



Consideration of a robust watermarking algorithm for color image using improved QR decomposition

Phuong Thi Nha¹ · Ta Minh Thanh¹ · Nguyen Tuan Phong¹

Accepted: 17 January 2022

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

In order to protect the digital image copyright, it is necessary to design a robust watermarking algorithm. To achieve this purpose, a novel color image watermarking scheme based on an improved QR decomposition is proposed in this paper. The proposed method gives a new algorithm to find elements of Q and R matrices instead of using the Gram–Schmidt algorithm for QR factorization. First, the 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, the Q matrix is computed based on the R matrix. In addition, a novel formula is proposed to improve the extracting time where the first element $R(1, 1)$ of the R matrix is found instead of computing QR decomposition as the previous proposals. Experimental results show that the proposed method outperforms other considered methods in this paper in terms of the quality of the watermarked images. Furthermore, the execution time is significantly improved, and the extracted watermark is more robust against almost tested attacks.

Keywords Digital watermarking · Copyright protection · Gram–Schmidt algorithm · QR factorization · Quantization index modulation (QIM)

1 Introduction

1.1 Overview

In recent years, exchanging digital data via the Internet has become more and more popular. The rapid development of digital technology techniques and devices has brought a lot of convenience for users. However, it is also a fertile land for attacks who want to steal or fake information. Therefore, inspecting the integrity and the authentication of data is an extremely important issue to tackle risks. For images, besides general rules of the law, there are some applied methods such as encryption, information hiding, and watermarking to help owners 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 an image by some constraints.

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 the pixel intensities of the cover image (Su and Chen 2018). Altering the 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 certain criteria, the transform domain coefficients are altered for embedding the watermark information. Finally, the inverse transform is applied to obtain the watermarked digital image. Although watermarking methods in the frequency domain have high computational complexity,

✉ Ta Minh Thanh
thanhtm@lqdtu.edu.vn

Phuong Thi Nha
phuongthinha@lqdtu.edu.vn

Nguyen Tuan Phong
tuanphongnguyen1974@gmail.com

¹ Le Quy Don Technical University, 236 Hoang Quoc Viet, Hanoi, Vietnam

they are always more robust than spatial domain-based watermarking schemes.

Image watermarking schemes based on DCT transformation often embed watermark on the median frequency to harmonize between the quality of the watermarked image and the robustness of the extracted information (Su et al. 2015; Hsu and Hu 2017). If the embedding is implemented on low frequency, the extracted watermark is good, but the invisibility of the 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 (Giri et al. 2015). However, to balance between the quality of the watermarked image and the robustness of the extracted watermark, the watermark is also embedded into HL and LH sub-bands.

For SVD decomposition, there are two trends for the embedding and extracting process. The first one used the first element $D(1, 1)$ of the D triangular matrix to change pixel values (Sun et al. 2002; Vaishnavi and Subashini 2015), whereas another way executed modifying elements of the first column of the U matrix (Lai 2011; Luo et al. 2020). Sun et al. in (2002) designed a novel watermarking scheme based on SVD, in which the watermark was embedded into 512×512 images by modifying the first coefficient of the D triangular matrix. This method had better performance in terms of robustness because the authors proposed an excellent formula to embed and extract the watermark. In addition, there is a novel scheme that was proposed by An-Wei Luo in (2020). 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 the U matrix reduced the quality of the watermarked images.

While the time required to conduct SVD computation is about $11n^3$ flops, Schur decomposition needs a fewer number of flops which is approximately $8n^3/3$ for an $n \times n$ matrix. That is the reason why some researchers focused on kind of this matrix analysis (Liu et al. 2017; Su et al. 2020). Su (2020) in 2020 described a new Schur decomposition-based algorithm where $U(2, k)$ 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.

In addition, QR decomposition, which decomposes a square matrix into an orthogonal matrix and an upper triangular matrix, is a very important matrix transformation for watermarking images. The advantage of QR decomposition lies in its low computational complexity and stable numerical feature. Because QR decomposition is an intermediate step in Schur decomposition, it requires a fewer number of computations than SVD and Schur factorization (Su et al. 2013; Sima et al. 2018). Therefore, it is appropriate for real-time systems. In many works, the elements in the first column of

the Q matrix not only have the same sign but also have similar values (Su et al. 2013). The QR decomposition-based watermarking also has satisfactory performance in terms of imperceptibility and robustness (Chen et al. 2021). Furthermore, due to concentrating energy of QR analysis on the first element of the R triangular matrix, the information is often embedded in this element (Qingtang et al. 2017, 2019). Discovering the suitable element to embed is one of the major factors for determining the effectiveness of an image watermarking scheme. This completely agrees with the purpose of watermarking technique which only focuses on some high-energy elements instead of embedding information on all elements of the pixel matrix to guarantee the quality of the watermarked image.

For the above importance of QR decomposition, many researchers utilized QR decomposition to transform pixel matrices in their watermarking techniques (Su et al. 2013; Naderahmadian and Hosseini-Khayat 2014; Su et al. 2014; Qingtang et al. 2017; Sima et al. 2018; Qingtang et al. 2019; Chen et al. 2021). From 2013 to 2019, Qingtang Su had three papers that focused on QR factorization. In the first method, Su *et. al* performed a watermarking process that depends on the relation between the second-row first-column coefficient and the third-row first-column coefficient of the Q unitary matrix (Su et al. 2013). After that, Su improved it by other choices in the years 2017 and 2019 where the author divided the host image into 3×3 blocks instead of a size of 4×4 as in the paper (Su et al. 2013). These improvements enhanced the embedded watermark capacity; however, the quality of the watermarked image was heavier affected.

Besides using separately above methods, many authors also had hybrid image watermarking schemes to strengthen the robustness of the watermark in recent years. That is a combination of DWT and SVD (Singh et al. 2017; Yadav et al. 2018; Roy and Pal 2019; Ernawan and Kabir 2020; Laxmanika and Singh 2020), DCT and SVD (Li et al. 2018), DWT and DCT (Abdulrahman and Ozturk 2019), DWT and QR (Jia 2017; Singh et al. 2018), or DWT and LU (Wang et al. 2016). The experimental results of these proposals showed that the robustness of the extracted watermark is more improved than previous researches. Normalized correlation (NC) value, which measures the robustness, is often up to 90% under all image attacks. However, the invisibility of the watermarked images 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 which involve quality of watermarked image, the robustness of extracted

watermark, and execution time. To balance these requirements, a novel method, which is based on the formula of Sun (2002) and QR decomposition, is proposed in our paper. Sun (2002) embedded information on the first element $D(1, 1)$ of D triangular matrix after decomposing SVD (called *SunSVD*). The embedding and extracting formula in this proposal was referred to 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*) was experimented. It used the Gram–Schmidt algorithm (Vandenberghe 2018) 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 the upper triangular matrix R is used to embed and extract information due to the concentration of energy on this element.

The previous methods often utilized the Gram–Schmidt (GS) algorithm to decompose the image matrix. However, GS -based QR decomposition exists some disadvantages as follows. Firstly, according to Stewart (1998) and Vandenberghe (2018), the algorithm requires approximately n^3 flops for a $n \times n$ matrix. It means that its complexity is $O(n^3)$ which is similar to SVD and Schur decomposition. Therefore, it is more effective if we find a novel solution to compute QR factorization with less complexity. Secondly, although the GS algorithm-based method is easier to set up, it gives out worse the invisibility and the robustness of the watermarked image than the one based on SVD decomposition according to experimental results. A reason for this is because the GS algorithm concurrently calculates Q and R column by column, so it does not inspect diagonal elements of the R matrix if these values are zero or negative. From the theoretical point of view, an important factor to make the Q matrix and R matrix unique is that all diagonal elements $R(i, i)$ of the R matrix must be positive (Vandenberghe 2018). However, the GS algorithm does not ensure this demand in some cases. There are two versions of the GS algorithm: the classical algorithm and the modified algorithm. While the modified version depends upon the condition of the original matrix and fails when the original matrix is singular, classic GS usually has very poor orthogonality (Stewart 1998). Therefore, GS is considered a less accurate and stable algorithm (Stewart 1998). Besides that, the GS algorithm is not recommended in practice due to being sensitive to rounding errors as mentioned on page 15 of (Vandenberghe 2018). Figure 1 is an example to illustrate the error of GS -based QR decomposition when the original matrix A is in an ill condition. A matrix in the example is a 4×4 pixel block of the “Girl” image (University of Granada 2022). We can see that some elements of Q and R matrices are unknown in this case. As the result, the product of $Q^T Q$ is infinite and it is not equal to the unit matrix I , so the GS algorithm does not guarantee

the orthogonality of the Q matrix. Moreover, an inverse QR factorization cannot be computed. In other words, GS -based QR decomposition fails in this situation.

1.3 Our contributions

Our contributions can be summarized as follows:

1. To overcome the drawbacks of *SunQR*, a new watermarking scheme based on QR decomposition is proposed in this paper where the R matrix is computed at first and diagonal elements of the R matrix are checked and modified if they are zero or negative. This action ensures that Q and R matrices are always unique without the loss of orthogonality. Due to this computation, the quality of the watermarked image can be significantly improved.
2. Calculating elements of R is performed by solving a set of linear equations. After that, Q is computed based on R . Therefore, the time complexity of the proposed QR decomposition is $O(n^2)$ instead of $O(n^3)$ as the GS -based QR factorization.
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 compared with other methods. That makes our method can be suitable for real-time applications.

1.4 Roadmap

The rest of this paper is organized as follows. Section 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.

Fig. 1 An example of GS-based QR decomposition is when the original matrix is in ill condition

```

A matrix before embedding
43.0000 41.0000 40.0000 39.0000
43.0000 41.0000 39.0000 39.0000
43.0000 41.0000 39.0000 39.0000
43.0000 41.0000 39.0000 39.0000

Q matrix before embedding
0.5000 -1.#IND -1.#IND -1.#IND
0.5000 -1.#IND -1.#IND -1.#IND
0.5000 -1.#IND -1.#IND -1.#IND
0.5000 -1.#IND -1.#IND -1.#IND

R matrix before embedding
86.0000 82.0000 78.5000 78.0000
0.0000 0.0000 -1.#IND -1.#IND
0.0000 0.0000 -1.#IND -1.#IND
0.0000 0.0000 0.0000 -1.#IND

Q_T*Q matrix
1.0000 -1.#IND -1.#IND -1.#IND
-1.#IND -1.#IND -1.#IND -1.#IND
-1.#IND -1.#IND -1.#IND -1.#IND
-1.#IND -1.#IND -1.#IND -1.#IND

Inverse QR matrix
-1.#IND -1.#IND -1.#IND -1.#IND
-1.#IND -1.#IND -1.#IND -1.#IND
-1.#IND -1.#IND -1.#IND -1.#IND
-1.#IND -1.#IND -1.#IND -1.#IND
    
```

For example, a matrix A of size 4×4 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 as follows.

