Two Embedding Algorithms in Schur-Based Image Watermarking Scheme

Anh Le-Thi[1(✉)], Bich Pham-Ngoc[2], and Dung Tran-Tien[1]

[1] Hanoi University of Industry, Hanoi, Vietnam
{leanh,trantd}@haui.edu.vn
[2] FPT College, Hanoi, Vietnam
bichpn@fpt.edu.vn

Abstract. Digital watermarking is a potential technique for copyright protection purposes that has been widely developed in recent years. In image watermarking schemes, the embedding algorithm plays an essential role in the process of embedding and extracting information. This paper presents a Schur-based image watermarking scheme with two different embedding algorithms. The first algorithm embeds the watermark into two components of the orthogonal matrix, while the second algorithm considers the embedding element. To increase the security of the proposed scheme, the watermark image (WMI) is encoded by the Arnold transform before it is embedded into the host image. The achieved outcomes reveal that both algorithms have good performance in terms of invisibility and robustness. Besides that, the two embedding algorithms are analyzed and compared by data and illustrative examples to see the difference between them as well as their advantages and disadvantages in the watermarking scheme based on Schur decomposition (SDC).

Keywords: Image watermarking · Schur decomposition · Transform domain · Embedding algorithm · Copyright protection

1 Introduction

With the rapid development of technology, the requirement of protection for copyright has also become an urgent topic that needs to be addressed. Among existing solutions, digital watermarking is considered an effective tool for image copyright protection. It is essentially a process of creating a binding through embedding and extraction formulas amid the origin picture and the watermark. One requirement is that the watermark must be resistant to common image attacks. Robust image watermarking models are frequently implemented by applying matrix transformations. Besides expansions such as DWT, SVD, QR,

Supported by Hanoi University of Industry.

N. Thi Dieu Linh et al. (Eds.): ADHOCNETS 2023, LNICST 558, pp. 25–40, 2024.
https://doi.org/10.1007/978-3-031-55993-8_3

LU, or Hessenberg, Schur analysis is well suited for use in this area. According to [1], the Schur decomposition has the computation complication as $O(8n^3/3)$ and SVD is $O(11n^3)$. It is obvious that the number of computations required for Schur decomposition is less than one-third of SVD required. In addition, in the paper [2], Mohan compared the performance of Schur and SVD and concluded that Schur decomposition has a lower execution time than SVD decomposition. This relation shows that the Schur decomposition will be more widely applied in digital watermarking. Meanwhile, the Schur vector has good scaling invariance, which can enhance the robustness of the watermarking technique. Therefore, there have been many studies on the application of Schur in the watermarking field in recent years. Image watermarking techniques based on the Schur transform usually focus on two main directions: single domain and hybrid domain. On the single domain, the researchers only embed the watermark through the Schur decomposition [3–12]. On the contrary, it is a combination of Schur factorization with other transformations (such as DWT, DCT, SVD, QHT, $NSCT$) in hybrid domain-based watermarking schemes [13–22].

For single domain-based watermarking schemes, the first example is a proposal by Su in 2017 [6]. This study embeds the watermark on two components $U(2, c)$ and $U(3, c)$ of the matrix U with $c = max(Dij)$ (D is the upper triangular matrix. To improve resistance to cropping attacks, embedded blocks are chosen based on a pseudo-random sequence. Thus, the elements of the WMI as R, G, and B were swapped by the AT with K_a. Authors in [10] proposal give positive watermark image (WMI) quality and durability results and ensure system security. Moreover, in [11], before using the Schur transformation to each block, Li divides the black and white WMIs into 8×8 blocks. Next, the watermark is shuffled by Arnold and Logistic Map before embedding in the element with the most significant energy $D(1, 1)$ of the upper triangular matrix. $PSNR$ values without attack are in the range of 40 dB–47 dB. Their results were found to be better than other comparable studies; however, low-pass filtering issues and resizing attacks are required to solve in the following research.

For hybrid domain-based watermarking schemes, Karajeh and colleagues published a study based on the DWT-Schur association scheme [14] in 2019. In this proposal, the two-level Haar filter transforms the DWT, and then the Schur is employed to $HL2$. The watermark is then embedded in the diagonal elements of the matrix D. In 2021, Prabha proposed a scheme using LWT (Lift Wavelet Transform) and Schur transformations on the B color channel of the original image [15]. LWT is a fast and efficient transformation. The original image is first transformed through the Haar transform, and then Schur is expanded on the sub-band LL. After being swapped by Arnold, the WMI will be embedded in the U matrix. The results show that the $PSNR$ value reaches 41.1241 dB and the NC value is close to 1.

From the above discussions, it is seen that the choice of location to embed information is very important in the process of building a watermarking scheme because it directly affects the stability of the watermark and the quality of the watermarked image. In other words, the embedding formula and the embedding

element largely determine the different performance of watermarking schemes if they use the same matrix transformation. Therefore, in this paper, we propose a blind image watermarking scheme which includes two embedding algorithms. Then, we focus on analyzing their role in Schur-based watermarking schemes. By experimental results and illustrative examples, two embedding algorithms are evaluated and compared in terms of robustness and invisibility. Finally, the advantages and limitations of each algorithm will be presented.

