Journal of Information Hiding and Multimedia Signal Processing ©2011 ISSN 2073-4212
Ubiquitous International Volume **2**, Number **4**, October **2011**

Optimal Image Watermarking Algorithm Based on LWT-SVD via Multi-objective Ant Colony Optimization

Khaled Loukhaoukha
Laval University, Quebec, QC, Canada, G1K 7P4
khaled.loukhaoukha.1@ulaval.ca

Jean-Yves Chouinard and Mohamed Haj Taieb
Laval University, Quebec, QC, Canada, G1K 7P4
jean-yves.chouinard@gel.ulaval.ca; mohamed.haj-taieb.1@ulaval.ca

Received April 2010; revised July 2010

Abstract. *In this paper, a new optimal watermarking scheme based on lifting wavelet transform (LWT) and singular value decomposition (SVD) using multi-objective ant colony optimization (MOACO) is presented. The singular values of the binary watermark are embedded in a detail subband of host image. To achieve the highest possible robustness without losing watermark transparency, multiple scaling factors (MSF) are used instead of a single scaling factor (SSF). Determining the optimal values of the multiple scaling factors (MSF) is a difficult problem. However, to determine these values, a multi-objective ant colony-based optimization method is used. Experimental results show much improved performances in terms of transparency and robustness for the proposed method compared to other watermarking schemes. Furthermore, the proposed scheme does not suffer from the problem of high probability of false positive detections of the watermarks.*

Keywords: Digital watermarking, multi-objective optimization, ant colony algorithm, singular value decomposition, lifting wavelet transform, false positive detection.

1. Introduction.

One of the most important advantage of the numeric era, is the widespread use of Internet and computers as the exchange of digital media became a very easy task. Moreover, voluminous digital media files can be easily exchanged by Peer-to-Peer techniques. This extraordinary technical revolution from analog to numerical technology was not achieved without generating anxiety in terms of the protection of the authors rights since digital media content including audio, video and image, can be quite easy to duplicated, modified and illegally attacked by anyone, and this without deterioration. Thus it became invreasingly important for authors of the digital media documents to protect themselves and secure multimedia documents as they were affected by significant revenue losses. Digital watermarking techniques provides a solution to this problem: the exchange of multimedia documents has been since more secure. Cryptography is used to protect the numerical documents during the transmission phase, but once deciphered they do not have any protection. Digital watermarking was introduced at the beginning of the 1990s, as a second level of technical security protection after encryption. It consists of inscribing invisible secret data (albeit in some cases visible) into the multimedia document to protect. This is done in two stages: watermark embedding and extraction.

Image watermarking begins with an embedding process where an original image I is watermarked using a watermark M and a watermark key K_W. Watermark message M can be either a random sequence, a copyrighted message, a gray-scale or black-and-white image, ownership identifiers or any other multimedia information data. This watermark can also be encrypted using a cryptographic function F_W. At the end of the embedding process, the original image I is slightly modified using the embedding function, denoted by F_M, giving the watermarked image I_W. To further protect the content, an additional cryptographical key, C_W, can be applied. The watermarked image I_W can then be sent via an unsecure transmission channel, or network, where it could be corrupted by attacks leading to an altered watermarked image, $\hat{I}_W$.

For the legitimate recipient, the watermark extraction process begins with the reception of the watermarked image, possibly attacked $\hat{I}_W$ by a third party. It is then decoded using the extracted function, denoted by F_X, to extract the estimated watermark $\hat{W}$. To produce an estimate of the original (watermark) message, $\hat{M}$, an estimated watermark $\hat{W}$ is decrypted using the inverse of the embedding function, F_W^{-1}. If the original image I is required to extract the watermark, then the watermarking scheme is referred to as a *blind watermarking* scheme, otherwise it is referred as a *no blind watermarking* scheme. Figures 1 and 2 depicts a blind watermarking scheme and no blind watermarking scheme.

