

Robust watermarking in DWT domain using SVD and opposition and dimensional based modified firefly algorithm

Elham Moeinaddini¹ · Fatemeh Afsari²

Received: 24 June 2017 / Revised: 27 December 2017 / Accepted: 26 February 2018 /
Published online: 12 April 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract The process of authenticating a digital image by embedding a watermark into it is called digital image watermarking, which protects the image from copyright infringement. This paper proposes an optimized watermarking scheme in the discrete wavelet transform (DWT) domain based on the singular value decomposition (SVD) using opposition and dimensional based modified firefly algorithm (ODFA). The host image is segmented into 8×8 non-overlapping blocks and the most suitable embedding blocks are selected for embedding. The selected blocks are then transformed into DWT domain and SVD is applied on LL1 sub-bands. Finally, the process of embedding the watermarked bits is optimized using ODFA, which applies the combination of imperceptibility and robustness as the objective function. The embedding algorithm possesses good similarity between the host and the watermarked images as well as the strong robustness against various image processing operations and attacks. Experimental results reveal that the proposed scheme has a higher degree of imperceptibility and robustness compare to related existing schemes.

Keywords Blind watermarking · Discrete wavelet transform · Opposition and dimensional based modified firefly algorithm · Robustness · Imperceptible

1 Introduction

Nowadays, the development of internet and computer networks has made multimedia (images, videos, audios, and so on) easily accessible and replicable. The quality of the replicated digital

✉ Fatemeh Afsari
afsari@uk.ac.ir; afsari.f@gmail.com

Elham Moeinaddini
el_moin@ujiroft.ac.ir

¹ Department of Electrical Engineering, University of Jiroft, Jiroft, Iran

² Department of Computer Engineering, Shahid Bahonar University of Kerman, PO Box: 76169-133, Afzalipoor square, Kerman, Iran

contents are exactly the same as of the original ones. This case sometimes causes critical negative conditions like abusing data. Digital watermarking is a technique to overcome such issues. It provides appropriate methods to protect the copyright of digital contents. Digital watermarking is an approach to hiding data, which insures media security. In this technique, a watermark, which can be a digital text or image, is embedded inside a digital content. In the watermarking process, two major criteria are mandatory to be fulfilled. These are (1) imperceptibility of embedded watermark and (2) robustness of the watermark embedding scheme. Indeed, the embedded watermark should be undetectable (fulfilling the first criteria), and should not be removed by malicious attacks or common communication channel processes (fulfilling the second criteria). It is worth noting, imperceptibility and robustness of the watermark are two main contradictory objectives to evaluate any watermarking algorithm.

In elementary watermarking methods, the watermark is directly embedded into image bits using easy methods like embedding in the least significant bits (LSB) of the covering image as well as the vector quantization (VQ) method. In addition to the simplicity of usage of these approaches, the required time to embed the watermark is very short. However, they are not robust enough against common image processing operations and attacks. In contrast, many techniques were proposed to embed the watermark in the frequency domain. These techniques mostly employ different transformations, including discrete Fourier transformation (DFT), discrete cosine transformation (DCT), discrete wavelet transformation (DWT), singular value decomposition (SVD), and their combinations, e.g. DCT-DWT, DCT-SVD, DWT-SVD [4, 5, 11, 15, 16, 18, 20, 22]. Among them, wavelet transformation can be known as one of the most powerful methods due to its transparency and robustness. By using the combination of SVD and DWT the imperceptibility and robustness of the watermarking methods have been increased [15].

In most watermarking techniques, there are variable amounts of changes on host images for embedding the watermark. Thus, it is crucial to determine the amount of these changes in a way that a good trade-off is achieved between imperceptibility and robustness of the watermarking method. These amounts of changes are usually adjusted by parameters calling strength factors. In most watermarking methods, single strength factor is used, which will not usually have good imperceptibility and robustness results due to ignoring the features of each part of the content [3, 6, 11, 13, 24]. Also, some other watermarking methods have utilized multiple strength factors. Selecting these factors is more difficult than it seems. High values of strength factors, increase the robustness of the watermark, while makes it tangible and thus reduces content transparency. On the other hand, low values of strength factors preserve imperceptibility however decrease robustness of the watermark. Therefore, an efficient and powerful algorithm is necessary to find the optimal multiple strength factors to reach a good trade-off between imperceptibility and robustness. Since, the goal is to find the optimal multiple strength factors, this issue can be addressed as an optimization problem and heuristic methods can be used to solve it. Recently, researchers have employed population-based and swarm intelligent methods (particle swarm, ant colony optimization, artificial bee colony, firefly algorithm, and etc.) to find optimal multiple strength factors in watermarking techniques, which have significantly improved the results [2, 17, 19, 27].

Musrat Ali et al. [2] have proposed an image watermarking method in the DWT-SVD domain with a fixed distribution in which the artificial bee colony algorithm (ABC) is applied to optimize the multiple strength factors (MSFs). Lai et al. [12] employed tiny genetic algorithm (tiny-GA) based on the SVD to find optimal MSF values. Ishtiaq et al. [8] utilized PSO to find optimal MSFs in the DCT domain. They used PSNR value as the objective function to evaluate the strengths. Loukhaoukha et al. [17] employed multi-objective ant colony optimization in the LWT-SVD domain to find optimal MSFs.

The aforementioned approaches are usually complex with several control parameters. The firefly algorithm is one of the collective intelligence based optimization methods, which was introduced by Yang [29]. Many researchers have used FA in different applications and have realized that this approach outperforms other aforementioned heuristic methods [7, 31]. The improved versions of the firefly algorithms, which were recently proposed, have eliminated the weaknesses of this method. The most important modification was the chaotic based [30] and opposition and dimensional based [26] firefly algorithms. The opposition and dimensional based firefly algorithm (ODFA) is proposed by Verma et al. [26] to promote the original FA. In this method, the oppositional-based learning is used to increase convergence rate and also the dimensional-based method is used to find the global optimum among all optimal solutions in all dimensions. Due to the purposeful movement of fireflies and the reduction in their random movements, this hybrid version of FA has a higher convergence speed and lower time complexity. Recently, researchers have utilized the firefly algorithm to create optimal trade-offs between imperceptibility and robustness in image watermarking methods. Mishra et al. [19] have proposed an image watermarking method in the DWT-SVD domain in which a simple FA is used to detect MSFs. In their approach, the objective function is a linear combination of the imperceptibility and robustness of the watermark. However, Musrrat Ali et al. [2] have shown that this method has false positive errors and the algorithm poses watermark ambiguity. Dong et al. [4] have proposed a gray scale image watermarking method in the DCT-SVD domain in which the chaotic FA is used to find MSFs. Their results indicate an increase in robustness and imperceptibility. Although, in their method, the existence of the original image is essential to reveal the watermark. Kazemivash and Ebrahimi Moghaddam [9] proposed a robust technique for gray scale image watermarking which is based on the lifting wavelet transform and firefly algorithm to select appropriate blocks in the host image for embedding watermark. Also they proposed in [10] another watermarking technique that utilizes the firefly algorithm. Their proposal technique is based on lifting wavelet transform and the watermark is embedded using predicted value of regression tree. The multiple scaling factors are achieved by the firefly algorithm. Although they increase the robustness in both techniques but image transparency is not well maintained.

This paper proposes an image watermarking scheme in the DWT domain based on SVD, and tries to discover multiple strength factors by using the opposition and dimensional based firefly algorithm. The proposed scheme presents a robust and imperceptible image watermarking scheme. The process begins by dividing the cover image into certain blocks, then by using the human visual system the appropriate embedding blocks are selected. Moreover, the optimized MSFs are detected by using the opposition and dimensional based firefly algorithm. The objective function of the optimization problem is a linear combination of the normal correlation coefficient (NC) and the value of peak signal-to-noise ratio (PSNR).

The rest of this paper is organized as follows. Section 2 briefly explains oppositional dimensional based firefly algorithm. In Section 3 a brief summary of the entropy, DWT and SVD are discussed. Section 4 introduces the embedding and extracting of proposed scheme, Section 5 presents the application of the ODFA to find thresholds and Section 6 discusses experiments and empirical results. Finally, Section 7 concludes the paper.

