International Journal of Mathematics and
Computer Science, **17**(2022), no. 2, 753–774

Combining Algebraic GSVD and Gravitational Search Algorithm for Optimizing Secret Image Watermark Sharing Technique

Areej M. Abduldaim, Anwar Khaleel Faraj

Mathematics and Computer Applications
Applied Sciences Department
University of Technology
Baghdad 10001, Iraq

email: 100044@uotechnology.edu.iq, 100290@uotechnology.edu.iq

(Received October 15, 2021, Accepted November 26, 2021)

Abstract

Linear algebra represents one of the main adequate tools for watermarking applications. Therefore, in this paper, we propose a new application of linear algebra to design an optimized secret image watermark sharing algorithm based on the Generalized Singular Value Decomposition (GSVD) method. The characteristic of this decomposition method is factorizing the matrix into five new matrices. Two of these matrices are nonnegative diagonal and the result of dividing them represents the important information of the image called the generalized singular values. These values are utilized to find the features of the original image that will be XORed with the binary watermark such that the embedded watermark is obtained itself after the extraction process. The experimental results display that the proposed optimized secret image watermark sharing algorithm is successfully resisted lots of attacks, like Salt and Pepper noise, Gaussian noise, JPEG compression, Speckle noise, Motion filter, Median filter, and Histogram equalization.

Key words and phrases: Generalized Singular Value Decomposition (GSVD), Secret Watermark Sharing, Binary Gravitational Search Algorithm (BGSA).
AMS (MOS) Subject Classifications: 15A99, 94A08, 15-xx.
ISSN 1814-0432, 2022, http://ijmcs.future-in-tech.net

1 Introduction

Linear algebra represents one of the mathematics sections that is extensively exploited in several sciences and engineering fields [1]. It deals with vectors, matrices, and linear transforms, among others. Therefore, it represents the essential concept in the image processing area. Digital watermarking takes the advantage of utilizing this mathematics section [2].
Secret sharing is a specific method of cryptography for safeguarding a secret via separating it into distinct parts named shadows or shares. The adequate parts are capable of efficiently revealing the secret. However, the inadequate ones are not [3]. The secret sharing concept was extended from the domain of numbers to the domain of images. This led to the scheme of secret image sharing in which the binary secret image is converted into several images (shares) of random-looking to be distributed to several participants [4, 5]. The transmitting of these generated noise-like shares over the internet makes them suspicious to invaders and can possibly be attacked. Therefore, the techniques of watermarking based on secret image sharing were developed to yield zero-watermarking algorithms to avoid threats [6]. The algorithms of zero-watermarking extract the essential features of original images without modification. Consequently, these algorithms are capable of addressing the inconsistency between robustness and imperceptibility of the conventional watermarking algorithms [7].
There are many schemes of linear algebra-based zero-watermarking (secret watermark sharing) algorithms that have been proposed in the past few years. Mohammed [8] introduced a new linear algebra application depending on Hessenberg Decomposition Algebraic Transform to optimize a zero watermarking scheme utilizing a Genetic algorithm. Waleed et al. [9] presented an optimized secret watermark sharing algorithm based on Singular Value Decomposition (SVD) in which the best blocks required for finding the best features from the color images are found using the Shuffled Frog Leaping Optimization Algorithm. This proposed algorithm has efficiently maximized the watermark robustness under several attacks. Kumar et al. [10] presented a zero watermarking algorithm for the digital biometric images which exploited the Low-frequency approximation of Discrete Wavelet Transform (DWT) to obtain important blocks. Then the SVD was enforced on each block to obtain important features from the palm-print images. By the uniqueness of palm-print attributes, the data duplicity was eliminated via this algorithm. In the experiment, the average value of NCC over several attacks was more than 0.8. Kang et al. [11] proposed a Polar Harmonic Transform (PHTs)

based distinguishable and robust secret watermark sharing algorithm where three moments of PHTs for the original color image are calculated at once and then the robust moments are chosen. After that, a binary feature sequence based on content is obtained via determining the relation between the adjacent moments' magnitudes. The acquired results demonstrate that the presented algorithm holds good discriminability and equalization, and resisting scaling and rotation attacks. However, the robustness requires to be improved withstanding Gaussian noise, JPEG compression, and median filter attacks. Zhiqiu Xia et al. [12], proposed a triple zero-watermarking algorithm in which the three forms of decimal-order polar harmonic transforms are utilized for constructing triple zero-watermarks. The obtained results demonstrated that the presented algorithm is highly robust against geometric attacks.

In this work, we adopt an algebraic decomposition method to design an optimized secret watermark sharing algorithm using algebraic transformation (GSVD). The organization of this paper is as follows: In section 2, we deal with preliminaries. In section 3, we state the proposed optimized secret watermark sharing in detail. In section 4, we describe the experiments. In the final section, we draw some conclusion and suggest some recommendations.

2 Preliminaries

In this section, we give basic information needed in this paper.

2.1 Generalized Singular Value Decomposition (GSVD)

Linear algebra treats any type of array of non-negative scalar entries. Therefore, any image matrix can be dealt with on this basis. The Singular Value Decomposition (SVD) is an effective numerical analysis tool used to analyze matrices in the field of linear algebra. The algebraic transformation SVD factorizes any matrix into the product of three matrices.

$$A = USV^T, \tag{2.1}$$

where the matrices U, V, and S are all in $R^{n\times n}$, U and V are both orthogonal and S is diagonal:

$$\mathbf{S} = \begin{pmatrix} \sigma & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n \end{pmatrix} \tag{2.2}$$

Here the diagonal elements σ' s are singular values and satisfy

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq \sigma_{r+1} \geq \sigma_n = 0 \; ...$$

The singular values $\sigma_1, \sigma_2, ...\sigma_n$ are unique but the matrices U and V are not unique. The SVD method is an optimal matrix decomposition technique in a least-square sense that it packs the maximum signal energy into as few coefficients as possible. It can adapt to the variations in local statistics of an image. Figure 1 illustrates 512×512 Baboon image after applying the SVD immediately [13, 14].

