Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2016, Article ID 2497379, 9 pages
http://dx.doi.org/10.1155/2016/2497379

Research Article

A New Double Color Image Watermarking Algorithm Based on the SVD and Arnold Scrambling

Ying Li,[1] Musheng Wei,[1,2] Fengxia Zhang,[1] and Jianli Zhao[1]

[1] *College of Mathematical Sciences, Liaocheng University, Shandong 252000, China*
[2] *College of Mathematics and Science, Shanghai Normal University, Shanghai 200234, China*

Correspondence should be addressed to Ying Li; liyingld@163.com

Received 12 July 2016; Revised 6 October 2016; Accepted 26 October 2016

Academic Editor: M. Montaz Ali

We propose a new image watermarking scheme based on the real SVD and Arnold scrambling to embed a color watermarking image into a color host image. Before embedding watermark, the color watermark image W with size of $M \times M$ is scrambled by Arnold transformation to obtain a meaningless image $\widetilde{W}$. Then, the color host image A with size of $N \times N$ is divided into nonoverlapping $N/M \times N/M$ pixel blocks. In each (i, j) pixel block $A_{i,j}$, we form a real matrix $C_{i,j}$ with the red, green, and blue components of $A_{i,j}$ and perform the SVD of $C_{i,j}$. We then replace the three smallest singular values of $C_{i,j}$ by the red, green, and blue values of $\widetilde{W}_{ij}$ with scaling factor, to form a new watermarked host image $\widetilde{A}_{ij}$. With the reserve procedure, we can extract the watermark from the watermarked host image. In the process of the algorithm, we only need to perform real number algebra operations, which have very low computational complexity and are more effective than the one using the quaternion SVD of color image.

1. Introduction

Security and copyright protection are important issues in multimedia applications and services. Among the protection techniques, digital watermarking is considered a powerful method, which is a technique that hides the secret information in a host image without affecting its normal usage. The watermark can be extracted and used for authentication and verification of ownership. Digital watermarking has been concerned since it has been proposed in [1] in 1994. Over the past 20 years, many watermarking methods have been developed [2–5]. One can use a sequence of random numbers or a recognizable binary pattern or an image as a digital watermark. Most of the existing color techniques use binary or greyscale images [6–9] as watermarks. Due to the rapid applications of the color image technique on the Internet, during the recent years, scientists and researchers have turned their attention to color image watermarks [10–15], which have become one of the hot research topics.

Compared with the binary and greyscale image, the color image can greatly improve the capacity and fidelity of the information. First, the color image can hide a great amount of data. Second, the color perception relies not only on the luminance but also on the chrominance. Color image can be broken into multiple prime color channels, so the color image watermarking is more challenging compared to single channel greyscale images. There are various breakup techniques available in the literature for color image like YIQ, YCbCr, RGB, and HSI, out of which RGB is apparently the most popular space, because this channel format is a natural scheme for representing real world color, and each of the three channels is highly correlated with the other two.

The typical algorithms for color image watermarking can be summarized as follows.

(1) Single Channel Embedding. In this approach, the watermark is embedded into the blue component part of the host image [16]. Such an approach might not be completely adequate since it does not take the implication of the Human Visual System into consideration and in particular it is sensitive to color brightness and perception.

(2) Multichannel Embedding. In this approach, the host and watermark color images are first decomposed into single

color channels and then treated independently [13]. Whether single channel processing or multichannel synthesis, its essence is to treat greyscale image. Therefore, this approach cannot reflect the connection between the channels very well.

(3) Algorithms Based on Quaternion. A color image can be considered as a quaternion matrix of pure imaginary numbers. In this way, the color image can be processed in a holistic manner without losing color information. Notice that these kinds of algorithms require more quaternion operations.

An effective watermarking algorithm should meet certain requirements, including transparency, robustness, adequate information capacity, and low computational complexity.

Watermarking algorithms utilizing the Singular Value Decomposition (SVD) have become popular during the last ten years [14, 17–19], because of the stability of the SVD. SVD-based algorithms embed a watermark by modifying either the singular values or the orthogonal matrices. The work in [17] proposed a SVD-based watermarking scheme where the singular values of the watermark image were used to modify the singular values of the subband images which were obtained by using DWT to the host image. The work in [18] proposed an algorithm to employ SVD on the blue channel of the host image to retrieve the singular values and embed the watermark in these singular values. The work in [19] proposed a SVD-based color image watermarking scheme, in which the singular values of original watermark were required to extract the embedded singular values, and then U and V orthogonal matrices of the original watermark were utilized to recover the watermark. The work in [14] proposed a block-SVD-based watermarking method to embed color watermark to color host image. All these algorithms can preserve the visual quality of the image even for large singular value changes.

In this paper, we propose a new image watermarking scheme based on the real SVD and Arnold scrambling to embed a color watermarking image into a color host image. Before embedding the watermark, the color watermark image W with size of $M \times M$ is scrambled by Arnold transformation to obtain a meaningless image $\widetilde{W}$. Then, the color host image A with size of $N \times N$ is divided into nonoverlapping $N/M \times N/M$ pixel blocks. In each (i, j) pixel block $A_{i,j}$, we form a real matrix $C_{i,j}$ with the red, green, and blue components of $A_{i,j}$ and perform the SVD of $C_{i,j}$. We then replace the three smallest singular values of $C_{i,j}$ by the red, green, and blue values of $\widetilde{W}_{ij}$ with scaling factor, to form a new watermarked host image $\widetilde{A}_{ij}$. With the reserve procedure, we can extract the watermark from the watermarked host image.