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

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

2.2 Arnold transform

For improving the security of watermarking method, Arnold transform is often used to permute the watermark image (Su et al. 2020), and its detailed permutation process is given by Eq. (2).

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

where x', y', x , and y are integers in $\{0, 1, 2, \dots, N-1\}$ and N is order of watermark image matrix. The modulus operation is denoted by *mod* with a divisor with N . The image pixel at the coordinate (x, y) can be permuted to a new coordinate (x', y') by Eq. (2), which disorganizes the order of the watermark image. Based on the Arnold transform, we can enhance the security in the 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

2.3.1 Finding the elements of 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:

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

(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 a size of 4×4 . Therefore, the M, A^T, R^T , and R matrices are also 4×4 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

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

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

In order to find the elements of the 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

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

$$r_{11}r_{12} = m_{12} \Rightarrow r_{12} = \frac{m_{12}}{r_{11}} \tag{5}$$

$$r_{11}r_{13} = m_{13} \Rightarrow r_{13} = \frac{m_{13}}{r_{11}} \tag{6}$$

$$r_{11}r_{14} = m_{14} \Rightarrow r_{14} = \frac{m_{14}}{r_{11}} \tag{7}$$

$$r_{12}r_{12} + r_{22}r_{22} = m_{22} \Rightarrow r_{22} = \sqrt{m_{22} - r_{12}^2} \tag{8}$$

$$r_{12}r_{13} + r_{22}r_{23} = m_{23} \Rightarrow r_{23} = \frac{m_{23} - r_{12}r_{13}}{r_{22}} \tag{9}$$

$$r_{12}r_{14} + r_{22}r_{24} = m_{24} \Rightarrow r_{24} = \frac{m_{24} - r_{12}r_{14}}{r_{22}} \tag{10}$$

$$r_{13}r_{13} + r_{23}r_{23} + r_{33}r_{33} = m_{33} \Rightarrow r_{33} = \sqrt{m_{33} - r_{13}^2 - r_{23}^2} \tag{11}$$

$$r_{13}r_{14} + r_{23}r_{24} + r_{33}r_{34} = m_{34} \Rightarrow r_{34} = \frac{m_{34} - r_{13}r_{14} - r_{23}r_{24}}{r_{33}} \tag{12}$$

$$r_{14}r_{14} + r_{24}r_{24} + r_{34}r_{34} + r_{44}r_{44} = m_{44} \Rightarrow r_{44} = \sqrt{m_{44} - r_{14}^2 - r_{24}^2 - r_{34}^2} \tag{13}$$

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

$$r_{11} = \sqrt{m_{11}} \text{ and } r_{1j} = \frac{m_{1j}}{r_{11}} \tag{14}$$

$$r_{ij} = \begin{cases} \sqrt{m_{ij} - \sum_{k=1}^{i-1} r_{kj}^2}, & i = j \\ \frac{(m_{ij} - \sum_{k=1}^{i-1} r_{ki}r_{kj})}{r_{ii}}, & i \neq j \end{cases} \tag{15}$$

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

2.3.2 Finding the elements of Q matrix

Calculating the elements of the Q matrix is based on the R matrix. The Q matrix can be expressed by columns 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 (Vandenberghe 2018), computes Q column by column. According to that, the columns of Q are calculated as follows.

$$\tilde{q}_1 = a_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} \text{ and } q_1 = \frac{1}{r_{11}} \tilde{q}_1 \tag{16}$$

$$\tilde{q}_2 = a_2 - r_{12}q_1 = \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ a_{42} \end{bmatrix} - r_{12} \begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \end{bmatrix} \text{ and}$$

$$q_2 = \frac{1}{r_{22}} \tilde{q}_2. \tag{17}$$

$$\tilde{q}_3 = a_3 - r_{13}q_1 - r_{23}q_2 = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \\ a_{43} \end{bmatrix}$$

$$-r_{13} \begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \end{bmatrix} - r_{23} \begin{bmatrix} q_{12} \\ q_{22} \\ q_{32} \\ q_{42} \end{bmatrix} \text{ and } q_3 = \frac{1}{r_{33}} \tilde{q}_3 \tag{18}$$

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

$$q_3 = \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \\ a_{44} \end{bmatrix} - r_{14} \begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \end{bmatrix} - r_{24} \begin{bmatrix} q_{12} \\ q_{22} \\ q_{32} \\ q_{42} \end{bmatrix} - r_{34} \begin{bmatrix} q_{13} \\ q_{23} \\ q_{33} \\ q_{43} \end{bmatrix} \text{ and}$$

$$q_4 = \frac{1}{r_{44}} \tilde{q}_4 \tag{19}$$

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

$$\tilde{q}_1 = a_1 \text{ and } q_1 = \frac{1}{r_{11}} \tilde{q}_1 \tag{20}$$

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

$$q_i = \frac{1}{r_{ii}} \tilde{q}_i \tag{21}$$

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

2.3.3 The computational complexity of the proposed solution

Algorithm 1 presents shortly steps to discover the elements of the R matrix and Q matrix. According to Algorithm 1, it is easy to see that the computational complexity of the proposed approach is the sum of four calculations. These calculations comprise transposing A matrix, multiplying A^T and A , calculating the elements r_{ij} and q_{ij} , respectively. Each calculation needs two nested for loops, so the execution time for each one is Cn^2 (C is a constant). Therefore, the time complexity of the whole Algorithm 1 is $O(n^2)$

Algorithm 1 Calculating R and Q matrices.

INPUT: An A matrix with size of $n \times n$

OUTPUT: The upper triangular matrix R and the orthogonal matrix Q

BEGIN:

1. Transpose A matrix
2. Calculate $M = A^T A$
3. Compute the element r_{ij} of R matrix based on Eq. (14) and Eq. (15)

4. Compute the elements q_{ij} of Q matrix based on Eq. (20) and Eq. (21)

return r_{ij} and q_{ij}

END.

3 The proposed watermarking method

In this section, we describe a new watermarking scheme based on the improved QR decomposition and the formula of Sun in the paper (2002). The image watermarking scheme includes two stages, embedding and extracting, respectively.

3.1 Watermark embedding scheme

In the embedding process, the host color image is divided into 4×4 non-overlapping blocks at first. Then, the gray watermark image is permuted by Arnold transform and is converted 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 the R matrix. The proposed watermark embedding scheme can be summarized as follows.

1. Divide the host color image H into 4×4 non-overlapping blocks. In this image, each pixel is represented by three components (R, G, B).
2. Permute the gray watermark image by Arnold transform and then convert to a one-dimensional array w_i with $i = 1, 2, \dots, M \times M$. $M \times M$ 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 matrix.
 - Calculate A^T by transposing the matrix A
 - Compute $M = A^T A$
 - Find out R and Q matrices by solving a set of equations as represented in Eqs. (14), (15), (20) and (21).

4. Embed a watermark bit into the triangular matrix R based on the formula of Sun (2002):

- Get the first element $R(1, 1)$ of R matrix
- Calculate $z = R(1, 1) \bmod q$ (with q is a positive integer)
- Case $w_i = "0"$

$$R'(1, 1) = \begin{cases} R(1, 1) + \frac{q}{4} - z, & z < 3\frac{q}{4} \\ R(1, 1) + 5\frac{q}{4} - z, & \text{elsewhere} \end{cases} \quad (22)$$

- Case $w_i = "1"$

$$R'(1, 1) = \begin{cases} R(1, 1) - \frac{q}{4} - z, & z < \frac{q}{4} \\ R(1, 1) + 3\frac{q}{4} - z, & \text{elsewhere} \end{cases} \quad (23)$$

Note that q is also the strength of watermark embedding.

5. Update matrix A by formula Eq. (1): $A' = QR'$ and assign A' back to B components of the blocks.
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' .

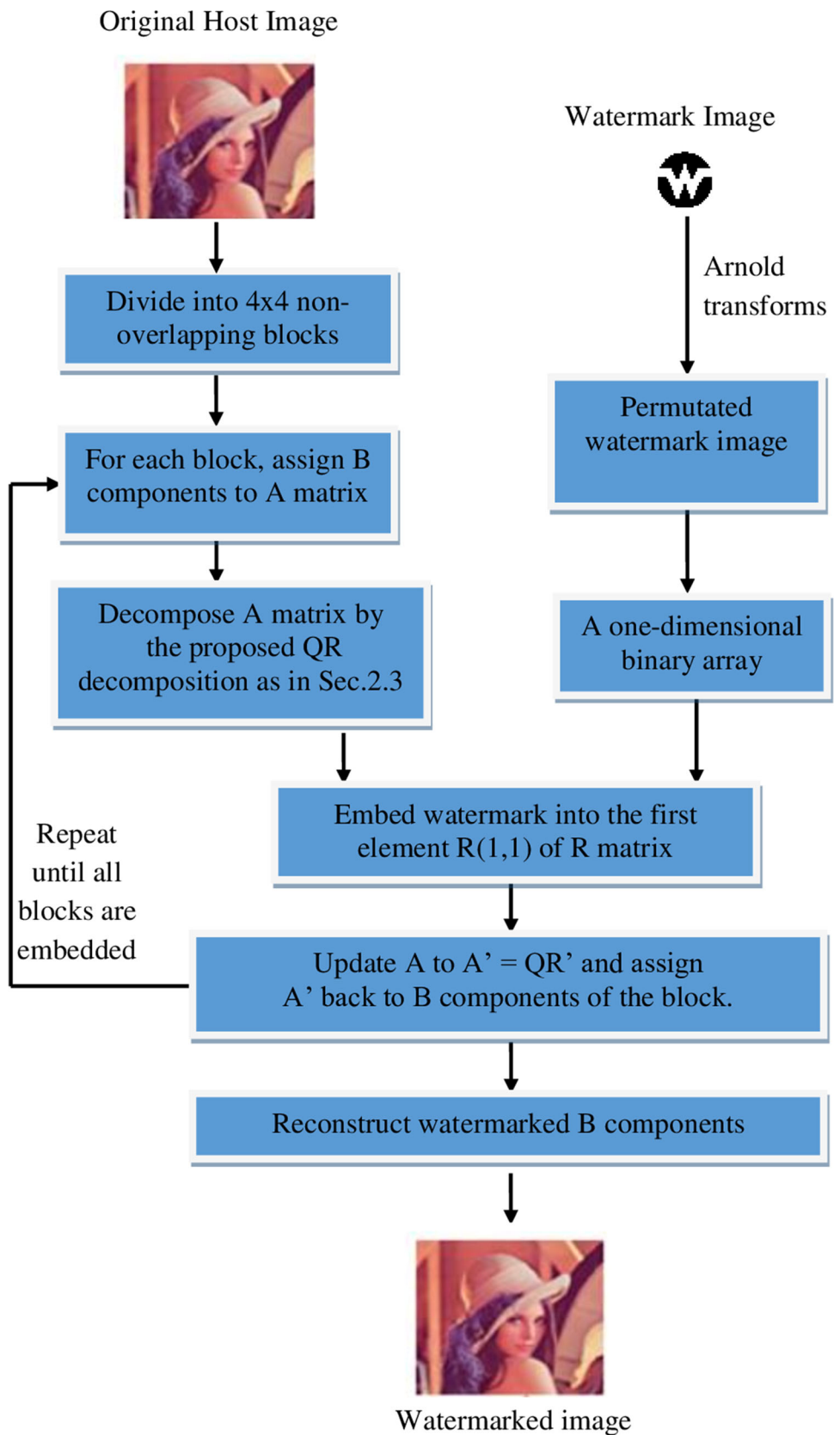
The detail of steps for the embedding stage can be represented in Fig. 2.