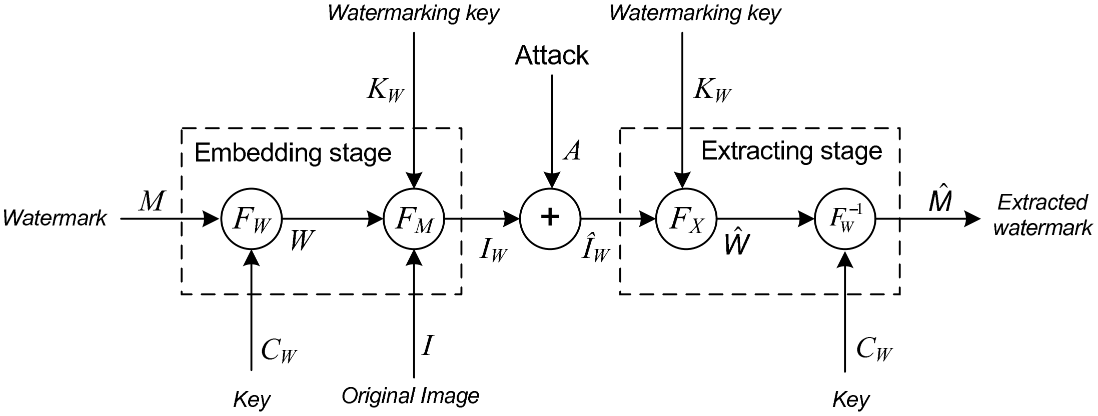


FIGURE 1. Blind watermarking scheme.

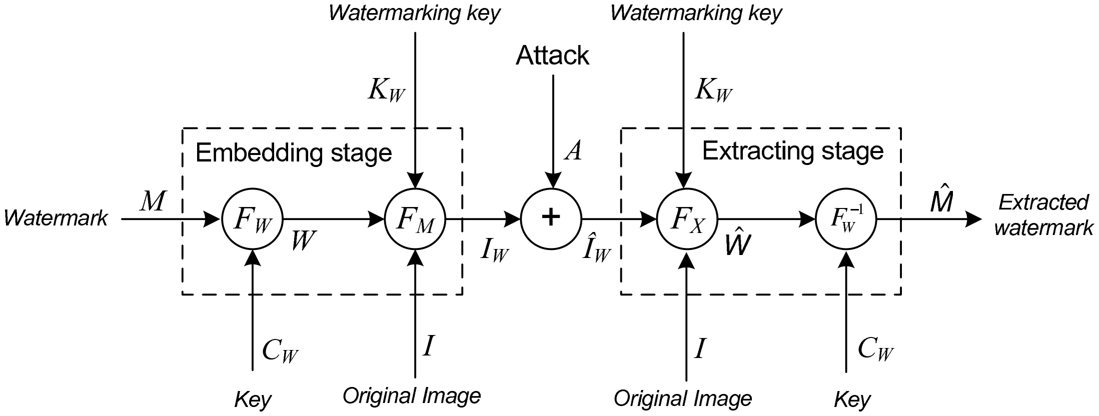


FIGURE 2. No blind watermarking scheme.

In terms of robustness, the watermarking algorithm can be classified into three groups: robust, fragile and semi-fragile. Robust watermarking is designed to be resistant against attacks that attempt to remove or destroy the watermark without degrading the visual quality of the watermarked image significantly. The main objective of robust watermarking is not the verification of the image authenticity, but rather the verification of its origin. Robust watermarking is typically employed for copyright protection and ownership verification. Conversely, fragile watermarking is employed to ensure the integrity and image authenticity rather than verifying the actual ownership. It is designed to detect any unauthorized modification in such a way that slight modifications or tampering on the watermarked image will alter or destroy the watermark. Semi-fragile watermarking combines the properties of fragile and robust watermarking in order to detect unauthorized manipulations while still being robust against authorized manipulations. Semi-fragile watermarking can also be used for authentication.

