

# Consideration of A Robust Watermarking Algorithm for Color Image Using Improved QR Decomposition

Phuong Thi Nha

LQD

Ta Minh Thanh (✉ [taminhjp@gmail.com](mailto:taminhjp@gmail.com))

Dai hoc Ky thuat Le Quy Don

Nguyen Tua Phong

LQD

---

## Research Article

**Keywords:** color image, negative

**Posted Date:** May 4th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-423766/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

**Version of Record:** A version of this preprint was published at Soft Computing on March 31st, 2022. See the published version at <https://doi.org/10.1007/s00500-022-06975-3>.

### **Compliance with Ethical Standards**

Hereby, I, Ta Minh THANH, consciously assure that for the manuscript "*Consideration of A Robust Watermarking Algorithm for Color Image Using Improved QR Decomposition*" the following is fulfilled:

- 1) This material is the authors' own original work, which has not been previously published elsewhere.
- 2) The paper is not currently being considered for publication elsewhere.
- 3) The paper reflects the authors' own research and analysis in a truthful and complete manner.
- 4) The paper properly credits the meaningful contributions of co-authors and co-researchers.
- 5) The results are appropriately placed in the context of prior and existing research.
- 6) All sources used are properly disclosed (correct citation). Literally copying of text must be indicated as such by using quotation marks and giving proper reference.
- 7) All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

I agree with the above statements and declare that this submission follows the policies of Solid State Ionics as outlined in the Guide for Authors and in the Ethical Statement.

Date: Apr 13, 2021

Corresponding author's signature:  
Ta Minh THANH

# Consideration of A Robust Watermarking Algorithm for Color Image Using Improved $QR$ Decomposition

Phuong Thi Nha<sup>1a</sup>, Ta Minh Thanh<sup>1b,\*</sup>, Nguyen Tuan Phong<sup>1c</sup>

<sup>1</sup>Le Quy Don University, 236 Hoang Quoc Viet, Ha Noi, Viet Nam

Received: date / Accepted: date

**Abstract** In order to protect the color image copyright protection of the digital multimedia, it is necessary to design a color image watermarking algorithm. To achieve this purpose, an improved color image watermarking scheme based on  $QR$  decomposition for color image matrix is proposed in this paper. The proposed method gives a new algorithm to find elements of  $Q$  and  $R$  matrices instead of using Gram-Schmidt algorithm for  $QR$  factorization. First,  $R$  matrix is performed by solving a set of linear equations where diagonal elements of  $R$  are checked and modified if they are zero or negative. After that,  $Q$  matrix is computed based on  $R$  matrix. In addition, a novel formula is proposed to improve extracting time where the first element  $R(1, 1)$  of  $R$  matrix is found instead of computing  $QR$  decomposition as the previous proposals. Experimental results show that the proposed method outperforms other methods considered in this paper in term of the quality of the watermarked images and the robustness of embedding method. Furthermore, the execution time is significantly improved and the watermark image is more robust under some tested attacks.

## 1 Introduction

### 1.1 Overview

In recent years, exchanging data via the internet has become more and more popular. With the rapidly development of digital technology techniques and devices, it has brought a lot of convenience for users. However, it is also a fertile land for attacks who want to steal/replace or fake information. Therefore, inspecting data's integrity and authentication is an extremely important issue to tackle risks. For images, beside general rules of the law, there are some applied methods such as encryption, information hiding, watermarking to help owners to protect their digital copyright. Among these techniques, the image watermarking technique has been known as the best one until now. Watermarking is a process of embedding digital information called watermark into image by some constraints.

\*Corresponding author

<sup>a</sup>e-mail: phuongthinha@lqdtu.edu.vn

<sup>b</sup>e-mail: thanhtm@mta.edu.vn

<sup>c</sup>e-mail: tuanphongnguyen1974@gmail.com

Depending on the watermark embedding domain, digital watermarking methods can be divided into two main categories: spatial domain and transform domain. In spatial domain techniques, the watermark is inserted by directly altering pixel intensities of the cover image [1]. Altering least significant bits (LSB) of the cover image is one of the common spatial domain based watermarking techniques. Spatial domain methods have low computational complexity, but they are not usually robust against almost image processing or other attacks. On the other hand, in transform domain methods, the original image is first transformed into the frequency domain by several transformation methods, such as discrete cosine transform (*DCT*), discrete wavelet transform (*DWT*) or matrix decomposition such as singular value decomposition (*SVD*), *QR* decomposition, *LU* decomposition and *Schur* decomposition. Then, according to the certain criteria, the transform domain coefficients are altered for embedding the watermark information. Finally, inverse transform is applied to obtain the watermarked digital image. Although watermarking methods in the frequency domain have high computational complexity, they are always more robust than spatial domain watermarking schemes.

Image watermarking schemes based on *DCT* transformation often embed watermark on median frequency to harmonize between the quality of watermarked image and the robustness of extracted information [2, 3]. If embedding is implemented on low frequency, the extracted watermark is good but the invisibility of watermarked image is bad, and vice versa. For *DWT* transformation, the watermark is commonly inserted on *LL* low sub-domain to archive a robust result [4]. However, for balancing the the quality of watermarked image and the robustness of extracted watermark, the watermark is commonly embedded into *HL* and *LH* sub-band.

For *SVD* decomposition, there are two trends for embedding and extracting process. The first one uses the first element  $D(1, 1)$  of *D* triangular matrix to change pixel values [5, 6]. Whereas, an other way executes modifying elements of the first column of *U* matrix [7, 8]. Sun *et al.* in [5] designed a novel watermarking scheme based on *SVD*, in which the watermark was embedded into  $512 \times 512$  images by modifying the first coefficient of the *D* triangular matrix. This method had better performance in term of robustness because the authors proposed an excellent formula to embed as well as extract watermark. In addition, there was a novel scheme which is proposed by An-Wei Luo in 2020 [8]. This proposal performed an optimal *SVD* blocks selection strategy to improve the imperceptibility and used different embedding strengths for each block. However, embedding on two elements  $U(2, 1)$  and  $U(3, 1)$  of *U* matrix reduced the quality of the watermarked image.

While the time required to conduct *SVD* computation is about  $11n^3$  flops, the Schur decomposition needs fewer number of flops which is approximately  $8n^3/3$  for a  $n \times n$  matrix. That is the reason why some researchers focused on kind of this matrix analysis [9, 10]. Su [10] in 2020 described a new based Schur algorithm where the  $U(2, k)$  and  $U(3, k)$  elements of *U* unitary matrix are chosen for embedding (with *k* is a row index of *D* triangular matrix that contains the biggest value). Although Schur decomposition takes execution time less than *SVD* factorization, it is still a complex transformation.

There were several researchers who utilized *QR* decomposition to transform pixel matrices in their watermarking techniques [11–16]. *QR* decomposition is an intermediate step of Schur decomposition, so it requires less number of computations. From 2013 to 2019, Qingtang Su had three papers which focused on *QR* factorization. In the first method, Su performed a watermarking process which depends on the relation between the second row first column coefficient and the third row first column coefficient of *Q* unitary matrix [11]. After that, Su improved it by alternative choices in years of 2017 and 2019 where the author divided the host image into  $3 \times 3$  blocks instead of size of  $4 \times 4$  in the papers [14, 16].

These improvements enhanced the embedded watermark capacity, however the quality of watermarked image was heavier affected.

Beside using separately above methods, many authors also had hybrid image watermarking schemes to strengthen the robustness of watermark in recent years. That is a combination of *DWT* and *SVD* [17–21], *DCT* and *SVD* [22], *DWT* and *DCT* [23], *DWT* and *QR* [24, 25], or *DWT* and *LU* [26]. The experimental results of these proposals showed that robustness of extracted watermark is more improved than previous researches. Normalized Correlation (*NC*) value, which measures robustness, is often up to 90% under almost image attacks. However, the invisibility of watermarked image is only around 40dB by Peak Signal to Noise Ratio (*PSNR*) index. Furthermore, these methods cost the computational complexity and they are not suitable for real-time systems.

## 1.2 Challenging issues

As above discussions, an effective image watermarking scheme needs to satisfy three main criteria that involve quality of watermarked image, robustness of extracted watermark, and execution time. In order to balance these requirements, a novel method, which is based on formula of Sun [5] and *QR* decomposition, is proposed in our paper. Sun [5] embedded information on the first element  $D(1, 1)$  of *D* triangular matrix after decomposing *SVD* (called *SunSVD*). In their method, the embedding and extracting formula in this proposal was referred by many researchers due to its stability. Because of high computational complexity, *SVD* decomposition should be replaced with *QR* decomposition. From this idea, a combination between *QR* decomposition and the formula of Sun (called *SunQR*) is experimented. It uses Gram-Schmidt algorithm [27] for *QR* factorization and the formula of Sun for embedding as well as extracting watermark. To be similar to *SVD* decomposition, the first element  $R(1, 1)$  of upper triangular matrix *R* is used to embed and extract information.

As experimental results, although this method increases speed of calculating, it gives out worse the invisibility and the robustness of the watermarked image than the one based on *SVD* decomposition. A reason for this is because Gram-Schmidt algorithm calculates *Q* and *R* column by column, so it does not inspect diagonal elements of *R* matrix if these values are zero or negative.

## 1.3 Our contributions

Our contributions can be summarized as follows:

1. To overcome drawbacks of *SunQR*, a new watermarking scheme based on *QR* decomposition is proposed in this paper where *R* matrix is computed at first and diagonal elements of *R* matrix are checked and modified if they are zero or negative.
2. Calculating elements of *R* is performed by solving a set of linear equations; after that, *Q* is computed based on *R*. Due to this computation, the quality of the watermarked image will be significantly improved.
3. In addition, for improving extracting time, a novel formula is designed to get out the first element  $R(1, 1)$  of *R* matrix instead of calculating *QR* factorization as the previous proposals. Based on our proposed extracting scheme, our method can reduce significantly execution time comparing with another methods. That makes our method can be suitable for real-time applications.

## 1.4 Roadmap

The rest of this paper is organized as follows. Sect. 2 describes the  $QR$  decomposition theory and its special features. Then, Sect. 3 introduces the details of our watermark embedding and our watermark extraction procedure. After that, Sect. 4 gives the experimental results and discussion. Finally, Sect. 5 concludes this paper.