3.2 Watermark extraction scheme

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

1. Divide the watermarked image H' into 4×4 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 4×4 matrix (matrix A^*).

Fig. 2 The embedding stage



- Obtain the first element $R^*(1, 1)$ of R^* matrix as follows: $R^*(1, 1) = \text{length of the first column vector of } A^*$ matrix (Vandenbergh 2018).

$$R^*(1, 1) = \sqrt{A^*(1, 1)^2 + A^*(2, 1)^2 + A^*(3, 1)^2 + A^*(4, 1)^2} \tag{24}$$

- Extract the information of watermark based on algorithm of Sun (2002):
 - Calculate $z = R^*(1, 1) \bmod q$
 - The watermark bit is extracted by using following equation.

$$w = \begin{cases} \text{"0"}, & z < \frac{q}{2} \\ \text{"1"}, & \text{elsewhere} \end{cases} \tag{25}$$

- 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 in Fig. 3. And an example of the proposed image watermarking algorithm is also illustrated in Fig. 4.

3.3 The computational complexity of the proposed scheme

According to Sect. 2.3.3, the time complexity for the proposed QR decomposition is $O(n^2)$ with an image block size of $n \times n$. For the proposed watermarking scheme, we only embed the watermark on the B channel instead of three color channels of the host image. The reasons for choosing component B are: (a) considering all components may distort the colors of the watermarked images; (b) human eyes are less sensitive to component B than components R and G according to the human visual system. Therefore, the overall complexity of the proposed embedding scheme for a color image of $N \times N$ pixels is estimated as $\frac{N \times N}{n \times n} O(n^2)$. And the overall complexity of the proposed extraction scheme is only $\frac{N \times N}{n \times n} O(1)$ for a color image of $N \times N$ pixels because we do not use QR decomposition as the previous QR -based methods in this stage. Instead of that, we calculate $R(1, 1)$ of the R matrix based on Eq. (24).

4 Experimental results

4.1 Evaluation criteria

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 $N \times N$ in this paper. PSNR is employed as a measure for evaluating the quality of the watermarked image. PSNR is described by the following equation.

$$\text{PSNR} = 10 \log_{10} \frac{255^2 \text{MSE}}{\dots} \tag{26}$$

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

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (H(i, j) - H'(i, j))^2 \tag{27}$$

Moreover, the SSIM is considered to be correlated with the quality perception of the human visual system (HVS). The SSIM, as denoted in Eq. (28), is also used to measure the similarity between the original color image H and the watermarked image H' (Jia 2017).

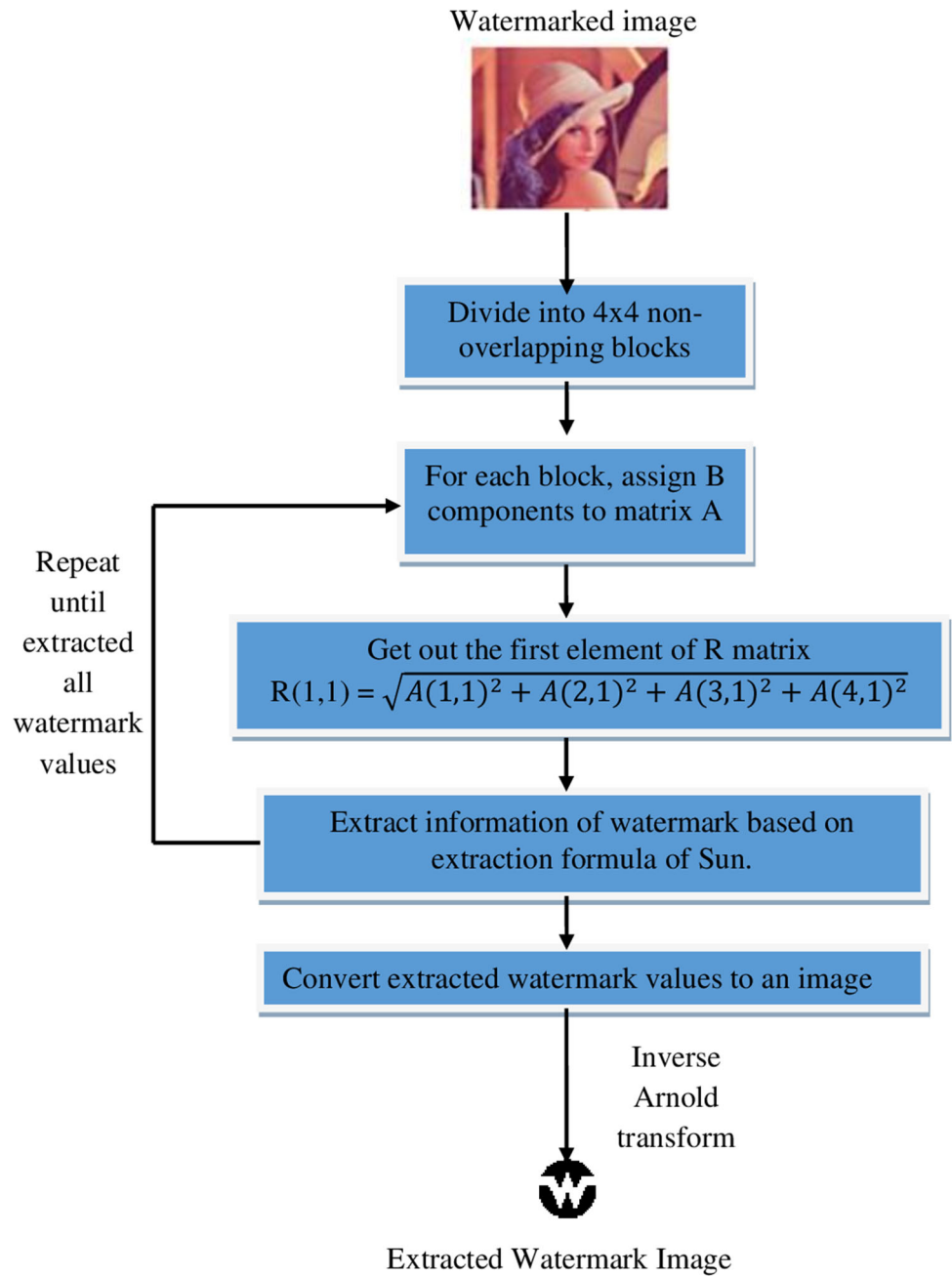
$$\text{SSIM}(H, H') = l(H, H')c(H, H')s(H, H'), \tag{28}$$

where

$$\begin{cases} l(H, H') = (2\mu_H \mu_{H'} + C_1) / (\mu_H^2 + \mu_{H'}^2 + C_1) \\ c(H, H') = (2\sigma_H \sigma_{H'} + C_2) / (\sigma_H^2 + \sigma_{H'}^2 + C_2) \\ s(H, H') = (\sigma_{HH'} + C_3) / (\sigma_H \sigma_{H'} + C_3) \end{cases} \tag{29}$$

The first term in Eq. (29) is the luminance comparison function which measures the closeness of the two images' mean luminance (μ_H and $\mu_{H'}$). The second term is the contrast comparison function which measures the closeness of the contrast of the two images. Here the contrast is measured by the standard deviation σ_H and $\sigma_{H'}$. The third term is the structure comparison function which measures the correlation coefficient between the two images H and H' . Note that $\sigma_{HH'}$ is the covariance between H and H' . The positive values of the SSIM index are in $[0, 1]$. A value of "0" means no correlation between images, and "1" means that $H = H'$. The positive constants C_1 , C_2 , and C_3 are used to avoid a null denominator.

Fig. 3 The extracting stage



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:

$$NC = \frac{\sum_{j=1}^4 \sum_{x=1}^m \sum_{y=1}^n (W(x, y, j)W'(x, y, j))}{\sqrt{\sum_{j=1}^4 \sum_{x=1}^m \sum_{y=1}^n (W(x, y, j))^2} \sqrt{\sum_{j=1}^4 \sum_{x=1}^m \sum_{y=1}^n (W'(x, y, j))^2}}, \tag{30}$$

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 and $m \times n$ 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 terms 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.