In general, watermarking algorithms operate either in the spatial domain or in a transform domain such as the discrete cosine transform (DCT) [1, 2, 3], the discrete Fourier transform (DFT) [4, 5, 6], the discrete wavelet transform (DWT) [7, 8] and the ridgelet transform (RIT) [9, 10]. Spatial domain watermarking has the advantage of low calculation complexity compared to that in transform domains. However, most watermarking schemes in the scientific literature operate in the transform domain because it provides enhanced imperceptibility and robustness compared to those in spatial domain. However, decomposition of images in a standard basis set using the transform domain does not necessarily leads to the optimal representation of an image. Therefore, different representations were investigated for watermarking: these include non-negative matrix factorization (NMF) [11] and singular value decomposition (SVD) [12, 13]. Watermarking schemes based on SVD are advantageous since slight changes in the singular values do not affect significantly the image quality. Unfortunately, several SVD-based watermarking schemes based suffer from typically high probability of false positive watermark detections.

In multiplicative and additive embedding schemes, the tradeoff between visual quality and the robustness of the watermarking scheme is controlled by a *single scaling factor* (SSF). Cox et al. [14] suggest to use *multiple scaling factors* (MSF) instead of one. They state that a single scaling factor (SSF) may not be applicable for altering all the pixel values of the original image. The determination of the optimal values for the multiple scaling factors (MSF) for watermarking can be viewed as optimization problem witch is unfortunately a difficult problem. In this paper, we investigate an artificial intelligence based technique to solve this optimization problem. A *multi-objective ant colony algorithm optimization method* (MOACO) is used to determine the optimal values of the multiple scaling factors (MSF) in order to improve the visual quality and to enhance the robustness of the watermarking algorithm.

The paper is organized as follow. Section 2 gives an overview of a MOACO: the ant colony optimization algorithm. In Section 3, we describe a SVD-LWT watermarking algorithm which not suffers from the vulnerability of false positive watermark detections. Section 4 demonstrates the SVD-LWT watermarking algorithm using the multi-objective ant colony optimization. Experimental results to validate the proposed watermarking scheme are discussed in Section 5. Concluding remarks are given in Section 6.

2. **Ant colony optimization principle.** The ant colony optimization (ACO) concept was introduced by Dorigo [15] as a new metaheuristic for the solution of hard optimization problems. It is inspired by observation of real ant colonies. Ants explore randomly the area surrounding their nest in order to find food. If an ant find a food source, it evaluates and carries some food to the nest. During the return travels the ant deposit on the ground

a chemical substance called pheromone trail. Other ants can smell the pheromone and follow it with some probability. This way, ants can communicated via pheromone and find the optimal path between the food source and the nest. This capability of real ant colony to find optimal paths has led to the definition of artificial ant colonies that can find the optimal solution for hard optimization problems [16] such as the traveling salesman problem. The outline of the generic ACO algorithm is presented in Figure 3.