The remainder of this paper is organised as follows. In Section 2, we present a brief overview regarding the SVD and image scrambling based on Arnold transformation. In Section 3, we compare the SVD algorithms for color images. In Section 4, we propose embedding and extracting procedures of the new algorithm. The experimental results and analysis are presented in Section 5. Finally, our conclusions are stated in Section 6.

2. Preliminaries

2.1. The SVD. The SVD [20] is one of the most powerful tools and is widely applied to digital image processing [21].

The SVD of an $m \times n$ matrix A is expressed as

$$A = U\Sigma V^H, \tag{1}$$

where both $U_{m\times m}$ and $V_{n\times n}$ are unitary matrices, V^H is conjugate transpose of V, and

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{2}$$

where $\Sigma_1 = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ are positive singular values of A, and r is the rank of matrix A. If A is a real matrix, then both $U_{m\times m}$ and $V_{n\times n}$ are real orthogonal matrices,

$$A = U\Sigma V^T, \tag{3}$$

where V^T is the transpose of V.

2.2. Arnold Scrambling. Scrambling is a pretreatment stage of watermarking, which transforms the meaningful image into another meaningless one and provides security in watermarking scheme. Even if the aggressors obtain the embedded watermarking image, they cannot extract the watermark without the knowledge of the scrambling algorithm.

In the proposed watermarking scheme, the 2D Arnold scrambling transformation [22] is used, which shuffles the pixels positions of watermark image as follows:

$$\begin{gathered} \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} \bmod N, \\ \begin{bmatrix} x \\ y \end{bmatrix} = A^{-1} \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} \bmod N, \\ A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \\ A^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}, \end{gathered} \tag{4}$$

where (x, y) is the old pixel coordinates of the original image and $(\tilde{x}, \tilde{y})$ is the new pixel coordinates after iterative computation scrambling and N is the size of the image. The transformation is periodic and shows periodicity after completing the period, which is shown in Figure 1.

3. Comparison of the SVD Algorithms for Color Images

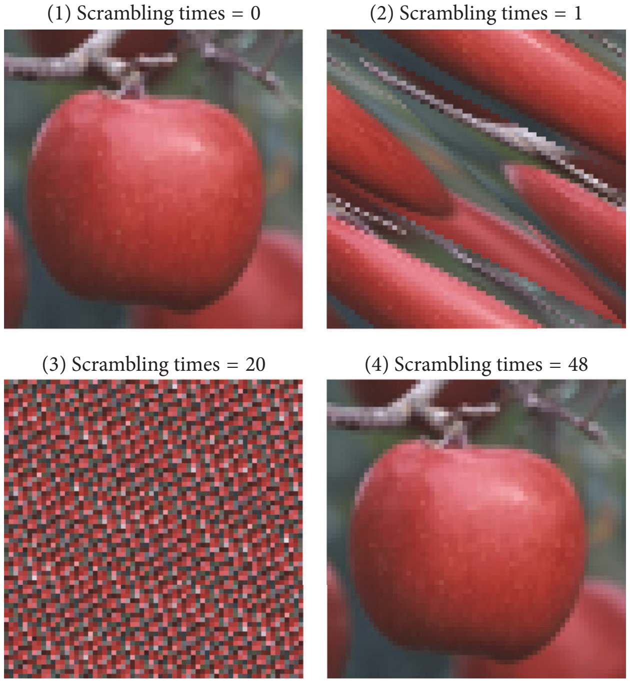
3.1. Quaternion Representation of a Color Image. A quaternion $q \in Q$ is expressed as

$$q = a + bi + cj + dk, \tag{5}$$

where $a, b, c, d \in R$ and three imaginary units i, j, and k satisfy

$$i^2 = j^2 = k^2 = ijk = -1. \tag{6}$$

When $a = 0$, q is called pure quaternion.



(1) Scrambling times = 0

(2) Scrambling times = 1

(3) Scrambling times = 20

(4) Scrambling times = 48

Figure 1: Arnold scrambling.

In [23], Sangwine proposed to encode the three channel components of a RGB image on the three imaginary parts of a pure quaternion; that is,

$$q(x,y) = r(x,y)\,i + g(x,y)\,j + b(x,y)\,k, \quad (7)$$

where $r(x,y)$, $g(x,y)$, and $b(x,y)$ are the red, green, and blue values of the pixel (x,y), respectively. Thus, a color image with m rows and n columns can be represented by a pure imaginary quaternion matrix

$$A = \left(q_{ij}\right)_{m\times n} = \mathrm{R}i + \mathrm{G}j + \mathrm{B}k, \quad q_{ij} \in \mathbb{Q}. \quad (8)$$

Since then, quaternion representation of a color image has attracted great attention. Many researchers applied the quaternion matrix to study the problems of color image processing due to the ability of quaternion matrices to treat the three color channels holistically without losing color information.

3.2. Comparison of Real SVD and Quaternion SVD. For a quaternion matrix $A \in \mathbb{Q}^{m\times n}$ of a color image, we can perform the quaternion SVD with different kinds of algorithms. For example, in [24], authors provided the function "svd" in Matlab Toolbox using quaternion arithmetics. In [25, 26], authors proposed a real structure-preserving algorithm based on the following results.

For any quaternion matrix $A = A_1 + A_2 i + A_3 j + A_4 k$, where $A_1, A_2, A_3, A_4 \in \mathbb{R}^{m\times n}$, its real representation can be defined as follows [25, 26]:

$$A^R \equiv \begin{bmatrix} A_1 & -A_2 & -A_3 & -A_4 \\ A_2 & A_1 & -A_4 & A_3 \\ A_3 & A_4 & A_1 & -A_2 \\ A_4 & -A_3 & A_2 & A_1 \end{bmatrix}. \quad (9)$$

The properties of A^R are as follows.