2 The modified oppositional dimensional-based firefly algorithm

The firefly algorithm (FA) is a meta-heuristic method inspired by the behaviors of fireflies, which presents good optimal solutions for many problems. The firefly algorithm has a fewer number of

the control parameters in comparison to the other methods in this domain. Therefore, it is more simple and effective approach. Despite the good performance of the FA, this algorithm is prone to the premature convergence like other collective intelligence algorithms and may be stuck in the local optima. In order to find the most optimal solution, this algorithm does not consider all dimensions of a firefly separately. This makes some dimensions closer to optimal values and the rest farther. Therefore, the global optimum is not achieved. The opposition and dimensional based FA (ODFA) improves the simple FA in two ways. First, using the oppositional method during initialization of the selected solutions provides to search all directions of the corresponding interval and consequently increases the convergence of the solutions. Second, using the dimensional-based method, fireflies are updated along different dimensions. This approach generates more optimal and less time complex solutions due to the effective updates and initialization. Here the ODFA presented by Verma et al. [26] is reviewed briefly.

Opposite numbers:

Let $x \in [m, n]$ be a real number and \tilde{x} be the opposite number of x and is given by:

$$\tilde{x} = m + n - x \quad (1)$$

In multidimensional space, the definition of the opposite number is slightly different. Suppose $x = (x_1, x_2, \dots, x_D)$ is a point in a D-dimensional space, where x_1, x_2, \dots, x_D are real numbers and $x_j \in [m_j, n_j]$ for all $j \in \{1, 2, \dots, D\}$. The opposite point \tilde{x} at dimension j is given by:

$$\tilde{x}_j = m_j + n_j - x_j, j = 1, \dots, D \quad (2)$$

Opposition-based optimization:

Suppose that $f(\cdot)$ be the objective function, x be a selected solution in D-dimensional space and \tilde{x} be its opposite point. If $f(\tilde{x}) > f(x)$ then x is replaced by \tilde{x} . Therefore, in the optimization process to specify the more suitable point, any point and its opposite are evaluated simultaneously.

In the FA, suppose that N fireflies are defined as $x = (x_1, x_2, \dots, x_N)$ in the D-dimensional space such that i^{th} firefly is defined as $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, the following changes have been applied on the original firefly algorithm:

- a) The oppositional method is used to initialize the fireflies. More specifically, in order to generate the N initial fireflies, for each randomly generated firefly, its opposite is also created. Finally, there are $2N$ fireflies and the best N ones are selected among them according to the objective function.
- b) Each firefly represents a solution of the problem. In the simple FA, all D dimensions of each firefly are updated simultaneously. This makes some dimensions reach close to the optimum and the rest go far from it. The values that are far from the optimum will become worse in the next iterations. Thus, it is impossible to achieve the global optimum in environments with a large number of dimensions. The ODFA proposes to update each dimension separately. This is realized through the introduction of the concept called Gbest. Gbest (global best) represents the most optimal firefly in the current generation. At each dimension, the GBest values is replaced separately according to the corresponding values of other fireflies. Subsequently, GBest is updated if each value increases the objective function. This process repeats for all dimensions of each firefly at each iteration, and at the end of each iteration, all fireflies will move towards GBest. The pseudo code of the ODFA is shown in Fig. 1 [26].

```

/* Opposition and dimensional based modified firefly algorithm*/
1. Generate initial population of fireflies  $x = (x_1, x_2, \dots, x_N)$  in the range  $[m, n]$ , such that
 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ .
2. Generate Opposition Population,  $\tilde{x}_{ij} = m_j + n_j - x_{ij}$   $i \in [1, N]$  &  $j \in [1, D]$ 
3. Select  $N$  fitted fireflies from the set  $\{x, \tilde{x}_{ij}\}$  as initial population of fireflies,  $x$ .
4. Light intensity at  $x_i$  is calculated using optimization function given by,  $f(x_i) = f(x_{i1}, x_{i2}, \dots, x_{iD})$ 
5. Initialize the value of  $\gamma$ .
6. Initialize  $Gbest$  as the best firefly from the initial fireflies' position.
7. while  $t < MaxGeneration$ 
8. For each dimension  $i=1$  to  $D$ 
9. For each firefly  $j=1$  to  $N$ 
10.  $Y = Gbestpos$ 
11.  $Y = x(j, i)$ 
12. If  $f(Y) > Gbest$ 
13.  $Gbest = f(Y)$ 
14.  $Gbestpos = Y$ 
15. End if
16. End for  $j$ 
17. End for  $i$ 
18. For  $i=1$  to  $N$ 
19. Move firefly  $i$  toward  $Gbest$  in  $D$ -dimension
20. Determine the distance between  $i^{th}$  firefly and  $Gbest$  by Euclidean
21. Determine Attractiveness which varies distance  $r$  via  $\beta e^{-\gamma r^2}$ 
22. Move firefly  $i$  toward  $Gbest$ 
23. End for  $i$ 
24. End while
25. End.

```

Fig. 1 Pseudo code of the oppositional dimensional-based firefly algorithm

2.1 Complexity analysis

In the FA the main program has two nested loops requiring $O(N^2)$ operations, where N is the number of fireflies. ODFFA also has two nested loops requiring $O(DN)$ operations; the outer loop requires $O(D)$ operations and the inner loop needs $O(N)$ operations, where D is the dimensions of the fireflies. These operations should be run for each iteration, thus the overall time complexity of FA and ODFFA are $O(MG \times N^2)$ and $O(MG \times ND)$, respectively, where MG is the maximum number of generations. In this way the time complexity of ODFFA will be less than FA when $D < N$. Although ODFFA converges much faster than FA, it is also possible to adjust a small value for MG to reduce the running time of ODFFA more. In our experiments the value of D , the dimensions of fireflies, is adjusted according to the watermark size. To be specific, for a watermark of size $n \times n$, dimensions of each firefly is considered to be $n^2/10$ instead of n^2 to reduce the overall running time.

3 Summary of concepts and algorithms

This section briefly presents the concepts and algorithms used to select proper blocks and to perform transformations.

3.1 Discrete wavelet transform

Wavelets are a group of mathematical functions for decompose signal to its frequency components of constant bandwidth using logarithmic scaling. Discrete wavelet transform is the result of sampling on wavelet function. DWT can be implemented as a multistage

transformation. An image in first stage is decomposed to the four sub bands denoted LL, LH, HL and HH. LL sub band can be decomposed again to obtain the next level of decomposition. This process can be continued on LL sub bands until the desired number of levels reached. Since the human visual systems are much more sensitive to the LL sub band, most of compression processes avoid variation on this sub band. Therefore, for higher robustness it is better to hide watermark in the LL sub band.

3.2 Singular value decomposition

Singular value decomposition (SVD) is an important concept in linear algebra to decompose matrices and has many theoretical and practical applications. SVD has several applications in watermarking and image processing. It provides structural information of an image that can be used for prediction of image quality. Formally, the SVD of an $M \times N$ rectangular matrix A is a decomposition of the form $A = USV^T$, where, U is an $M \times M$ orthogonal matrix that its columns are called the left-singular vectors of A , S is an $M \times N$ rectangular diagonal matrix with non-negative real numbers on the diagonal that are known as the singular values of A and V is an $N \times N$ orthogonal matrix that its columns are called the right-singular vectors of A .

3.3 HVS based block selection

The human visual system (HVS) extracts perceptual information from a group of pixels instead of a single pixel. There exists a special correlation between the neighboring pixels in images, known as structural information. Image entropy is a quantity, which is used to describe the “business” of an image, i.e. the amount of information, which must be embedded by a watermarking algorithm. Low entropy images, such as those containing a lot of black sky, have very little contrast and large runs of pixels with the same or similar values. An image that is perfectly flat will have an entropy of zero. Consequently, the correlation of neighboring pixels are very high. On the other hand, high entropy images such as an image of heavily cratered areas on the moon have a great deal of contrast from one pixel to the next and consequently the neighboring pixels are correlated highly. Thus, entropy is an appropriate measure of spatial correlation of neighboring pixels. Shannon’s definition of entropy to calculate the entropy of an image is given by [25]:

$$E = - \sum_{i=1}^n p_i \log p_i \quad (3)$$

where p_i is the probability of occurrence of the event i , $0 \leq p_i \leq 1$ and $\sum_{i=1}^n p_i = 1$. At this Eq., the signal is considered as a sequence of symbols. Entropy depends on the relative occurrence of the symbols, irrespective of the positions of occurrence. Obviously, it is necessary to have some information about image features to reach an imperceptible and robust watermark. Pal and Pal in [21] defined average edge information as an exponential form to calculate the entropy that can capture two dimensional spatial correlation of images better than Shannon’s entropy. The entropy is defined in [21] as:

$$E = \sum_{i=1}^n p_i e^{1-p_i} \quad (4)$$

where $1 - p_i$ is the uncertainty of the pixel value i . This definition of entropy gives more information about pixel dispersion and edges of an image.