The Generalized Singular Value Decomposition (GSVD) is a matrix decomposition that is more general than the singular value decomposition. When we apply the GSVD on two matrices A and B, the yield consists of five matrices: U and V which are both unitary matrices, two diagonal matrices with nonnegative values C and S, and one square matrix X. Thus, defining the GSVD as

$$GSVD[A, B] = [U, V, X, C, S], \tag{2.3}$$

the original matrices A and B can be recovered from these matrices as follows:

$$A = U * C * X^T \text{ and } B = V * S * X^T \tag{2.4}$$

With the diagonal matrix of singular values:

$$\Sigma = C * S^{-1} \tag{2.5}$$

such that,

$$C^T * C + S^T * S = I \tag{2.6}$$

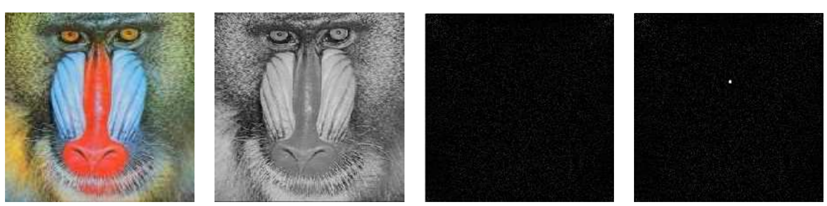
Note that $[U, V, X, C, S] = gsvd(A, B)$ returns unitary matrices U and V, a (usually) square matrix X, and nonnegative diagonal matrices C and S. When B is square and nonsingular, the generalized singular values, $gsvd(A, B)$, are equal to the ordinary singular values, $svd(A/B)$, but they are sorted in the opposite order. Their reciprocals are $gsvd(B, A)$ [15].

The SVD and GSVD have diverse applications that involve areas such as signal processing, numerical computation, and statistics. Figure 2 below illustrates 512×512 Baboon image after applying the GSVD immediately.

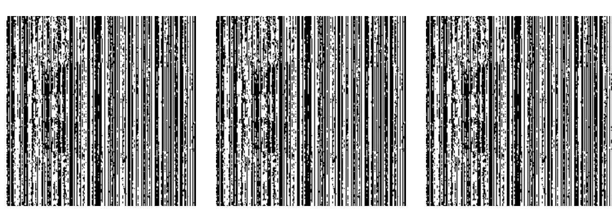
Figure 3 shows the intensity values of parts of the original Baboon image matrix after being addressed by GSVD.

(a) Color Image (b) Gray Image (c) U (d) S (e) V^T

Figure 1: The Effect of the SVD on Baboon Image



(a) Color Image (b) Gray Image (c) U (d) V



(e) X (f) C (g) S

Figure 2: The Effect of the GSVD on Baboon Image

	1	2	3	4	5	6	7	8	9	10
1	99	65	91	55	69	97	109	89	150	80
2	99	68	74	57	87	62	34	35	100	82
3	99	101	70	60	84	78	91	35	70	77
4	51	123	98	58	96	96	94	85	70	84
5	39	85	113	66	106	152	115	125	81	111
6	40	48	73	82	112	163	153	116	104	111
7	39	92	62	87	139	120	93	103	130	123
8	64	105	66	26	93	128	61	82	80	134
9	75	132	87	51	38	69	81	103	92	95
10	75	52	80	62	101	110	68	84	140	48

(a) Part of the Matrix of Original Baboon Image Used in GSVD

	1	2	3	4	5	6	7	8	9	10
1	0.3369	-0.5563	-0.1968	-0.0020	-0.1163	-0.0753	0.0600	-0.0013	0.0311	0.0513
2	-0.2343	-0.1347	0.0836	0.0111	-0.2532	0.2025	-0.2377	0.0522	0.0552	-0.0446
3	0.0221	-0.0450	-0.0414	0.1280	0.0702	0.1539	-0.1582	-0.0349	0.0506	-0.0071
4	0.0993	0.0328	0.0668	-0.1477	0.0011	0.2851	0.1267	-0.2395	-0.0186	0.2575
5	0.0226	0.0504	0.0218	-0.0819	-0.2496	-0.1640	-0.0338	0.0890	-0.0239	0.1605
6	-0.0761	-0.0094	-0.0251	0.0319	-0.0889	-0.0029	-0.2376	-0.0680	0.1542	-0.0676
7	-0.0684	0.0204	-0.2636	0.1650	0.0830	0.0983	0.0522	-0.1916	0.1060	-0.1872
8	0.1291	0.2609	-0.0378	0.1726	-0.3727	-0.1415	0.1499	-0.1286	-0.0230	0.0093
9	-0.1922	-0.1347	0.0647	-0.0686	-0.0628	-0.1338	0.0738	-0.1778	-0.2107	-0.0188
10	0.1901	0.0611	0.1551	-0.1020	0.1218	0.0010	0.0168	-0.1213	0.1060	0.0047