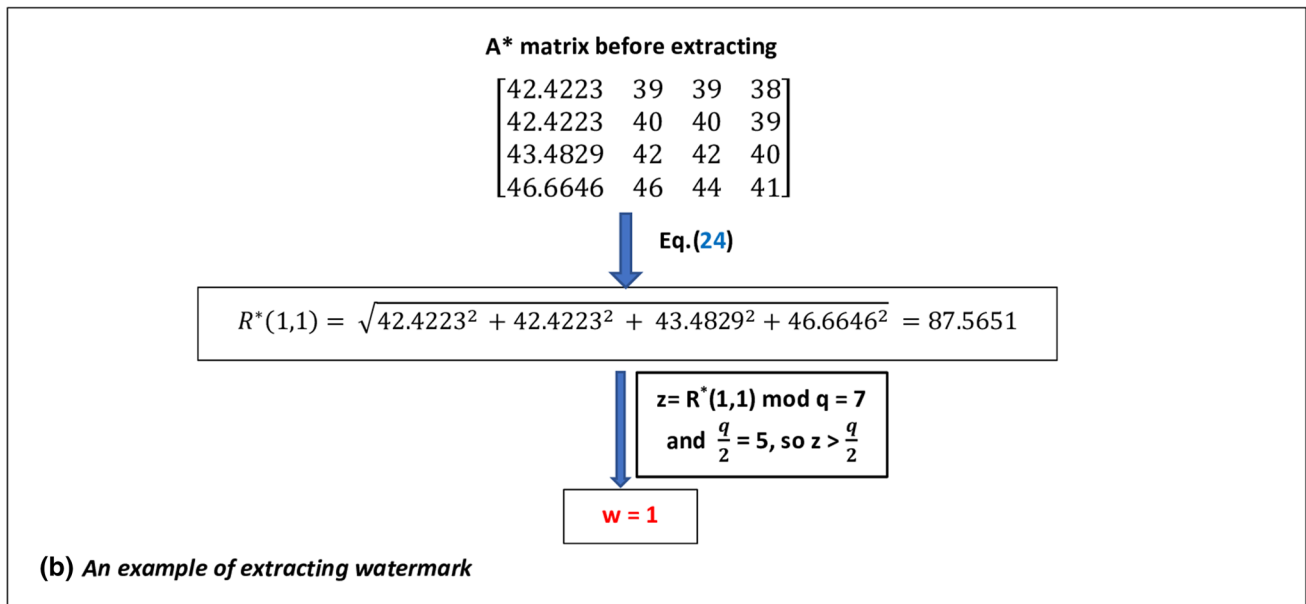
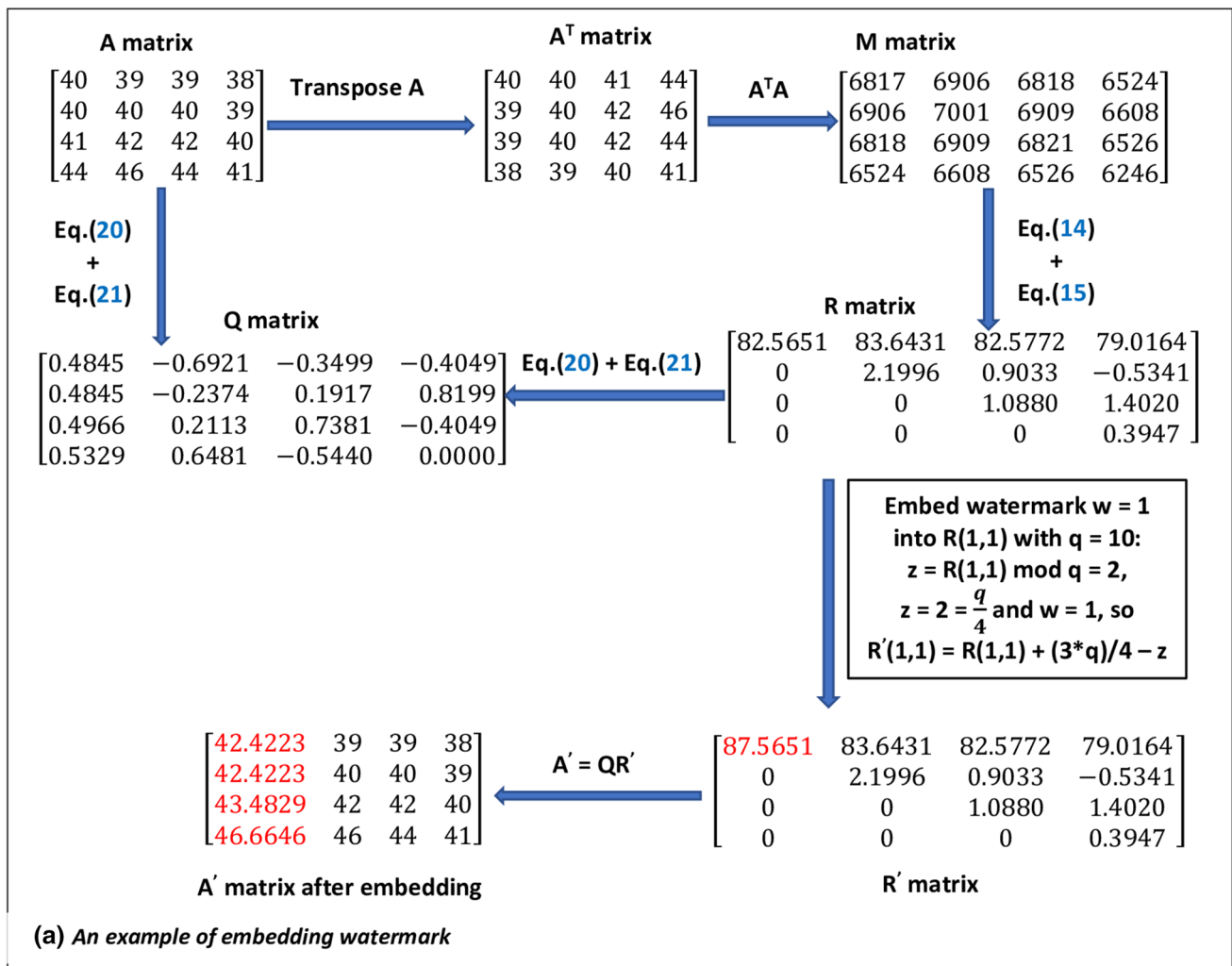


Fig. 4 An example of the proposed watermarking algorithm

Table 1 Various attacks used in our experiments

Index	Attacks	Description
1	Blur	Blur the watermarked image with radius = 0 and sigma is designed to 0.2 and 0.5, respectively
2	Sharpen	Sharpen the watermarked image with radius = 0 and sigma is designed to 0.2 and 0.5, respectively
3	Cropping	Remove 25% and 50% of the watermarked image on the top left corner
4	Scaling (1/2)	Resize the watermarked image from 512×512 to 256×256 and subsequently restore it to 512×512 .
5	Scaling (2)	Resize the watermarked image from 512×512 to 1024×1024 and subsequently restore it to 512×512
6	Gaussian noise	Add Gaussian noise to the watermarked image with $\mu = 0$ and variance $\sigma^2 = 0.001, 0.003$
7	S&P noise	Add salt and pepper noise to the watermarked image with a noise density den = 0.002, 0.005, 0.01
8	Rotation 5°	Rotate the watermarked image clockwise at the angle = 5 and subsequently rotate it counterclockwise at the angle = - 5
9	Rotation 10°	Rotate the watermarked image clockwise at the angle = 10 and subsequently rotate it counterclockwise at the angle = - 10
10	JPEG	Compress the watermarked images by using <i>DCT</i> transform with size of window 8×8 and 16×16
11	Mean filter	Use the filter with size of window from 2×2 and 5×5

4.2 The simulation setting

In order to evaluate objectively the stability and the effectiveness of the proposed method, twelve 24-bit color images with a size of 512×512 in the CVG-UGR image database (University of Granada 2022) are selected as the host images, and two 32×32 gray images are used as original watermarks as shown in Fig. 5. The host images are standard color images that involve various types such as portrait, landscapes photograph, animal photograph, and fruit photograph. Pixel distribution of these images is different from each other. All tests are implemented by Visual Studio v15 and are performed on a laptop with Intel® Core™ i5-6200U CPU at 2.30 GHz, 4.00 GB RAM and 64-bit OS. Table 1 describes shortly different image attacks which are used in our robustness tests.

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). Table 2 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 Table 2, 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 invisibility and robustness, a value of q is set to 10 for evaluating the performance of the proposed method.

A comparison between the methods of Sun (2002), Luo (2020), Su (2020), Su (2018), Chen (2021), Hu (2020), Chen

Table 2 Values of SSIM and NC under different embedding coefficients

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

Best values are indicated in bold

(2021), Qin (2021), Kumar (2021) and our proposal is performed to simulate for the effectiveness of the proposed algorithms. Figure 6 reviews some important information of the related works in terms of utilized technique, evaluation tools, data set, performance metrics, advantages, and disadvantages. The first one is a scheme of Sun (2002) which used SVD decomposition as the main technique to embed a 64×64



Fig. 5 The host images: **a** avion, **b** baboon, **c** Balloon, **d** couple, **e** Girl, **f** house, **g** lena, **h** milkdrop, **i** parrots, **j** peppers, **k** sailboat, **l** tree. The watermarks: **m** w1, **n** w2