The first step at the watermarking process is to select appropriate blocks to embed watermark. Das et al. [3] applied the entropy to choose the blocks. More specifically, for each block, mean gray information and mean edges information are calculated by using Eqs. (3) and (4), respectively and then summed together. The information obtained from each block is sorted in an ascending order to form a chain. This chain is divided into three equal parts; where the first, second, and third parts consist of blocks with low, average, and high irregularities, respectively. Information embedding in blocks with high irregularities is less detectable and the HVS is less capable to detect them. However, embedding in this part makes the information vulnerable against malicious attacks, particularly against image compressions. In contrast, blocks with low irregularities are more resistant against attacks and image compressions. However, embedding information in these areas reduces the image quality and causes detectable changes. In this study, to reach a robust watermarking scheme the areas with low and average irregularities are selected to embed the watermark.

4 The proposed watermarking scheme

There are three main steps in a watermarking scheme, which are choosing the appropriate blocks to embed the watermark, the process of embedding and the process of extracting the watermark. The first step is explained at Section 3.3. In Sections 4.1 and 4.2, the embedding and extracting processes in DWT domain and using SVD techniques are presented, respectively. Finally in Section 4.3, the main contribution of the proposed study which reveals the robustness and imperceptibility of the watermarking scheme is presented.

4.1 The process of embedding the watermark

Suppose a binary $n \times n$ image, the watermark, is given to embed in an $M \times N$ image, and here after is called the host image. At first, the host image is segmented into equal non-overlapping 8×8 blocks. Then the information of each block is calculated according to Section 3.3 and the suitable blocks, i.e., blocks with low and average irregularities, are chosen to embed the watermark. Since all blocks are categorized into three parts (blocks with low, average and high irregularities) and just two parts (low and average irregularities blocks) are selected, thus the number of the selected blocks are equal to $2/3((M/8) \times (N/8))$. Therefore, the number of watermark bits could not be higher than this value. The coordinates (x, y) of the selected blocks are stored in two matrices denoted by A and B for further considerations.

The $n \times n$ watermark image is rearranged into an n^2 dimensional vector and the orders of its elements are shuffled using a key that is generated using the key-based Hash pseudo-random permutation algorithm based on MD5 [23]. This key will be used as the first secret key. For each bit of the watermark, the followings are applied to embed this bit in the selected blocks of the host image.

One level DWT is applied to the current selected block to obtain the LL_1 sub-band of this block. Then, SVD is applied to the obtained LL_1 sub-band to achieve matrices U , S , and V . Fan et al. in [6] show that first column of U and V component of SVD are most stable and invariable under general image processing so in the proposed scheme to embed the watermark bit, second and third elements of the first column of the obtained U matrix (i.e., $u_{2,1}$ and $u_{3,1}$

of the matrix U) are used to achieve more robust watermark. Let us call the difference between these two values $u_{2,1}$ and $u_{3,1}$ as Δ , i.e.,

$$\Delta = u_{2,1} - u_{3,1} \tag{5}$$

In our proposed scheme, if $u_{2,1}$ is greater than $u_{3,1}$, then a watermark bit value 1 is embedded in the current block. To this end, it is desired that $u_{2,1}$ be greater than $u_{3,1}$, i.e., $\Delta > 0$. Also to embed a watermark bit value 0 in the current block, it is desired that $u_{2,1}$ be less than $u_{3,1}$, i.e., $\Delta < 0$. If these conditions already presents no modification is needed otherwise, $u_{2,1}$ should be increased or decreased to satisfy above conditions. Usually in the blocks with sharp edges, Δ has large values, so changing $u_{2,1}$ will have a large variation and causes the image quality to reduce. Therefore, in both cases, if variation of Δ exceeds a predefined threshold then the embedding process is reversed. That means the conditions for embedding watermark bit 1 and 0 are exchanged. Thus, in this case, to embed a watermark bit 1 the value of Δ will be negative instead of positive and vice versa for a watermark bit 0. Accordingly, the approach proposed in [3] is used to overcome this problem. First, the interval that Δ is allowed to vary in it (possible values for Δ) is divided into five areas according to the value of the threshold as shown in Fig. 2. The threshold value that controls the Δ value is called Th . In Section 5, an efficient optimization algorithm will be presented to find the optimal threshold values for each block. These threshold values stored in a matrix denoted as C . Matrices A and B (The coordinates (x, y) of the selected blocks) and C have been encrypted using AES-192 and stored as the second secret key for extraction process.

In the case of embedding a watermark bit 1, if Δ becomes negative and is greater than $-\frac{Th}{2}$ (in the first area), then $u_{2,1}$ should be modified to bring Δ in area 2. Also, if Δ becomes negative and is less than $-\frac{Th}{2}$ (Δ is in area 4), in order to satisfy the desired condition, i.e., $\Delta > 0$ the variation of Δ will exceed Th . Thus to have less modification on $u_{2,1}$, $u_{2,1}$ is modified to bring Δ in area 5 (reversing the embedding strategy). Moreover, when the value of Δ is itself positive (Δ is in area 1) and the desired condition is already present the value of $u_{2,1}$ is modified to bring Δ in area 2, too. This modification is made to decrease the value of Δ as desired. Similarly, in order to embed a watermark bit 0, $u_{2,1}$ is modified to bring Δ in area 1 if it is in area 2 (Δ value is greater than $+\frac{Th}{2}$), and bring Δ in area 4 if it is in area 3 or 5 (Δ is less than $+\frac{Th}{2}$). Despite of these modifications, the value of the elements of matrix U may be changed and also the change will happen by compressions or other image processing operations

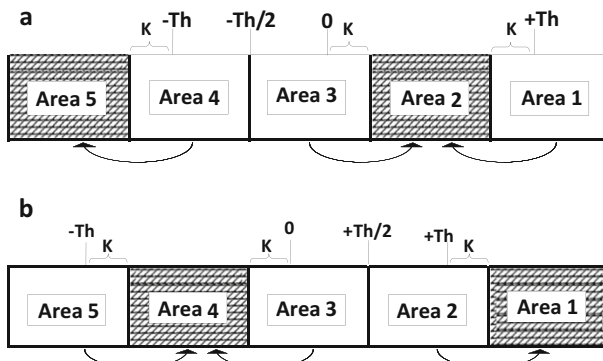


Fig. 2 The interval that Δ can vary; (a) Five areas to embed a bit 1, and (b) Five areas to embed a bit 0

or attacks. Therefore, the Δ values that are closer to the boundary areas may move to the adjacent areas, which causes errors during the extraction process. In order to eliminate this issue, Δ is distanced from boundary areas by a value k . If k has a larger value, the watermark becomes more robust; while the image quality will diminish. Thus choosing an appropriate value for k as well as for Th is challenging. In fact, these two values, Th and k values make a trade-off between imperceptibility and robustness of the watermark. In the proposed scheme, the value of k is considered as one-tenth of the value of Th .

The next step of the proposed embedding process is to obtain LL'_1 , which is

$$LL'_1 = U'.S.V^T \quad (6)$$

where U' is obtained by modifying the two elements of U as explained above. Finally, the inverse of DWT is applied on LL'_1 to achieve the watermarked image.

Figures 3 and 4 show the pseudo-code and the schematic view of the process of embedding the watermark, respectively.

4.2 The process of extracting the watermark

In the proposed scheme, the watermark is recovered blindly. That means during the extraction process, the original image is not necessary at all. The watermark image and

Input: $M \times N$ host image I , $n \times n$ binary watermark W , and the threshold values (Th) that have been optimized by ODFA

Output: The watermarked image

1. Select the appropriate blocks using the approach presented in Section 3.3
2. for each selected block i //embed a watermark bit in a block
 - 2.1. Apply DWT, and SVD on the LL_1 sub-band of the block i ,
// $u_{2,1}, u_{3,1}$ are second and third elements of the first left-singular vector of SVD of LL_1 sub-band,
 - 2.2. $k = Th(i)/10$;
 - 2.3. $p = k/5$;
 - 2.4. $\Delta = u_{2,1} - u_{3,1}$;
 - 2.5. if $W(i) == 1$ //embedding bit value 1
 - if $\Delta > Th(i) - k$
 - while $\Delta \geq Th(i) - k$
 - $u_{2,1} = u_{2,1} - p$;
 - else if $\Delta > -Th(i)/2$
 - while $\Delta \leq k$
 - $u_{2,1} = u_{2,1} + p$;
 - else
 - while $\Delta \geq -Th(i) - k$
 - $u_{2,1} = u_{2,1} - p$;
 - 2.6. else if $W(i) == 0$ //embedding bit value 0
 - if $\Delta < -Th(i) + k$;
 - while $\Delta \leq -Th(i) + k$
 - $u_{2,1} = u_{2,1} + p$;
 - else if $\Delta < Th(i)/2$
 - while $\Delta \geq -k$
 - $u_{2,1} = u_{2,1} - p$;
 - else
 - while $\Delta \leq Th(i) + k$
 - $u_{2,1} = u_{2,1} + p$;
 - 2.7. Apply SVD and DWT inverse operations to construct the block i of the output watermarked image.