(b) Part of the Matrix U of Baboon Image After Applying the GSVD

	1	2	3	4	5	6	7	8	9	10
1	0.3369	-0.5563	-0.1968	-0.0020	-0.1163	-0.0753	0.0600	-0.0013	0.0311	0.0513
2	-0.2343	-0.1347	0.0836	0.0111	-0.2532	0.2025	-0.2377	0.0522	0.0552	-0.0446
3	0.0221	-0.0450	-0.0414	0.1280	0.0702	0.1539	-0.1582	-0.0349	0.0506	-0.0071
4	0.0993	0.0328	0.0668	-0.1477	0.0011	0.2851	0.1267	-0.2395	-0.0186	0.2575
5	0.0226	0.0504	0.0218	-0.0819	-0.2496	-0.1640	-0.0338	0.0890	-0.0239	0.1605
6	-0.0761	-0.0094	-0.0251	0.0319	-0.0889	-0.0029	-0.2376	-0.0680	0.1542	-0.0676
7	-0.0684	0.0204	-0.2636	0.1650	0.0830	0.0983	0.0522	-0.1916	0.1060	-0.1872
8	0.1291	0.2609	-0.0378	0.1726	-0.3727	-0.1415	0.1499	-0.1286	-0.0230	0.0093
9	-0.1922	-0.1347	0.0647	-0.0686	-0.0628	-0.1338	0.0738	-0.1778	-0.2107	-0.0188
10	0.1901	0.0611	0.1551	-0.1020	0.1218	0.0010	0.0168	-0.1213	0.1060	0.0047

(c) Part of the Matrix V of Baboon Image After Applying the GSVD

	1	2	3	4	5	6	7	8	9	10
1	-10.9645	-111.9434	15.4686	40.9094	-109.8764	137.5120	11.7385	-140.2362	-38.6949	72.9683
2	-7.6009	-61.0221	-1.7328	25.2182	-173.0671	182.2604	16.9911	-197.1435	-47.0594	77.7411
3	18.2307	-70.6576	-21.3602	36.1900	-153.2260	154.6757	-67.3709	-169.7119	-27.1935	115.9612
4	-15.6919	-55.1547	-52.9172	54.9861	-92.7260	167.7959	-88.1389	-151.6893	-27.6024	106.5467
5	20.1588	-59.6794	-38.8613	81.4757	-149.6555	226.8790	-77.9671	-203.4131	-45.1714	100.3946
6	15.6521	-89.4839	-16.9768	92.8657	-155.2959	160.3629	-49.8999	-211.3229	-38.4908	121.9146
7	11.6547	-114.0348	-38.8526	57.6003	-127.5770	183.8336	-60.3705	-190.6985	-45.0459	78.3021
8	19.9805	-85.2786	-12.9948	1.8689	-113.3919	136.0839	-67.5382	-214.4468	-65.0108	75.1893
9	-24.3506	-169.5626	-27.4283	49.5109	-96.6057	151.3250	-57.0938	-237.0438	-50.0652	118.1894
10	-60.2105	-91.4851	-21.1577	101.7591	-187.1356	195.2261	9.3298	-199.9016	-47.8358	104.5178

(d) Part of the Matrix X of Baboon Image After Applying the GSVD

	1	2	3	4	5	6	7	8	9	10
1	0.7071	0	0	0	0	0	0	0	0	0
2	0	0.7071	0	0	0	0	0	0	0	0
3	0	0	0.7071	0	0	0	0	0	0	0
4	0	0	0	0.7071	0	0	0	0	0	0
5	0	0	0	0	0.7071	0	0	0	0	0
6	0	0	0	0	0	0.7071	0	0	0	0
7	0	0	0	0	0	0	0.7071	0	0	0
8	0	0	0	0	0	0	0	0.7071	0	0
9	0	0	0	0	0	0	0	0	0.7071	0
10	0	0	0	0	0	0	0	0	0	0.7071

(e) Part of the Matrix C of Baboon Image After Applying the GSVD

	1	2	3	4	5	6	7	8	9	10
1	0.7071	0	0	0	0	0	0	0	0	0
2	0	0.7071	0	0	0	0	0	0	0	0
3	0	0	0.7071	0	0	0	0	0	0	0
4	0	0	0	0.7071	0	0	0	0	0	0
5	0	0	0	0	0.7071	0	0	0	0	0
6	0	0	0	0	0	0.7071	0	0	0	0
7	0	0	0	0	0	0	0.7071	0	0	0
8	0	0	0	0	0	0	0	0.7071	0	0
9	0	0	0	0	0	0	0	0	0.7071	0
10	0	0	0	0	0	0	0	0	0	0.7071

(f) Part of the Matrix S of Baboon Image After Applying the GSVD

Figure 3: The Intensity Values of Parts of the Original Baboon Image Matrix After Being Addressed by GSVD

2.2 Binary Gravitational Search Algorithm (BGSA)

The Gravitational Search Algorithm (GSA) was suggested in [16] and is considered as a major heuristic optimization algorithm that operate according to the metaphor of the gravitational interactions among masses. GSA is motivated by Newton's theory stating that: "Every particle in the universe attracts every other particle with a force which is inversely proportional to the square of the distance among them and directly proportional to the product of their masses". Gravity can be considered as force, pulling together all matter.
The initial form of GSA has been developed for the search spaces related to real-valued vectors. Yet, a lot of optimization problems exist in the discrete (binary) space. In the previous form of GSA, algorithmic "gravitational forces" result in modifications in the search point's position in multidimensional continuous space (search space). Concerning Binary Gravitational Search Algorithm (BGSA), the result regarding such forces has been transformed to probability value related to each one of the elements in the binary vector which will decide if the element is going to have a value of 1 or 0. The force equation will be defined in Equation (2.7).

$$F = G\frac{M_1 M_2}{R^2}, \tag{2.7}$$

where M_1 and M_2 represent the particles' masses, R represents the distance between them, G is the gravitational constant, the gravitational force among big particles that are closer will be large, and its real value based on the universe age is shown in Equation (2.8).

$$G(t) = G_0(1 - \frac{t}{t_{max}}), \tag{2.8}$$

where t_{max} represents the total number of iterations. G_0 represents the value related to the gravitational constant at the start.