The remainder of the manuscript is organized into four sections: Sect. 2 presents primary Schur decomposition (SDC) and Arnold transform (AT); Sect. 3 shows the image watermarking scheme; The experiments and analysis will be presented in Sect. 4; the final Sect. 5 is conclusion.

2 Preliminary

2.1 Schur Decomposition (SDC)

In SDC, a matrix A of size $n \times n$ is analyzed as the product of three matrices U, D, and U^T as follows.

$$A = UDU^T, \tag{1}$$

here U, D is the orthogonal and upper triangular matrices respectively, and U^T is the transpose matrix of U.

For instance, suppose one matrix A of size 4×4 as follows.

$$A = \begin{bmatrix} 98 & 108 & 108 & 114 \\ 202 & 204 & 203 & 199 \\ 197 & 200 & 199 & 198 \\ 204 & 207 & 207 & 201 \end{bmatrix}$$

By applying Eq. 1, we can reach D and U as below:

$$D = \begin{bmatrix} 713.7100 & 156.8300 & 18.9500 & 3.4027 \\ 0 & -9.8128 & -3.2578 & -5.7126 \\ 0 & 0 & 0.2065 & 0.2476 \\ 0 & 0 & 0 & -2.0990 \end{bmatrix}$$

and

$$U = \begin{bmatrix} -0.2957 & -0.9470 & -0.1168 & 0.0471 \\ -0.5519 & 0.2560 & -0.7644 & -0.2132 \\ -0.5429 & 0.0678 & 0.5824 & -0.6012 \\ -0.5597 & 0.1820 & 0.2506 & 0.7687 \end{bmatrix}$$

As shown in the above example, the biggest values belong to the component $D(1,1)$ of the matrix D, so it is considered to embed the information. Besides, in the matrix U, two components, $U(2,1)$ and $U(3,1)$, have the same sign and are the closest elements. Thus, these two elements are also a good choice.

2.2 Arnold Transform (AT)

AT is applied to swap pixels of the original WMI before embedding with the purpose of improving the secrecy watermarking approach. The change of location of a pixel from (x, y) to another location (s, t) [23] is performed by this method, in Eq. (2).

$$\begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (mod\ N) \tag{2}$$

here, N is a factor of size $N \times N$ of the watermark picture.

Then, using an inversion of the AT in Eq. (3) to recover the original image.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix} (mod\ N) \tag{3}$$

In the process of permuting the pixels, the parameter determines the AT's security level. This parameter will not be the same for different image sizes and depends on the number of repetitions. In this paper, the number of repetitions is 24 because the WMI size is 32×32. Therefore, the parameter is a number between 1 and 24.

3 The Proposed Watermarking Scheme

A Schur-based watermarking (SBW) scheme with two different embedding algorithms will be introduced in this part. The first algorithm performs embedding the watermark into two components $U(2, 1)$ and $U(3, 1)$ of the orthogonal matrix U. The first algorithm uses these two components because they are the closest [8]. The second algorithm selects the first component $D(1, 1)$ of D to imbed information because of its energy concentration [20]. To be fair, both algorithms are implemented on the same color channel and the same size of the block. The scheme consists of two main processes, namely the embedding phase and the extracting phase, as follows.

3.1 Embedding Phase

The embedding procedure of the image watermarking is presented as follows.

- First step:
  - AT technique is applied to permute the WMI.
  - Then, converting the permuted image into a one-dimensional binary array w_i.
- Second step
  - R, G and B is created by dividing the host image.
  - Next, B is split into 4×4 non-overlapping blocks.
- Third step:
  - Each block is assigned to a matrix A.

- SDC is executed on the matrix A of each block by Eq. 1 to acquire the matrices U and D.

– Fourth step:
- Embedding a watermark bit into the matrices U or D based on two algorithms as follows.
  * (a) Algorithm 1:
    If $(w_i = 0)$ and $(|U(2,1)| > |U(3,1)|)$, then

$$\begin{cases} U'(2,1) = sign(U(2,1) * (U_{avg} - T/2) \\ U'(3,1) = sign(U(3,1) * (U_{avg} + T/2) \end{cases} \tag{4}$$

    If $(w_i = 1)$ and $(|U(2,1)| \leq |U(3,1)|)$, then

$$\begin{cases} U'(2,1) = sign(U(2,1) * (U_{avg} + T/2) \\ U'(3,1) = sign(U(3,1) * (U_{avg} - T/2) \end{cases} \tag{5}$$

    where sign(x) represents the sign of x, $U_{avg} = (|U(2,1)| + |U(3,1)|)/2$, $|x|$ means the absolute value of x, and T is one threshold value.
  * (b) Algorithm 2:
    If $(w_i = 0)$, then

$$D'(1,1) = \begin{cases} D(1,1) - \lambda + \frac{1}{4}\sigma, \lambda \in [0, \frac{3}{4}\sigma) \\ D(1,1) - \lambda + \frac{5}{4}\sigma, \lambda \in [\frac{3}{4}\sigma, \sigma) \end{cases} \tag{6}$$

    If $(w_i = 1)$, then

$$D'(1,1) = \begin{cases} D(1,1) - \lambda - \frac{1}{4}\sigma, \lambda \in [0, \frac{1}{4}\sigma) \\ D(1,1) - \lambda + \frac{3}{4}\sigma, \lambda \in [\frac{1}{4}\sigma, \frac{3}{4}\sigma) \end{cases} \tag{7}$$

    where σ indicates quantification step and $\lambda = mod(D(1,1), \sigma)$

– Fifth step:
- Inverse SDC and repeating from the thrid step to the fifth step until all blocks are embedded.