## 2 Preliminaries

### 2.1 $QR$ decomposition

$QR$  decomposition (also called  $QR$  factorization) of a matrix  $A$  is a decomposition of the matrix into two matrices as Eq. (1).

$$A = QR, \quad (1)$$

where  $Q$  is an orthogonal matrix (*i.e.*  $Q^T Q = Q Q^T = I$ ) and  $R$  is an upper triangular matrix. If  $A$  is non-singular, then this factorization is unique.

For example, a matrix  $A$  of size  $4 \times 4$  as

$$A = \begin{bmatrix} 40 & 39 & 39 & 38 \\ 40 & 40 & 40 & 39 \\ 41 & 42 & 42 & 40 \\ 44 & 46 & 44 & 41 \end{bmatrix}$$

can be factored into an orthogonal matrix  $Q$  and an upper triangular matrix  $R$  by  $QR$  decomposition like this,

$$Q = \begin{bmatrix} 0.4845 & -0.6921 & -0.3499 & -0.4049 \\ 0.4845 & -0.2374 & 0.1917 & 0.8199 \\ 0.4966 & 0.2113 & 0.7381 & -0.4049 \\ 0.5329 & 0.6481 & -0.5440 & 0.0000 \end{bmatrix}$$

$$R = \begin{bmatrix} 82.5651 & 83.6431 & 82.5772 & 79.0164 \\ 0 & 2.1996 & 0.9033 & -0.5341 \\ 0 & 0 & 1.0880 & 1.4020 \\ 0 & 0 & 0 & 0.3947 \end{bmatrix}$$

### 2.2 Arnold transform

For improving the security of watermarking method, Arnold transform is often used to permute watermark image [10], and its detailed permutation process is given by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N}, \quad (2)$$

where  $x'$ ,  $y'$ ,  $x$  and  $y$  are integers in  $\{0, 1, 2, \dots, N-1\}$  and  $N$  is order of watermark image matrix. The image pixel at the coordinate  $(x, y)$  can be permuted to new coordinate  $(x', y')$  by Eq. (2), which disorganizes the order of watermark image and enhances the security in visual identification of watermark images. Moreover, the number of permutation times of Arnold transform is often used as the secret key.

### 2.3 The special feature of $Q$ and $R$ matrix

By multiplying two sides of the equation with  $A^T$  (is a transposition of  $A$ ), Eq. (1) in Sect. 2.1 becomes as follows:

$$\begin{aligned}
 A^T A &= A^T Q R = M \\
 \Rightarrow A^T A &= (Q R)^T Q R = M \\
 \Rightarrow A^T A &= R^T Q^T Q R = M \\
 \Rightarrow A^T A &= R^T R = M
 \end{aligned} \tag{3}$$

(since  $Q$  is an orthogonal matrix, so  $Q^T Q = I$ , where  $I$  is the identity matrix.)

Since  $R$  is an upper triangular matrix,  $R$  can be computed easily by solving a set of linear equations. Supposedly, the host matrix  $A$  has size of  $4 \times 4$ . Therefore, the  $M$ ,  $A^T$ ,  $R^T$  and  $R$  matrices are also  $4 \times 4$  matrices. The elements of  $A$  are represented as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = [a_1 \ a_2 \ a_3 \ a_4],$$

where  $a_1, a_2, a_3, a_4$  are column vectors of  $A$ , respectively. We have

$$A^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{32} & a_{42} \\ a_{31} & a_{23} & a_{33} & a_{43} \\ a_{41} & a_{24} & a_{34} & a_{44} \end{bmatrix}, \quad M = A^T A = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ 0 & r_{22} & r_{23} & r_{24} \\ 0 & 0 & r_{33} & r_{34} \\ 0 & 0 & 0 & r_{44} \end{bmatrix}, \quad R^T = \begin{bmatrix} r_{11} & 0 & 0 & 0 \\ r_{12} & r_{22} & 0 & 0 \\ r_{13} & r_{23} & r_{33} & 0 \\ r_{14} & r_{24} & r_{34} & r_{44} \end{bmatrix}$$

In order to find the elements of  $R$  matrix, we will begin with Eq. (3).

$$R^T R = M \Leftrightarrow \begin{bmatrix} r_{11} & 0 & 0 & 0 \\ r_{12} & r_{22} & 0 & 0 \\ r_{13} & r_{23} & r_{33} & 0 \\ r_{14} & r_{24} & r_{34} & r_{44} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ 0 & r_{22} & r_{23} & r_{24} \\ 0 & 0 & r_{33} & r_{34} \\ 0 & 0 & 0 & r_{44} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}$$

Therefore, we have

$$r_{11} r_{11} = m_{11} \Rightarrow r_{11} = \sqrt{m_{11}} \tag{4}$$

$$r_{11} r_{12} = m_{12} \Rightarrow r_{12} = \frac{m_{12}}{r_{11}}$$

$$r_{11} r_{13} = m_{13} \Rightarrow r_{13} = \frac{m_{13}}{r_{11}}$$

$$r_{11} r_{14} = m_{14} \Rightarrow r_{14} = \frac{m_{14}}{r_{11}}$$

$$r_{12} r_{12} + r_{22} r_{22} = m_{22} \Rightarrow r_{22} = \sqrt{m_{22} - r_{12}^2} \tag{5}$$

$$r_{12} r_{13} + r_{22} r_{23} = m_{23} \Rightarrow r_{23} = \frac{m_{23} - r_{12} r_{13}}{r_{22}}$$

$$r_{12}r_{14} + r_{22}r_{24} = m_{24} \Rightarrow r_{24} = \frac{m_{24} - r_{12}r_{14}}{r_{22}}$$

$$r_{13}r_{13} + r_{23}r_{23} + r_{33}r_{33} = m_{33} \Rightarrow r_{33} = \sqrt{m_{33} - r_{13}^2 - r_{23}^2} \quad (6)$$

$$r_{13}r_{14} + r_{23}r_{24} + r_{33}r_{34} = m_{34} \Rightarrow r_{34} = \frac{m_{34} - r_{13}r_{14} - r_{23}r_{24}}{r_{33}}$$

$$\begin{aligned} r_{14}r_{14} + r_{24}r_{24} + r_{34}r_{34} + r_{44}r_{44} &= m_{44} \\ \Rightarrow r_{44} &= \sqrt{m_{44} - r_{14}^2 - r_{24}^2 - r_{34}^2} \end{aligned} \quad (7)$$

In general, if the host matrix  $A$  has size of  $n \times n$ , we have

$$\begin{aligned} r_{11} &= \sqrt{m_{11}} \\ r_{1j} &= \frac{m_{1j}}{r_{11}} \\ r_{ij} &= \begin{cases} \sqrt{m_{ij} - \sum_{k=1}^{i-1} r_{kj}^2}, & i = j \\ \frac{(m_{ij} - \sum_{k=1}^{i-1} r_{ki}r_{kj})}{r_{ii}}, & i \neq j \end{cases} \end{aligned} \quad (8)$$

where  $i, j = 2, 3, \dots, n$ .

Calculating  $Q$  is based on  $R$  matrix.  $Q$  matrix can be expressed by column as follows:

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix} = [q_1 \ q_2 \ q_3 \ q_4]$$

The Gram-Schmidt algorithm, which was introduced in [27], computes  $Q$  column by column. According to that, the columns of  $Q$  are calculated as follows:

$$\tilde{q}_1 = a_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix}$$

and

$$\begin{aligned} q_1 &= \frac{1}{r_{11}} \tilde{q}_1 \\ \tilde{q}_2 &= a_2 - r_{12}q_1 = \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ a_{42} \end{bmatrix} - r_{12} \begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} q_2 &= \frac{1}{r_{22}} \tilde{q}_2. \\ \tilde{q}_3 &= a_3 - r_{13}q_1 - r_{23}q_2 = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \\ a_{43} \end{bmatrix} - r_{13} \begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \end{bmatrix} - r_{23} \begin{bmatrix} q_{12} \\ q_{22} \\ q_{32} \\ q_{42} \end{bmatrix} \end{aligned}$$



and

$$q_3 = \frac{1}{r_{33}} \tilde{q}_3$$

$$\tilde{q}_4 = a_4 - r_{14}q_1 - r_{24}q_2 - r_{34}q_3$$

$$= \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \\ a_{44} \end{bmatrix} - r_{14} \begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \end{bmatrix} - r_{24} \begin{bmatrix} q_{12} \\ q_{22} \\ q_{32} \\ q_{42} \end{bmatrix} - r_{34} \begin{bmatrix} q_{13} \\ q_{23} \\ q_{33} \\ q_{43} \end{bmatrix}$$

and

$$q_3 = \frac{1}{r_{33}} \tilde{q}_3$$

In general, if the host matrix  $A$  has size of  $n \times n$ , we have

$$\tilde{q}_1 = a_1 \quad \text{and} \quad q_1 = \frac{1}{r_{11}} \tilde{q}_1$$

$$\tilde{q}_i = a_i - (r_{1i}q_1 + r_{2i}q_2 + \dots + r_{i-1,i}q_{i-1}) \quad \text{and} \quad q_i = \frac{1}{r_{ii}} \tilde{q}_i \quad (9)$$

where  $a_i, \tilde{q}_i$  and  $q_i$  (with  $i = 2$  to  $n$ ) are  $n \times 1$  vectors

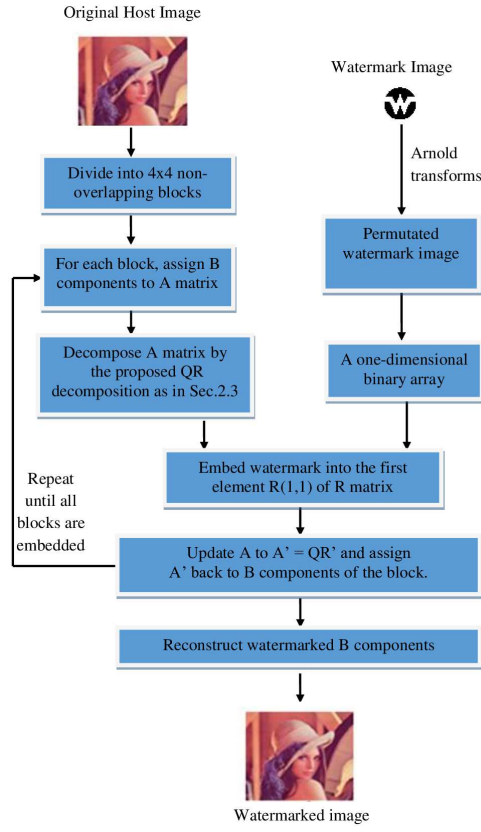
### 3 The proposed watermarking method

In this section, we describe a new watermarking scheme which is based on the improved  $QR$  decomposition and the formula of Sun in paper [5]. A image watermarking includes two stages, embedding and extracting, respectively.