Computer science employs the law of Newtonian gravitational in the issues of optimization as many solutions of the stochastic candidate, named agents, which are generated for optimization problems as an initial population. Then every agent moves other agents according to the law of Newtonian gravitational. So, the problem space is searched to discover the best possible solution. Based on GSA, at the beginning of the algorithm, the population of n agents will be created. Each agent's position that is considered as a solution is described as [17]:

$$X_i = (x_{il}, ..., x_{id}, ..., x_{im});\ i = 1, 2, ..., n, \tag{2.9}$$

in which x_{id} represents the position of i^{th} mass. After that, via estimating and using each particle's fitness value, their masses will be calculated as (maximization problem):

$$M_i = \frac{q_i}{\Sigma_{i=1}^{n} q_i(t)} \tag{2.10}$$

$$q_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, \tag{2.11}$$

in which $fit_i(t)$ represents the i^{th} agent's fitness value in the i^{th} iteration (i.e., specific-time), $best(t)$ and $worst(t)$ indicate best and worst values in the t^{th} iteration and are displayed in the following equations:

$$best(t) = max_{j\in\{1,\ldots,n\}} fit_j(t) \tag{2.12}$$

$$worst(t) = min_{j\in\{1,\ldots,n\}} fit_j(t). \tag{2.13}$$

In the following stage, force act from j^{th} to i^{th} agent in t^{th} iteration will be assessed through the use of the following equation:

$$F_{ij}^{d} = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij} + \varepsilon}(x_j^d(t) - x_i^d(t)), \tag{2.14}$$

where M_i and M_j indicate the mass of the agents i and j, e represents a small constant. R_{ij} represents hamming distance among two agents i and j. Total force act from other agents to the i^{th} agent will be calculated via summing arbitrarily weighted forces which will be obtained from d^{th} dimension of the other agents:

$$F_i^d(t) = \Sigma_{j\in Kbest, j\neq iu}\gamma_j F_{ij}^d(t), \tag{2.15}$$

where γ_i represents an arbitrary value in the range of (0 to 1). The K optimum agents having the maximum fitness value will be just used for attracting others because such constraint is going to enhance the effectiveness of GSA through handling explorations and exploitations to avoid trapping in the local optimum solution. At the initial phase, the set of K best consists of whole agents and reduces the linearly to one at a time. In the following step, acceleration related to i^{th} agent is estimated depending on the law of motion through the use of the following equation:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}, \tag{2.16}$$

Then the agent's velocity in t^{th} the iteration at dimension d will be estimated via its acceleration, current velocity will be estimated through the use of the following equation:

$$\upsilon_i^d(t+1) = a_i \upsilon_i^d(t) + a_i^d(t), \tag{2.17}$$

where a_i is a random value in the range $[0, 1]$.
Concerning BGSA, the agent's position in each one of the dimensions could hold a value of one or zero. The agent's position is updated depending on the mass velocity probability using the following equation:

$$x_i^d(t+1) = \begin{Bmatrix} complement(x_i^d(t)); & \gamma_i < S(\upsilon_i^d(t)) = |\tanh(\upsilon_i^d(t))| \\ x_i^d(t), & \text{Otherwise} \end{Bmatrix} \tag{2.18}$$

To attain a good convergence, rate the velocity bounded by $|\ \upsilon_i^d(t)\ | < \upsilon_{max} = 6$. Previous steps will be repeated until maximum iteration is achieved or good solutions are obtained [18].

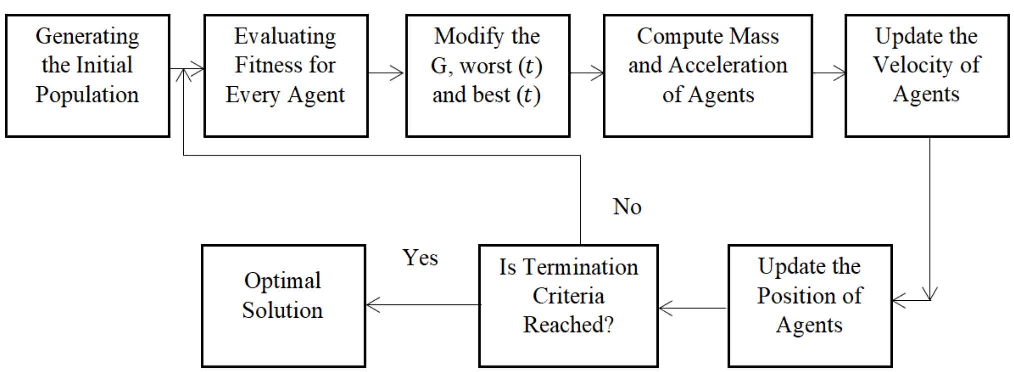


Figure 4: The Diagram of BGSA

3 The Proposed Optimized Secret Watermark Sharing

In this proposed algorithm, GSVD is utilized to extract the features of the original image and the optimal features are selected based on the BGSA optimization algorithm. Then these optimal selected features are XORed with the binary watermark such that the embedded watermark is obtained itself after the extraction process.

3.1 The Influence and the Importance of GSVD in the Proposed Algorithm

The GSVD was applied to decompose each block of the original image into five matrices to obtain the two diagonal matrices C_i and S_i for each block, each of C_i and S_i includes the generalized singular values in the diagonal. Hence, sum the diagonal elements of the matrices C_i to construct L_i, the same steps are applied on the matrices S_i to construct N_i. All L_i are divided by the corresponding N_i to take out the matrix K_i and then the collected matrix K that XORed (after conversion into a binary matrix K_b) with the binary watermark to generate the secret share.
The core idea is the operation used to compute the division between L and N. Algebraically,

- $mldivide(L, N) \cong L/N$ is the left division (forward slash) such that the matrices L and N have the same number of rows except L is a scalar. So, L/N is element-wise division represented by $L/N = L./N$.
- If L is a square matrix, then $L/N \cong L * inv(N)$.
- If N is a matrix of size $n \times n$ and L is a row vector of n components, then the solution to the equation $X * N = L$ is $X = L/N$ that obtained by using Gaussian elimination with partial pivoting.