– Sixth step:
- Rebuild watermarked elements to get the watermarked image.

3.2 Extracting Phase

Figure 1 shows the extraction procedure. In our image watermarking scheme, both the original image and the watermark do not need to be considered in the watermark extraction phase, so this is a blind watermarking scheme. The procedure of the extracting stage is presented as follows.

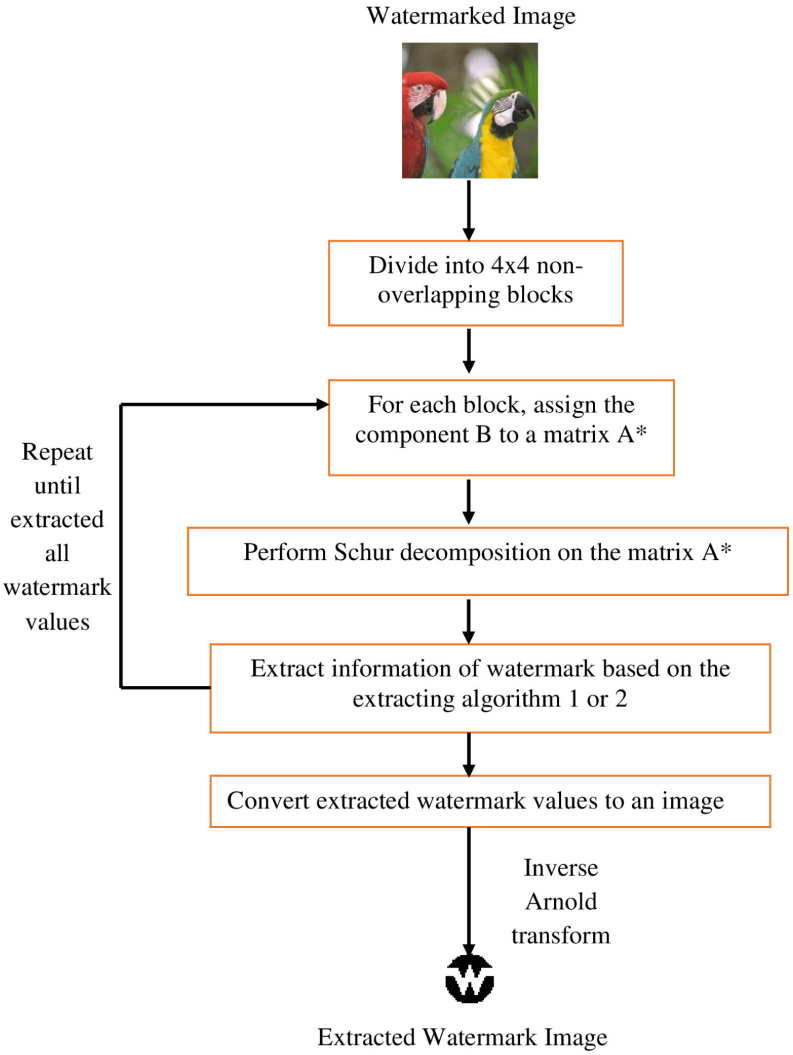

Fig. 1. Extraction process.

– First step:
  - R, G, and B is created by dividing the watermarked image.
  - Next, B is split into 4×4 non-overlapping blocks.
– Second step:
  - Each block is assigned to a matrix A^*.
  - SDC is applied on the matrix A^* of each block by Eq. (1) to acquire U^* and D^*
– Third step:
  - Extracting the information bit of watermark has been established on Algorithm 1 and Algorithm 2:

* (a) Algorithm 1:

$$w_i^* = \begin{cases} \text{“0”} & , \quad U^*(2,1) \leq U^*(3,1) \\ \text{“1”} & , \quad \text{elsewhere} \end{cases} \tag{8}$$

* (b) Algorithm 2:

$$w_i^* = \begin{cases} \text{“0”} & , \quad \lambda^* < \frac{1}{2}\sigma \\ \text{“1”} & , \quad \text{elsewhere} \end{cases} \tag{9}$$

where $\lambda^* = mod(D^*(1,1), \sigma)$

- Repeating both steps 2 and 3 until all watermark values.

– Forth step:
- The extracted watermark values are converted to an image.
- Getting the final extracted WMI by using Inverse AT.

4 Experimental Analysis

In this section, a comparison of Algorithm 1 and Algorithm 2 will be presented based on the results of invisibility tests and robustness tests. Previously, the simulation setting, which includes evaluation criteria, image data, and parameters setting, will be introduced.