#### 3.1 Watermark embedding scheme

In the embedding process, the host color image is divided into  $4 \times 4$  non-overlapping blocks at first. Then, the gray watermark image is permuted by Arnold transform and transformed to a binary sequence after that. Finally, the improved  $QR$  decomposition is performed on the image blocks in succession and the watermark is embedded into  $R$  matrix. The proposed watermark embedding scheme can be summarized as follows:

1. Divide the host color image  $H$  into  $4 \times 4$  non-overlapping blocks. In this image, each pixel is represented by three components ( $R, G, B$ ).
2. Transform the gray watermark image to a one-dimensional array  $w_i$  with  $i = 1, 2, \dots, N \times N$ .  $N \times N$  is the size of gray watermark image.
3. Perform  $QR$  decomposition on one block based on Sect. 2.3 as follows:
  - Assign  $B$  components of the block to a  $4 \times 4$  matrix (matrix  $A$ ).
  - Calculate  $A^T$  by transposing  $A$
  - Compute  $M = A^T A$
  - Find out  $R$  and  $Q$  matrices by solving a set of equations as represented in Eq. (10) and Eq. (9).
4. Embed watermark into the triangular matrix  $R$  based on the formula of Sun [5]:
  - Get the  $R(1, 1)$ , first element of  $R$  matrix
  - Calculate  $z = R(1, 1) \bmod q$  (with  $q$  is a positive integer)



**Fig. 1** The embedding stage.

– Case  $w_i = "0"$

$$R'(1,1) = \begin{cases} R(1,1) + \frac{q}{4} - z, & z \leq 3\frac{q}{4} \\ R(1,1) + 5\frac{q}{4} - z, & \text{elsewhere} \end{cases} \quad (10)$$

– Case  $w_i = "1"$

$$R'(1,1) = \begin{cases} R(1,1) - \frac{q}{4} - z, & z \leq \frac{q}{4} \\ R(1,1) + 3\frac{q}{4} - z, & \text{elsewhere} \end{cases} \quad (11)$$

Note that,  $q$  is also the strength of watermark embedding.

5. Update  $A$  by formula Eq. (1):  $A' = QR'$  and assign  $A'$  back to  $B$  components of the block.
6. Repeat steps 3–5 until all blocks are embedded watermark values. Finally, the watermarked  $B$  components are reconstructed to obtain the watermarked image  $H'$ .

The detail of steps for the embedding stage can be represented by Fig. 1.

### 3.2 Watermark extraction scheme

Since the watermark is only embedded into  $R$  matrix in the embedding process,  $QR$  decomposition is not needed in the watermark extraction procedure. The main purpose of this extraction scheme is finding out the first element  $R(1, 1)$  of  $R$  matrix. This improvement makes our paper is different from another one based on  $QR$  decomposition. Therefore, the watermark extraction steps are described as follows.

1. Divide the watermarked image  $H'$  into  $4 \times 4$  non-overlapping blocks. In this image, each pixel is represented by three components ( $R, G, B$ ).
2. Assign  $B$  components of the block to a  $4 \times 4$  matrix (matrix  $A$ ).
3. Obtain the first element  $R(1, 1)$  of  $R$  matrix as follows:  
 $R(1, 1) = \text{length of the first column vector of } A \text{ matrix [27].}$

$$R(1, 1) = \sqrt{A(1, 1)^2 + A(2, 1)^2 + A(3, 1)^2 + A(4, 1)^2} \quad (12)$$

4. Extract information of watermark based on algorithm of Sun [5]:
  - Calculate  $z = R(1, 1) \bmod q$
  - The watermark bit is extracted by using following equation.

$$w = \begin{cases} \text{"0"} & , \quad z \leq \frac{q}{2} \\ \text{"1"} & , \quad \text{elsewhere} \end{cases} \quad (13)$$

5. Repeat steps 2-4 until watermark values are extracted on all blocks. Finally, collect all extracted watermark values into an image and use inverse Arnold transform to get the final watermark.

The detail of the extracting stage can be represented by Fig. 2.

## 4 Experimental results

In general, the efficiency of image watermarking schemes is usually measured by their invisibility, robustness and computing time. For evaluating the invisibility capability, not only the peak signal-to-noise ratio ( $PSNR$ ) is used, but also the structural similarity index measurement ( $SSIM$ ) is utilized to measure the similarity between the original color image  $H$  and the watermarked image  $H'$  with size of  $M \times N$  in this paper.  $PSNR$  is employed as a measure for evaluating the quality of the watermarked image.  $PSNR$  is described by the following equation:

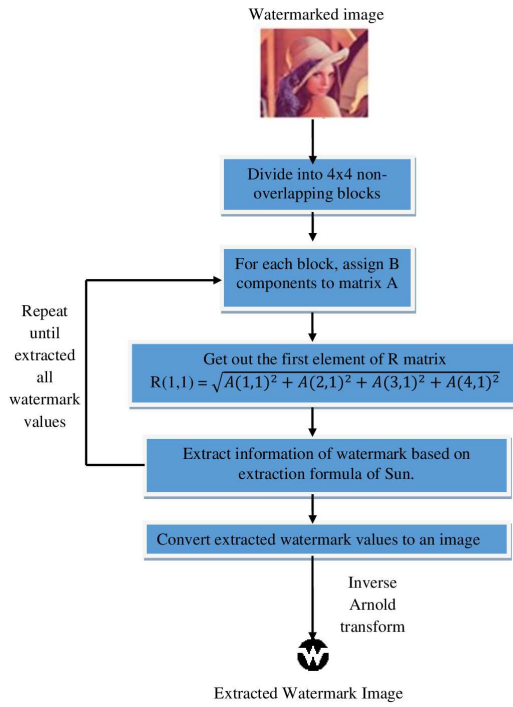
$$PSNR = 10 \log_{10} \frac{255^2}{MSE}, \quad (14)$$

where the mean square error ( $MSE$ ) between the original and watermarked image is defined as:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (H(i, j) - H'(i, j))^2 \quad (15)$$

Furthermore, the Normalized Correlation ( $NC$ ) coefficient is computed for evaluating robustness by using the original watermark  $W$  and the extracted watermark  $W'$ , which is denoted as follows:

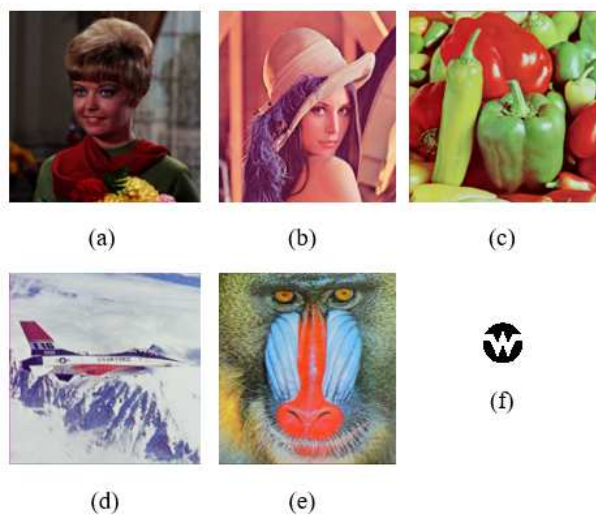
$$NC = \frac{\sum_{j=1}^4 \sum_{x=1}^m \sum_{y=1}^n (W(x, y, j) W'(x, y, j))}{\sqrt{\sum_{j=1}^4 \sum_{x=1}^m \sum_{y=1}^n (W(x, y, j))^2} \sqrt{\sum_{j=1}^4 \sum_{x=1}^m \sum_{y=1}^n (W'(x, y, j))^2}}, \quad (16)$$



**Fig. 2** The extracting stage.

where  $W(x, y, j)$  and  $W'(x, y, j)$  present the value of pixel  $(x, y)$  in component  $j$  of the original watermark and the extracted one,  $m \times n$  denote size of row and column of the watermark image, respectively.

In general, a larger *PSNR* or *SSIM* value denotes the watermarked image is very near to the original host image, which means that the watermarking method has better performance in term of invisibility. A higher *NC* value reveals that the extracted watermark is alike to the original watermark, which shows that the watermarking method is more robust. In our experiments, seven 24-bit color images with size of  $512 \times 512$  in the CVG-UGR image database [28] are selected as the host images, and a  $32 \times 32$  gray image as shown in Fig. 3 is used as original watermark. In order to select a suitable embedding parameter, the watermark is embedded into all host images with different embedding coefficients  $q$  (from 5 to 20 with the step length 1). Tab. 4 gives a part of *SSIM* of the watermarked images and the *NC* of the extracted watermark with the different embedding coefficients  $q$  ( $q = 5, 10, 15, 20$ , respectively). As shown in Tab. 4, each watermarked image has various *SSIM* and *NC* values with the same  $q$  and they are also different from other images. However, it is clear that when the coefficient  $q$  increases, the *SSIM* is smaller whereas the *NC* is bigger, and vice versa. This means that if the robustness of the watermark is better, then the invisibility of the watermarked image is worse when  $q$  goes up. Therefore, to balance between the invisibility and robustness, a value of  $q$  is set to 10 for evaluating the performance of the proposed method.