Theorem 1 (see [27]). *Let $A, B \in \mathbb{Q}^{m\times n}$, $C \in \mathbb{Q}^{n\times s}$, and $a \in \mathbb{R}$. Then, one has the following:*

(1) $(A+B)^R = A^R + B^R$, $(aA)^R = aA^R$, $(AC)^R = A^R C^R$.

(2) $(A^H)^R = (A^R)^T$.

(3) *$A \in \mathbb{Q}^{m\times m}$ is a unitary matrix if and only if A^R is an orthogonal matrix.*

From (9) and Theorem 1, for the matrix A^R, we only need to store the first column block of A^R, denoted as

$$A^R_c = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix}. \quad (10)$$

From this notation and Theorem 1, we have the following result.

Theorem 2. *Let $A, B \in \mathbb{Q}^{m\times n}$, $C \in \mathbb{Q}^{n\times s}$, $q \in \mathbb{Q}^m$, and $a \in \mathbb{R}$. Then, one has the following:*

(1) $(A+B)^R_c = A^R_c + B^R_c$, $(aA)^R_c = aA^R_c$, $(AC)^R_c = A^R C^R_c$.

(2) $(A^H)^R_c = [(A^R)^T]_c$.

(3) $\|A\|_F = \|A^R_c\|_F$, $\|q\|_2 = \|q^R_c\|_2$.

In addition, we have the following results about the SVD of a quaternion matrix and its real representation.

Theorem 3. *Let $A \in \mathbb{Q}^{m\times n}$. Then, the singular values of A^R appear in fours.*

Therefore, if we want to compute the SVD of an $m \times n$ quaternion matrix A, then we can deal with the SVD of the $4m \times 4n$ real representation matrix A^R. In fact, under the orthogonal transformations, the real representation of a quaternion matrix has the standard form stated in the next theorem.

Theorem 4 (see [25]). *Suppose that $A \in \mathbb{Q}^{m\times n}$ and A^R is the real representation of A. Then, there exist orthogonal matrices $U \in \mathbb{R}^{4m\times 4m}$ and $V \in \mathbb{R}^{4n\times 4n}$ such that*

$$U^T A^R V = \begin{bmatrix} D & 0 & 0 & 0 \\ 0 & D & 0 & 0 \\ 0 & 0 & D & 0 \\ 0 & 0 & 0 & D \end{bmatrix}, \quad (11)$$

where $D \in \mathbb{R}^{m\times n}$ is a bidiagonal matrix.

Table 1: Computation amounts and assignment numbers for real SVD and quaternion SVD.

	Real flops	Assignment number
Real SVD	$12mn^2 + 9n^3$	$6n$
Quaternion SVD	$96mn^2 + (64/3)n^3$	$10n$

From Theorems 3 and 4, the SVD of A^R can be obtained by computing the SVD of bidiagonal matrix D. So, we should first get bidiagonal matrix D using Householder based transformation. In [26], we listed three forms of Householder based transformations that appeared in the literatures and proposed a new form of quaternion Householder based transformation. And then, we gave the real structure-preserving algorithms for these quaternion Householder based transformations. By comparison on computation amounts and assignment numbers, we obtained the most flexible and efficient one which is described as follows.

Theorem 5. *Suppose that* $0 \neq y \in \mathbb{Q}^n$ *is not a multiple of* e_1*; denote* $u = (y - \alpha e_1)/\|y - \alpha e_1\|$*, where*

$$\alpha = \begin{cases} -\dfrac{y_1}{|y_1|}\|y\|, & y_1 \neq 0, \\ -\|y\|, & \text{otherwise}, \end{cases} \tag{12}$$

$\alpha_M = \operatorname{diag}(\overline{\alpha}/|\alpha|, I_{n-1})$*, and* $H = \alpha_M(I - 2uu^H)$*, and then* H *maps* y *to* $|\alpha|e_1$.

This kind of Householder based transformation is $H4$ introduced in [26]. For its specific real structure-preserving algorithm, refer to Algorithms 4.1, 4.2, and 4.8 in [26].

After the bidiagonal matrix D is calculated, we can perform a sequence of iterations using Givens rotations on it, and then the SVD of quaternion matrix is obtained.

On the other hand, we can extract the three imaginary parts R, G, and B of matrix A to rearrange as a new real rectangle matrix

$$C = \begin{bmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{bmatrix}, \tag{13}$$

and then directly use the function "svd" on the real matrix C.

Remark 6. Not only is the computation time related to the number of floating points arithmetics, but also it has a lot to do with the assignment number. In Table 1, we list the numbers of real flops and assignment numbers for computing real SVD for $C \in \mathbb{R}^{3m\times n}$ and quaternion SVD for $A \in \mathbb{Q}^{m\times n}$ using a real structure-preserving algorithm, where assignment numbers refer to the number of calling subroutines or performing matrix operations. In matrix operations, say $B = AX + Y$, we adopt the assignment $B = A * X + Y$ to utilize vector pipelining arithmetic operations rather than explicitly using triply nested for-end loops, to speed up computations remarkably. Therefore, real arithmetic numbers as well as assignment numbers are important measures. See, for example, Chapter 1 of [20].