4.1 Simulation Setting

Evaluation Criteria. To measure the implementation quality of our watermarking scheme, we use the peak signal-to-noise ratio ($PSNR$) and normalized correlation (NC). $PSNR$ is determined in Eq. 10:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}, \tag{10}$$

here, MSE (mean square error) is formulated as follows

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (H(i,j) - H'(i,j))^2 \tag{11}$$

NC is applied to evaluate the resemblance between the original and extracted watermarks. It is constantly not greater than one and can be computed by Eq. (12):

$$NC = \frac{\sum_{j=1}^{4} \sum_{x=1}^{m} \sum_{y=1}^{n} (W(x,y,j)W'(x,y,j))}{\sqrt{\sum_{j=1}^{4} \sum_{x=1}^{m} \sum_{y=1}^{n} (W(x,y,j))^2} \sqrt{\sum_{j=1}^{4} \sum_{x=1}^{m} \sum_{y=1}^{n} (W'(x,y,j))^2}}, \tag{12}$$

here $W(x,y,j)$ and $W'(x,y,j)$ represent the value of pixel (x,y) in element j of the original and extracted watermarks, respectively. And, $m \times n$ is the watermark size.

Fig. 2. 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.

Image Data and Parameters. To evaluate the performance of the watermarking model, twelve color images with sizes of 512×512 and 256×256 from [24] are utilized as host images. The original watermarks are displayed in Fig. 2 with characteristics: two gray-scale and size of 32×32.

In embedding algorithms, selecting a suitable parameter value is extremely necessary. To balance between robustness and imperceptibility, the threshold T of Algorithm 1 is estimated to be 0.04, and the quantification step σ of Algorithm 2 is set to 70 in our experiments.

Besides, to investigate the efficiency of the two proposed algorithms, we perform the comparison with two other approaches that were published in 2022, namely Soualmi [12] and Sun [22]. While Soualmi [12] *et al.* examined a blind image watermarking scheme that has been established on Schur decomposition and chaotic sequence (CS).

4.2 Invisibility Experiment

In this subsection, the watermarks w1 and w2 are embedded into twelve color images to estimate the invisibility of the image after embedding. Then, the $PSNR$ values are calculated based on Eq. 10 to give a comparison of four algorithms as shown in Table 1. The results in this table indicate that the values of $PSNR$ of Soualmi [12] *et al.* are higher than other algorithms for both watermarks. The average PSNR value of this algorithm is larger than 60dB for all tested images. A reason for this result is that Soualmi *et al.* only hides the watermark in one element D(2, 1) of D by adding 1 to this element for all blocks. Thus, the distance between the pixel value after embedding and the original value is not large. Meanwhile, the algorithm of Sun [22] *et al.* gives the lowest result among the four algorithms since the authors embedded information in the whole first column of Q. Therefore, all elements of the pixel matrix after embedding are modified, leading to a reduction in the watermark's quality.

In addition, this table displays that the values of $PSNR$ in Algorithm 1 are greater than that of Algorithm 2 except in the scenarios of the "baboon" picture. This shows that Algorithm 1 gives better quality of the watermarked images than Algorithm 2. $PSNR$ values are in the range of 50.4527 dB and 58.1620 dB when Algorithm 1 (Alg1) is applied, while they are always around 51 dB when Algorithm 2 (Alg2) is used. These results can be explained as follows.

Table 1. $PSNR$ values of the two algorithms without attacks.

Image	w1				w2			
	Soualmi [12]	Sun [22]	Alg1	Alg2	Soualmi [12]	Sun [22]	Alg1	Alg2
Girl	62.5318	38.2071	58.1143	51.6357	62.1344	38.9001	58.1620	51.5537
avion	63.1120	39.4210	52.9759	51.5691	63.5017	39.1266	53.0076	51.4619
baboon	62.7699	38.7309	50.5138	51.5093	62.4586	38.8233	50.4527	51.4408
House	61.3142	36.5586	51.9829	51.6655	61.7209	36.4508	52.2222	51.4720
milkdrop	64.0081	40.1102	55.8592	51.4481	63.8999	40.2160	56.0307	51.3932
peppers	64.2173	38.9233	55.1237	51.5927	64.3216	38.7567	54.8419	51.5912
sailboat	62.5605	37.1538	52.2410	51.6451	63.0025	37.3214	52.1667	51.4180
lena	61.7249	38.2170	53.9790	51.4801	61.6510	38.0571	54.1249	51.4239
Balloon	64.1126	40.1256	54.1353	51.5559	64.3021	40.4012	54.0019	51.4442
couple	65.0410	38.5241	57.1718	51.8779	64.9712	38.6318	56.7774	51.7497
Parrots	61.2536	38.3096	53.6720	51.6619	61.3324	38.4177	54.0833	51.6928
tree	62.9002	37.8993	52.1963	51.3993	63.1806	37.9054	52.0995	51.3839

First, although Algorithm 1 modifies the value of $U(2,1)$ and $U(3,1)$ of the matrix U by Eq. 4 and Eq. 5,only the middle two rows are changed in the achieved matrix after embedding. Meanwhile, Algorithm 2 only uses one element $D(1,1)$ of the matrix D for embedding, but it changes the value of all the elements of the resulting matrix A'. Figure 3 is an example to illustrate this argument. In this instance, A is the 6^{th} block of "baboon". It is easy to see that after the achieved elements are converted to unsigned 8-bit pixel values, the matrix A'_{int} of Algorithm 2 has more difference from Algorithm 1 compared to the matrix.

