. *Appl. Sci.* **2018**, *8*, 410. [[CrossRef](#)]

7. Abdulfetah, A.A.; Sun, X.M.; Yang, H.F.; Mohammad, N. Robust adaptive image watermarking using visual models in DWT and DCT domain. *Inf. Technol. J.* **2010**, *9*, 460–466. [[CrossRef](#)]
8. Gupta, R.; Mishra, A.; Jain, S. A semi-blind HVS based image watermarking scheme using elliptic curve cryptography. *Multimed. Tools Appl.* **2018**, *77*, 19235–19260. [[CrossRef](#)]
9. Erkucuk, S.; Krishnan, S.; Zeytinoglu, M. A robust audio watermark representation based on linear chirps. *IEEE Trans. Multimed.* **2006**, *8*, 925–936. [[CrossRef](#)]
10. Mohammed, A.A.; Ali, N.A. Robust video watermarking scheme using high efficiency video coding attack. *Multimed. Tools Appl.* **2018**, *77*, 2791–2806. [[CrossRef](#)]
11. Sathya, S.P.A.; Ramakrishnan, S. Fibonacci based key frame selection and scrambling for video watermarking in DWT–SVD domain. *Wireless Pers. Commun.* **2018**, *102*, 2011–2031. [[CrossRef](#)]
12. Himeur, Y.; Boukabou, A. A robust and secure key-frames based video watermarking system using chaotic encryption. *Multimedia Tools Appl.* **2018**, *77*, 8603–8627. [[CrossRef](#)]
13. Chou, C.H.; Wu, T.L. Embedding color watermarks in color images. *EURASIP J. Appl. Signal Process.* **2003**, *1*, 32–40. [[CrossRef](#)]
14. Su, Q.T.; Yuan, Z.H.; Liu, D.C. An approximate Schur decomposition-based spatial domain color image watermarking method. *IEEE Access* **2019**, *7*, 4358–4370. [[CrossRef](#)]
15. Su, Q.T.; Liu, D.C.; Yuan, Z.H.; Wang, G.; Zhang, X.F.; Chen, B.J.; Yao, T. New rapid and robust color image watermarking technique in spatial domain. *IEEE Access* **2019**, *7*, 30398–30409. [[CrossRef](#)]
16. Barni, M.; Bartolini, F.; Piva, A. Multichannel watermarking of color image. *IEEE Trans. Circuits Syst. Video Technol.* **2002**, *12*, 142–156. [[CrossRef](#)]
17. Liu, K.C. Wavelet-based watermarking for color images through visual masking. *AEU - Int. J. Electron. Commun.* **2010**, *64*, 112–124. [[CrossRef](#)]
18. Patvardhan, C.; Kumar, P.; Lakshmi, C.V. Effective color image watermarking scheme using YCbCr color space and QR code. *Multimed. Tools Appl.* **2018**, *77*, 12655–12677. [[CrossRef](#)]
19. Roy, S.; Pal, A.K. A blind DCT based color watermarking algorithm for embedding multiple watermarks. *Aeu-Int. J. Electron. Commun.* **2017**, *72*, 149–161. [[CrossRef](#)]
20. Niu, P.P.; Wang, X.Y.; Liu, Y.N.; Yang, H.Y. A robust color image watermarking using local invariant significant bitplane histogram. *Multimed. Tools Appl.* **2017**, *76*, 3403–3433.