**Fig. 3** Original host images: (a) girl.bmp, (b) lena.bmp, (c) peppers.bmp, (d) avion, (e) baboon; Original watermark image: (f) logo.png.

| image   | q         | PSNR           | SSIM          | NC            |
|---------|-----------|----------------|---------------|---------------|
| Girl    | 5         | 65.6435        | 0.9997        | 0.9656        |
|         | <b>10</b> | <b>62.5665</b> | <b>0.9996</b> | <b>0.9999</b> |
|         | 15        | 53.8237        | 0.9986        | 0.9980        |
|         | 20        | 45.4135        | 0.9932        | 0.9748        |
| lena    | 5         | 65.3886        | 0.9998        | 0.9716        |
|         | <b>10</b> | <b>62.4570</b> | <b>0.9996</b> | <b>0.9981</b> |
|         | 15        | 60.6977        | 0.9995        | 0.9980        |
|         | 20        | 59.3773        | 0.9993        | 0.9961        |
| peppers | 5         | 60.6537        | 0.9998        | 0.9547        |
|         | <b>10</b> | <b>55.7789</b> | <b>0.9991</b> | <b>0.9787</b> |
|         | 15        | 54.2037        | 0.9986        | 0.9768        |
|         | 20        | 52.0914        | 0.9978        | 0.9778        |
| avion   | 5         | 65.6314        | 0.9998        | 0.9696        |
|         | <b>10</b> | <b>62.5527</b> | <b>0.9997</b> | <b>0.9980</b> |
|         | 15        | 60.8013        | 0.9995        | 0.9990        |
|         | 20        | 59.3868        | 0.9993        | 0.9961        |
| baboon  | 5         | 64.5499        | 0.9999        | 0.9809        |
|         | <b>10</b> | <b>62.0187</b> | <b>0.9999</b> | <b>0.9904</b> |
|         | 15        | 60.4548        | 0.9998        | 0.9942        |
|         | 20        | 59.2404        | 0.9997        | 0.9952        |

**Table 1** Values of *SSIM* and *NC* under different embedding coefficients

#### 4.1 Invisibility test

The quality of watermarked images is evaluated by *PSNR* and *SSIM* indexes. A comparison between the method of SunSVD [5], SunQR in Sect. 1.2, Su [29], Luo [8] and our proposal is performed to simulate for effectiveness of the proposed algorithm. The details of the result are represented in Fig. 4. In general, the watermarked image is more invisible when the value of *PSNR* is bigger or the value of *SSIM* is near to 1. Theoretically, extracted watermark



















































| Method                        | SunSVD [5]  | SunQR in Sect. 1.2  | Su [29]   | Luo [8]   | The proposed method  |
|-------------------------------|---|---|---|---|--|
| Watermarked image (PSNR/SSIM) | <br>58.0311/0.9989   | <br>55.7257/0.9964   | <br>43.4108/0.9893   | <br>53.9879/0.9978   | <br><b>62.5665/0.9996</b>   |
| Extracted watermark (NC)      | <br>0.9952           | <br>0.9451           | <br>0.8522           | <br>0.8977           | <br><b>0.9999</b>           |
| Watermarked image (PSNR/SSIM) | <br>57.9641/0.9992   | <br>55.1997/0.9996   | <br>44.3169/0.9931   | <br>47.9291/0.9927   | <br><b>62.4570/0.9996</b>   |
| Extracted watermark (NC)      | <br>0.9942           | <br>0.9980           | <br>0.9414           | <br>0.9828           | <br><b>0.9981</b>           |
| Watermarked image (PSNR/SSIM) | <br>45.2276/0.9892   | <br>55.7789/0.9991   | <br>45.9144/0.9959   | <br>47.6466/0.9953   | <br><b>57.9257/0.9996</b>   |
| Extracted watermark (NC)      | <br>0.9787          | <br>0.9729          | <br>0.9158          | <br>0.9527          | <br><b>0.9856</b>          |
| Watermarked image (PSNR/SSIM) | <br>57.9867/0.9991 | <br>48.5683/0.9997 | <br>34.9126/0.9650 | <br>52.9882/0.9985 | <br><b>62.5527/0.9997</b> |
| Extracted watermark (NC)      | <br>0.9923         | <br>0.9971         | <br>0.9609         | <br>0.9827         | <br><b>0.9980</b>         |
| Watermarked image (PSNR/SSIM) | <br>57.8236/0.9997 | <br>58.0014/0.9998 | <br>44.5487/0.9970 | <br>46.0408/0.9976 | <br><b>62.0187/0.9999</b> |
| Extracted watermark (NC)      | <br>0.9799         | <br>0.9900         | <br>0.9667         | <br>0.9035         | <br><b>0.9904</b>         |

Fig. 4 A comparison of the quality of the watermarked images between the methods.

| Value/Method | SunSVD [5] | SunQR in Sect. 1.2 | Su [29] | Luo [8] | The proposed method |
|--------------|------------|--------------------|---------|---------|---------------------|
| PSNR         | 55.4066    | 54.6548            | 42.6207 | 49.7185 | 61.5041             |
| SSIM         | 0.9972     | 0.9989             | 0.9881  | 0.9964  | 0.9997              |
| NC           | 0.9881     | 0.9806             | 0.9274  | 0.9439  | 0.9944              |

**Table 2** Average comparison between the different methods in term of PSNR, SSIM and NC values.

should be the same original watermark under no attacks. It means that  $NC$  value must be 1. However, it is not completely correct in our experimental tests. In fact,  $NC$  value are less than 1 because it depends on structure of host image as well as the embedding strength. In almost methods, the embedding strength is chosen to a suitable value in order to balance between the quality of watermarked image and extracted watermark.

Fig. 4 shows that  $PSNR/SSIM$  values of Su [29] and Luo [8] are lower than the other methods because the authors embedded watermark on two elements of  $L$  and  $U$  matrix, respectively. This causes a big change in two rows of the corresponding block after embedding. Therefore, the pixel values of watermarked image will not be close to the original image. As the result, the quality of watermarked image of these methods is worse. Meanwhile, the methods of SunSVD [5] and SunQR in Sect. 1.2 have higher  $PSNR/SSIM$  values. These methods use embedding formula of Sun which impacts on one element of the triangular matrix, so the embedded block only changes on one row instead of two rows as of Su [29] and Luo [8]. That is a reason why the formula of Sun is utilized in the proposed method. It can be seen clearly in Tab. 2 that the proposed method gives much higher  $PSNR/SSIM$  and NC values than others. This means that the invisibility of watermarked image is much better in the proposed scheme. In addition, the result table brings out that the proposed scheme not only overcomes on the quality of watermarked image, but also can effectively extract the embedded watermark.

Our results can be explained as follows. Theoretically,  $R$  is an upper triangular matrix with nonzero diagonal elements. According to Eq. (4) in Sect. 2.3, the first diagonal element  $r_{11}$  is always positive because pixel value is also positive. However, the other diagonal ones can be zero or negative which based on Eq. (5), Eq. (6) and Eq. (7) in Sect. 2.3. From Eq. (5) in Sect. 2.3, we have  $r_{22} = \sqrt{m_{22} - r_{12}^2}$ . If  $(m_{22} - r_{12}^2) = 0$  then  $r_{22} = 0$ . Thus,  $r_{23}$  and  $r_{22}$  are infinite. Otherwise, if  $(m_{22} - r_{12}^2) < 0$ ,  $r_{22}$  is infinite too. It is similar to  $r_{33}$  and  $r_{44}$ . This leads to infinite values of  $R$ ,  $Q$  and  $A'$  matrices (with  $A'$  is the matrix  $A$  after embedding watermark). This is a reason for reducing quality of watermarked image. Therefore, to solve this issue, we check the value below the square root before calculating  $r_{ii}$  ( $i = 2, 3, 4$ ). If this value is zero or negative,  $r_{ii}$  is set up to 1. This action not only gives out the better invisibility of the watermarked image, but also do not completely affect embedding process as well as extraction because the two processes only use the first diagonal element  $r_{11}$ .

#### 4.2 Execution time test

It is easy to see that execution time of watermarking image algorithms which are based on transform domain, depends on decomposition of a matrix a lot.  $SVD$  decomposition needs about  $11n^3$  flops for a  $n \times n$  matrix whereas time required to compute  $LU$  factorization is  $n^2$ . And  $QR$  decomposition is considered as an intermediate stage between  $SVD$  and  $LU$ .

In these experiments, a computer with Intel® Core™ i5-6200U CPU at 2.30GHz, 4.00GB RAM, 64-bit OS and Visual Studio v15 is used as the computing platform. The

| Method             | Embedding time | Extracting time | Total time |
|--------------------|----------------|-----------------|------------|
| SunSVD[5]          | 0.4850         | 0.0870          | 0.5720     |
| SunQR in Sect. 1.2 | 0.3718         | 0.0296          | 0.4014     |
| Su[29]             | 0.1774         | 0.0074          | 0.1848     |
| Luo[8]             | 2.4566         | 0.1730          | 2.6296     |
| Proposed method    | 0.3448         | 0.0060          | 0.3508     |

**Table 3** Average execution time of different methods (in second).

embedding time and extraction time of the proposed methods is 0.3448s and 0.006s, respectively. Tab. 3 shows a comparison of the execution time between different methods. According to Tab. 3, the total of execution time of the proposed method is bigger than the time of Su[29] which uses *LU* decomposition, but it is smaller than the others. In the scheme of Luo [8], the author proposed a combination between *DWT* and *SVD*. That is the reason why its running time is the biggest.

For embedding time, the proposed algorithm is proximate SunQR in Sect. 1.2 because the both uses *QR* factorization. For extraction time, the proposed method gives an effective result because it calculates length of the first column vector of *A* matrix to find out the first element  $R(1, 1)$  of *R* matrix instead of using *QR* decomposition as SunQR in Sect. 1.2. This demonstrates that the proposed scheme is extremely significant to improve speed of watermarking process, especially extraction time.

#### 4.3 Robustness test

For testing the robustness of the proposed method, nine operations are used to attack three watermarked images. And then, the extracted results from the attacked images are compared to the related works with different kinds of matrix decomposition. (SunSVD [5], SunQR in Sect. 1.2, Su[29], and Luo[8]). In [5], Sun used *SVD* decomposition, while SunQR in Sect. 1.2 used *QR* factorizing based on Gram-Schmit algorithm. Beside that, *LU* decomposition is developed by Su in [29], and the method of Lou[8] is a combination of *DWT* and *SVD* decomposition.

First of all, blurring and sharpening are two of the common image processes. The blurring technique is set up by two arguments like radius and sigma. The first value radius, is also important as it controls how big an area the operator should look at when spreading pixels. This value should typically be either '0' or at a minimum double that of the sigma. The second sigma value can be thought of as an approximation of just how much your want the image to blur in pixels. In the experiments, radius is fixed to '0' and sigma is designed to 0.2 and 0.5, respectively. The sharpening operation is a sort of an inverted blurring. In fact, both the operations work in just about the same way. Therefore, its arguments are similarly set to blur. Fig. 5 and Fig. 6 give the results of visual comparison and quantitative values. *NC* values from two these figures inform us about the superiority of the proposed method over the others.