Because the secret watermark sharing schemes were built essentially on the features of the original image (the most significant information), the GSVD method was successful in embedding and extracting the watermark exactly. This is due to the important property of the GSVD method that generates two nonnegative diagonal matrices used to construct the secret share.

3.2 Embedding Algorithm

The process of constructing the secret share to design an optimized secret watermark sharing algorithm using the GSVD method is explained in Figure 5. The itemized steps are inserted as follow:s

Step1: Input the $n \times n$ original image and turn it into grayscale.

Step2: Divide the grayscale image into 4×4 non-overlapping blocks.

Step3: Select the blocks that match the random key generated by the BGSA algorithm and apply the GSVD method on each block to obtain $\{U_i, V_i, X_i, C_i, S_i\}$.

Step4: Sum the diagonal elements C_i of each selected block to form L_i and also sum the diagonal elements S_i of each block to form N_i. Divide L_i on N_i to obtain the feature K_i.

Step5: Convert the features matrix K into a binary matrix K_b as follows:

$$\begin{aligned} &K_b(i,j) = 1 && \text{if } K(i,j) \geq K(i,j+1) \\ &K_b(i,j) = 0 && \text{otherwise} \end{aligned} \tag{3.19}$$

Step6: Input and convert the watermark matrix W into binary to get the matrix W_b.

Step7: Perform XOR logical operation between the binary features of the original image K_b and the binary watermark W_b to get M.

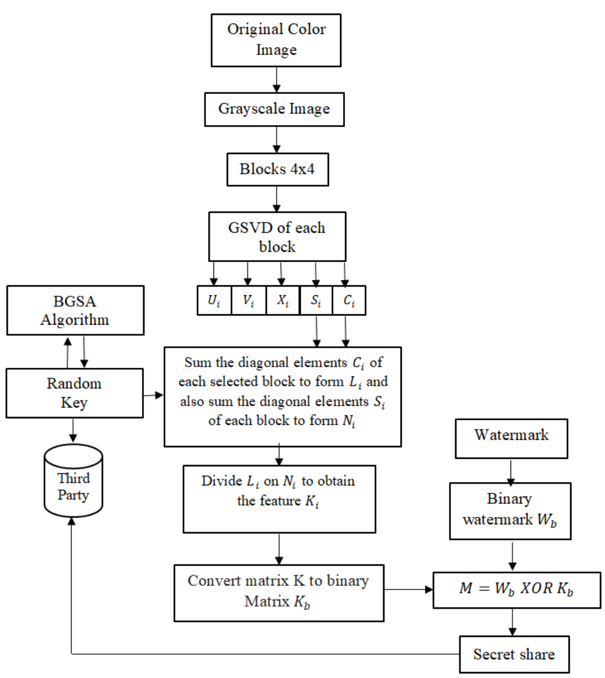


Figure 5: The Diagram of Embedding Algorithm

3.3 Optimal Secret Watermark Sharing Based on Binary Gravitational Search Algorithm (BGSA)

The BGSA algorithm is exploited to find the best blocks (optimal agents) in the original image to extract robust features. The BGSA optimization algorithm based secret watermark sharing algorithm encompasses several steps as follows:

Step1: Create the initial random population of n agents which indicates random numbers of blocks' locations required for extracting the robust feature bits.

Step2: Calculate the function of valuation which indicates the summation of NC values for Six attacks "Gaussian noise, JPEG compression, Speckle noise, Motion filter, Median filter, and Histogram equalization" on the original color image.

Step3: Complete the processes of optimization. Modify the $G, worst(t)$, and $best(t)$. Compute mass and acceleration of agents; Update the velocity of agents; Update the position of agents.

Step4: If the termination criteria are reached then obtain the optimal solution otherwise go to step 2.

The selected agents that hold the highest valuation function indicate the optimal blocks required to extract optimal features.

3.4 Extraction Algorithm

The extraction of the watermark is explained in Figure 6. The itemized steps are inserted as the following:

Step1: Input the $n \times n$ suspicious image and turn it into grayscale.

Step2: Generate the binary feature matrix K_b as in the steps of the embedding algorithm.

Step3: Apply the XOR logical operation between binary features of the original image K_b and the secret share M to get Watermark.