Fig. 3 The Pseudo-code of the embedding process

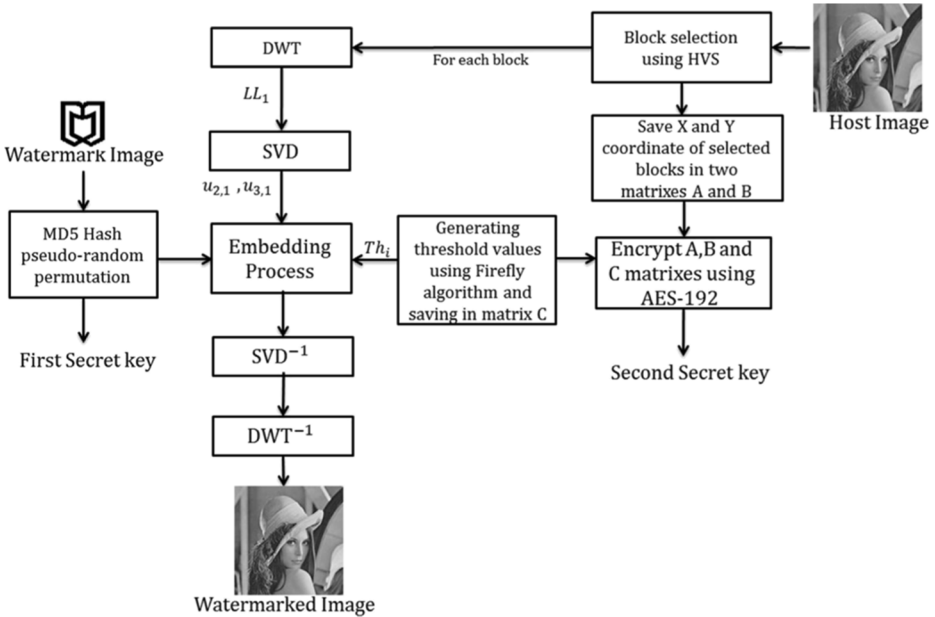


Fig. 4 The schematic view of the proposed embedding process

two secret keys are necessary to extract the watermark. Secret keys should be sent to authorize party using a covert channel. The watermark extraction process is as follows. At first, the watermarked image is divided into equal non-overlapping 8×8 blocks and the blocks with low and average irregularity levels are selected. Also, the threshold values are specified by decrypting the encoded matrices A , B and C . One watermark bit have been embedded in each block, thus for each target block the followings are applied to extract the corresponding watermark bit. The one-level DWT is applied to the specified block of the watermarked image and the LL'_1 sub-band is calculated. Then, SVD is applied on LL'_1 sub-band to achieve the matrix U . After, the difference of elements $u_{2,1}$ and $u_{3,1}$ (Δ) are computed using Eq. (5). According to Fig. 5, if Δ is in area 2 or 4, the bit value 1 is extracted, otherwise the bit value 0 is extracted. At last, after extracting all bits hash inverse permutation is performed on the watermark bits using the first secret key, to reconstruct the watermark image.

Also the pseudo-code and the schematic view of the process of extracting the watermark are shown in Figs. 6 and 7, respectively.

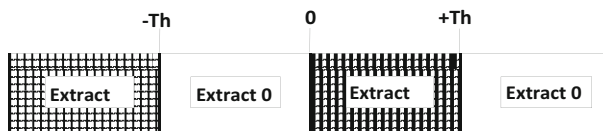


Fig. 5 Extraction areas for bit value 0 and 1

```

Input:  $M \times N$  watermarked image  $I$ , second secret key
Output: The extracted watermarked
1. Find target blocks using the second secret key and AES-192,
2. for each selected block  $i$  //extract a watermark bit from a block
  2.1. Apply DWT, and SVD on the  $LL_1$  sub-band of the block  $i$ ,
      //  $u_{2,1}, u_{3,1}$  are second and third elements of the first left-singular vector of SVD of  $LL_1$  sub-band,
  2.2.  $\Delta = u_{2,1} - u_{3,1}$ ;
  2.3. if  $(\Delta > Th(i))$  or  $(-Th(i) < \Delta < 0)$  //extracting bit value 0
       $W(i) = 0$ ;
  2.4. if  $(\Delta < -Th(i))$  or  $(0 < \Delta < Th(i))$  //extracting bit value 1
       $W(i) == 1$ ;
    
```

Fig. 6 The Pseudo-code of the extracting process

4.3 Finding optimal threshold values by ODFA

As it was mentioned previously, a threshold value determines the strength of the watermark and controls its robustness and imperceptibility. Most existing methods use the same threshold value for all blocks. Since, all blocks do not have the same texture, using the same threshold value causes that the image quality to reduce. In order to overcome this issue in this paper, for each block, a separate threshold value is applied in a specific range. Therefore, for each watermark bit one threshold value should be defined. Since an optimal values for threshold is desired, using an optimization algorithm is inevitable. The ODFA as an optimization algorithm is used to find the optimal threshold values adaptively. Thus, each firefly is a D -dimensional vector, where D is the number of watermark bits which is at most $2/3((M/8) \times (N/8))$ as mentioned in Section 4.1.

To define the objective function of ODFA, different attacks are applied to the watermarked image to evaluate the quality of the thresholds and the watermark is then extracted from them. The objective function of the optimization problem is a linear combination of the quality of the watermarked image (PSNR) and the quality of the extracted watermark (NC) after encountering attacks.

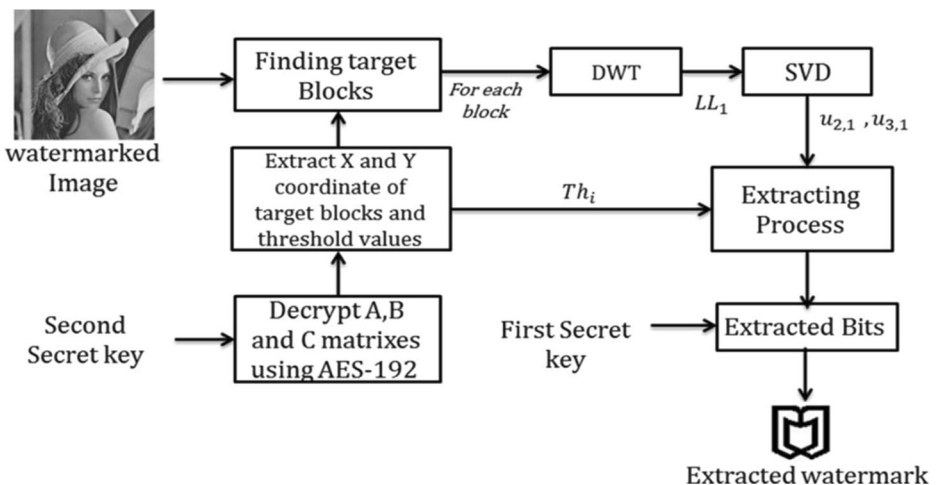


Fig. 7 The schematic view of the proposed embedding process

The peak signal to noise ratio is a ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation and is measured in decibels. The Root Mean Square Error (RMSE) between the $M \times N$ original image I and the $M \times N$ modified images \bar{I} is calculated as,

$$RMSE = \sqrt{\frac{1}{MN} \sum_{M,N} [I(i,j) - \bar{I}(i,j)]^2} \quad (7)$$

and then peak signal to noise ratio given by:

$$PSNR = 10 \log_{10} (255/RMSE) \quad (8)$$

The Normalized Correlation (NC) is the similarity between the reference (original) watermark w and extracted watermark \bar{w} and is defined as:

$$NC = \frac{\sum_{i=1}^M \sum_{j=1}^N [w(i,j) \cdot \bar{w}(i,j)]}{\sum_{i=1}^M \sum_{j=1}^N [w(i,j)]^2} \quad (9)$$

NC is a value between 0 and 1. The closer this value to 1, there is higher similarity between the original and extracted watermark [14]. The objective function of the optimization problem is given by:

$$f = |PSNR_{ideal} - PSNR_t| + \sum_{i=1}^N (1 - NC_i) \quad (10)$$

where N is the number of attacks, NC_i is the normalized correlation of extracted watermark corresponding to the i^{th} attack, and $PSNR_t$ is the PSNR value that is obtained after embedding the watermark using the optimum (minimum) values of the thresholds. $PSNR_{ideal}$ is the ideal value of the PSNR, which is set to 52 in our experiments. This is a minimization problem. Thus, as the value of f decreases, the $PSNR_t$ value approaches $PSNR_{ideal}$ and the NC_i value approaches 1 in Eq. (10).