Adding noise is also one of the common operations in the image processing. In this experiment, we select the Salt & Peppers noise and Gaussian noise as the attack noises. In the adding Salt & Peppers noise, the noise quantity is from 1% to 10% increasing with 1%, and Fig. 7 shows part of *NC* values and visual perception, respectively. Moreover, adding Gaussian white noise of mean 0 and variances from 0.001 to 0.005 increasing with 0.001 to

































| Image | Attack         | SunSVD [5]   | SunQR in Sect.1.2  | Su [29]  | Luo [8]  | The proposed method   |
|-------|----------------|--|--|--|--|---|
| avion | Blurring (0.2) | <br>0.9923  | <br>0.9971  | <br>0.9609  | <br>0.9827  | <br><b>0.9980</b>  |
|       | Blurring (0.5) | <br>0.9417  | <br>0.9674  | <br>0.9330  | <br>0.9694  | <br><b>0.9827</b>  |
| lena  | Blurring (0.2) | <br>0.9943  | <br>0.9980  | <br>0.9415  | <br>0.9827  | <br><b>0.9981</b>  |
|       | Blurring (0.5) | <br>0.9677  | <br>0.9808  | <br>0.9292  | <br>0.9808  | <br><b>0.9818</b>  |
| Girl  | Blurring (0.2) | <br>0.9952  | <br>0.9451  | <br>0.8522  | <br>0.8977  | <br><b>0.9999</b>  |
|       | Blurring (0.5) | <br>0.9933 | <br>0.9370 | <br>0.8436 | <br>0.8957 | <br><b>0.9999</b> |

Fig. 5 Extracted watermarks of the different methods under the blurring attacks.

process the watermarked images. Fig. 8 shows the  $NC$  values and visual perception results of extracted watermark after adding Gaussian white noise of mean 0 and different variances (0.001, 0.003, respectively). As can be seen from these figures, the proposed method has better robustness to against the process of adding noise.

In the filtering attack, the mean filter method is performed on the watermarked images. At first, the median filtering with different window sizes from  $2 \times 2$  to  $5 \times 5$  increasing with 1 is used to process the watermarked images. Fig. 9 shows the extracted watermarks and  $NC$  values with the filtering sizes of  $2 \times 2$  and  $3 \times 3$ , respectively. It is seen from this figure that when the window size is  $3 \times 3$ ,  $NC$  values are the best for our method. Under this filtering attack, our method also gives more effective results and the watermark image can be recognized for "avion" and "Girl" images.

For testing the cropping robustness, two cases are simulated to crop the three watermarked images. The first case is cropped in the upper left corner by 25%, while the second case is cropped in the upper half by 50%. Although  $NC$  value cannot be measure in this operation because it does not correctly reflect the quality of the extracted watermark. As the results displayed in Fig. 10, the visibility of the extracted watermarks of the proposed method is clearer than other methods. Obviously, the proposed method can be effectively against this cropping process.































| Image | Attack           | SunSVD [5]  | SunQR in Sect. 1.2  | Su [29]   | Luo [8]   | The proposed method   |
|-------|------------------|---|---|---|---|---|
| avion | Sharpening (0.2) | <br>0.9923 | <br>0.9971 | <br>0.9609 | <br>0.9827 | <br><b>0.99805</b> |
|       | Sharpening (0.5) | <br>0.9270 | <br>0.9192 | <br>0.9406 | <br>0.9239 | <br><b>0.9557</b>  |
| lena  | Sharpening (0.2) | <br>0.9943 | <br>0.9981 | <br>0.9415 | <br>0.9827 | <br><b>0.99806</b> |
|       | Sharpening (0.5) | <br>0.9407 | <br>0.9367 | <br>0.9274 | <br>0.9365 | <br><b>0.9743</b>  |
| Girl  | Sharpening (0.2) | <br>0.9952 | <br>0.9451 | <br>0.8522 | <br>0.8977 | <br><b>0.9999</b>  |
|       | Sharpening (0.5) | <br>0.9887 | <br>0.9340 | <br>0.8421 | <br>0.8968 | <br><b>0.9990</b>  |

Fig. 6 Extracted watermarks of the different methods under the sharpening operations.

An other type of the well-known image operations is geometry attacks, which mainly include rotation and scaling. There are two rotation experiments to show the robustness in Fig. 11. One involves rotating the watermarked image to the right by 5 degrees. The other involves rotating the watermarked image to the right by 10 degrees. The images are first rotated a certain number of degrees clockwise and then are rotated the same number of degrees counterclockwise. Fig. 12 shows the quantitative results and visual perception results for the case of scaling, respectively. In this experiment, two scaling operations of 200% and 50% are used to deteriorate the watermarked image. From data in Fig. 11 and Fig. 12, although the proposed method gives less robustness than the methods of SunSVD [5] and Luo[8], it is more effective than the schemes of SunQR in Sect. 1.2 and Su[29].

Finally, *JPEG* compression is also known as a image process. In this experiment, the watermarked images are compressed by *JPEG* compression with the window size is  $8 \times 8$  and  $16 \times$ , respectively. As the results shown in Fig. 13, although the proposed method is not the best method under *JPEG* compression, it has higher *NC* values than the methods of SunQR, Su[29] and Luo[8] for "avion" image. It is also better than the other methods for "lena" and "Girl" images, excepting the method of Lou[8].

To sum up, the average *NC* values in the above mentioned figures, which obtained by the methods (SunSVD [5], SunQR in Sect. 1.2, Su[29] and Luo[8]) and the proposed one, are 0.8711, 0.8406, 0.7925, 0.8283 and 0.8905, respectively. In addition, there are 33/51 *NC* values that the proposed method is the highest among compared methods. These data show














































| Image | Attack                     | SunSVD [5]  | SunQR in Sect.1.2   | Su [29]   | Luo [8]   | The proposed method  |
|-------|----------------------------|---|---|---|---|--|
| avion | Salt & Peppers noise (2%)  | <br>0.9676   | <br>0.9875   | <br>0.9569   | <br>0.9663   | <br><b>0.9904</b>   |
|       | Salt & Peppers noise (5%)  | <br>0.9281   | <br>0.9715   | <br>0.9513   | <br>0.9410   | <br><b>0.9724</b>   |
|       | Salt & Peppers noise (10%) | <br>0.8622   | <br>0.9376   | <br>0.9390   | <br>0.9134   | <br><b>0.9448</b>   |
| lena  | Salt & Peppers noise (2%)  | <br>0.9714   | <br>0.9923   | <br>0.9396   | <br>0.9751   | <br><b>0.9924</b>   |
|       | Salt & Peppers noise (5%)  | <br>0.9486  | <br>0.9784  | <br>0.9350  | <br>0.9495  | <br><b>0.9819</b>  |
|       | Salt & Peppers noise (10%) | <br>0.8993 | <br>0.9676 | <br>0.9245 | <br>0.9342 | <br><b>0.9704</b> |
| Girl  | Salt & Peppers noise (2%)  | <br>0.9583 | <br>0.9391 | <br>0.8470 | <br>0.8882 | <br><b>0.9942</b> |
|       | Salt & Peppers noise (5%)  | <br>0.8987 | <br>0.9330 | <br>0.8448 | <br>0.8689 | <br><b>0.9866</b> |
|       | Salt & Peppers noise (10%) | <br>0.8250 | <br>0.9221 | <br>0.8344 | <br>0.8353 | <br><b>0.9791</b> |

Fig. 7 Extracted watermarks of the different methods under the Salt & Peppers noise adding.

| Image | Attack                   | SunSVD [5] | SunQR in Sect. 1.2 | Su [29] | Luo [8] | The proposed method |
|-------|--------------------------|------------|--------------------|---------|---------|---------------------|
| avion | Gaussian noise (0,0.001) | 0.9611     | 0.9704             | 0.9310  | 0.9788  | <b>0.9828</b>       |
|       | Gaussian noise (0,0.003) | 0.8696     | 0.8337             | 0.8454  | 0.8206  | <b>0.9375</b>       |
| lena  | Gaussian noise (0,0.001) | 0.9495     | 0.9762             | 0.9284  | 0.5127  | <b>0.9773</b>       |
|       | Gaussian noise (0,0.003) | 0.8795     | 0.8166             | 0.7930  | 0.5061  | <b>0.8853</b>       |
| Girl  | Gaussian noise (0,0.001) | 0.9502     | 0.9261             | 0.7919  | 0.8824  | <b>0.9799</b>       |
|       | Gaussian noise (0,0.003) | 0.8372     | 0.7878             | 0.7237  | 0.8241  | <b>0.8834</b>       |

Fig. 8 Extracted watermarks of the different methods under the Gaussian noise adding.

that the proposed method has better robust than other methods in almost cases. Furthermore, the proposed method outperforms SunQR in all cases although both these methods use QR decomposition. The reason for this is because the proposed method has improvement to find out elements of  $Q$  and  $R$  matrices instead of using Gram-Schmidt algorithm as SunQR. By experimental results, we also can see that the schemes which use  $SVD$  decomposition (SunSVD [5] and Luo[8]) is more robust than other methods (SunQR in Sect. 1.2, Su[29] and the proposed one) under geometry attacks (rotation, scaling). However, the method of Luo[8] is less effective under mean filter and cropping operations.