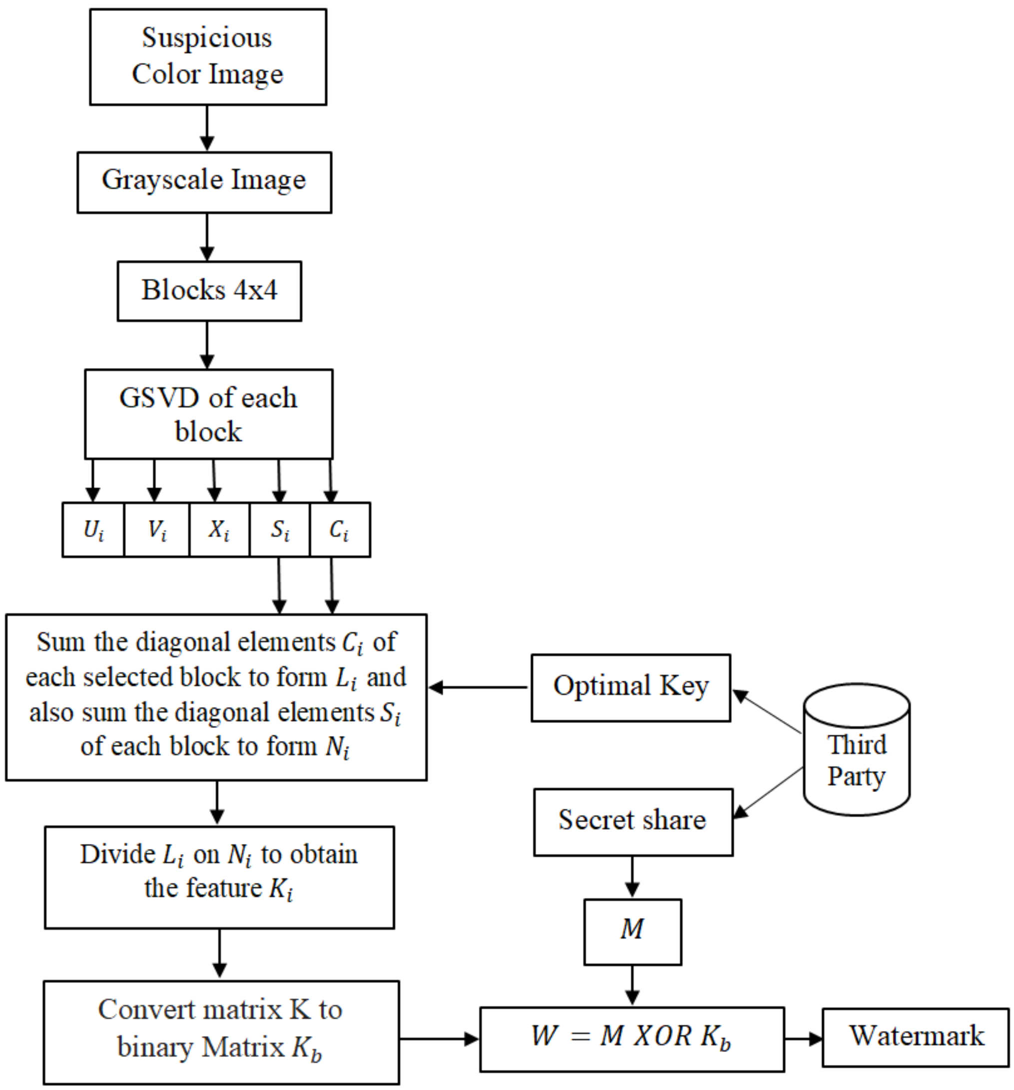


Figure 6: The Diagram of Extraction Algorithm

4 Experimental Results

To evaluate the performance of the watermarked images, there are some types of measurements to determine the quality of the image including, peak signal-noise ratio (PSNR) and Mean Square Error (MSE). The PSNR is most commonly used as a measure of the quality of image reconstruction, where the PSNR formula is:

$$PSNR = 20 \log(\frac{MAXI^2}{MSE}) \tag{4.20}$$

where $MAXI$ is the maximum possible pixel value of the image. The high PSNR value indicates high security because it indicates the minimum difference between the original and watermarked data. So, no one can suspect the hidden information. PSNR is measured in decibels (db).
Mathematically, the normalized correlation (NC) is adopted to check that the proposed technique is working quite correctly such that the obtained results are logical and precise. The optimal value of the NC is equal to 1.

$$NC(W;W') = \frac{\Sigma_{m=1}\Sigma_{n=1} W \times W'}{\sqrt{\Sigma_{m=1}\Sigma_{n=1} W^2} \times \sqrt{\Sigma_{m=1}\Sigma_{n=1} W'^2}} \tag{4.21}$$

where W and W' represent the $n \times m$ original and extracted watermarks respectively. The original images and the watermark utilized in this proposed algorithm are illustrated in Figure 7.

(a) Leann Image (b) Baboon Image (c) Girl Image (d) Peppers Image (e) Watermark

Figure 7: The Images Used in the Work

To evaluate the performance of the proposed secret watermark sharing algorithm, the following attacks were applied, see Table 1.

It is noticeable that the proposed algorithm with and without optimization was worked successfully, whereas the watermarks were extracted exactly depending on the NC values, see Table 2.

In this proposed algorithm after dividing the 512×512 original image into 4×4 non-overlapping blocks, 128×128 blocks are obtained. The GSVD

Table 1: The Utilized Attacks and their Descriptions

No.	**Attack**	**Descriptions**
1	JPEG compression	quality factor (QF) of 25
2	salt and pepper noise	density equal to 0.01
3	Speckle Noise	density equal to 0.01
4	Histogram equalization	-
5	Gaussian noise	mean equal to 0.01
6	Motion filter	Linear motion of camera was Len = 20
7	The median filter	Neighborhood size is 22

Table 2: The NC Values for Extracted Watermark Images

Image	**Lena Image**	**Baboon Image**	**Girl Image**	**Peppers Image**
NC Values (with optimization)	1	1	1	1
NC Values (without optimization)	1	1	1	1

method was applied on each selected block of the original image to separate each block into five 4×4 matrices U_i, V_i, X_i, C_i, S_i. In this algorithm the matrices C_i, S_i are utilized to obtain the features matrix K and convert K into a binary matrix K_b. Then the XOR logical operation between the binary features K_b and 64×64 binary watermark W_b is performed to create a secret watermark share.

Before any image is attacked, the results of the direct extraction of the watermark that are given in Table 2 presents the NC of the extracted watermark images which hold the value 1. This illustrates that the watermark was extracted exactly and the GSVD method worked successfully before any attack is being launched. Table 3 shows the original images, the watermark image, and the extracted watermark (E.W.).