Eventually, the ODFA that is shown in Fig. 1 is applied to find optimal threshold values for each selected block. At each solution that is represented by a firefly, the watermark image is embedded by using the threshold values given by this solution according to the method presented in Section 4.1. Then at the objective function evaluation, N different attacks are applied to this watermarked image and the objective function is calculated by Eq. (10) and the steps of the algorithm given in Fig. 1 will continue.

5 Experimental results

In this Section the results of the proposed watermarking scheme naming DWT-SVD-ODFA are compared to the same method where a simple firefly algorithm is used as optimizer, which is called here DWT-SVD-FA and the base method where a constant threshold value (0.006) and consequently no optimizer is used, that is called DWT-SVD. Six standard test images of size 512×512 from USC-SIPI database [1] are used as host images that are given in Fig. 8a-f. Binary logo of the University of Jirort-Jiroft-IRAN of size 32×32 that is shown in Fig. 8g is taken as the watermark to verify image copyright. In all experiments, the host and watermark image sizes are the same. In

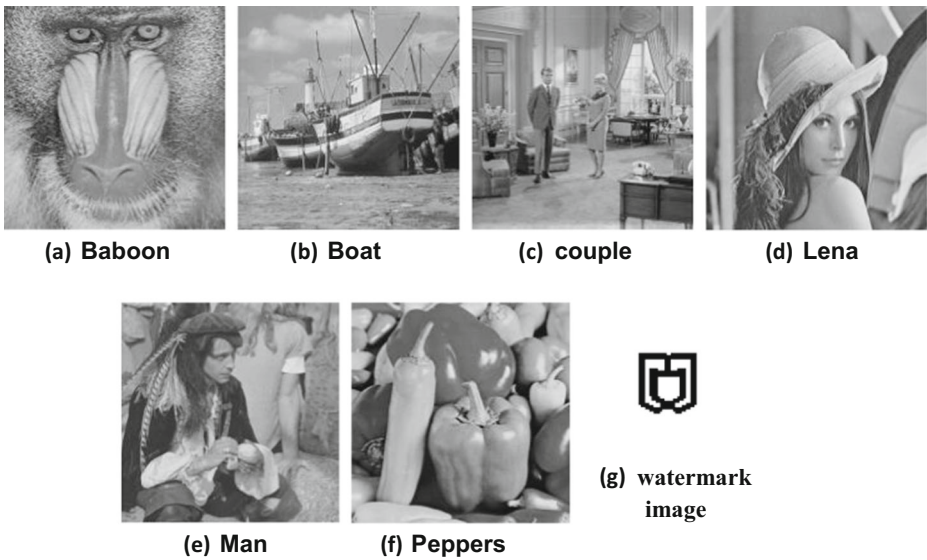


Fig. 8 (a-f) Test images, (g) watermark image

order to evaluate the robustness of the proposed scheme, ten common image processing operations are applied as attacks. Table 1 describes these attacks briefly.

The parameter settings of ODFA and FA are $\beta_0 = 1$, $\alpha = 0.01$ and $\gamma = 1$ that are adjusted from [26]. The total number of initial fireflies is 10, and also the number of iterations is 100. Number of dimensions of each firefly should be the same as the number of watermark bits. Indeed, to embed an $n \times n$ watermark, n^2 threshold values are needed that implies each firefly should have n^2 dimensions. In the proposed scheme to reduce the time complexity the dimension of each firefly is considered as $n^2/10$, thus for each ten adjacent blocks in the chain an equal threshold value is taken. The rational is that, the adjacent blocks in a chain have similar entropy and texture, therefore using the same threshold value for them does not affect on the efficiency, while decreasing the time complexity.

Threshold values are the most important values that have great influence on the imperceptibility and robustness of the watermarking scheme. Therefore finding the appropriate interval to vary these values is another important aspect. To achieve this goal, we designed an experiment that examines the effect of different values of the threshold on the two objectives;

Table 1 Description of ten different attacks used for the robustness analysis

Attack#	Attack description
1	Median filtering with window size 3×3
2	Gaussian low pass filter with window size 9×9
3	Gamma correction with gamma value 0.2
4	Histogram equalization
5	Sharpening
6	Salt & pepper noise with noise density 0.01
7	Cropping 1/8 from center of image
8	Rescaling 512_256_512
9	Anticlockwise rotation by 5°
10	JPEG compression with quality factor 75

the PSNR and the NC values as the measure of imperceptibility and robustness of the watermark. Figure 9 shows two plots of PSNR values and mean values of NC versus different threshold values. In this experiment, the DWT-SVD watermarking scheme is performed on the six test images with threshold values varies from 0.002 to 0.016. PSNR plots show the PSNR values of each image with different values of thresholds and NC plots show the mean of NC values obtained by applying ten attacks mentioned on Table 1 on each image with different values of thresholds.

As Fig. 9a shows for all test images, the PSNR value is higher than 43 db where the threshold values approach to 0.01 and the threshold values greater than 0.01 worsen the PSNR value. Thus threshold values less than 0.01 are suitable. On the other hand as shown in Fig. 9b, increasing the threshold values increases the NC value, too. Let us accept the NC values greater than 0.8, where the threshold value goes upper than 0.004. Thus the interval [0.004, 0.01] for threshold values is an acceptable range.

In order to show the overall preference of the proposed scheme the comparison of the objective function values of two watermarking schemes based on the FA (DWT-SVD-FA and DWT-SVD-ODFA) is presented in Fig. 10. These results are obtained on the Lena test image as the host image. The results obtained on 30 different runs and the best ones are shown. According to the Fig. 10, the proposed scheme using DWT-SVD-ODFA outperforms DWT-SVD-FA at all iterations. Specially, the DWT-SVD-FA scheme is at least 0.521 after 53 runs, while the DWT-SVD-ODFA reaches lower than 0.504 after just 25 runs. This improvement is due to the initial population generation based on the opposition theory, as well as updating all dimensions of fireflies in each iteration. After the 53 runs, DWT-SVD-FA converges to 0.521. This is due to the fact that this approach does not search all dimensions that causes it to get in the local optimum. DWT-SVD-ODFA converges to 0.504 at the 25th run, which indicates its faster convergence while finding the more optimal solution. Also the first watermarking scheme, i.e., DWT-SVD with the fixed threshold value for all blocks that is 0.006 is examined on the Lena image and the best objective function value that it achieves is 1.0769 which is worse than both watermarking schemes with threshold optimization.