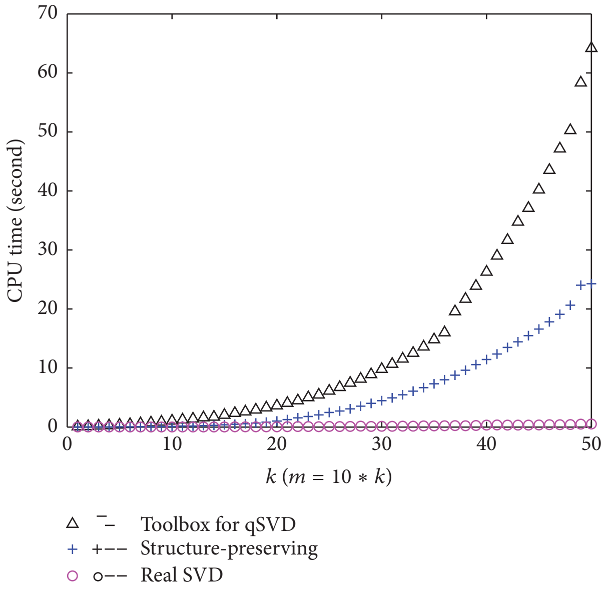


Figure 2: Comparison of the SVD algorithms.

We now provide a numerical example to compare the efficiencies of the three algorithms mentioned above. All these computations are performed on an Intel Core i5@ 2.20 GHz/8 GB computer using Matlab R2013a.

Example 7. For $k = 1 : 50$, $m = 10 * k$, $A \in \mathbb{Q}^{m\times m}$, we apply the above three different algorithms to compute the SVD. We compare the CPU times of three algorithms: qSVD in toolbox for A [24], the structure-preserving SVD for A, and the real SVD for C.

In [25, 26], we have already shown that structure-preserving algorithm is superior to quaternion command "svd" in Matlab. Figure 2 shows that the CPU time of the real SVD algorithm for C is the smallest. Particularly, when the matrix is bigger, its superiority is more obvious.

From the above discussion and Figure 2, we see that the algorithm performing the SVD for C is the most efficient. In the next section, we will propose a new double color image watermarking algorithm based on the real SVD for C.

4. The Proposed Color Image Watermarking Algorithm

In this section, we describe our color image watermarking algorithm, in which a color watermark image is embedded as copyright message into a color host image.

Assume that an original host image A is a RGB color image of size $N \times N$, where $N = 2^n$, and watermark image W is also a RGB color image of size $M \times M$, where $M = 2^m$ and $N \geq 3M$.

4.1. Watermark Embedding

Step 1 (color watermark preprocessing). We shuffle the color watermark image W k times by Arnold scrambling to obtain

$\widetilde{W}$, where k can be used as the secret key for watermark recovery.

Step 2 (partition). The original RGB host color image A is divided into nonoverlapping blocks A_{ij} of size $d \times d$ pixels, where $i, j = 1, 2, \ldots, M,\ d = N/M$.

Step 3 (pixels block rearranging). The R, G, and B color components of A_{ij} are rearranged as rectangle matrices C_{ij} of $3d \times d$.

Step 4 (performing the SVD). Perform the SVD for C_{ij}: $C_{ij} = U_{ij}\Sigma_{ij}V_{ij}^T$, where $\Sigma_{ij} = \operatorname{diag}(\sigma_1^{(ij)}, \ldots, \sigma_d^{(ij)})$.

Step 5 (embedding). Form $\widetilde{\Sigma}_{ij} = \operatorname{diag}(\sigma_1^{(ij)}, \ldots, \sigma_{d-3}^{(ij)}, \alpha\widetilde{W}_{i,j}^{\mathrm{R}}, \alpha\widetilde{W}_{i,j}^{\mathrm{G}}, \alpha\widetilde{W}_{i,j}^{\mathrm{B}})$, where $\widetilde{W}_{i,j}^{\mathrm{R}}$, $\widetilde{W}_{i,j}^{\mathrm{G}}$, and $\widetilde{W}_{i,j}^{\mathrm{B}}$ are red, green, and blue values at (i, j) pixel of $\widetilde{W}$ and α is the scaling factor. Form $\widetilde{C}_{ij} = U_{ij}\widetilde{\Sigma}_{ij}V_{ij}^T$, and then obtain $\widetilde{A}_{i,j}$ from $\widetilde{C}_{ij}$.

4.2. Watermark Extraction. The watermark is extracted as follows.

Step 1. Both the original host image A and the watermarked image $\widetilde{A}$ are divided into nonoverlapping blocks A_{ij} and $\widetilde{A}_{ij}$ of size $d \times d$ pixels, respectively, where $i, j = 1, 2, \ldots, M,\ d = N/M$.

Step 2. The R, G, and B color components of A_{ij} and $\widetilde{A}_{ij}$ are rearranged as rectangle matrices C_{ij} and $\widetilde{C}_{ij}$ of $3d \times d$, respectively.

Step 3. Perform the SVD for C_{ij}: $C_{ij} = U_{ij}\Sigma_{ij}V_{ij}^T$, and

$$\widetilde{\Sigma}_{ij} = \operatorname{diag}\left(\operatorname{diag}\left(U_{ij}^T\widetilde{C}_{ij}V_{ij}\right)\right) = \operatorname{diag}\left(\widetilde{\sigma}_1^{i,j}, \ldots, \widetilde{\sigma}_d^{i,j}\right). \quad (14)$$

Step 4. Extract the watermarks. Set

$$W_{i,j}^{\mathrm{R}} = \frac{\widetilde{\sigma}_{d-2}^{i,j}}{\alpha},$$
$$W_{i,j}^{\mathrm{G}} = \frac{\widetilde{\sigma}_{d-1}^{i,j}}{\alpha}, \quad (15)$$
$$W_{i,j}^{\mathrm{B}} = \frac{\widetilde{\sigma}_{d}^{i,j}}{\alpha}.$$

Step 5. The inverse Arnold scrambling is applied to $W = (W_{ij})$ to construct the watermark.