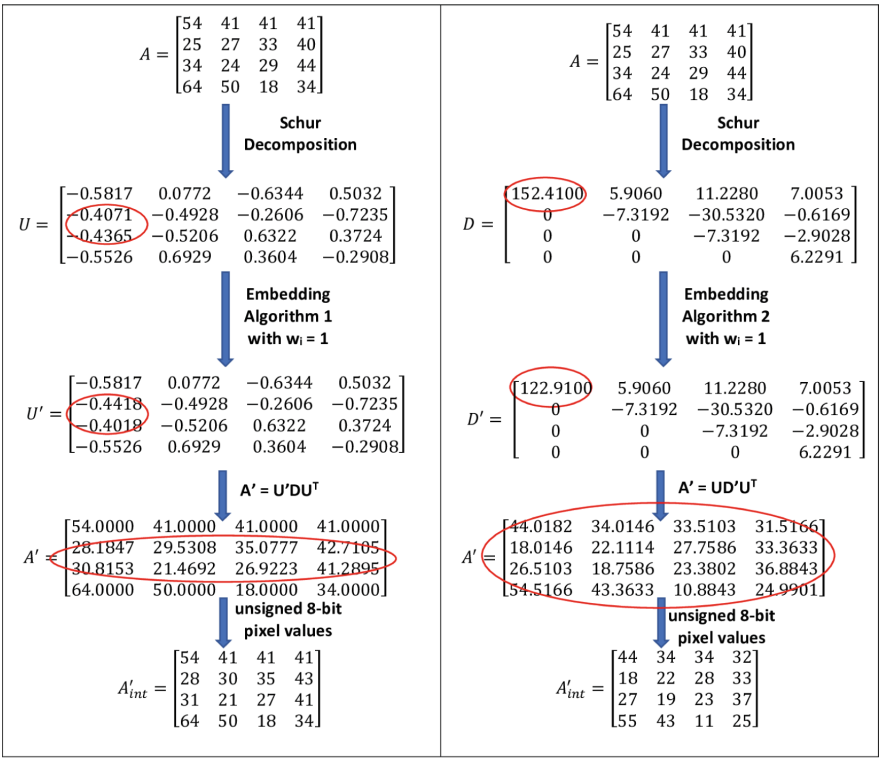

Fig. 3. An example indicates the difference between the Algorithm 1 and the Algorithm 2 regarding invisibility.

Table 2. The rate of unchanged blocks of the two algorithms.

Image	Size of image	Algorithm 1		Algorithm 2	
		Number of unchanged blocks	Percentage (%)	Number of unchanged blocks	Percentage (%)
Girl	512 × 512	8176	49.90	2623	16.00
avion	512 × 512	8209	50.10	2859	17.45
baboon	512 × 512	8213	50.13	2712	16.55
House	512 × 512	8113	49.52	3134	19.13
milkdrop	512 × 512	8135	49.65	2426	14.81
peppers	512 × 512	7992	48.78	2446	14.93
sailboat	512 × 512	8091	49.38	3026	18.47
lena	512 × 512	8019	48.94	2620	15.99
Balloon	256 × 256	2059	50.27	743	18.14
couple	256 × 256	2064	50.39	669	16.33
Parrots	256 × 256	2065	50.41	668	16.31
tree	256 × 256	2016	49.22	519	12.67

Second, the rate of change of pixel values after embedding from the original value affects the quality of the watermarked picture. Digital watermarking is an approach to create the binding between the host image and the watermark through embedding and extraction formulas. Each watermark bit of the host image will be embedded in a block and extracted on the same block in the extraction phase. However, blocks are not always changed through embedding. For example, if $w_i = 0$ and a block has $|U(2,1)| \leq |U(3,1)|$, they do not satisfy the conditions of Eq. 4 and Eq. 5. Therefore, there is not any change in this case, and $U'(2,1) = U(2,1)$ and $U'(3,1) = U(3,1)$. It means that pixel values of this block after embedding are the same as the original values. The watermark bit in the extraction phase will be performed via Eq. 8, and because of $|U^*(2,1)| \leq |U^*(3,1)|$, $w_i^* = 0$. In other words, the watermark bit is still exactly extracted in this case. Table 2 counts some unchanged blocks of twelve host images when applying Algorithm 1 and Algorithm 2, and the watermark is w1. For 512×512 images, the total number of blocks is 16384 blocks, and 4096 blocks for 256×256 images. Accordingly, the rate of unchanged blocks is calculated. As displayed in Table 2, the rate of unchanged blocks of Algorithm 1 is much larger than that of Algorithm 2. Thus, the $PSNR$ values of Algorithm 1 are bigger than those of Algorithm 2.

4.3 Robustness Experiment

To compare the efficiency of the two algorithms regarding robustness, nine common types of attacks are applied to watermarked images. These attacks consist of Gaussian noise, salt & pepper noise, blurring, sharpening, rotation, resizing, cropping, mean filter, and $JPEG$ compression. After that, the watermarks are extracted based on Eq. 8 and Eq. 9. Moreover, we have calculated NC values of two previous algorithms, namely Soualmi [12], and Sun [22] under these attacks to assess the superiority of the proposed method. NC values are calculated via Eq. 12.