The rest of this Section discusses the imperceptibility and robustness of the proposed scheme in comparison to the results of Musrati et al. [2], Loukhoukha et al. [17], Lai et al. [11] and Kazemivash and Ebrahimi Moghaddam [10]. Watermarking schemes proposed by [2, 10, 17], were briefly reviewed in the introduction. They employ the artificial bee colony, firefly and ant colony algorithms to optimize the multiple scaling factors, respectively. Also in

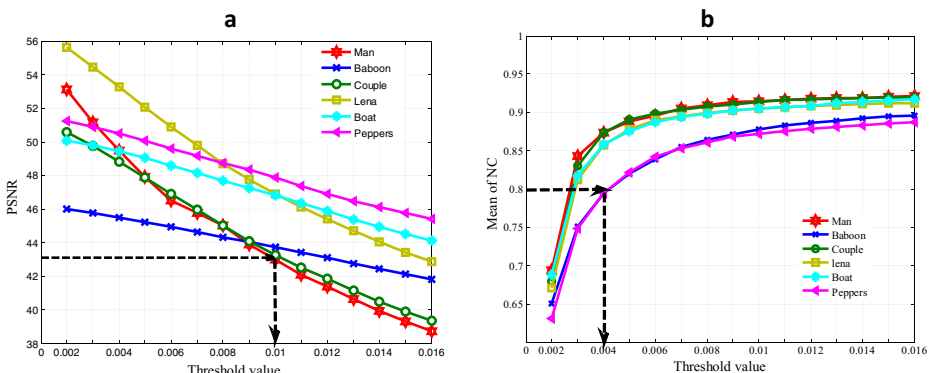


Fig. 9 Effects of different threshold values (a) on PSNR, (b) on NC values

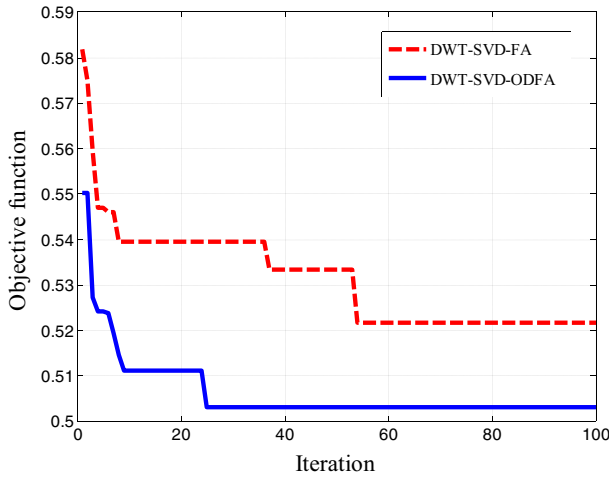


Fig. 10 Average objective function values of DWT-SVD-FA compared to DWT-SVD-ODFA for 100 iterations

[11], in order to embed the watermark, coefficients of matrix U resulting from SVD are changed using a single strength factor. This strength factor (single strength factor) is determined manually and it is constant for all blocks. The recommended parameter settings from the corresponding studies are taken. In what follows, the results are investigated from three standpoints; imperceptibility, robustness, and security. The best results among the comparison in Tables 3, 4, 5 and 6 are highlighted in bold faced.

In order to evaluate the imperceptibility of a watermarked image, it is compared with the original host image. In this evaluation, peak signal-to-noise ratio (PSNR) is usually used. The full description of calculating the PSNR was presented at Section 5. Here, the structural similarity index measurement (SSIM) as a measure of the quality of the watermarked images is applied, too. Despite of MSE or PSNR, SSIM considers image degradation as *perceived change in structural information*, incorporating important perceptual phenomena, including loss of correlation, luminance masking and contrast masking terms. SSIM of an $M \times N$ host image I and watermarked image \bar{I} of same size is given by:







$$SSIM(I, \bar{I}) = l(I, \bar{I}) \cdot c(I, \bar{I}) \cdot s(I, \bar{I}) \tag{11}$$

where

$$\begin{cases} l(I, \bar{I}) = \frac{2\mu_I\mu_{\bar{I}} + c_1}{\mu_I^2 + \mu_{\bar{I}}^2 + c_1} \\ c(I, \bar{I}) = \frac{2\sigma_I\sigma_{\bar{I}} + c_2}{\sigma_I^2 + \sigma_{\bar{I}}^2 + c_2} \\ s(I, \bar{I}) = \frac{\sigma_{I\bar{I}} + c_3}{\sigma_{\bar{I}\bar{I}} + c_3} \end{cases} \tag{12}$$

$l(I, \bar{I})$ measures the closeness, $c(I, \bar{I})$ measures the closeness of the contrast and $s(I, \bar{I})$ measures the correlation coefficient between the two images. μ_I and $\mu_{\bar{I}}$ are the mean luminance

Table 2 The PSNR, the SSIM and the NC values and the extracted watermark (visual) of different proposed schemes

Proposed Schemes	Watermarked image and PSNR	The Watermark (logo image) and NC
DWT-SVD	 PSNR = 50.8634 SSIM = 0.9820	 NC = 1
DWT-SVD-FA	 PSNR = 51.9805 SSIM = 0.9842	 NC = 1
DWT-SVD-ODFA	 PSNR = 52.1906 SSIM = 0.9877	 NC = 1

and σ_I and $\sigma_{\bar{I}}$ are the standard deviation of I and \bar{I} , respectively. $\sigma_{I\bar{I}}$ is the covariance between I and \bar{I} . c_1, c_2 , and c_3 are three positive constants to stabilize the division with weak denominator. SSIM is a value between $[0, 1]$ and values close to 1 represent higher similarity between I and \bar{I} .

At first the visual results of Lena image after watermarking are shown in Table 2, they are the extracted watermark, the PSNR, the SSIM and the NC values of three proposed comparative watermarking schemes (DWT-SVD, DWT-SVD-FA and DWT-SVD-ODFA). In all these three schemes the extracted watermark is identical to original one and $NC = 1$, while the PSNR and SSIM values of the proposed DWT-SVD-ODFA scheme is the highest value.

Also, Table 3 presents the PSNR and the SSIM values of the proposed DWT-SVD-ODFA scheme comparing to DWT-SVD and DWT-SVD-FA, for different test images. It is obvious from the results of Tables 2 and 3 that the PSNR and the SSIM values of the proposed DWT-

Table 3 The PSNR and the SSIM results on different host images of DWT-SVD, DWT-SVD-FA and DWT-SVD-ODFA

Test image	Proposed Schemes					
	DWT-SVD		DWT-SVD-FA		DWT-SVD-ODFA	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Baboon	45.4059	0.9805	53.2033	0.9889	53.3506	0.9890
boat	46.8707	0.9822	52.8992	0.9885	53.0788	0.9883
Couple	47.4611	0.9812	51.3201	0.9831	51.0125	0.9860
Lena	50.9063	0.9820	51.9911	0.9842	52.1906	0.9877
Man	46.5113	0.9801	52.9855	0.9869	54.1380	0.9896
Pepper	49.6039	0.9813	52.5020	0.9880	51.9802	0.9814
Average	47.7932	0.9812	52.4835	0.9866	52.6251	0.9870

Table 4 The PSNR and the SSIM results on different host images of DWT-SVD-ODFA comparing to the existing schemes

Test image	Watermarking Schemes									
	Lai. et al. scheme [11]		Loukhaoukha et al. scheme [17]		Ali. et al. scheme [2]		Kazemivash and Ebrahimi Moghaddam [10]		DWT-SVD-ODFA	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Baboon	40.3577	0.9741	52.1355	0.9870	45.0148	0.9799	38.6945	0.9725	53.3506	0.9890
boat	35.9505	0.9725	54.8104	0.9888	46.3311	0.9826	38.4532	0.9705	53.0788	0.9883
Couple	42.9106	0.9782	51.3022	0.9845	45.0038	0.9795	38.9544	0.9750	51.0125	0.9860
Lena	43.0013	0.9792	49.9190	0.9811	47.0207	0.9803	38.9981	0.9766	52.1906	0.9877
Man	44.5119	0.9800	50.5641	0.9815	49.7290	0.9818	38.5562	0.9736	54.1380	0.9896
Pepper	43.2366	0.9785	50.1409	0.9802	45.1866	0.9792	38.7377	0.9759	51.9802	0.9814
Average	41.6614	0.9770	51.4786	0.9838	47.0207	0.9805	38.7399	0.9740	52.6251	0.9870

SVD-FA and DWT-SVD-ODFA schemes are all higher than 51 db and 0.9814 which indicates the acceptable imperceptibility given by the proposed schemes. In the DWT-SVD approach, since all threshold values are taken the same and also the image texture features are ignored, it has achieved poor results.

Table 4 presents the PSNR and SSIM results of the proposed DWT-SVD-ODFA scheme in comparison to the schemes proposed in [2, 10, 11, 17] for different images. The scheme that is proposed by Lai. et al. [11] uses single strength factor to embed the watermark and consequently its results are poor. The approaches proposed in [2, 17] get better results than Lai. et al. [11], since they have used multiple threshold values for embedding and they have applied artificial bee and ant colony optimizations (ABC and ACO), respectively. Although scheme [10] has used firefly algorithm as optimizer but the PSNR values are low and it is because of its high robustness that affects its imperceptibility. However, the proposed DWT-SVD-ODFA outperforms all these schemes. The rational is behind of using ODFA for optimization of multiple threshold values as well as inverting the bit embedding process when large coefficient variations are necessary.