Scheme of	Used techniques	Evaluation tools	Host image	Watermark image	Parameters	Position to embed	Size of block	Advantages	Disadvantages
Sun[5]	SVD	T _{AF}	Gray, 512x512	Grayscale, 64x64	The quantization coefficient Q is set to 10.	D(1,1) of the D matrix	8x8	The proposed scheme has the advantages of distinguishing the JPEG lossy compression from other malicious manipulation (such as replacing) and identifying the modified portion of the image.	Big time complexity ($O(n^3)$). The experiments were not evaluated on color images.
Luo [8]	DWT, SVD, Logistic map	PSNR, NC	Gray, 512x512	Grayscale, 32x32	The controlling parameters k_1 and k_2 are designated as 0.02 and 2, respectively	U_s and U_o of U matrix on the LL sub-band	4x4	The method has an optimal SVD blocks selection strategy which improves the imperceptibility and robustness	Big time complexity ($O(n^3)$). The performance of resisting rotation operation will be further studied and the work will need to be extended to color images
Su[10]	Schur and Arnold transform	PSNR, SSIM, NC	Color, 512x512	Color, 32x32	The embedding coefficient $T = 0.03$ and the embedding strength $\Delta = 25$	U(2,c) and U(3,c) of U matrix where $c = \max(D_{max})$ or D_{max} of the D matrix	4x4	The method has a strategy to select the embedding image blocks from the image layer, so the watermark invisibility has been significantly improved.	The embedded flag is uploaded to the cloud service provider with the watermarked image. This may be a security gap of this approach.
Su [29]	LU, Arnold Transform, MD5	PSNR, SSIM, NC	Color, 512x512	Color, 32x32	The embedding strength T is set to 0.0275	L(2,1) and L(3,1) of the L matrix	4x4	The low computational complexity ($O(n^2)$) and high embedding payload	The performance of resisting image rotating attack will be further considered in future work
Chen[31]	Quaternion QR (QQRD)	PNSR, BER, NC	Color, 512x512	Color, 32x32	The strength control parameter T is set in the interval [0.003, 0.03]	Q(2,1) and Q(3,1) of the Q matrix	4x4	The algorithm avoids layered processing that causes abnormal sensitivity to attacks and less correlation that causes excessive modification of coefficient values	This scheme has limited resistance to 50% scaling correction and Gaussian noise
Hu[32]	SVD, Arnold transform, mixed modulation	SSIM, BER, NC	Color, 512x512	Color, 64x32 and 32x16	The value of level shifting λ is set to 216 and quantization step size D is set to 0.032	U(2,1) and U(3,1) of the U matrix	4x4	The proposed watermark embedding process helps to preserve orthonormality in the unitary matrix and compensate for the resulting distortion. Iterative regulation ensures the accurate retrieval of the embedded watermark, while mixed modulation helps to improve robustness without compromising image quality.	Big time complexity ($O(n^3)$). The proposed algorithm did not fare particularly well against JPEG compression, Gaussian noise, or speckle noise
Chen[35]	WHT, Arnold transform, MD5	PSNR, SSIM, NC	Color, 512x512	Color, 32x32	The quantization step size $T = 5$	The first row of the H matrix	4x4	This algorithm also has higher embedding capacity and better real-time performance. In addition, the proposed algorithm improves the security of the algorithm by transforming and rearranging the pixel values and randomly selecting the embedded block	The robustness of this method is less than the others in some cases. And the quality of the watermarked images is at a mean level.
Qin[36]	DWT-QDFT, QR	PSNR, SSIM, CDR	Color, 512x512	Grayscale, 64x64	The quantization step $t = 40000$	R(1,4) of R matrix on sub-band LL of three R, G, and B channels	4x4	The watermarking algorithm has good robustness against attacks such as JPEG compression, cropping, and median filtering	The transparency of the watermarked image is very low because the PSNR values are only between 20 dB and 30 dB
Kumar[37]	LWT, ACM	PSNR, SNR, SSIM, NCC, MAE, MSE	Color, 512x512	Grayscale, 64x64	The scaling factor $\alpha = 0.7$	Sub-band HH of the Y component	32x32	The Alpha blending technique is used to balance the trade-offs between imperceptibility and robustness of the watermarking techniques and Arnold's Cat Map is used to enhance the security of watermarking technique	This method is less robust against median and filtering attacks
The proposed method	Improved QR (IQRD) and Arnold transform	PSNR, SSIM, NC	Color, 512x512	Grayscale, 32x32	The quantization coefficient q is set to 10	R(1,1) of the R matrix	4x4	By using an improved QR decomposition, the algorithm inspected the legality of the diagonal elements of the R matrix which enhances the quality of the watermarked image. Furthermore, the formula of calculating R(1,1) instead of performing QR factorization in the extracting stage reduced significantly computational time.	The method has a restriction on coping with geometric attacks and JPEG compression.

Fig. 6 A short description of the related works

binary image into a gray 512×512 image on the element $D(1, 1)$ of the D matrix. This scheme has the advantages of distinguishing the JPEG lossy compression from other malicious manipulation and identifying modified portions of the image but it also has a big computational complexity. It is similar to the schemes of Lou (2020) and Hu (2020) because both algorithms applied SVD analysis in their image watermarking methods too. In (Luo et al. 2020), Lou combined DWT with SVD decomposition and logistic map to create a novel watermarking scheme. The method has an optimal SVD blocks selection strategy which improves the imperceptibility and robustness. Lou (2020) and Hu (2020) inserted the watermark bits into the $U(2, 1)$ and $U(3, 1)$; however, Lou used gray and binary images while Hu utilized color images for the host image and the watermark image. In addition, Su had two proposals based on Arnold transform, LU decomposition, and Schur factorization in (Qingtang et al. 2018; Su et al. 2020), respectively. The approach of Su (2018) has a low computational complexity ($O(n^2)$) and high embedding payload but the performance of resisting image rotating attack will be further considered in future work. QR decomposition is selected in the method of Chen (2021) and our method. However, while Chen embedded color watermark bits into two elements $Q(2, 1)$ and $Q(3, 1)$ of Q matrix, we chose $R(1, 1)$ of R matrix as the modified element. In addition, three state-of-the-art algorithms were published in 2021 by Chen (2021), Qin (2021), and Kumar (2021), respectively. Firstly, Chen (2021) used WHT transform as a main technique for the proposed scheme. In this publish, the author calculated a correlation between the matrix coefficients to find out the embedded elements of the WHT blocks on three R , G , and B channels. This algorithm has higher embedding capacity and improves security by rearranging the pixel values and randomly selecting the embedded blocks. Secondly, a combination of DWT, QDFT, and QR decomposition was built by Qin (2021) to create a novel watermarking scheme. Qin performed the first-level DWT transform on R , G , and B channels at first. Next, three low-frequency components constituted a pure quaternion matrix and the left quaternion Fourier transform was performed to obtain the real part. Then, the watermark was embedded into the r_{14} element of the R matrix by QR decomposition. The algorithm has good robustness against attacks such as JPEG compression, cropping, and median filtering but the transparency of the watermarked image is very low because the PSNR values are only between 20 dB and 30 dB. Finally, an LWT-based image watermarking method, which hid the watermark into sub-band HH of Y component, was developed by Kumar (2021). In this article, Kumar applied the alpha blending technique to balance the trade-offs between the imperceptibility and the robustness, and Arnold's Cat Map was used to enhance security of the watermarking technique. Unfortunately, this method is less robust against median and filtering

attacks. In summary, all the above papers divide the host image into the blocks of 4×4 except Kumar (2021) and the evaluation tools are often PSNR, SSIM, NC , and BER.

























4.3 Invisibility test

The quality of watermarked images is evaluated by PSNR and SSIM indexes. 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, the extracted watermark should be the same original watermark under no attacks. It means that the NC value must be 1. However, it is not completely correct in our experimental tests. In fact, the NC value is less than 1 because it depends on the structure of the host image as well as the embedding strength. In all methods, the embedding strength is chosen to a suitable value in order to balance between the quality of the watermarked image and extracted watermark. Figure 7 shows that PSNR values of the proposed method are at a stable level of over 62 dB for all tested images except "peppers" and "Parrots". For each host image, NC indexes of two watermarks are similar to each other.

The detailed results of the imperceptibility are also compared in Fig. 8. According to Fig. 8, PSNR/SSIM values of Su (2018), Luo (2020), Hu (2020), and Chen (2021) are lower than the other methods because the authors embedded the watermark on two elements of the L matrix, U matrix, and Q matrix, respectively. This causes a big change in two rows of the corresponding block after embedding. Therefore, the pixel values of the watermarked image will not be close to the original image. As the result, the quality of the watermarked image of these methods is worse. In addition, the figures in Fig. 8 display that the algorithms of Su (2018) and Hu (2020) bring a low result in terms of PSNR and SSIM. The reason for this is because these methods embedded the information on three channels instead of one channel as the others. Although embedding on three channels enhances the embedded capacity, it leads to distortion of the pixel values. Meanwhile, the methods of SunSVD (2002) and SunQR in Sect. 1.2 have higher PSNR/SSIM values. These methods used the 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 the cases of Su (2018), Luo (2020), Hu (2020), and Chen (2021). That is a reason why the formula of Sun is utilized in the proposed method.

Figure 9 is a piece of evidence to demonstrate the influence of the embedded elements on the pixel matrix. In this figure, the proposed algorithm only embeds watermark bits on $R(1, 1)$ of the R matrix. Thus, the matrix after embedding only is modified on the first column in comparison with the original matrix. Unlike that, the watermark bit is embedded into two elements $Q(2, 1)$ and $Q(3, 1)$ of the Q matrix in

Fig. 7 PSNR and *NC* values of the proposed method for twelve host images and two watermark images in the absence of attack

Host images	Watermark is 'w1'		Watermark is 'w2'	
	PSNR	NC	PSNR	NC
house	62.6034	 0.9985	62.5317	 0.9980
milkdrop	62.5203	 0.9935	62.3717	 0.9894
sailboat	62.6911	 0.9985	62.6590	 0.9980
Girl	62.6127	 1.0000	62.5665	 0.9999
lena	62.4939	 0.9978	62.4570	 0.9981
peppers	55.1929	 0.9841	57.9257	 0.9856
avion	62.5458	 0.9964	62.5527	 0.9980
baboon	62.0940	 0.9935	62.0187	 0.9904
couple	62.6295	 0.9978	62.4845	 0.9990
tree	62.6295	 0.9957	62.7430	 0.9981
Balloon	62.5298	 0.9877	62.5573	 0.9856
Parrots	49.8066	 0.9992	50.8003	 0.9990

the method of Chen (2021) which leads to change in the second row and the third row of the embedded matrix. To some extent, Fig. 9 explains why the proposed method has better imperceptibility than the scheme of Chen (2021).

In addition, it can be seen clearly in Table 3 that the proposed method gives much higher PSNR/SSIM and NC values than others. This means that the invisibility of the watermarked image is much better in the proposed scheme. In addition, the result table brings out that the proposed method can not only overcome the quality of the watermarked image but also 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. (22) 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 are based on Eqs. (8), (11), (13) in Sect. 2.3. From Eq. (8) 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 the watermark). This is a reason for reducing the quality of the 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 does not completely affect the embedding process as well as extraction because the two processes only use the first diagonal element r_{11} . This solution makes the proposed method more stable and effective in terms of the quality of the watermarked image.