Table 3: The Watermark that Embedded and the Extracted without Attacks

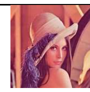
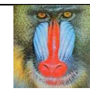
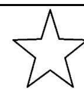
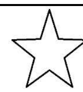
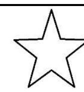
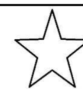
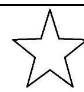
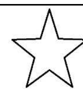
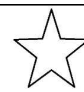
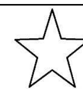
Image Type	Lena	Girl	Baboon	Peppers
Original Image				
The Watermark				
E.W.				

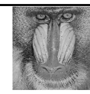
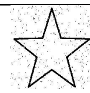
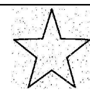
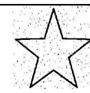
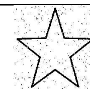
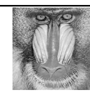
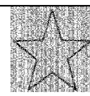
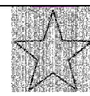
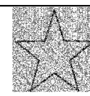
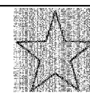
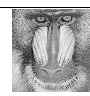
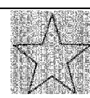
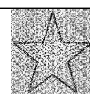
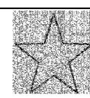
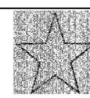
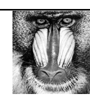
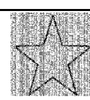
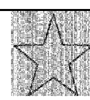
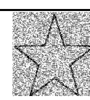
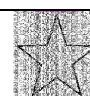
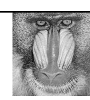
Table 4 shows the robustness (NC) and the imperceptibility (PSNR) tests of the secret watermark sharing algorithm of the extracted watermark after using various kinds of attacks.

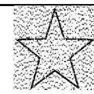
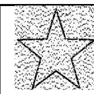
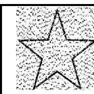
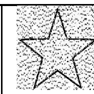
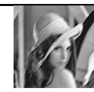
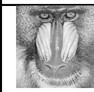
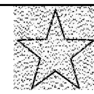
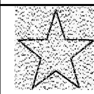
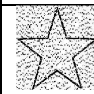
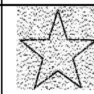
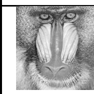
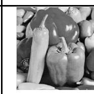
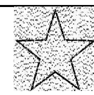
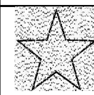
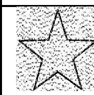
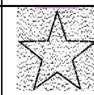
Table 4: The Values of the PSNR and the NC for Images after Attacks without Using Optimization

Attacks	Lenna		Girl		Baboon		Peppers	
	PSNR	NC	PSNR	NC	PSNR	NC	PSNR	NC
Salt and Pepper Noise	26.765	0.914	26.156	0.923	27.187	0.902	26.609	0.900
JPEG Compression	60.708	0.893	60.715	0.892	60.709	0.898	60.143	0.891
Hist. Equalization	19.074	0.882	10.120	0.884	16.665	0.878	20.584	0.891
Gaussian Noise	28.073	0.855	32.191	0.861	25.187	0.857	28.262	0.8529
Speckle Noise	35.665	0.955	41.052	0.951	35.533	0.955	35.724	0.9557
Motion Filter	22.058	0.933	24.454	0.932	21.260	0.933	21.095	0.9335
Median Filter	28.720	0.886	32.356	0.896	24.241	0.888	29.153	0.890

Table 5 shows the corresponding attacked images under various attacks and the watermarks that are extracted after extract the features from the images after being attacked. We see that the attacked images are similar to the host images under all attacks.

Table 5: Extracted Watermark under Attacks without Optimization

Attacks	**Lena**	**Girl**	**Baboon**	**Peppers**
Salt and Pepper				
E.W.				
JPEG Compression				
E.W.				
Gaussian Noise				
E.W.				
Histogram Equalization				
E.W.				
Speckle Noise				

E.W.				
Histogram Equalization				
E.W.				
Speckle Noise				
E.W.				

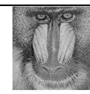
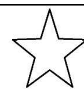
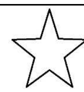
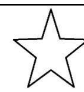
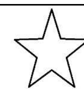
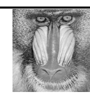
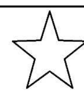
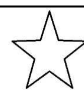
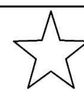
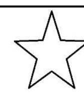
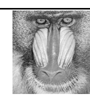
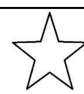
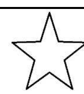
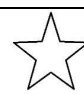
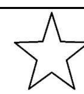
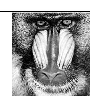
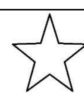
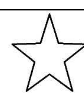
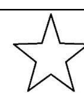
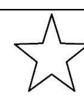
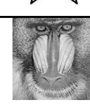
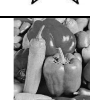
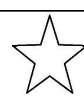
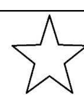
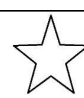
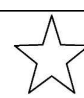
However, we note that the constructed watermark after attacks is not exactly like the original one. So, to improve this algorithm the optimization algorithm is suggested to be used. Table 6 shows the robustness (NC) and the imperceptibility (PSNR) tests of the optimized secret watermark sharing algorithm of the extracted watermark after using various kinds of attacks.

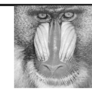
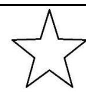
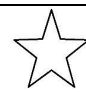
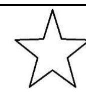
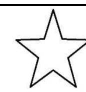
Table 6: The Values of the PSNR and the NC for Images after Attacks Using Optimization

Attacks	**Lenna**		**Girl**		**Baboon**		**Peppers**	
	PSNR	**NC**	**PSNR**	**NC**	**PSNR**	**NC**	**PSNR**	**NC**
Salt and Pepper Noise	26.765	1	26.156	1	27.187	1	26.609	1
JPEG Compression	60.708	1	60.715	1	60.709	1	60.143	1
Hist. Equalization	19.074	1	10.120	1	16.665	1	20.584	1
Gaussian Noise	28.073	1	32.191	1	25.187	1	28.262	1
Speckle Noise	35.665	1	41.052	1	35.533	1	35.724	1
Motion Filter	22.058	1	24.454	1	21.260	1	21.095	1
Median Filter	28.720	1	32.356	1	24.241	1	29.153	1

The values of the PSNR of all the images are equal to the values of the PSNR obtained in Table 4. This is because of the attacks launched on the same original images in both algorithms with and without optimization. Table 7 presents the extracted watermarks after attacks for each image.