Fig. 11 Watermarked Lena image after applying 10 attacks listed in Table 1

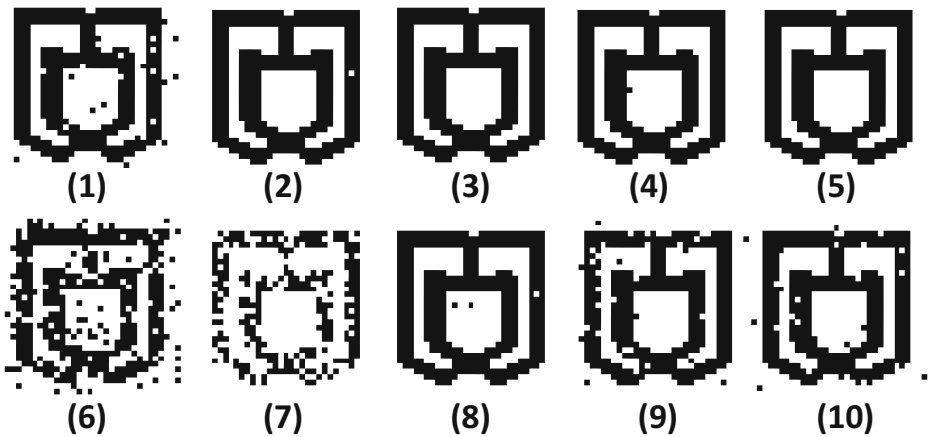


Fig. 12 The extracted watermarks from Lena image after applying 10 attacks listed in Table 1

The next experiments are designed to investigate the robustness of the watermarking process against the several attacks mentioned in Table 1. Accordingly, attacks are applied to the watermarked image one by one and the watermark is extracted and then evaluated. The similarity of the original and extracted watermarks is computed by normal correlation (NC) (by using Eq. (9)).

The NC results of the proposed DWT-SVD-ODFA scheme, always becomes 1 without applying any attacks. Figure 11 presents the watermarked Lena image after applying the attacks given in Table 1 and Fig. 12 presents the extracted watermarks. It is clear that in all cases after applying each attack, the recovered watermark is clear and visible. However, by applying the attack number 7, which is cropping 1/8 of the image from center of it (that is a harmful attack), the quality of the extracted watermark is diminished drastically. To clarify, Table 5 presents the numerical values of the NC results of the proposed schemes for Lena image after applying 10 attacks listed in Table 1. At all cases the NC results of the proposed schemes are encouraging especially at the third proposed scheme, DWT-SVD-ODFA the obtained NC results are the highest in case of average. The proposed scheme proved experimentally that by using multiple threshold values and using ODFA as the optimizer the robustness will improve.

Table 5 The NC results of the proposed schemes DWT-SVD, DWT-SVD-FA, and DWT-SVD-ODFA for Lena image

Attack No.	Proposed Schemes		
	DWT-SVD	DWT-SVD-FA	DWT-SVD-ODFA
1	0.9697	0.9769	0.9725
2	0.9980	0.9980	0.9998
3	1.0000	1.0000	1.0000
4	0.9970	0.9980	0.9989
5	0.9717	1.0000	1.0000
6	0.8701	0.8763	0.8830
7	0.8301	0.8938	0.8950
8	0.8320	0.9933	0.9990
9	0.9638	0.9630	0.9705
10	0.9307	0.9679	0.9752
Average	0.9363	0.9665	0.9693

Table 6 The *NC* results of the proposed scheme DWT-SVD-ODFA comparing with existing schemes

Attack No.	Proposed Schemes				
	Lai.et al. scheme [11]	Loukhaoukha et al. scheme [17]	Ali.et al. scheme [2]	Kazemivash and Ebrahimi Moghaddam [10]	DWT-SVD-ODFA
1	0.9134	0.9102	0.9002	0.9899	0.9725
2	0.9801	0.9825	0.9971	1	0.9998
3	0.9831	0.9950	0.9990	0.9977	1.0000
4	0.9792	0.9910	0.9982	1	0.9989
5	0.9705	0.9816	0.9980	1	1.0000
6	0.9016	0.8617	0.9105	0.9295	0.8830
7	0.8388	0.9158	0.8547	0.9912	0.8950
8	0.9508	0.9212	0.9341	1	0.9990
9	0.9457	0.9606	0.9547	0.9025	0.9705
10	0.9375	0.8960	0.9580	0.9965	0.9752
Average	0.9400	0.9415	0.9504	0.98073	0.9693






Also the proposed scheme DWT-SVD-ODFA is compared with three mentioned existing schemes [2, 10, 11, 17]. Table 6 presents the *NC* results after applying 10 different attacks to the watermarked Lena images. The average *NC* results are also presented to show a fair comparison. According to the results of Table 6, the least average *NC* value corresponds to scheme proposed in [11], which uses single scaling factors, while schemes [2], [10, 17] show higher robustness against attacks. That is due to finding multiple threshold values using meta-heuristic approaches. Scheme [10] has highest *NC* values but in cost of low transparency. Anyway, in most cases, the robustness of our proposed schemes is high. Selecting blocks based on the image texture and optimal selection of threshold values using oppositional dimensional based optimization in the firefly algorithm has led to the high robustness of the proposed scheme.

Most watermarking methods, which use SVD, are prone to false positive errors due to embedding singular value of the watermark within the image. In other words, a watermark may be extracted from an image that really any watermark does not embedded in it. In the proposed scheme, the watermark is revealed blindly and since watermark bits are fully embedded in the image, it is not likely that this error occurs. During the extraction process, the chain of blocks with low and average irregularities and threshold values are required as the second secret key to find the blocks containing watermark bits. Moreover, the first secret key that is used to shuffle the order of watermark bits is necessary to reveal the watermark image. The existence of these two keys prevent unauthorized parties to reveal the watermark and increases the security level of the proposed watermarking scheme.

Table 7 Capacity results of the proposed scheme DWT-SVD-ODFA comparing with existing schemes






	Proposed Schemes				
	Lai.et al. scheme [11]	Loukhaoukha et al. scheme [17]	Ali.et al. scheme [2]	Kazemivash and Ebrahimi Moghaddam [10]	DWT-SVD-ODFA
Watermark Size	32 × 32	32 × 32	32 × 32	32 × 32	52 × 52
Host Image Size	512 × 512	256 × 256	512 × 512	512 × 512	512 × 512
Capacity	0.0039	0.0156	0.0039	0.0039	0.0103

Table 8 The *PSNR*, the *SSIM* and the *NC* values for five high texture images

High texture images					
PSNR	47.9624	48.4162	46.2222	49.9385	49.9284
SSIM	0.9968	0.9984	0.9913	0.9954	0.9965
Attack No.	NC Values				
1	0.6992	0.6768	0.6436	0.6201	0.6035
2	0.8506	0.8350	0.7363	0.7988	0.7676
3	0.7646	0.6875	0.6943	0.7256	0.6611
4	0.9763	0.9611	0.9564	0.9685	0.9772
5	0.9457	0.9379	0.8725	0.9832	0.9246
6	0.8682	0.8584	0.8496	0.8789	0.8701
7	0.8988	0.8793	0.9256	0.8617	0.8842
8	0.7256	0.6543	0.6123	0.6367	0.6182
9	0.9563	0.9814	0.9535	0.9645	0.9553
10	0.9763	0.9848	0.9645	0.9743	0.9706
Average	0.8662	0.8457	0.8209	0.8412	0.8232

The embedding rate is used to analyze the capacity of the watermarking scheme. The embedding rate is the amount of embedded bits divided by the amount of all pixels in host image and is presented by bit per pixel (b/p) [28]. For example, for a 512×512 host image and a 52×52 binary image (the watermark), the capacity is $(52 \times 52)/(512 \times 512) = 0.0103$ (b/p). As mentioned in Section 4.1, in the proposed watermarking scheme the length of the watermark is at most $2/3((M/8) \times (N/8))$ bits. Table 7 presents the capacity of the proposed DWT-SVD-ODFA and three mentioned existing schemes [11, 13, 24, 31]. Using low and medium informative blocks and ignoring high informative blocks causes low capacity of the proposed watermarking scheme, however the capacity of our proposed scheme is greater than the capacity of the methods proposed in [11, 13, 31]. The embedding rate of Loukhaoukha et al. scheme [24] outperforms ours, instead according to the results of Tables 4 and 6 our

Table 9 The *PSNR*, the *SSIM* and the *NC* values for five low texture images

Low texture images					
PSNR	48.8731	50.3354	47.8089	49.3527	49.3233
SSIM	0.9897	0.9921	0.9882	0.9889	0.9889
Attack No.	NC Values				
1	0.9980	0.9404	0.9824	1.0000	1.0000
2	1.0000	0.9404	1.0000	1.0000	1.0000
3	1.0000	0.9229	1.0000	0.9414	0.9658
4	0.6093	0.6504	0.0820	0.5133	0.4834
5	1.0000	0.9404	1.0000	1.0000	1.0000
6	0.9336	0.8506	0.9277	0.9199	0.9297
7	0.7715	0.6973	0.7803	0.8428	0.8467
8	1.0000	0.9404	1.0000	1.0000	1.0000
9	0.9844	0.9277	0.9766	0.9902	0.9844
10	1.0000	0.9404	1.0000	1.0000	1.0000
Average	0.9297	0.8751	0.8749	0.9208	0.9210

proposed scheme has higher watermark invisibility and also stronger robustness against the common image processing attacks.