## 5 Conclusion

In this paper, a novel image watermarking scheme, which is based on  $QR$  decomposition, is presented. In embedding stage, the color host image is divided into  $4 \times 4$  blocks at first. For each block,  $QR$  decomposition is applied on the  $B$  channel where calculating  $Q$  and  $R$  matrices is executed in succession. First, the elements of  $R$  matrix are found by performing a





























































| Image | Attack            | SunSVD [5]   | SunQR in Sect. 1.2   | Su [29]  | Luo [8]  | The proposed method   |
|-------|-------------------|--|--|--|--|---|
| avion | Mean Filter (2x2) | <br>0.8005  | <br>0.8252  | <br>0.5587  | <br>0.6872  | <br><b>0.8261</b>  |
|       | Mean Filter (3x3) | <br>0.7999  | <br>0.8303  | <br>0.4246  | <br>0.4754  | <br><b>0.8314</b>  |
| lena  | Mean Filter (2x2) | <br>0.6324  | <br>0.6232  | <br>0.6366  | <br>0.5062  | <br><b>0.6512</b>  |
|       | Mean Filter (3x3) | <br>0.7693  | <br>0.7687  | <br>0.4971  | <br>0.4930  | <br><b>0.7946</b>  |
| Girl  | Mean Filter (2x2) | <br>0.7580  | <br>0.7934  | <br>0.6954  | <br>0.7032  | <br><b>0.8589</b>  |
|       | Mean Filter (3x3) | <br>0.9211 | <br>0.8873 | <br>0.6066 | <br>0.5187 | <br><b>0.9658</b> |

Fig. 9 Extracted watermarks of the different methods under the Mean Filter attacks.

set of linear equations as Eq. (10). Second,  $Q$  matrix is computed column by column which based on  $A$  and  $R$  matrices. Third, the watermark information is embedded into the first element  $R(1,1)$  of  $R$  matrix by using the Eq. (11). Finally, we have the watermarked image after taking a reverse  $QR$  decomposition to update pixel values. In extracting stage, we get  $R(1,1)$  by only one operation Eq. (12) without  $QR$  factorization as the previous methods. After that, the binary values of the watermark are extracted via Eq. (13). As the result, the image of watermark is reconstructed from the received information. The tests are experimented on five color images and one gray-scale watermark image with  $PSNR/SSIM$  and  $NC$  indexes to evaluate effectiveness of the proposed method. The results of comparison in Sect. 4 show that our scheme overcomes the others in term of the quality of the watermarked image. In addition, because we do not need to use  $QR$  decomposition in the extracting stage, the execution time is significantly improved. And our proposal is also more robust than the methods of SunSVD [5], SunQR in Sect. 1.2, Su [29] and Luo [8] under attacks such as Salt&Peppers noise, Gaussian noise, blurring, sharpening, cropping and Mean filter. In future, a combination of  $DWT$  and  $QR$  can be developed to improve the robustness of the watermark under geometry attacks.

| Image | Attack         | SunSVD [5]  | SunQR in Sect.1.2   | Su [29]   | Luo [8]   | The proposed method   |
|-------|----------------|---|---|---|---|---|
| avion | Cropping (25%) |  |  |  |  |  |
|       | Cropping (50%) |  |  |  |  |  |
| lena  | Cropping (25%) |  |  |  |  |  |
|       | Cropping (50%) |  |  |  |  |  |
| Girl  | Cropping (25%) |  |  |  |  |  |
|       | Cropping (50%) |  |  |  |  |  |

**Fig. 10** Extracted watermarks of the different methods under the Cropping operations.

**Acknowledgements** This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2019.12.

## References

1. Su, Q., Chen, B. "Robust color image watermarking technique in the spatial domain". *Soft Comput.*, 22, pp. 91–106 (2018). <https://doi.org/10.1007/s00500-017-2489-7>
2. Su Q, Wang G, Jia S, Zhang X, Liu Q and Liu LX (2015), "Embedding color image watermark in color image based on two-level DCT", *Signal Image Video Process*, 9(5), pp. 991–1007
3. L.-Y. Hsu and H.-T. Hu (2017), "Robust blind image watermarking using crisscross inter-block prediction in the DCT domain", *Journal of Visual Communication and Image Representation*, vol.46, pp. 33-47.
4. Kaiser J. Giri, Mushtaq Ahmad Peer and P. Nagabhushan (2015), "A Robust Color Image Watermarking Scheme Using Discrete Wavelet Transformation", *I.J. Image, Graphics and Signal Processing*, pp. 47-52.
5. R. Sun, H. Sun, T. Yao, "A SVD and quantization based semi-fragile watermarking technique for image authenticatio", *Proc. Internat. Conf. Signal Process.* 2, pp. 1952-1955, 2002































| Image | Attack         | SunSVD [5]  | SunQR in Sect.1.2  | Su [29]  | Luo [8]  | The proposed method  |
|-------|----------------|---|--|--|--|--|
| avion | Rotation (5°)  | <br>0.7614         | <br>0.6668  | <br>0.7118  | <br><b>0.8107</b> | <br>0.6994  |
|       | Rotation (10°) | <br>0.7028         | <br>0.6924  | <br>0.6791  | <br><b>0.7565</b> | <br>0.6644  |
| lena  | Rotation (5°)  | <br>0.7781         | <br>0.6347  | <br>0.7480  | <br><b>0.8181</b> | <br>0.6360  |
|       | Rotation (10°) | <br>0.7524         | <br>0.6039  | <br>0.7138  | <br><b>0.7661</b> | <br>0.6046  |
| Girl  | Rotation (5°)  | <br><b>0.8375</b>  | <br>0.6991  | <br>0.7284  | <br>0.7616        | <br>0.7825  |
|       | Rotation (10°) | <br><b>0.7897</b> | <br>0.6670 | <br>0.7032 | <br>0.7211       | <br>0.7504 |

Fig. 11 Extracted watermarks of the different methods under the Rotation attacks.

6. D. Vaishnavi and T. S. Subashini (2015), "Robust and Invisible Image Watermarking in RGB Color Space Using SVD", *Procedia Computer Science*, 46, pp. 1770-1777.
7. C.C Lai, "An improved SVD-based watermarking scheme using human visual characteristic", *Optics Communicaion*, 284, pp. 938-944, 2011.
8. Luo, A., Gong, L., Zhou, N. et al. "Adaptive and blind watermarking scheme based on optimal SVD blocks selection", *Multimed Tools Appl*, 79, 243–261 (2020). <https://doi.org/10.1007/s11042-019-08074-2>
9. F. Liu, H. Yang and Q. Su (2017), "Color image blind watermarking algorithm based on Schur decomposition", *Application Research of Computers*, 34, pp. 3085-3093.
10. Su, Q., Zhang, X. & Wang, G., "An improved watermarking algorithm for color image using Schur decomposition", *Soft Comput*, 24, pp. 445–460 (2020). <https://doi.org/10.1007/s00500-019-03924-5>
11. Su, Q., Niu, Y., Zou, H. et al. "A blind double color image watermarking algorithm based on QR decomposition". *Multimed Tools Appl*, 72, pp. 987–1009 (2013). <https://doi.org/10.1007/s11042-013-1653-z>
12. Naderahmadian Y, Hosseini-Khayat S (2014), "Fast and robust watermarking in still images based on QR decomposition". *Multimed Tools Appl*, 72(3) pp.2597–2618

| Image | Attack        | SunSVD [5] | SunQR in Sect.1.2 | Su [29]    | Luo [8]    | The proposed method |
|-------|---------------|------------|-------------------|------------|------------|---------------------|
| avion | Scaling (1/2) | <br>0.7615 | <br>0.6222        | <br>0.5939 | <br>0.7514 | <br>0.6299          |
|       | Scaling (2)   | <br>0.8390 | <br>0.7976        | <br>0.8099 | <br>0.9351 | <br>0.7995          |
| lena  | Scaling (1/2) | <br>0.8306 | <br>0.5244        | <br>0.6717 | <br>0.7508 | <br>0.5257          |
|       | Scaling (2)   | <br>0.9429 | <br>0.8038        | <br>0.8627 | <br>0.9582 | <br>0.8073          |
| Girl  | Scaling (1/2) | <br>0.8826 | <br>0.6211        | <br>0.6702 | <br>0.6951 | <br>0.7018          |
|       | Scaling (2)   | <br>0.9832 | <br>0.8967        | <br>0.7851 | <br>0.8937 | <br>0.9609          |

Fig. 12 Extracted watermarks of the different methods under the Scaling attacks.

13. Su, Q., Niu, Y., Wang, G., Jia, S., & Yue, J. (2014). "Color image blind watermarking scheme based on QR decomposition". *Signal Processing*, 94, 219-235.
14. Qingtang Su, Gang Wang, Xiaofeng Zhang, Gaohuan Lv and Beijing Chen (2017), "An improved color image watermarking algorithm based on QR decomposition", *Multimed Tools Appl*, 76, pp. 707-729. <https://doi.org/10.1007/s11042-015-3071-x>
15. Sima Arasteh, Mojtaba Mahdavi, Pegah Nikbakht Bideh, Samira Hosseini and AmirAhmad Chapnevis (2018), "Security Analysis of Two Key based Watermarking Schemes Based on QR Decomposition", *26th Iranian Conference on Electrical Engineering (ICEE2018)*, pp. 1499-1504.
16. Qingtang Su, Yonghui Liu, Decheng Liu, Zihan Yuan and Hongye Ning (2019), "A new watermarking scheme for colour image using QR decomposition and ternary coding", *Multimedia Tools and Applications*, 78(7), pp. 8113-8132.
17. Ranjeet Kumar Singh, Dillip Kumar Shaw and Jayakrushna Sahoo (2017), "A secure and robust block based DWT-SVD image watermarking approach", *Journal of Information and Optimization Sciences*, 38(6), pp. 911-925.



| Image | Attack       | SunSVD [5] | SunQR in Sect.1.2 | Su [29]    | Luo [8]    | The proposed method |
|-------|--------------|------------|-------------------|------------|------------|---------------------|
| avion | JPEG (8x8)   | <br>0.9556 | <br>0.8559        | <br>0.6299 | <br>0.8346 | <br>0.8680          |
|       | JPEG (16x16) | <br>0.9702 | <br>0.8646        | <br>0.6779 | <br>0.8719 | <br>0.8693          |
| lena  | JPEG (8x8)   | <br>0.6404 | <br>0.6348        | <br>0.7255 | <br>0.8812 | <br>0.6425          |
|       | JPEG (16x16) | <br>0.6278 | <br>0.6433        | <br>0.7596 | <br>0.9140 | <br>0.6470          |
| Girl  | JPEG (8x8)   | <br>0.6295 | <br>0.7268        | <br>0.6970 | <br>0.7810 | <br>0.7739          |
|       | JPEG (16x16) | <br>0.6314 | <br>0.6944        | <br>0.7202 | <br>0.8343 | <br>0.7583          |

Fig. 13 Extracted watermarks of the different methods under the JPEG compression.