5. Experimental Results and Analysis

To verify the effectiveness of the proposed algorithm, a series of experiments were conducted where different host images are adopted. We carried out our experiments in Matlab R2013a environment on a laptop with Intel i5 processor rated at 2.5 GHz.

In the first experiment, color image Pepper of size 512 × 512 is taken as the host image, shown in Figure 3(a). A color image Apple of size 64 × 64 is taken as watermark image, shown in Figure 3(b). The detailed implementation procedure of the proposed watermarking model is depicted in Figures 3(c)–3(f). The scrambled watermark after 5-fold Arnold transformation is shown in Figure 3(c). Figure 3(d) shows the watermarked image based on the SVD and Arnold transformation with watermarking scaling factor $\alpha = 0.14$, in which the embedded watermark is invisible. Figure 3(e) shows the extracted scrambling watermark from the watermarked image. Figure 3(f) shows the recovered watermark by the inverse Arnold scrambling, which is similar to the original watermark.

In the second experiment, we use 256 × 256 color image Lena as host image and 64 × 64 color image Apple as watermark. After 5-fold Arnold scrambling, the watermark is embedded with scaling factor 0.28. The detailed implementation procedure is depicted in Figures 4(a)–4(f).

In the third experiment, the watermark is the same as in the above two experiments. We use 1024 × 1024 color image Butterfly as host image and scaling factor 0.14. The detailed implementation procedure is depicted in Figures 5(a)–5(f). We observe from the figures that visually extracted watermarks are quite good compared with the origin ones.

The visual fidelity can be measured by calculating a parameter known as peak signal-to-noise ratio (PSNR) and the structured similarity index (MSSIM) between the original host image A and the watermarked image $\widetilde{A}$. PSNR is expressed in decibel (dB) and is defined as

$$\mathrm{PSNR} = 10\lg\frac{3N^2\left(\max A(x, y, k)\right)^2}{\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{k=1}^{3}\left(A(x, y, k) - \widetilde{A}(x, y, k)\right)^2}, \quad (16)$$

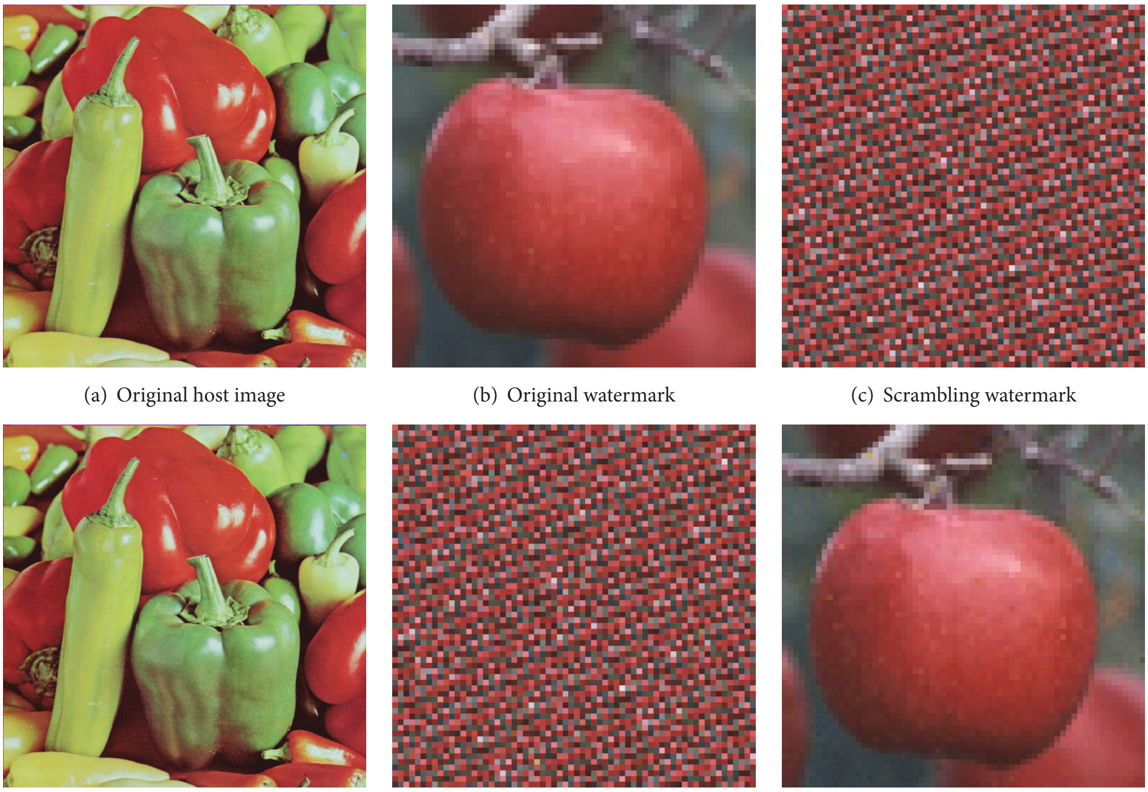
where $\max A(x, y, k)$ represents the maximum pixel value of a color image, and here it is 255. $A(x, y, k)$ and $\widetilde{A}(x, y, k)$ are the pixel values location at position (x, y, k) in the original host image and the watermarked image, respectively. In general, the larger the PSNR value is, the more invisible the watermark is.