Table 7: Extraction Watermark under Attacks with Optimization

Attacks	**Lena**	**Girl**	**Baboon**	**Peppers**
Salt and Pepper				
E.W.				
JPEG Compression				
E.W.				
Gaussian Noise				
E.W.				
Histogram Equalization				
E.W.				
Speckle Noise				
E.W.				
Histogram Equalization				
E.W.				

Speckle Noise				
E.W.				

5 Conclusions

We presented an application of linear algebra using a matrix decomposition method (Generalized Singular Value Decomposition-GSVD) in image processing for a secret watermark sharing algorithm. Moreover, we used the gravitational search algorithm to demonstrate the efficiency of the application and to improve the performance of using GSVD to find the optimal secret image watermark sharing such that the watermark can be extracted successfully and the algorithm can resist most of the common attacks. Furthermore, the proposed optimized secret watermark sharing algorithm satisfied several properties: First, the robustness in which the NC values are equal to one (optimal robustness) before and after shedding any attack. Secondly, the imperceptibility in which the PSNR values are also optimal.

References

[1] M. H. Khudhur, J. Waleed, H. Hatem, A. M. Abduldaim, D. A. Abdullah, An Efficient and Fast Digital Image Copy-Move Forensic Technique, 2nd International Conference for Engineering, Technology and Sciences of Al-Kitab, IEEE, (2018), 78–82.

[2] A. M. Abduldaim, J. Waleed, A. N. Mazher, An Efficient Scheme of Digital Image Watermarking Based on Hessenberg Factorization and DWT, International Conference on Computer Science and Software Engineering, IEEE, (2020), 180–185.

[3] R. Eriguchi, N. Kunihiro, Strongly Secure Ramp Secret Sharing Schemes from Any Linear Secret Sharing Schemes, IEEE Information Theory Workshop, (2019), 1–5.

[4] X. Wu, C. N. Yang, Partial Reversible Ambtc-Based Secret Image Sharing with Steganography, Digital Signal Processing, **93,** (2019), 22–33.

[5] R. I. Sabri, A. M. Abduldaim, J. Waleed, Mamdani FIS Combined with LU Decomposition Method and Two-Level LWT for Image Watermarking Technique, 3rd International Conference on Engineering Technology and its Applications, IEEE, (2020), 12–17.

[6] J. Waleed, H. D. Jun, S. Hameed, A Robust Optimal Zero-Watermarking Technique for Secret Watermark Sharing, International Journal of Security and Its Applications, **8,** no. 5, (2014), 349–360.

[7] R. Wang, H. Shaocheng, P. Zhang, M. Yue, Z. Cheng, Y. Zhang, A Novel Zero-Watermarking Scheme Based on Variable Parameter Chaotic Mapping in NSPD-DCT Domain, IEEE Access, **8,** (2020), 182391–182411.

[8] N. S. Mohammed, A. M. Abduldaim, Algebraic Hessenberg Decomposition Method Optimized by Genetic Algorithm for Zero Watermarking Technique, International Journal of Mathematics and Computer Science, **16,** no. 4, (2021), 1497–1514.

[9] J. Waleed, A. M. Abduldaim, H. H. Alyas, A. Q. Mohammed, An Optimized Zero-Watermarking Technique Based on SFL Algorithm, 2nd International Conference on Electrical, Communication, Computer, Power and Control Engineering, IEEE, (2019), 171–175.

[10] A. Kumar, A. Dwivedi, M. K. Dutta, A Zero Watermarking Approach for Biometric Image Security, International Conference on Contemporary Computing and Applications, (2020), 53–58.

[11] X. Kang, F. Zhao, Y. Chen, G. Lin, C. Jing, Combining Polar Harmonic Transforms and 2D Compound Chaotic Map for Distinguishable and Robust Color Image Zero-Watermarking Algorithm, Journal of Visual Communication and Image Representation, **70,** (2020), 102804.

[12] Z. Xia, X. Wang, B. Han, Q. Li, X. Wang, C. Wang, T. Zhao, Color Image Triple Zero-Watermarking Using Decimal-Order Polar Harmonic Transforms and Chaotic System, Signal Processing, **180,** (2021), 107864.

[13] G. W. STEWART, On The Early History of The Singular Value Decomposition, SIAM REVIEW, **35,** no. 4, (1993), 551–566.

[14] C. C. Paige, M. A. Saunders, Towards a Generalized Singular Value Decomposition, SIAM Journal on Numerical Analysis, **18,** (1981), 398–405.

[15] G. H. Golub, C. V. Loan, Matrix Computations, Third Edition, Johns Hopkins University Press, Baltimore, 1996.

[16] J. P. Papa, A. Pagnin, S. A. Schellini, A. Spadotto, R. C. Guido, M. Ponti, G. Chiachia, A. X. Falcao, Feature Selection through Gravitational Search Algorithm, IEEE International Conference on Acoustics, Speech and Signal Processing, (2011), 2052–2055.

[17] M. Sheikhan, H. Bostani, Binary Gravitational Search Algorithm: Improved Efficiency, in Encyclopedia of Information Assurance, Taylor & Francis, (2016), 1–19.

[18] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Binary gravitational search algorithm, Natural Computing, **9,** no. 3, (2010), 727–745.