4.4 Execution time test

It is easy to see that the execution time of watermarking image algorithms, which are based on the transform domain, depends mainly on the type of used matrix decomposition. SVD decomposition needs about $11n^3$ flops for a $n \times n$ matrix, whereas the required time to compute LU factor-







































































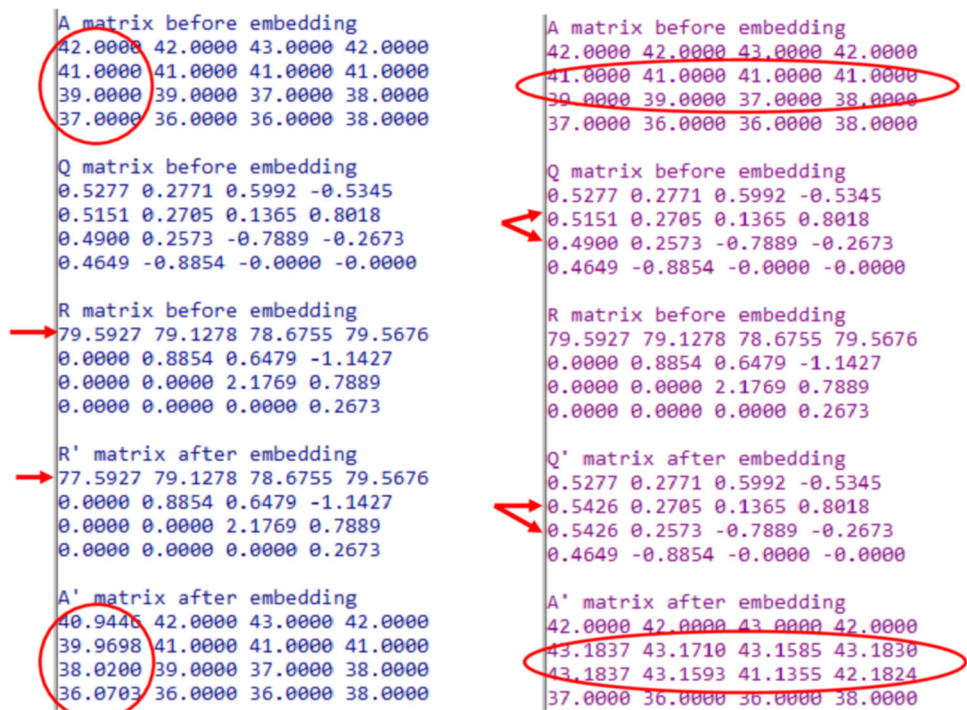
Method	SunSVD [5]	SunQR in Sect.1.2	Su [29]	Luo [8]	Hu[32]	Chen[31]	The proposed method
Watermarked image (PSNR/SSIM)	 58.0311/0.9989	 55.7257/0.9964	 43.4108/0.9893	 53.9879/0.9978	 42.6125/0.9862	 50.6604/0.9972	 62.5665/0.9996
Extracted watermark (NC)	 0.9952	 0.9451	 0.8522	 0.8977	 0.9912	 0.9923	 0.9999
Watermarked image (PSNR/SSIM)	 57.9641/0.9992	 55.1997/0.9996	 44.3169/0.9931	 47.9291/0.9927	 41.8759/0.9778	 52.682/0.9986	 62.4570/0.9996
Extracted watermark (NC)	 0.9942	 0.9980	 0.9414	 0.9828	 0.9956	 0.9962	 0.9981
Watermarked image (PSNR/SSIM)	 45.2276/0.9892	 55.7789/0.9991	 45.9144/0.9959	 47.6466/0.9953	 40.9531/0.9710	 51.0417/0.9980	 57.9257/0.9996
Extracted watermark (NC)	 0.9787	 0.9729	 0.9158	 0.9527	 0.9815	 0.9768	 0.9856
Watermarked image (PSNR/SSIM)	 57.9867/0.9991	 48.5683/0.9997	 34.9126/0.9650	 52.9882/0.9985	 38.3628/0.9693	 49.6744/0.9982	 62.5527/0.9997
Extracted watermark (NC)	 0.9923	 0.9971	 0.9609	 0.9827	 0.9945	 0.9952	 0.9980
Watermarked image (PSNR/SSIM)	 57.8236/0.9997	 58.0014/0.9998	 44.5487/0.9970	 46.0408/0.9976	 41.1352/0.9760	 53.7594/0.9990	 62.0187/0.9999
Extracted watermark (NC)	 0.9799	 0.9900	 0.9667	 0.9035	 0.9918	 0.9923	 0.9904

Fig. 8 A comparison of the quality of the watermarked images between the methods

Table 3 An average comparison between the different methods in terms of PSNR, SSIM and NC values

Value/method	SunSVD (Sun et al. 2002)	SunQR in Sect. 1.2	Su (2018)	Luo (2020)	Hu (2020)	Chen (2021)	The proposed method
PSNR	55.4066	54.6548	42.6207	49.7185	40.9879	51.5636	61.5041
SSIM	0.9972	0.9989	0.9881	0.9964	0.9761	0.9982	0.9997
NC	0.9881	0.9806	0.9274	0.9439	0.9909	0.9906	0.9944

Fig. 9 An example for comparison of original and embedded matrices between the proposed method and the approach of Chen (2021)



(b) An example of the proposed method

(a) An example in the approach of Chen[31]

ization is n^2 flops. And QR decomposition is considered as an intermediate stage between SVD and LU . For GS-based QR factorization, it costs $(n^3 - \frac{n^2}{3})$ floating-point additions and multiplications (Stewart 1998). In (Chen et al. 2021), Chen used quaternion QR decomposition ($QQRD$) which is based on Householder transformation for matrix calculation. According to Vandenberghe (2018), the Householder algorithm requires approximately $\frac{4n^3}{3}$ flops for an $n \times n$ matrix. On the other hand, the scheme of Hu (2020) is based on SVD factorization with mixed modulation incorporated. And its time complexity is the sum of the five basic processing modules with an image block size of $n \times n$, as follows: $SVD(O(4n^3))$, orthonormal restoration ($O(2n^3)$), distortion compensation ($O(3n^2)$), matrix recomposition ($O(2n^3)$), and tentative verification ($O(4n^3)$). The overall complexity of this method for a color image of $N \times N$ pixels is estimated as $\frac{N \times N}{n \times n}(12n^3 + 6n^2)$. Therefore, the SVD and QR decomposition-based approaches such as SunSVD (2002), Lou (2020), SunQR in Sect. 1.2, Chen (2021), and Hu (2020) have time complexity of $O(n^3)$ for an $n \times n$ matrix. For the proposed scheme, as shown in Sect. 3, the embedding time complexity is $\frac{N \times N}{n \times n}O(n^2)$ while the extracting time complexity of the proposed approach is only $\frac{N \times N}{n \times n}O(1)$ for a color image of $N \times N$ pixels.

In these experiments, a computer with Intel® Core™ i5-6200U CPU at 2.30 GHz and Visual Studio *v15* is used as the computing platform. The embedding time and extraction time of the proposed methods are 0.2448s and 0.006 s, respectively. Table 4 shows a comparison of the execution time between different methods.

According to Table 4, the total execution time of the proposed method is bigger than the time of Su (2018) which uses LU decomposition, but it is smaller than the others. In the scheme of Luo (2020), the author proposed a combination of DWT and SVD, so this method spends more time than the others except for the approach of Hu (2020). Meanwhile, the algorithm of Hu (2020) applied many processing modules such as level shifting with dither noise, SVD, sign correction, mixed modulation, orthonormal restoration, distortion compensation, tentative verification, and iterative regulation. It is the reason why its running time is the biggest.

For embedding time, the algorithm SunQR in Sect. 1.2 and the method of Chen (2021) are similar because they use QR factorization based on GS and Household algorithms, respectively. However, the algorithm of Chen (2021) needs more time to measure the correlation between elements q_{ij} of the Q matrix and select an optimal quaternion embedding position, so it costs a higher number. For extraction time, the proposed method gives an effective result because it calculates the length of the first column vector of the A matrix to find out the first element $R(1, 1)$ of the R matrix instead of using QR decomposition as the previous QR based schemes. This

Table 4 Average execution time of different methods (in second)

Method	Embedding time	Extracting time	Total time
SunSVD (2002)	0.4850	0.0870	0.5720
SunQR in Sect. 1.2	0.3718	0.0296	0.4014
Su (2018)	0.1774	0.0074	0.1848
Luo (2020)	2.4566	0.1730	2.6296
Hu (2020)	5.2116	1.0348	6.2464
Chen (2021)	0.4001	0.0279	0.4280
Proposed method	0.2448	0.0060	0.2508











































Image	Attack	SunSVD [5]	SunQR in Sect.1.2	Su [29]	Luo [8]	Hu[32]	Chen[31]	The proposed method
avion	Blurring (0.2)	 0.9923	 0.9971	 0.9609	 0.9827	 0.9945	 0.9952	 0.9980
	Blurring (0.5)	 0.9417	 0.9674	 0.9330	 0.9694	 0.9898	 0.9913	 0.9827
lena	Blurring (0.2)	 0.9943	 0.9980	 0.9415	 0.9827	 0.9956	 0.9952	 0.9981
	Blurring (0.5)	 0.9677	 0.9808	 0.9292	 0.9808	 0.9885	 0.9871	 0.9818
Girl	Blurring (0.2)	 0.9952	 0.9451	 0.8522	 0.8977	 0.9912	 0.9923	 0.9999
	Blurring (0.5)	 0.9933	 0.9370	 0.8436	 0.8957	 0.9789	 0.9865	 0.9999

Fig. 10 Extracted watermarks and *NC* values of the different methods under the blurring attack

demonstrates that the proposed approach can significantly improve the speed of the watermarking process, especially the extraction time.

4.5 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 such as SunSVD (2002), SunQR in Sect. 1.2, Su(2018), Luo(2020), Hu (2020), and Chen(2021). The schemes of Sun (2002) and Hu (2020) used SVD decomposition for depositing the pixel matrix. While SunQR in Sect. 1.2 applied the GS algorithm for *QR* factorizing, the Household algorithm-based *QR* factorization was utilized by Chen (2021). Besides, *LU* decomposition was developed

by Su in (2018), and the method of Lou(2020) was 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 since 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 you want the image to blur in pixels. In the experiments, the radius is fixed to '0,' and sigma is designed to 0.2 and 0.5, respectively. The sharpening operation is also a sort of inverted blurring. Both operations work in just about the same way. Therefore, its arguments are similarly set to blur. Figures 10 and 11 give the results of the visual comparison and quantitative values. *NC* values from these figures inform us about the superiority of the proposed











































Image	Attack	SunSVD [5]	SunQR in Sect. 1.2	Su [29]	Luo [8]	Hu[33]	Chen[31]	The proposed method
avion	Sharpening (0.2)	 0.9923	 0.9971	 0.9609	 0.9827	 0.9945	 0.9952	 0.9980
	Sharpening (0.5)	 0.9270	 0.9192	 0.9406	 0.9239	 0.9634	 0.9750	 0.9557
lena	Sharpening (0.2)	 0.9943	 0.9981	 0.9415	 0.9827	 0.9956	 0.9952	 0.9981
	Sharpening (0.5)	 0.9407	 0.9367	 0.9274	 0.9365	 0.9691	 0.9914	 0.9743
Girl	Sharpening (0.2)	 0.9952	 0.9451	 0.8522	 0.8977	 0.9912	 0.9923	 0.9999
	Sharpening (0.5)	 0.9887	 0.9340	 0.8421	 0.8968	 0.9802	 0.9923	 0.9990

Fig. 11 Extracted watermarks and NC values of the different methods under the sharpening operation

method over the others. Furthermore, it is easy to see that all schemes can prevent damage from blurring and sharpening attacks. The extracted watermark of Hu (2020) and Chen (2021) is clearer to recognize than the others for the “avion” and “lena” images when the sigma is set to 0.5.