The mean structured similarity index (MSSIM) [28] is an objective measure originally developed to assess perceptual image quality. It is superior to PSNR for image quality comparison and better at reflecting the overall similarity of two pictures in terms of appearance rather than simple mathematical point-to-point difference. In the practical application, the image can be divided into blocks by using the sliding window, and the total number of blocks is L, and MSSIM is defined as

$$\mathrm{MSSIM}\left(A, \widetilde{A}\right) = \frac{1}{L}\sum_{k=1}^{L}\mathrm{SSIM}\left(x_k, y_k\right), \quad (17)$$

and the formula used for calculating SSIM is as follows:

$$\mathrm{SSIM}(x, y) = \frac{\left(2\mu_x * \mu_x + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}, \quad (18)$$



(a) Original host image

(b) Original watermark

(c) Scrambling watermark

(d) Watermarked host image

(e) Extracted watermark

(f) Recovery watermark

FIGURE 3: 512 × 512 host image with scale factor 0.14.

where μ_x, μ_y, σ_x^2, σ_y^2, and σ_{xy} are the average, variance, and covariance of x and y, respectively. c_1 and c_2 are constants. The more the MSSIM value gets close to one, the more similar to the original host image the watermarked image is.

In order to measure the quality of the embedded and extracted watermark, the Normalized Correlation (NC) is calculated between the original watermark W and extracted watermark $\widetilde{W}$, which is defined as

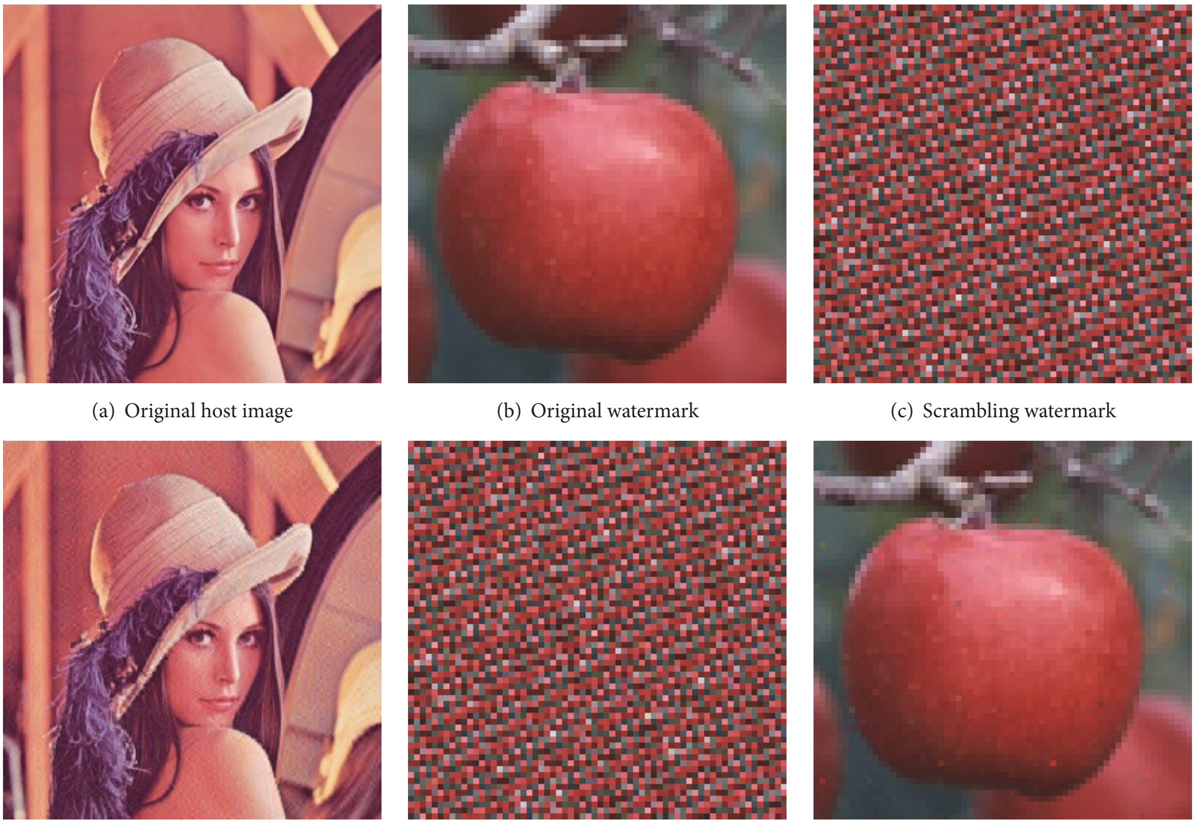
$$\mathrm{NC} = \frac{\sum_{x=1}^{M}\sum_{y=1}^{M}\sum_{k=1}^{3} W(x,y,k) \times \widetilde{W}(x,y,k)}{\sqrt{\sum_{x=1}^{M}\sum_{y=1}^{M}\sum_{k=1}^{3} W^2(x,y,k)} \times \sqrt{\sum_{x=1}^{M}\sum_{y=1}^{M}\sum_{k=1}^{3} \widetilde{W}^2(x,y,k)}}. \quad (19)$$

A higher NC reveals that the extracted watermark resembles the original watermark more closely. If a method has a higher NC value, it is more robust.

In Table 2, we list the PSNR values, MSSIM values, NC values, and the CPU times, in which N, ST, SF, and CPU stand for size of original host image, scrambling times of Arnold transformation, scaling factor, and execution time, respectively. In all the experiments, we observe the good visual quality of watermarked images and good similarity of extracted and original watermarks with the proposed algorithm; the execution time for preprocessing, embedding, and extraction procedure was only 0.5110 seconds for 512 × 512 host image when the number of Arnold iterations is 5. Other examples also show that the execution times of this algorithm are short, and the algorithm has low computational cost and is easily implemented.