21. Liu, X.Y.; Wang, Y.F.; Du, J.Y.; Liao, S.H.; Lou, J.T.; Zou, B.J. Robust hybrid image watermarking scheme based on KAZE features and IWT-SVD. *Multimed. Tools Appl.* **2019**, *78*, 6355–6384. [[CrossRef](#)]
22. Chang, T.J.; Pan, I.H.; Huang, P.S.; Hu, C.H. A robust DCT-2DLDA watermark for color images. *Multimed. Tools Appl.* **2019**, *78*, 9169–9191. [[CrossRef](#)]
23. Lakrissi, Y.; Saaïdi, A.; Essahlaoui, A. Novel dynamic color image watermarking based on DWT-SVD and the human visual system. *Multimed. Tools Appl.* **2018**, *77*, 13531–13555. [[CrossRef](#)]
24. Li, J.Z.; Yu, C.Y.; Gupta, B.B.; Ren, X.C. Color image watermarking scheme based on quaternion Hadamard transform and Schur decomposition. *Multimed. Tools Appl.* **2018**, *77*, 4545–4561. [[CrossRef](#)]
25. Rosales-Roldan, L.; Chao, J.H.; Nakano-Miyatake, M.; Perez-Meana, H. Color image ownership protection based on spectral domain watermarking using QR codes and QIM. *Multimed. Tools Appl.* **2018**, *77*, 16031–16052. [[CrossRef](#)]
26. Roy, S.; Pal, A.K. A hybrid domain color image watermarking based on DWT-SVD. *Iran. J. Sci. Technol.-Trans. Electr. Eng.* **2019**, *43*, 201–217. [[CrossRef](#)]
27. Vaidya, P.S.; Mouli, C.P.V.S.S.R. A robust semi-blind watermarking for color images based on multiple decompositions. *Multimed. Tools Appl.* **2017**, *76*, 25623–25656.
28. Mohammad, M.; Gholamhossein, E. A new DCT-based robust image watermarking method using teaching-learning-based optimization. *J. Inf. Secur. Appl.* **2019**, *47*, 28–38.
29. Laur, L.; Rasti, P.; Agoyi, M.; Anbarjafari, G. A robust color image watermarking scheme using entropy and QR decomposition. *Radio Eng.* **2015**, *24*, 1025–1032. [[CrossRef](#)]
30. Cedillo-Hernandez, M.; Cedillo-Hernandez, A.; Garcia-Ugalde, F.J.; Nakano-Miyatake, M.; Perez-Meana, H. Copyright protection of color imaging using robust-encoded watermarking. *Radio Eng.* **2015**, *24*, 240–251. [[CrossRef](#)]
31. Su, Q.T.; Wang, G.; Jia, S.L.; Zhang, X.F.; Liu, Q.M.; Liu, X.X. Embedding color image watermark in color image based on two-level DCT. *Signal Image Video Process.* **2015**, *9*, 991–1007. [[CrossRef](#)]

32. Su, Q.T.; Chen, B.J. An improved color image watermarking scheme based on Schur decomposition. *Multimed. Tools Appl.* **2017**, *76*, 24221–24249. [[CrossRef](#)]
33. Pandey, M.K.; Parmar, G.; Gupta, R.; Sikander, A. Non-blind Arnold scrambled hybrid image watermarking in YCbCr color space. *Microsyst. Technol.* **2019**, *25*, 3071–3081. [[CrossRef](#)]
34. Jia, S.L.; Zhou, Q.P.; Zou, H. A novel color image watermarking scheme based on DWT and QR decomposition. *J. Appl. Sci. Eng.* **2017**, *20*, 193–200.
35. Hou, Z.X.; Wang, C.Y.; Yang, A.P. All phase biorthogonal transform and its application in JPEG-like image compression. *Signal Process. Image Commun.* **2009**, *24*, 791–802. [[CrossRef](#)]
36. Zhou, X.; Zhang, H.; Wang, C.Y. A robust image watermarking technique based on DWT, APDCBT, and SVD. *Symmetry* **2018**, *10*, 77. [[CrossRef](#)]
37. Zhang, Z.; Wang, C.Y.; Zhou, X. Image watermarking scheme based on DWT-DCT and SSVD. *Int. J. Secur. Its Appl.* **2016**, *10*, 191–205.
38. Cai, Y.J.; Niu, Y.G.; Su, Q.T. Blind watermarking algorithm for color images based on DWT-SVD and Fibonacci transformation. *Appl. Res. Comput.* **2012**, *29*, 3025–3028.
39. University of Granada, Computer Vision Group: CVG-UGR Image Database. Available online: <http://decsai.ugr.es/cvg/dbimagenes/c512.php> (accessed on 12 July 2019).
40. Scholar Homepage of Chengyou Wang. Available online: <http://www.scholat.com/wangchengyou> (accessed on 15 July 2019).
41. Ladislav, P.; Jan, K.; Ondrej, Z.; Ondrej, K.; Libor, B.; Martin, S.; Tomas, K. Study of advanced compression tools for stereoscopic video by objective metrics. In Proceedings of the 26th International Conference Radioelektronika, Kosice, Slovakia, 19–20 April 2016.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).