Adding noise is also one of the common operations in 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% increase by 1%, and Fig. 12 shows a 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 process the watermarked images. Figure 13 shows the NC values and visual perception results of the 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 than the others against the process of adding noise. In some cases, the scheme of Chen (2021) has advantages but the gap between this approach and the proposed one is not too big. While adding Salt & Peppers noise seems not to make all methods difficult, the Gaussian noise attack brings a poor performance to the approaches of Su (2018), Luo (2020), and Hu (2020). The extracted watermark of Luo (2020) is even presented in an unrecognized shape for the “lena” image.

For the filtering attack, the mean filter method is performed on the watermarked images. A mean filter with different window sizes from 2×2 to 5×5 is used to process the watermarked images. Figure 14 shows the extracted

watermarks and NC values with the filtering sizes of 2×2 and 3×3 , respectively. It is seen from this figure that the method of Chen (2021) is the most effective among all studies. Besides, the algorithm of Hu (2020) is more robust than the methods of SunSVD (2002), SunQR, Su (2018), and Luo (2020). Our method also gives positive results and the extracted watermarks can be recognized for the “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 values cannot be measured in this operation because they do not correctly reflect the quality of the extracted watermark. As the results displayed in Fig. 15, the invisibility of the extracted watermarks of the proposed method is clearer than the other methods. Obviously, the proposed method is robust against this cropping process. And the algorithm of Luo (2020) seems to be less effective for the “lena” image because the extracted watermark can not be recognized in this case.

Another type of well-known image operation is geometry attack, which mainly includes rotation and scaling. There are two rotation experiments to show the robustness in Fig. 16. 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. Figure 17 shows the quantitative results and visual perception results for the case of scaling, respectively. In this experiment, two scaling
































































Image	Attack	SunSVD [5]	SunQR in Sect.1.2	Su [29]	Luo [8]	Hu[32]	Chen[31]	The proposed method
avion	Salt & Peppers noise (2%)	 0.9676	 0.9875	 0.9569	 0.9663	 0.9872	 0.9865	 0.9904
	Salt & Peppers noise (5%)	 0.9281	 0.9715	 0.9513	 0.9410	 0.9705	 0.9721	 0.9724
	Salt & Peppers noise (10%)	 0.8622	 0.9376	 0.9390	 0.9134	 0.9356	 0.9386	 0.9448
lena	Salt & Peppers noise (2%)	 0.9714	 0.9923	 0.9396	 0.9751	 0.9823	 0.9895	 0.9924
	Salt & Peppers noise (5%)	 0.9486	 0.9784	 0.9350	 0.9495	 0.9779	 0.9847	 0.9819
	Salt & Peppers noise (10%)	 0.8993	 0.9676	 0.9245	 0.9342	 0.9599	 0.9743	 0.9704
Girl	Salt & Peppers noise (2%)	 0.9583	 0.9391	 0.8470	 0.8882	 0.9802	 0.9807	 0.9942
	Salt & Peppers noise (5%)	 0.8987	 0.9330	 0.8448	 0.8689	 0.9567	 0.9661	 0.9866
	Salt & Peppers noise (10%)	 0.8250	 0.9221	 0.8344	 0.8353	 0.9241	 0.9307	 0.9791

Fig. 12 Extracted watermarks and NC values of the different methods under the Salt & Peppers noise adding

operations of 200% and 50% are used to deteriorate the watermarked image. From data in Figs. 16 and 17, although the proposed method gives less robustness than the methods of SunSVD (2002), Luo (2020), Hu (2020), and Chen (2021), it is more effective than the schemes of SunQR in Sect. 1.2 and Su (2018). In these experiments, SVD decomposition-based studies such as SunSVD (2002), Luo (2020), and Hu (2020) bring better quality than others. Especially, the proposal of Hu (2020) is the most robust against geometric attacks.

Finally, JPEG compression is also known as an image process. In this experiment, the watermarked images are compressed by JPEG compression with the window size is 8×8 and 16×16 , respectively. JPEG is a common ‘lossy’

compression algorithm for digital images, which allows a selectable trade-off between storage size and image quality by discarding perceptually unimportant information in middle and high frequencies. Pixel alterations in a local area (i.e., the 4×4 blocks used for image watermarking) can be treated as middle-to-high-frequency noise in the 8×8 and 16×16 blocks used for image compression. This makes it easy for JPEG compression to destroy watermark information hidden in the middle-to-high frequency region. Therefore, as the results are shown in Fig. 18, most schemes are weak against JPEG compression, especially the algorithms of SunSVD (2002), and Su (2018). The method of Chen (2021) has the best performance for the “avion” and “Girl” images, while the











































Image	Attack	SunSVD [5]	SunQR in Sect.1.2	Su [29]	Luo [8]	Hu[32]	Chen[31]	The proposed method
avion	Gaussian noise (0,0.001)	 0.9611	 0.9704	 0.9310	 0.9788	 0.9133	 0.9789	 0.9828
	Gaussian noise (0,0.003)	 0.8696	 0.8337	 0.8454	 0.8206	 0.8176	 0.9421	 0.9375
lena	Gaussian noise (0,0.001)	 0.9495	 0.9762	 0.9284	 0.5127	 0.8915	 0.9762	 0.9773
	Gaussian noise (0,0.003)	 0.8795	 0.8166	 0.7930	 0.5061	 0.7749	 0.8940	 0.8853
Girl	Gaussian noise (0,0.001)	 0.9502	 0.9261	 0.7919	 0.8824	 0.9003	 0.9704	 0.9799
	Gaussian noise (0,0.003)	 0.8372	 0.7878	 0.7237	 0.8241	 0.8071	 0.8816	 0.8834

Fig. 13 Extracted watermarks and *NC* values of the different methods under the Gaussian noise adding











































Image	Attack	SunSVD [5]	SunQR in Sect.1.2	Su [29]	Luo [8]	Hu[32]	Chen[31]	The proposed method
avion	Mean Filter (2x2)	 0.8005	 0.8252	 0.5587	 0.6872	 0.8672	 0.8846	 0.8261
	Mean Filter (3x3)	 0.7999	 0.8303	 0.4246	 0.4754	 0.8910	 0.9221	 0.8314
lena	Mean Filter (2x2)	 0.6324	 0.6232	 0.6366	 0.5062	 0.8005	 0.8173	 0.6512
	Mean Filter (3x3)	 0.7693	 0.7687	 0.4971	 0.4930	 0.9127	 0.9406	 0.7946
Girl	Mean Filter (2x2)	 0.7580	 0.7934	 0.6954	 0.7032	 0.8452	 0.9819	 0.8589
	Mean Filter (3x3)	 0.9211	 0.8873	 0.6066	 0.5187	 0.9344	 0.9923	 0.9658

Fig. 14 Extracted watermarks and *NC* values of the different methods under the Mean Filter attack

Image	Attack	SunSVD [5]	SunQR in Sect.1.2	Su [29]	Luo [8]	Hu[32]	Chen[31]	The proposed method
avion	Cropping (25%)							
	Cropping (50%)							
lena	Cropping (25%)							
	Cropping (50%)							
Girl	Cropping (25%)							
	Cropping (50%)							

Fig. 15 Extracted watermarks and *NC* values of the different methods under the Cropping operation

Image	Attack	SunSVD [5]	SunQR in Sect.1.2	Su [29]	Luo [8]	Hu[32]	Chen[31]	The proposed method
avion	Rotation (5°)	 0.7614	 0.6668	 0.7118	 0.8107	 0.9002	 0.7565	 0.6994
	Rotation (10°)	 0.7028	 0.6924	 0.6791	 0.7565	 0.8356	 0.7376	 0.6644
lena	Rotation (5°)	 0.7781	 0.6347	 0.7480	 0.8181	 0.8933	 0.7483	 0.6360
	Rotation (10°)	 0.7524	 0.6039	 0.7138	 0.7661	 0.8279	 0.7120	 0.6046
Girl	Rotation (5°)	 0.8375	 0.6991	 0.7284	 0.7616	 0.8718	 0.8433	 0.7825
	Rotation (10°)	 0.7897	 0.6670	 0.7032	 0.7211	 0.8198	 0.7877	 0.7504

Fig. 16 Extracted watermarks and *NC* values of the different methods under the Rotation attack











































Image	Attack	SunSVD [5]	SunQR in Sect.1.2	Su [29]	Luo [8]	Hu[32]	Chen[31]	The proposed method
avion	Scaling (1/2)	 0.7615	 0.6222	 0.5939	 0.7514	 0.8780	 0.6589	 0.6299
	Scaling (2)	 0.8390	 0.7976	 0.8099	 0.9351	 0.9465	 0.9213	 0.7995
lena	Scaling (1/2)	 0.8306	 0.5244	 0.6717	 0.7508	 0.8871	 0.6569	 0.5257
	Scaling (2)	 0.9429	 0.8038	 0.8627	 0.9582	 0.9631	 0.9347	 0.8073
Girl	Scaling (1/2)	 0.8826	 0.6211	 0.6702	 0.6951	 0.9112	 0.7182	 0.7018
	Scaling (2)	 0.9832	 0.8967	 0.7851	 0.8937	 0.9895	 0.9373	 0.9609

Fig. 17 Extracted watermarks and *NC* values of the different methods under the Scaling attack











































Image	Attack	SunSVD [5]	SunQR in Sect.1.2	Su [29]	Luo [8]	Hu[32]	Chen[31]	The proposed method
avion	JPEG (8x8)	 0.8318	 0.8559	 0.6299	 0.8346	 0.7863	 0.9379	 0.8680
	JPEG (16x16)	 0.8467	 0.8646	 0.6779	 0.8719	 0.8271	 0.9393	 0.8693
lena	JPEG (8x8)	 0.6404	 0.6348	 0.7255	 0.8812	 0.7569	 0.8773	 0.6425
	JPEG (16x16)	 0.6278	 0.6433	 0.7596	 0.9140	 0.7695	 0.8720	 0.6470
Girl	JPEG (8x8)	 0.6295	 0.7268	 0.6970	 0.7810	 0.8018	 0.9443	 0.7739
	JPEG (16x16)	 0.6314	 0.6944	 0.7202	 0.8343	 0.8234	 0.9527	 0.7583

Fig. 18 Extracted watermarks and *NC* values of the different methods under the JPEG compression

Fig. 19 *NC* values of the proposed method for two watermarked images and two watermark images under different attacks

Attacks	The host image is 'milkdrop' and the watermark is 'w1'	The host image is 'Girl' and the watermark is 'w2'
Blurring (0.2)	0.9935	0.9999
Blurring (0.5)	0.9101	0.9999
Sharpening (0.2)	0.9935	0.9999
Sharpening (0.5)	0.8254	0.9990
Salt & Peppers noise (2%)	0.9906	0.9942
Salt & Peppers noise (10%)	0.9689	0.9791
Gaussian noise (0,0.001)	0.9806	0.9799
Gaussian noise (0,0.003)	0.8516	0.8834
Rotation (5°)	0.7658	0.7825
Rotation (10°)	0.7072	0.7504
Scaling (1/2)	0.7153	0.7018
Scaling (2)	0.9378	0.9609
Cropping (25%)	0.9378	0.9609
Cropping (50%)	0.9378	0.9609
JPEG (8x8)	0.8228	0.7739
JPEG (16x16)	0.8472	0.7583
Mean Filter (2x2)	0.8303	0.8589
Mean Filter (3x3)	0.9130	0.9658

approach of Luo (2020) overcomes the others for the “lena” image. In the contrast, the proposal of Hu (2020) has limited resistance to this kind of attack. Although the proposed method is not the best method under JPEG compression, it has higher *NC* values than the methods of SunQR, Su (2018) and Luo (2020) for the “avion” image. Our study is also better than some methods such as SunSVD (2002), SunQR, and Su (2018) for the “lena” and “Girl” images.