18. Yadav B., Kumar A., Kumar Y. (2018), "A Robust Digital Image Watermarking Algorithm Using DWT and SVD". In: Pant M., Ray K., Sharma T., Rawat S., Bandyopadhyay A. (eds) *Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing*, vol 583. Springer, Singapore. <https://doi.org/10.1007/978-981-10-5687-1-3>
19. Roy, S., Pal, A.K. "A Hybrid Domain Color Image Watermarking Based on DWT-SVD". *Iran J Sci Technol Trans Electr Eng*, 43, 201–217 (2019). <https://doi.org/10.1007/s40998-018-0109-x>
20. Ernawan, F., Kabir, M.N. "A block-based RDWT-SVD image watermarking method using human visual system characteristics". *Vis Comput*, 36, 19–37 (2020). <https://doi.org/10.1007/s00371-018-1567-x>
21. Laxmanika, Singh A.K., Singh P.K. (2020), "A Robust Image Watermarking Through Bi-empirical Mode Decomposition and Discrete Wavelet Domain". In: Singh P., Panigrahi B., Suryadevara N., Sharma S., Singh A. (eds) *Proceedings of ICETIT 2019. Lecture Notes in Electrical Engineering*, vol 605. Springer, Cham

- 
22. Li, J., Lin, Q., Yu, C. et al. "A QDCT- and SVD-based color image watermarking scheme using an optimized encrypted binary computer-generated hologram". *Soft Comput*, 22, 2018, pp.47–65. <https://doi.org/10.1007/s00500-016-2320-x>
  23. Abdulrahman, A.K., Ozturk, S., "A novel hybrid DCT and DWT based robust watermarking algorithm for color images". *Multimed Tools Appl*, 78, 2019, pp. 17027–17049. <https://doi.org/10.1007/s11042-018-7085-z>
  24. Shaoli Jia, Qingpo Zhou and Hong Zhou (2017), "A Novel Color Image Watermarking Scheme Based on DWT and QR Decomposition", *Journal of Applied Science and Engineering*, 20(2), 2017, pp. 193-200.
  25. Kamred Udham Singh, Vineet Kumar Singh and Achintya Singhal (2018), "Color Image Watermarking Scheme Based on QR Factorization and DWT with Compatibility Analysis on Different Wavelet Filters", *Jour of Adv Research in Dynamical & Control Systems*, 10(6), pp. 1796-1811.
  26. Dongyan Wang, Fanfan Yang and Heng Zhang (2016), "Blind Color Image Watermarking Based on DWT and LU Decomposition", *Journal of Information Processing System*, 12(4), pp. 765-778.
  27. L. Vandenberghe, "6.QR factorization", *ECE133A*, pp.6-1-6-39, 2018.
  28. University of Granada. "Computer Vision Group". *CVG-UGR Image Database*. [2012-10-22]. <http://decsai.ugr.es/cvadbimagesc512.php>
  29. Qingtang Su, Gang Wang, Xiaofeng Zhang, Gaohuan Lv and Beijing Chen (2018), "A new algorithm of blind color image watermarking based on LU decomposition", *Multidimensional Systems and Signal Processing*, 29(3), pp. 1055-1074.

# Figures

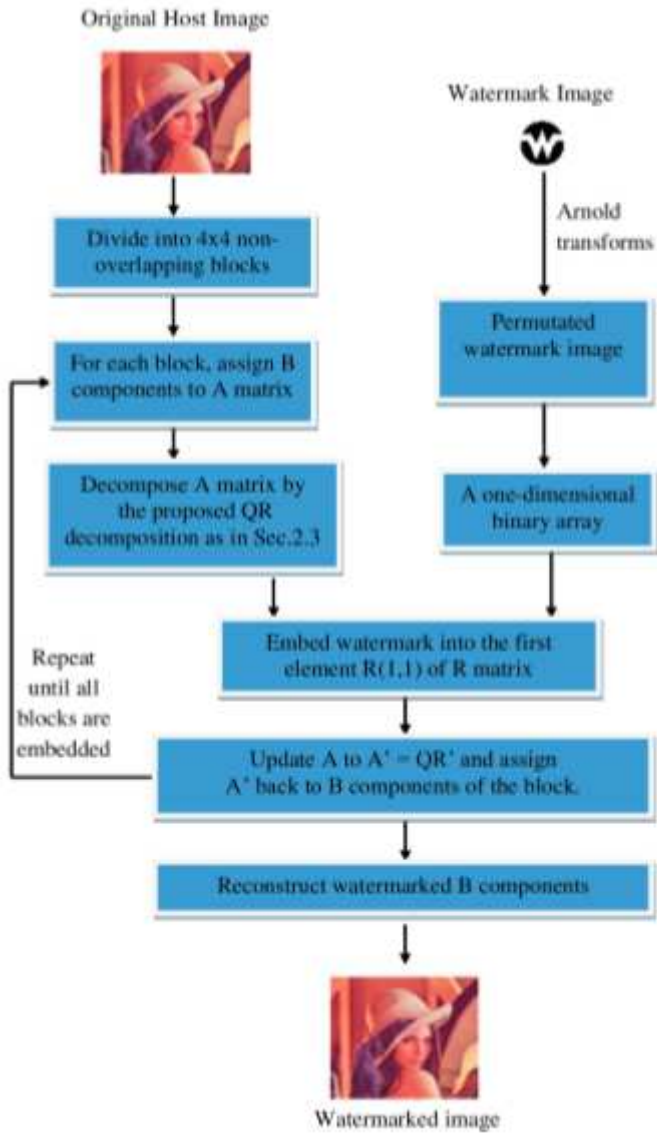


Figure 1

The embedding stage.

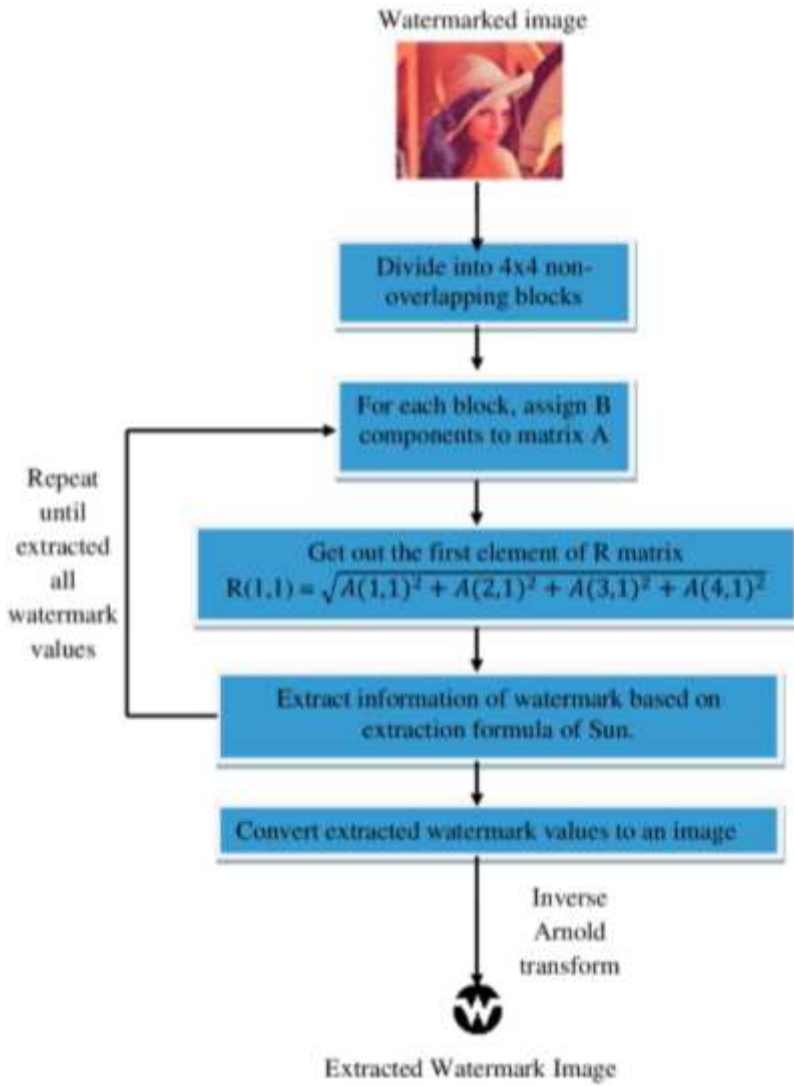


Figure 2

The extracting stage.



(a)



(b)



(c)



(d)



(e)



(f)

**Figure 3**

Original host images: (a) girl.bmp, (b) lena.bmp, (c) peppers.bmp, (d) avion, (e) baboon; Original watermark image: (f) logo.png.



















































| Method                        | SunSVD [5]  | SunQR in Sect. 1.2  | Su [29]   | Luo [8]   | The proposed method   |
|-------------------------------|---|---|---|---|---|
| Watermarked image (PSNR/SSIM) | <br>58.0311/0.9989   | <br>55.7257/0.9964   | <br>43.4108/0.9893   | <br>53.9879/0.9978   | <br><b>62.5665/0.9996</b>   |
| Extracted watermark (NC)      | <br>0.9952           | <br>0.9451           | <br>0.8522           | <br>0.8977           | <br><b>0.9999</b>           |
| Watermarked image (PSNR/SSIM) | <br>57.9641/0.9992   | <br>55.1997/0.9996   | <br>44.3169/0.9931   | <br>47.9291/0.9927   | <br><b>62.4570/0.9996</b>   |
| Extracted watermark (NC)      | <br>0.9942           | <br>0.9980           | <br>0.9414           | <br>0.9828           | <br><b>0.9981</b>           |
| Watermarked image (PSNR/SSIM) | <br>45.2276/0.9892   | <br>55.7789/0.9991   | <br>45.9144/0.9959   | <br>47.6466/0.9953   | <br><b>57.9257/0.9996</b>   |
| Extracted watermark (NC)      | <br>0.9787           | <br>0.9729           | <br>0.9158           | <br>0.9527           | <br><b>0.9856</b>           |
| Watermarked image (PSNR/SSIM) | <br>57.9867/0.9991 | <br>48.5683/0.9997 | <br>34.9126/0.9650 | <br>52.9882/0.9985 | <br><b>62.5527/0.9997</b> |
| Extracted watermark (NC)      | <br>0.9923         | <br>0.9971         | <br>0.9609         | <br>0.9827         | <br><b>0.9980</b>         |
| Watermarked image (PSNR/SSIM) | <br>57.8236/0.9997 | <br>58.0014/0.9998 | <br>44.5487/0.9970 | <br>46.0408/0.9976 | <br><b>62.0187/0.9999</b> |
| Extracted watermark (NC)      | <br>0.9799         | <br>0.9900         | <br>0.9667         | <br>0.9035         | <br><b>0.9904</b>         |

Figure 4

A comparison of the quality of the watermarked images between the methods.