6. Conclusions

In this paper, we have proposed a new double RGB color image watermarking algorithm based on the real SVD and Arnold scrambling. First, the color watermark image is scrambled by Arnold transformation to obtain a meaningless image. Then, the original host image is divided into nonoverlapping pixel blocks. We form a real matrix with the red, green, and blue components in each pixel block and perform the SVD of the real matrices. We then replace the three smallest singular values of each real matrix by the red, green,



(a) Original host image

(b) Original watermark

(c) Scrambling watermark

(d) Watermarked host image

(e) Extracted watermark

(f) Recovery watermark

Figure 4: 256 × 256 host image with scale factor 0.28.

Table 2: PSNR, MSSIM, and NC values and CPU times.

Host image	N	ST	SF	PSNR	MSSIM	NC	CPU (second)
Lena	256	0	0.28	26.5018	0.8942	0.9998	0.3346
Lena	256	1	0.28	26.1693	0.8892	0.9998	0.3363
Lena	256	5	0.28	26.2047	0.8840	0.9998	0.3380
Pepper	512	0	0.14	42.1996	0.9946	0.9995	0.5012
Pepper	512	1	0.14	42.1940	0.9949	0.9994	0.5075
Pepper	512	5	0.14	42.0375	0.9947	0.9995	0.5110
Butterfly	1024	0	0.14	44.3700	0.9954	0.9995	1.4372
Butterfly	1024	1	0.14	44.3636	0.9949	0.9995	1.4226
Butterfly	1024	5	0.14	44.3654	0.9949	0.9995	1.4257

and blue values of corresponding pixel of the scrambled watermark with scaling factor, to form a new watermarked host image. With the reserve procedure, we can extract the watermark from the watermarked host image.

The experimental results show that the proposed algorithm achieves high PSNR, high MSIIM of the watermarked image, and high NC of the extracted watermark. In addition, in the process of the algorithm, we only need to perform real number algebra operations, which have very low computational complexity and therefore are more effective than the one using the quaternion SVD of color image, which costs a large amount of quaternion operations.

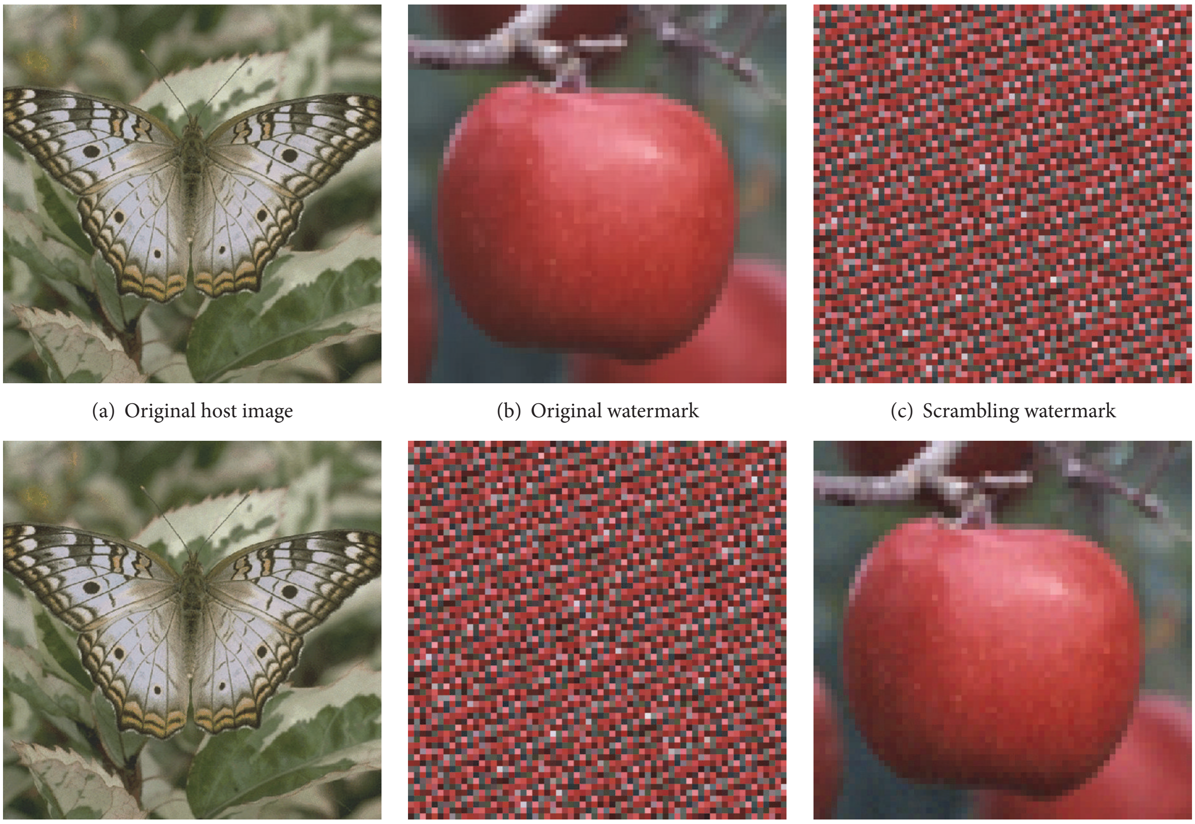
The idea described in this paper can also be applied to other kinds of methods concerning double color image watermarking problems.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grants 11171226 and 11301247,



(a) Original host image

(b) Original watermark

(c) Scrambling watermark

(d) Watermarked host image

(e) Extracted watermark

(f) Recovery watermark

Figure 5: 1024 × 1024 host image with scale factor 0.14.

the Natural Science Foundation of Shandong under Grant ZR2012FQ005, Science and Technology Project of Department of Education, Shandong Province (J15LI10), and the Science Foundation of Liaocheng University under Grants 31805 and 318011318.