To sum up, by the experimental results, we can see that the schemes which use SVD decomposition (SunSVD 2002, Luo (2020), and especially Hu (2020)) are more robust than other methods (SunQR in Sect. 1.2, Su (2018), Chen (2021) and the proposed one) under geometry attacks. However, the method of Lou (2020) is less effective under mean filter and cropping operations. Whereas, the algorithm of Chen (2021) is considered as a strong solution against most attacks, especially the mean filter. Figure 19 also displays *NC* values and

the extracted watermarks of two watermarked images of the proposed method under different attacks. The figures indicate that the proposed approach is robust against the most used attacks such as blurring, sharpening, salt & pepper noise, Gaussian noise, cropping, and mean filter for both watermark images. 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 an improvement to find out elements of Q and R matrices instead of using the GS algorithm as SunQR.

5 Conclusion

In this paper, a novel image watermarking scheme, which is based on QR decomposition, is presented. In the embedding stage, the color host image is divided into non-overlapping 4×4 blocks at first. For each block, the improved QR decomposition is applied on the B channel where calculating Q and R matrices is executed in succession. First, the elements of the R matrix are found by performing a set of linear equations as Eq. (15). Second, the Q matrix is computed column by column based on A and R matrices. Third, the watermark information is embedded into the first element $R(1, 1)$ of the R matrix by using Eq. (23). 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. (24) without QR factorization as the previous methods. After that, the binary values of the watermark are extracted via Eq. (25). As the result, the image of the 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 the effectiveness of the proposed method. The results of the comparison in Sect. 4 show that our scheme overcomes the others in terms 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 (2002), SunQR in Sect. 1.2, Su (2018) and Luo (2020) under attacks such as salt & peppers noise, Gaussian noise, blurring, sharpening, cropping, and mean filter.

In the future, a hybrid scheme should be developed to improve the robustness of the watermark under geometry attacks and JPEG compression. In recent years, many hybrid digital image watermarking methods have been expanded to enhance robustness, capacity, and security while still maintaining the quality of the watermarked image (Mahbuba and Mohammad 2020). In hybrid domain approaches, two or more image transformations are used for watermarking. These methods provide more imperceptibility and

high robustness to multimedia data and are mainly used for multimedia security and copyright protection. Therefore, a combination of the proposed QR decomposition and another transform domain is necessary and will be further considered in the future study.

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

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

References

- Abdulrahman AK, Ozturk S (2019) A novel hybrid DCT and DWT based robust watermarking algorithm for color images. *Multimed Tools Appl* 78:17027–17049. <https://doi.org/10.1007/s11042-018-7085-z>
- Arasteh S, Mahdavi M, Bideh PN, Hosseini S, Chapnevis AA (2018) Security analysis of two key based watermarking schemes based on QR decomposition. In: 26th Iranian conference on electrical engineering (ICEE2018)
- Begum M, Uddin MS (2020) Analysis of digital image watermarking techniques through hybrid method. *Adv Multimedia* 2020:12. <https://doi.org/10.1155/2020/7912690>
- Chen Y, Jia Z-G, Peng Y, Peng Y-X, Zhang D (2021) A new structure-preserving quaternion QR decomposition method for color image blind watermarking. *Signal Process*. <https://doi.org/10.1016/j.sigpro.2021.108088>
- Chen S, Su Q, Wang H et al (2021) A high-efficiency blind watermarking algorithm for double color image using Walsh Hadamard transform. *Vis Comput*. <https://doi.org/10.1007/s00371-021-02277-1>
- Ernawan F, Kabir MN (2020) A block-based RDWT-SVD image watermarking method using human visual system characteristics. *Vis Comput* 36:19–37. <https://doi.org/10.1007/s00371-018-1567-x>
- Giri Kaiser J, Peer Mushtaq Ahmad, Nagabhusan P (2015) A robust color image watermarking scheme using discrete wavelet transformation. *I.J. Image, Graphics and Signal Processing*, pp 47–52
- Hsu L-Y, Hu H-T (2017) Robust blind image watermarking using criss-cross inter-block prediction in the DCT domain. *J Vis Commun Image Represent* 46:33–47
- Hwai-Tsu H, Hsu L-Y, Chou H-H (2020) An improved SVD-based blind color image watermarking algorithm with mixed modulation incorporated. *Inf Sci*. <https://doi.org/10.1016/j.ins.2020.01.019>
- Jia S, Zhou Q, Zhou H (2017) A novel color image watermarking scheme based on DWT and QR decomposition. *J. Appl. Sci. Eng.* 20(2):193–200
- Kumar S, Singh BK (2021) An improved watermarking scheme for color image using alpha blending. *Multimed Tools Appl* 80:13975–13999. <https://doi.org/10.1007/s11042-020-10397-4>
- Lai CC (2011) An improved SVD-based watermarking scheme using human visual characteristic. *Opt Commun* 284:938–944
- 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
- Li J, Lin Q, Yu C et al (2018) A QDCT- and SVD-based color image watermarking scheme using an optimized encrypted binary computer-generated hologram. *Soft Comput* 22:47–65. <https://doi.org/10.1007/s00500-016-2320-x>
- Liu F, Yang H, Su Q (2017) Color image blind watermarking algorithm based on Schur decomposition. *Appl Res Comput* 34:3085–3093
- Luo A, Gong L, Zhou N et al (2020) Adaptive and blind watermarking scheme based on optimal SVD blocks selection. *Multimed Tools Appl* 79:243–261. <https://doi.org/10.1007/s11042-019-08074-2>
- Naderahmadian Y, Hosseini-Khayat S (2014) Fast and robust watermarking in still images based on QR decomposition. *Multimed Tools Appl* 72(3):2597–2618
- Qingtang S, Wang G, Zhang X, Lv G, Chen B (2017) An improved color image watermarking algorithm based on QR decomposition. *Multimed Tools Appl* 76:707–729. <https://doi.org/10.1007/s11042-015-3071-x>
- Qingtang S, Wang G, Zhang X, Lv G, Chen B (2018) A new algorithm of blind color image watermarking based on LU decomposition. *Multidimension Syst Signal Process* 29(3):1055–1074
- Qingtang S, Liu Y, Liu D, Yuan Z, Ning H (2019) A new watermarking scheme for colour image using QR decomposition and ternary coding. *Multimed Tools Appl* 78(7):8113–8132
- Qin L, Ma L, Fu X (2021) DWT-DQFT-based color image blind watermark with QR decomposition. In: Lu W., Sun K., Yung M., Liu F. (eds) Science of cyber security. SciSec 2021. Lecture Notes in Computer Science, vol 13005. Springer, Cham
- Roy S, Pal AK (2019) A Hybrid Domain Color Image Watermarking Based on DWT-SVD. *Iran J Sci Technol Trans Electr Eng* 43:201–217. <https://doi.org/10.1007/s40998-018-0109-x>
- Singh RK, Shaw DK, Sahoo J (2017) A secure and robust block based DWT-SVD image watermarking approach. *J Inf Optim Sci* 38(6):911–925
- Singh KU, Singh VK, Singhal A (2018) Color image watermarking scheme based on QR factorization and DWT with compatibility analysis on different wavelet filters. *J Adv Res Dyn Control Syst* 10(6):1796–1811
- Stewart GW (1998) Matrix algorithms. Volume 1: basic decompositions. The Society for Industrial and Applied Mathematics, ISBN 0-89871-414-1
- Su Q, Chen B (2018) Robust color image watermarking technique in the spatial domain. *Soft Comput* 22:91–106. <https://doi.org/10.1007/s00500-017-2489-7>
- Su Q, Niu Y, Zou H et al (2013) A blind double color image watermarking algorithm based on QR decomposition. *Multimed Tools Appl* 72:987–1009. <https://doi.org/10.1007/s11042-013-1653-z>
- Su Q, Niu Y, Wang G, Jia S, Yue J (2014) Color image blind watermarking scheme based on QR decomposition. *Signal Process* 94:219–235
- Su Q, Wang G, Jia S, Zhang X, Liu Q, Liu LX (2015) Embedding color image watermark in color image based on two-level DCT. *Signal Image Video Process* 9(5):991–1007
- Su Q, Zhang X, Wang G (2020) An improved watermarking algorithm for color image using Schur decomposition. *Soft Comput* 24:445–460. <https://doi.org/10.1007/s00500-019-03924-5>
- Su Q, Wang H, Liu D et al (2020) A combined domain watermarking algorithm of color image. *Multimed Tools Appl* 79:30023–30043. <https://doi.org/10.1007/s11042-020-09436-x>
- Sun R, Sun H, Yao T (2002) A SVD and quantization based semi-fragile watermarking technique for image authentication. *Proc Internat Conf Signal Process* 2:1952–1955
- University of Granada. Computer Vision Group. CVG-UGR image database. [2012-10-22]. <http://decsai.ugr.es/cv/bimimagenesc512.php>
- Vaishnavi D, Subashini TS (2015) Robust and invisible image watermarking in RGB color space using SVD. *Proc Comput Sci* 46:1770–1777
- Vandenberghe L (2018) 6. QR factorization. ECE133A, pp.6-1-6-39
- Wang D, Yang F, Zhang H (2016) Blind color image watermarking based on DWT and LU Decomposition. *J Inf Process Syst* 12(4):765–778
- 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*. 583. Springer, Singapore. <https://doi.org/10.1007/978-981-10-5687-1-3>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.