Figure 4 displays the results of four algorithms when the watermarked images are "Balloon" and "Milkdrop" in term of robustness. As shown in Fig. 4, the algorithm of Soualmi [12] *et al.* is less robust against most attacks. Extracted watermarks can not be recognized under many attacks, such as Gaussian noise, rotation, scaling, and mean filter. The reason for this result is that Soualmi [12] *et al.* did not apply any parameter in the embedding formula. Therefore, this algorithm can not ensure a balance of invisibility and robustness. As a result, although this algorithm has image's good quality and high embedding capacity, it is weak against common image attacks.

Figure 4 also indicates that the algorithm of Sun [22] *et al.* has good resistance to most attacks. In [22], the quaternion combines the split color channels, so the correlation between the color channels and the synchronization of watermark embedding is not lost. After that, the watermark is embedded and extracted blindly by modifying the relative relationship between multiple coefficient pairs. Therefore, although the algorithm of Sun [22] *et al.* can meet the robustness requirement, it has low image quality as analyzed in Sect. 4.2.

In addition, Fig. 4 shows that Algorithm 2 has the best robustness in comparison with three other algorithms. Algorithm 1 and Algorithm 2 can also extract information efficiently because the watermarks can be recognized in most cases. NC values of Algorithm 2 are bigger than those of Algorithm 1 and the algorithm of Sun [22], except for adding salt & pepper noise. It means that Algorithm 2 is more robust against attacks than Algorithm 1. In particular, Algorithm 2 is quite stable under attacks such as Gaussian noise and $JPEG$ compression. One of the reasons is that the constraints of Algorithm 1 are easier to change than those of Algorithm 2. In other words, the relationship between $U(2,1)$ and $U(3,1)$ is easily broken under the impact of attacks. This can be explained by an example in Fig. 5 as follows.

In Fig. 5, matrix A is the 3^{rd} block of the original image "Balloon", and the watermark bit $w_i = 1$. In the embedding stage, after decomposing the matrix A by Schur decomposition, we receive $U(2,1) = 0.4977$, $U(3,1) = 0.4968$, $D(1,1) = 585.79$, and so $\lambda = mod(D(1,1), \sigma) = 25$. Because $|U(2,1)| > |U(3,1)|$, by using Eq. 5 of the algorithm 1, we have $U'(2,1) = U(2,1) = 0.4977$ and $U'(3,1) = U(3,1) = 0.4968$. Hence, the matrix A'_{int} keeps the same values as matrix A. Meanwhile, after applying Eq. 7 of Algorithm 2, we find out $D'(1,1) = 613.29$ and $\lambda' = 53$. As a result, the A'_{int} is obtained as in the figure. Under the attack "Gaussian noise 0.003", the matrix A^* of the 3^{rd} block of the watermarked image is modified on some elements for both algorithms, highlighted in red. For Algorithm 1, we get $U^*(2,1) = 0.4908$ and $U^*(3,1) = 0.4967$, so $|U(2,1)| < |U(3,1)|$. It means that the relationship of $U(2,1)$ and $U(3,1)$ have been changed. Based on Eq. 8, we have $w_i^* = 0$. Therefore, this result does not match the original bit w_i. On the contrary, Algorithm 2 gives $w_i^* = w_i = 1$ since $\lambda^*(= 50) > 0.5\sigma(= 35)$ according to Eq. 9.

Attack	Parameter	Watermarked image is "Balloon"				Watermarked image is "milkdrop"			
		Soualmi [12]	Sun [22]	Algorithm 1	Algorithm 2	Soualmi [12]	Sun [22]	Algorithm 1	Algorithm 2
Gaussian noise	0.001	 0.7452	0.9547	0.9489	**0.9964**	 0.7510	 0.9650	 0.9707	 **0.9971**
	0.003	 0.6571	0.9068	0.9137	**0.9906**	 0.6892	 0.9219	 0.9339	 **0.9971**
Salt & Pepper noise	0.005	 0.9508	**0.9884**	0.9834	0.9084	 0.9489	 **0.9707**	 0.9534	 0.9060
	0.01	 0.8812	**0.9669**	0.9499	0.8148	 0.8765	 **0.9679**	 0.9359	 0.8437
JPEG	8x8	 0.8234	0.9547	0.8564	**0.9986**	 0.8009	 0.9441	 0.8106	 **0.9990**
	16x16	0.8376	0.9823	0.8509	**0.9986**	0.8289	0.9842	0.8445	**0.9990**
Cropping	25%								
	50%								
Blurring	0.2	0.9120	0.9804	**0.9993**	**0.9993**	0.9289	0.9902	**0.9990**	**0.9990**
	0.5	0.8773	0.8996	0.9630	**0.9935**	0.8871	0.9029	0.8337	**0.9895**
Sharpening	0.2	0.9850	0.9945	**0.9993**	**0.9993**	0.9762	0.9931	**0.9990**	**0.9990**
	0.5	0.9503	0.9826	0.9819	**0.9964**	0.9318	0.9835	0.9551	**0.9971**
Rotation	5°	0.7002	0.9324	0.9237	**0.9509**	0.7136	0.9110	0.8925	**0.9001**
	10°	0.6786	0.8871	0.8985	**0.9257**	0.6872	0.8568	0.8462	**0.8467**
Scaling	1/2	0.7809	0.9532	0.8772	**0.9680**	0.7823	0.9420	0.8525	**0.9454**
	2	0.8315	0.9917	0.9697	**0.9978**	0.8440	0.9932	0.9338	**0.9962**
Mean Filter	2x2	0.8663	0.8995	0.8820	**0.9384**	0.7582	0.8356	0.7989	**0.8448**
	3x3	0.8217	0.9213	0.9406	**0.9673**	0.8188	0.8872	0.9209	**0.9416**

Fig. 4. The extracted watermarks and NC values of four algorithms under nine different types of attacks.