In order to examine the performance of the proposed method on the high and low texture images, five images from each category are selected and tested under proposed algorithm. Visual and numerical results as the *SSIM*, *PSNR* and *NC* values of high and low texture images after watermarking are shown in Tables 8 and 9, respectively. As the results point out, *SSIM* values for high texture images are higher than low texture ones in most cases but *PSNR* values are almost in the same ranges. High texture images are more vulnerable against gamma correction, resizing, median and Gaussian filtering while low texture ones have lower results for histogram equalization and cropping. In general for all images with different textures, the results are acceptable.

6 Conclusions

This paper proposed a watermarking scheme using multiple thresholds in the SVD-DWT domain. Image blocks, to embed watermark bits, are selected according to the human visual system. The watermark bits are embedded by changing the coefficients of singular matrices. The amounts of these variations are determined by a threshold parameter. Since, blocks textures are not the same, different thresholds are used to embed the watermark bits. Moreover, the threshold values are optimized using the recently published oppositional dimensional-based firefly algorithm (ODFA), which is a modified version of the FA. By using the ODFA the imperceptibility and robustness of the watermark increases significantly. The robustness of the proposed scheme was evaluated using 10 different processing tasks on six standard test images. Experimental results indicate an increase in transparency and robustness of the proposed scheme in comparison to other similar approaches. The proposed scheme is the first scheme that applies the ODFA in image watermarking to optimize the multiple threshold values. Furthermore, the watermark is extracted blindly and thus, it is not likely to occur the false positive errors. Despite the good robustness of the proposed scheme against attacks, the robustness is limited against rotation and cropping. Our future works will include improving the robustness against such attacks and will generalize this approach to color images and videos.

References

1. (2014) <http://sipi.usc.edu/database/database.php>
2. Ali M, Ahn CW, Pant M, Siarry P (2015) An image watermarking scheme in wavelet domain with optimized compensation of singular value decomposition via artificial bee colony. *Inf Sci* 301:44–60
3. Das C, Panigrahi S, Sharma VK, Mahapatra KK (2014) A novel blind robust image watermarking in DCT domain using inter-block coefficient correlation. *AEU-Int J Electron C* 68(3):244–253
4. Dong H, He M, Qiu M (2015) Optimized Gray-Scale Image Watermarking Algorithm Based on DWT-DCT-SVD and Chaotic Firefly Algorithm. In: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), p 310–313
5. Elshazly EH, Faragallah OS, Abbas AM, Ashour MA, El-Rabaie ESM, Kazemian H, ..., Elsayed HS (2015) Robust and secure fractional wavelet image watermarking. *Signal, Image and Video Processing* 9(1):89–98
6. Fan MQ, Wang HX, Li SK (2008) Restudy on SVD-based watermarking scheme. *Appl Math Comput* 203(2):926–930

7. Horng M-H, Jiang T-W (2010) Multilevel image thresholding selection based on the firefly algorithm. In: Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2010 7th International Conference on, p 58–63
8. Ishtiaq M, Sikandar B, Jaffar MA, Khan A (2010) Adaptive watermark strength selection using particle swarm optimization. *ICIC Express Letters* 4(5):1–6
9. Kazemivash B, Moghaddam ME (2017) A robust digital image watermarking technique using lifting wavelet transform and firefly algorithm. *Multimed Tools Appl* 76(20):20499–20524
10. Kazemivash B, Moghaddam ME (2017) A predictive model-based image watermarking scheme using Regression Tree and Firefly algorithm. Springer, Berlin, Heidelberg, pp 1–16
11. Lai CC (2011) An improved SVD-based watermarking scheme using human visual characteristics. *Opt Commun* 284(4):938–944
12. Lai CC (2011) A digital watermarking scheme based on singular value decomposition and tiny genetic algorithm. *Digital Signal Processing* 21(4):522–527
13. Lai CC, Tsai CC (2010) Digital image watermarking using discrete wavelet transform and singular value decomposition. *IEEE Trans Instrum Meas* 59(11):3060–3063
14. Lewis J (1995) Fast normalized cross-correlation. In: Vision interface, p 120–123
15. Li Q, Yuan C, Zhong YZ (2007) Adaptive DWT-SVD Domain Image Watermarking Using Human Visual Model. In: The 9th International Conference on Advanced Communication Technology, p 1947–1951
16. Lin TC, Lin CM (2009) Wavelet-based copyright-protection scheme for digital images based on local features. *Information Sciences* 179(19):3349–3358
17. Loukhaoukha K, Chouinard JY, Taieb MH (2011) Optimal image watermarking algorithm based on LWT-SVD via multi-objective ant colony optimization. *J Inf Hiding Multimed Signal Process* 2(4):303–319
18. Maity SP, Maity S (2009) Multistage spread spectrum watermark detection technique using fuzzy logic. *IEEE signal processing letters* 16(4):245–248
19. O'Ruanaidh JJ, Pun T (1997) Rotation, scale and translation invariant digital image watermarking. In: Proceedings International Conference on Image Processing 1:536–539
20. O'Ruanaidh J, Pun T (1997) Rotation, translation and scale invariant digital image watermarking. In: International Conference on Image Processing. Santa Barbara, p 536–536
21. Pal NR, Pal SK (1989) Object-background segmentation using new definitions of entropy. *IEE Proceedings E (Computers and Digital Techniques)* 136(4):284–295
22. Ramkumar M, Akansu N, Alatan A (1999) A Robust Data Hiding Scheme for Images using DFT. In: International Conference on Image Processing. Japan, p 211–215
23. Rivest RL (1992) The MD5 message-digest algorithm. Request for comments (RFC) 1321. Available: <https://www.ietf.org/rfc/rfc1321.txt>
24. Santhi V, Thangavelu A (2009) DWT-SVD combined full band robust watermarking technique for color images in YUV color space. *Int J Comput Theory Eng* 1(4):424
25. Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech J* 27(3):379–423
26. Verma OP, Aggarwal D, Patodi T (2016) Opposition and dimensional based modified firefly algorithm. *Expert Syst Appl* 44:168–1
27. Wang YR, Lin WH, Yang L (2011) An intelligent watermarking method based on particle swarm optimization. *Expert Syst Appl* 38(7):8024–8029
28. Wu HT, Huang J (2012) Reversible image watermarking on prediction errors by efficient histogram modification. *Signal Process* 92(12):3000–3009
29. Yang X-S (2008) Nature-inspired metaheuristic algorithms. Luniver Press, Beckington, pp 242–246
30. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. *Int J Bio Inspir Com* 2(2):78–84
31. Zhang Y, Wu L (2012) A novel method for rigid image registration based on firefly algorithm. *International Journal of Research and Reviews in Soft and Intelligent Computing (IJRRSIC)* 2(2):141–146



Elham Moeinaddini received her B.Sc. and M.Sc. degree in computer science from Shahid Bahonar University of Kerman, Iran in 2004, and Delhi University, Delhi, India, in 2007 respectively. She is working in the Department of Electrical Engineering at Jiroft University, Jiroft, Iran, as an instructor since 2012. Her research interests are image processing, watermarking, evolutionary computation and soft computing.



Fatemeh Afsari received her B.Sc. in hardware computer engineering from Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran in 1999. She received her M.Sc. in artificial intelligence from the same department in 2003. She joined soon to Computer Engineering Department of Shahid Bahonar University of Kerman, Kerman, Iran as a faculty member. She received her Ph.D. on intuitionistic fuzzy sets and distance metric learning from Shahid Bahonar University of Kerman, Kerman, Iran in 2013. She is currently an assistant professor in Computer Engineering Department of Shahid Bahonar University of Kerman. Her research interests include Fuzzy sets and systems, Evolutionary computation, Image processing, Machine learning and pattern recognition with primary emphasis on semi-supervised learning.