| Image | Attack         | SunSVD [5]   | SunQR in Sect.1.2  | Su [29]   | Luo [8]  | The proposed method   |
|-------|----------------|--|--|---|--|---|
| avion | Blurring (0.2) | <br>0.9923  | <br>0.9971  | <br>0.9609  | <br>0.9827  | <br><b>0.9980</b>  |
|       | Blurring (0.5) | <br>0.9417  | <br>0.9674  | <br>0.9330  | <br>0.9694  | <br><b>0.9827</b>  |
| lena  | Blurring (0.2) | <br>0.9943  | <br>0.9980  | <br>0.9415  | <br>0.9827  | <br><b>0.9981</b>  |
|       | Blurring (0.5) | <br>0.9677  | <br>0.9808  | <br>0.9292  | <br>0.9808  | <br><b>0.9818</b>  |
| Girl  | Blurring (0.2) | <br>0.9952  | <br>0.9451  | <br>0.8522  | <br>0.8977  | <br><b>0.9999</b>  |
|       | Blurring (0.5) | <br>0.9933 | <br>0.9370 | <br>0.8436 | <br>0.8957 | <br><b>0.9999</b> |

Figure 5

Extracted watermarks of the different methods under the blurring attacks.

































| Image | Attack           | SunSVD [5]  | SunQR in Sect.1.2   | Su [29]  | Luo [8]   | The proposed method   |
|-------|------------------|---|---|--|---|---|
| avion | Sharpening (0.2) | <br>0.9923 | <br>0.9971 | <br>0.9609 | <br>0.9827 | <br><b>0.99805</b> |
|       | Sharpening (0.5) | <br>0.9270 | <br>0.9192 | <br>0.9406 | <br>0.9239 | <br><b>0.9557</b>  |
| lena  | Sharpening (0.2) | <br>0.9943 | <br>0.9981 | <br>0.9415 | <br>0.9827 | <br><b>0.99806</b> |
|       | Sharpening (0.5) | <br>0.9407 | <br>0.9367 | <br>0.9274 | <br>0.9365 | <br><b>0.9743</b>  |
| Girl  | Sharpening (0.2) | <br>0.9952 | <br>0.9451 | <br>0.8522 | <br>0.8977 | <br><b>0.9999</b>  |
|       | Sharpening (0.5) | <br>0.9887 | <br>0.9340 | <br>0.8421 | <br>0.8968 | <br><b>0.9990</b>  |

Figure 6

Extracted watermarks of the different methods under the sharpening operations.
















































| Image | Attack                     | SunSVD [5]   | SunQR in Sect.1.2  | Su [29]  | Luo [8]  | The proposed method   |
|-------|----------------------------|--|--|--|--|---|
| avion | Salt & Peppers noise (2%)  |  0.9676   |  0.9875   |  0.9569   |  0.9663   |  <b>0.9904</b>   |
|       | Salt & Peppers noise (5%)  |  0.9281   |  0.9715   |  0.9513   |  0.9410   |  <b>0.9724</b>   |
|       | Salt & Peppers noise (10%) |  0.8622   |  0.9376   |  0.9390   |  0.9134   |  <b>0.9448</b>   |
| lena  | Salt & Peppers noise (2%)  |  0.9714   |  0.9923   |  0.9396   |  0.9751   |  <b>0.9924</b>   |
|       | Salt & Peppers noise (5%)  |  0.9486   |  0.9784   |  0.9350   |  0.9495   |  <b>0.9819</b>   |
|       | Salt & Peppers noise (10%) |  0.8993  |  0.9676  |  0.9245  |  0.9342  |  <b>0.9704</b>  |
| Girl  | Salt & Peppers noise (2%)  |  0.9583 |  0.9391 |  0.8470 |  0.8882 |  <b>0.9942</b> |
|       | Salt & Peppers noise (5%)  |  0.8987 |  0.9330 |  0.8448 |  0.8689 |  <b>0.9866</b> |
|       | Salt & Peppers noise (10%) |  0.8250 |  0.9221 |  0.8344 |  0.8353 |  <b>0.9791</b> |

Figure 7

Extracted watermarks of the different methods under the Salt & Peppers noise adding.































| Image | Attack                   | SunSVD [5]  | SunQR in Sect. 1.2  | Su [29]   | Luo [8]   | The proposed method  |
|-------|--------------------------|---|---|---|---|--|
| avion | Gaussian noise (0,0.001) | <br>0.9611   | <br>0.9704   | <br>0.9310   | <br>0.9788   | <br><b>0.9828</b>   |
|       | Gaussian noise (0,0.003) | <br>0.8696   | <br>0.8337   | <br>0.8454   | <br>0.8206   | <br><b>0.9375</b>   |
| lena  | Gaussian noise (0,0.001) | <br>0.9495   | <br>0.9762   | <br>0.9284   | <br>0.5127   | <br><b>0.9773</b>   |
|       | Gaussian noise (0,0.003) | <br>0.8795   | <br>0.8166   | <br>0.7930   | <br>0.5061   | <br><b>0.8853</b>   |
| Girl  | Gaussian noise (0,0.001) | <br>0.9502   | <br>0.9261   | <br>0.7919   | <br>0.8824   | <br><b>0.9799</b>   |
|       | Gaussian noise (0,0.003) | <br>0.8372 | <br>0.7878 | <br>0.7237 | <br>0.8241 | <br><b>0.8834</b> |

Figure 8

Extracted watermarks of the different methods under the Gaussian noise adding.

| Image | Attack            | SunSVD [5] | SunQR in Sect.1.2 | Su [29] | Luo [8] | The proposed method |
|-------|-------------------|------------|-------------------|---------|---------|---------------------|
| avion | Mean Filter (2x2) | 0.8005     | 0.8252            | 0.5587  | 0.6872  | <b>0.8261</b>       |
|       | Mean Filter (3x3) | 0.7999     | 0.8303            | 0.4246  | 0.4754  | <b>0.8314</b>       |
| lena  | Mean Filter (2x2) | 0.6324     | 0.6232            | 0.6366  | 0.5062  | <b>0.6512</b>       |
|       | Mean Filter (3x3) | 0.7693     | 0.7687            | 0.4971  | 0.4930  | <b>0.7946</b>       |
| Girl  | Mean Filter (2x2) | 0.7580     | 0.7934            | 0.6954  | 0.7032  | <b>0.8589</b>       |
|       | Mean Filter (3x3) | 0.9211     | 0.8873            | 0.6066  | 0.5187  | <b>0.9658</b>       |

Figure 9

Extracted watermarks of the different methods under the Mean Filter attacks.































| Image | Attack         | SunSVD [5]  | SunQR in Sect. 1.2  | Su [29]   | Luo [8]   | The proposed method   |
|-------|----------------|---|---|---|---|---|
| avion | Cropping (25%) |  |  |  |  |  |
|       | Cropping (50%) |  |  |  |  |  |
| lena  | Cropping (25%) |  |  |  |  |  |
|       | Cropping (50%) |  |  |  |  |  |
| Girl  | Cropping (25%) |  |  |  |  |  |
|       | Cropping (50%) |  |  |  |  |  |

Figure 10

Extracted watermarks of the different methods under the Cropping operations.































| Image | Attack         | SunSVD [5]  | SunQR in Sect. 1.2   | Su [29]  | Luo [8]  | The proposed method  |
|-------|----------------|---|--|--|--|--|
| avion | Rotation (5°)  | <br>0.7614         | <br>0.6668  | <br>0.7118  | <br><b>0.8107</b> | <br>0.6994  |
|       | Rotation (10°) | <br>0.7028         | <br>0.6924  | <br>0.6791  | <br><b>0.7565</b> | <br>0.6644  |
| lena  | Rotation (5°)  | <br>0.7781         | <br>0.6347  | <br>0.7480  | <br><b>0.8181</b> | <br>0.6360  |
|       | Rotation (10°) | <br>0.7524         | <br>0.6039  | <br>0.7138  | <br><b>0.7661</b> | <br>0.6046  |
| Girl  | Rotation (5°)  | <br><b>0.8375</b>  | <br>0.6991  | <br>0.7284  | <br>0.7616        | <br>0.7825  |
|       | Rotation (10°) | <br><b>0.7897</b> | <br>0.6670 | <br>0.7032 | <br>0.7211       | <br>0.7504 |

Figure 11

Extracted watermarks of the different methods under the Rotation attacks.































| Image | Attack        | SunSVD [5]   | SunQR in Sect. 1.2   | Su [29]  | Luo [8]  | The proposed method  |
|-------|---------------|--|--|--|--|--|
| avion | Scaling (1/2) | <br>0.7615  | <br>0.6222  | <br>0.5939  | <br>0.7514  | <br>0.6299  |
|       | Scaling (2)   | <br>0.8390  | <br>0.7976  | <br>0.8099  | <br>0.9351  | <br>0.7995  |
| lena  | Scaling (1/2) | <br>0.8306  | <br>0.5244  | <br>0.6717  | <br>0.7508  | <br>0.5257  |
|       | Scaling (2)   | <br>0.9429  | <br>0.8038  | <br>0.8627  | <br>0.9582  | <br>0.8073  |
| Girl  | Scaling (1/2) | <br>0.8826  | <br>0.6211  | <br>0.6702  | <br>0.6951  | <br>0.7018  |
|       | Scaling (2)   | <br>0.9832 | <br>0.8967 | <br>0.7851 | <br>0.8937 | <br>0.9609 |

Figure 12

Extracted watermarks of the different methods under the Scaling attacks.



| Image | Attack       | SunSVD [5] | SunQR in Sect.1.2 | Su [29]    | Luo [8]    | The proposed method |
|-------|--------------|------------|-------------------|------------|------------|---------------------|
| avion | JPEG (8x8)   | <br>0.9556 | <br>0.8559        | <br>0.6299 | <br>0.8346 | <br>0.8680          |
|       | JPEG (16x16) | <br>0.9702 | <br>0.8646        | <br>0.6779 | <br>0.8719 | <br>0.8693          |
| lena  | JPEG (8x8)   | <br>0.6404 | <br>0.6348        | <br>0.7255 | <br>0.8812 | <br>0.6425          |
|       | JPEG (16x16) | <br>0.6278 | <br>0.6433        | <br>0.7596 | <br>0.9140 | <br>0.6470          |
| Girl  | JPEG (8x8)   | <br>0.6295 | <br>0.7268        | <br>0.6970 | <br>0.7810 | <br>0.7739          |
|       | JPEG (16x16) | <br>0.6314 | <br>0.6944        | <br>0.7202 | <br>0.8343 | <br>0.7583          |

Figure 13

Extracted watermarks of the different methods under the JPEG compression.