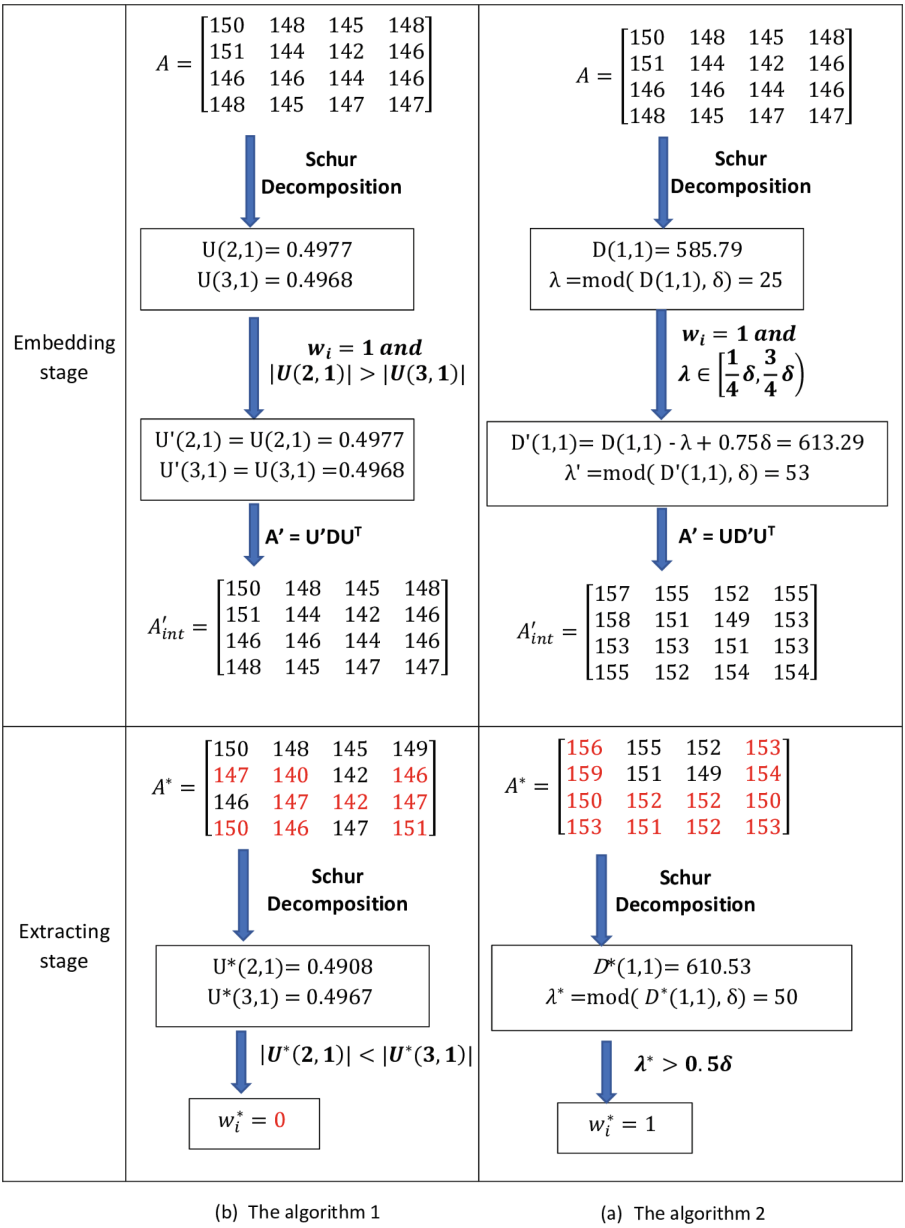

Fig. 5. A comparison of the instability between the Algorithm 1 and Algorithm 2.

5 Conclusion

This paper introduced a blind image watermarking scheme that has been established on Schur decomposition with two different embedding algorithms. In the first algorithm, the watermark was embedded into $U(2,1)$ and $U(3,1)$ of matrix U by modifying their relation. Meanwhile, the second algorithm selects a suitable position to embed in matrix D that is the first element $D(1,1)$. In the experiments, both algorithms are applied on the same condition of the host images, the watermarks, and the size of the blocks. The results have pointed out that in the case of evaluation of the watermarked image's quality, Algorithm 1 performs better than Algorithm 2; however, Algorithm 1 is less effective in the aspect of robustness. Besides, in comparison with previous studies, the achieved results indicated that our proposed algorithms can better balance the watermarked image's quality and the extracted watermark's robustness.

Acknowledgment. This research is funded by the Hanoi University of Industry, and thanks to Nha Phuong-Thi, who supported us during the process.