References

[1] R. G. Schyndel, A. Z. Tirkel, and C. F. Osborne, “A digital watermark,” in *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. 86–91, 1994.

[2] K. Bailey and K. Curran, “An evaluation of image based steganography methods,” *Multimedia Tools and Applications*, vol. 30, no. 1, pp. 55–88, 2006.

[3] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, “Digital image steganography: survey and analysis of current methods,” *Signal Processing*, vol. 90, no. 3, pp. 727–752, 2010.

[4] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, “Watermarking digital image and video data,” *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20–46, 2000.

[5] F. A. P. Petitcolas and R. J. Anderson, “Evaluation of copyright marking systems,” in *Proceedings of the 6th International Conference on Multimedia Computing and Systems (ICMCS ’99)*, pp. 574–579, Florence, Italy, June 1999.

[6] C. Deng, X. Gao, X. Li, and D. Tao, “A local tchebichef moments-based robust image watermarking,” *Signal Processing*, vol. 89, no. 8, pp. 1531–1539, 2009.

[7] C. Deng, X. Gao, X. Li, and D. Tao, “Local histogram based geometric invariant image watermarking,” *Signal Processing*, vol. 90, no. 12, pp. 3256–3264, 2010.

[8] S. Rawat and B. Raman, “A new robust watermarking scheme for color images,” in *Proceedings of the IEEE 2nd International Advance Computing Conference (IACC ’10)*, pp. 206–209, Patiala, India, February 2010.

[9] C. G. Thorat and B. D. Jadhav, “A blind digital watermark technique for color image based on integer wavelet transform and SIFT,” *Procedia Computer Science*, vol. 2, pp. 236–241, 2010.

[10] Q. Su, Y. Niu, G. Wang, S. Jia, and J. Yue, “Color image blind watermarking scheme based on QR decomposition,” *Signal Processing*, vol. 94, no. 1, pp. 219–235, 2014.

[11] C.-H. Chou and T.-L. Wu, “Embedding color watermarks in color images,” *EURASIP Journal on Advances in Signal Processing*, vol. 2003, Article ID 548941, pp. 32–40, 2003.

[12] S. Rastegar, F. Namazi, K. Yaghmaie, and A. Aliabadian, “Hybrid watermarking algorithm based on Singular Value Decomposition and Radon transform,” *AEU—International Journal of Electronics and Communications*, vol. 65, no. 7, pp. 658–663, 2011.

[13] Q. Su, Y. Niu, H. Zou, Y. Zhao, and T. Yao, “A blind double color image watermarking algorithm based on QR decomposition,” *Multimedia Tools and Applications*, vol. 72, no. 1, pp. 987–1009, 2014.

[14] N. E. H. Goléa, R. Seghir, and R. Benzid, “A bind RGB color image watermarking based on singular value decomposition,” in *Proceedings of the ACS/IEEE International Conference on*

Computer Systems and Applications (AICCSA '10), pp. 1–5, Hammamet, Tunisia, May 2010.

[15] Y. Xing and J. Tan, "A color watermarking scheme based on block-SVD and Arnold transformation," in *Proceedings of the 2nd Workshop on Digital Media and Its Application in Museum and Heritage*, pp. 3–8, December 2007.

[16] M. Kutter, F. D. Jordan, and F. Bossen, "Digital signature of color images using amplitude modulation," in *Storage and Retrieval for Image and Video Databases V*, vol. 3022 of *Proceedings of SPIE*, pp. 518–526, San Jose, Calif, USA, February 1997.

[17] E. Ganic and A. M. Eskicioglu, "Robust DWT-SVD domain image watermarking: embedding data in all frequencies," in *Proceeding of the ACM Multimedia and Security Workshop*, pp. 166–174, Magdeburg, Germany, September 2004.

[18] D. Vaishnavi and T. S. Subashini, "Robust and invisible image watermarking in RGB color space using SVD," *Procedia Computer Science*, vol. 46, pp. 1770–1777, 2015.

[19] C.-Q. Yin, L. Li, A.-Q. Lv, and L. Qu, "Color image watermarking algorithm based on DWT-SVD," in *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL '07)*, pp. 2607–2611, Jinan, China, August 2007.

[20] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 4th edition, 2013.

[21] H. C. Andrews and C. L. Patterson, "Singular value decompositions and digital image processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 1, pp. 26–53, 1976.

[22] V. J. Arnold and A. Avez, *The Mathematical Physics Monograph Series:ergodic Problems of Classical Mechanics*, Benjamin, New York, NY, USA, 1968.

[23] S. J. Sangwine, "Fourier transforms of colour images using quaternion or hypercomplex, numbers," *Electronics Letters*, vol. 32, no. 21, pp. 1979–1980, 1996.

[24] S. Sangwine and N. L. Bihan, Quaternion toolbox for matlab, http://qtfm.sourceforge.net/.

[25] Y. Li, M. Wei, F. Zhang, and J. Zhao, "A fast structure-preserving method for computing the singular value decomposition of quaternion matrices," *Applied Mathematics and Computation*, vol. 235, pp. 157–167, 2014.

[26] Y. Li, M. Wei, F. Zhang, and J. Zhao, "Real structure-preserving algorithms of Householder based transformations for quaternion matrices," *Journal of Computational and Applied Mathematics*, vol. 305, pp. 82–91, 2016.

[27] M. Wang and W. Ma, "A structure-preserving algorithm for the quaternion Cholesky decomposition," *Applied Mathematics and Computation*, vol. 223, pp. 354–361, 2013.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.