References

1. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press, Baltimore (2013)
2. Mohan, B.C., Swamy, K.V., Kumar, S.S.: A Comparative performance evaluation of SVD and Schur Decompositions for Image Watermarking. In: IJCA Proceedings on International Conference on VLSI, Communications and Instrumentation (ICVCI), vol. 14, pp. 25–29 (2011)
3. Su, Q., Niu, Y., Liu, X., Zhu, Y.: Embedding color watermarks in color images based on Schur decomposition. Opt. Commun. **285**(7), 1792–1802 (2012)
4. Mohammad, A.A.: A new digital image watermarking scheme based on Schur decomposition. Multimed. Tools Appl. **59**, 851–883 (2012). https://doi.org/10.1007/s11042-011-0772-7
5. Liu, F., Yang, H., Su, Q.: Color image blind watermarking algorithm based on Schur decomposition. Appl. Res. Comput. **34**(10), 3085–3089 (2017)
6. Su, Q., Chen, B.: An improved color image watermarking scheme based on Schur decomposition. Multimed. Tools Appl. **76**, 24221–24249 (2017). https://doi.org/10.1007/s11042-016-4164-x
7. Su, Q., Zhang, X., Wang, G.: An improved watermarking algorithm for color image using Schur decomposition. Soft Comput. **24**(1), 445–460 (2020). https://doi.org/10.1007/s00500-019-03924-5
8. Su, Q., Su, L., Wang, G., Li, L., Ning, J.: A novel colour image watermarking scheme based on Schur decomposition. Int. J. Embed. Syst. **12**(1), 31–38 (2020)
9. Hsu, L.Y., Hu, H.T.: A reinforced blind color image watermarking scheme based on Schur decomposition. IEEE Access **7**, 107438–107452 (2019). https://doi.org/10.1109/ACCESS.2019.2932077
10. Liu, D., Yuan, Z., Su, Q.: A blind color image watermarking scheme with variable steps based on Schur decomposition. Multimed. Tools Appl. **79**, 7491–7513 (2020)
11. Liu, D., Su, Q., Yuan, Z., Zhang, X.: A color watermarking scheme in frequency domain based on quaternary coding. Vis. Comput. **37**, 2355–2368 (2021)

12. Soualmi, A., Alti, A., Laouamer, L.: A novel blind medical image watermarking scheme based on Schur triangulation and chaotic sequence. Concurr. Comput.: Pract. Exp. **34**(1), 6480 (2022)
13. Li, J., Yu, C., Gupta, B.B., Ren, X.: Color image watermarking scheme based on quaternion Hadamard transform and Schur decomposition. Multimed. Tools Appl. **77**, 4545–4561 (2018). https://doi.org/10.1007/s11042-017-4452-0
14. Karajeh, H., Khatib, T., Rajab, L., et al.: A robust digital image watermarking scheme based on DWT and Schur decomposition. Multimed. Tools Appl. **78**, 18395–18418 (2019). https://doi.org/10.1007/s11042-019-7214-3
15. Prabha, K., Shatheesh Sam, I.: Lifting scheme and Schur decomposition based robust watermarking for copyright protection. In: Sheth, A., Sinhal, A., Shrivastava, A., Pandey, A.K. (eds.) Intelligent Systems. AIS, pp. 143–151. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-2248-9_15
16. Ye, G., Pan, C., Dong, Y., Jiao, K., Huang, X.: A novel multi-image visually meaningful encryption algorithm based on compressive sensing and Schur decomposition. Trans. Emerg. Telecommun. Technol. **32**(2), e4071 (2021)
17. Barouqa, H., Al-Haj, A.: Watermarking E-government document images using the discrete wavelets transform and schur decomposition. In: 2021 7th International Conference on Information Management (ICIM), pp. 102–106. IEEE (2021)
18. Abdullah, M.J.: The Trade-off between robustness and impercepbility performance of watermarking technique with DWT and Schur decomposition for medical images. J. Theor. Appl. Inf. Technol. **100**(1) (2022). ISSN 1992-8645
19. Marjuni, A., Nurhayati, O.D.: Robustness improvement against a non-geometrical attacks of lifting scheme-based image watermarking through singular value and Schur decompositions. Int. J. Intell. Eng. Syst. **14** (2021)
20. Li, J.Y., Zhang, C.Z.: Blind watermarking scheme based on Schur decomposition and non-subsampled contourlet transform. Multimed. Tools Appl. **79**, 30007–30021 (2020). https://doi.org/10.1007/s11042-020-09389-1
21. Hu, F., Cao, H., Chen, S., Sun, Y., Su, Q.: A robust and secure blind color image watermarking scheme based on contourlet transform and Schur decomposition. Vis. Comput. 1–20 (2022). https://doi.org/10.1007/s00371-022-02610-2
22. Sun, Y., Su, Q., Chen, S., Zhang, X.: A double-color image watermarking algorithm based on quaternion Schur decomposition. Optik **269**, 169899 (2022)
23. Satish, A., Prasad, E.V., Tejasvi, R., Swapna, P., Vijayarajan, R.: Image scrambling through two level Arnold transform. In: Alliance International Conference on Artificial Intelligence and Machine Learning (AICAAM) (2019)
24. University of Granada, Computer Vision Group, CVG-UGR Image Database. https://decsai.ugr.es/cvg/dbimagenes/c512.php. Accessed December 2022