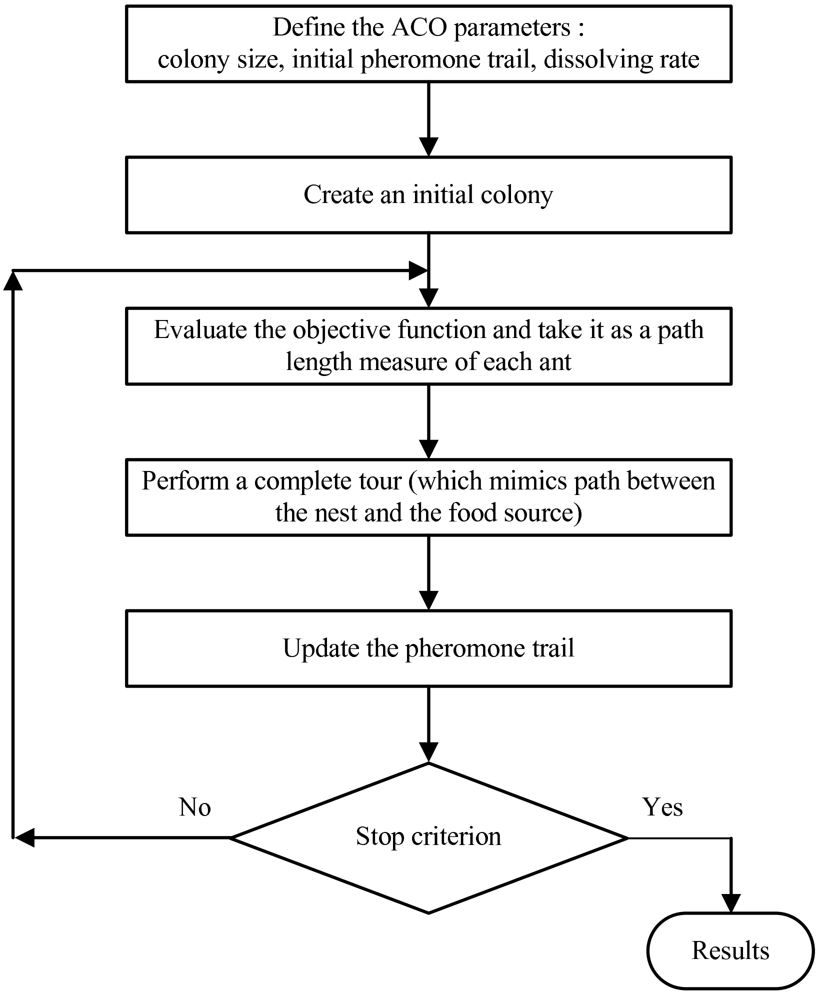

Define the ACO parameters :
colony size, initial pheromone trail, dissolving rate
Create an initial colony
Evaluate the objective function and take it as a path length measure of each ant
Perform a complete tour (which mimics path between the nest and the food source)
Update the pheromone trail
Stop criterion
No
Yes
Results

FIGURE 3. Generic ant colony optimization scheme [17].

The first point to take into account in the ACO algorithm is how the colony is represented. For continuous variables, a colony of m ants is represented as $m \times n$ matrix C, where $C = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_m]^T$ such that $\mathbf{x} = [x_1 \quad x_2 \quad \cdots x_n]$ is a vector of n design variables that corresponds to a single ant. The second point to consider is how to model the pheromone communication scheme. Socha and Dorigo [16] suggest to use a normal distribution for a continuous model implementation:

$$f_{\text{pheromone}}(x) = e^{\frac{-(x-x_{min})^2}{2\sigma^2}} \tag{1}$$

where x_{min} is the optimal point found within the design space and the standard deviation σ as an index of the ants aggregation around the current minimum. To initialize the algorithm, x_{min} is randomly chosen in the design space, using a uniform distribution, and σ is taken at least three times greater than the length of the design space, to uniformly locate the ants within it. As shown in Figure 3, at each iteration, the ACO algorithm updates the values of each design variable and this, for all the ants of the colony, i.e. at each iteration each ant sets the values for the trial solution as per the distribution in (1). At the end, the pheromone distribution over the design space is updated by collecting the information acquired throughout the optimization steps. Since the pheromone is modeled by (1), it is necessary only to update x_{min} and σ as:

$$\sigma = std(\text{colony}) \tag{2}$$

where std(colony) makes use of the colony of ants (candidate solution) to return a vector containing the standard deviation for each design variable [18]. The accumulation of pheromone increases in the vicinity of the candidate towards the optimum solution. However to avoid premature convergence, negative update procedures are not discarded: for this a simple method is used, which consists in dissolving the pheromone. The principle is to spread the amount of pheromone by changing the current standard deviation (for each variable) according to:

$$\sigma_{new} = \gamma \cdot \sigma_{old} \tag{3}$$

where $\gamma > 1$ is the dissolving rate.

3. **Watermarking algorithm based on SVD and LWT.** The SVD-LWT watermarking algorithm presented in [13] is described by the following watermark embedding and extraction processes.

3.1. **Watermark embedding.** Consider an original image I of size $N \times N$ and let the watermark W be a binary image of size $M \times M$. The embedding process is as follows:

Step 1 : Decompose the original image I into $3\ell + 1$ subbands by applying a ℓ-level lifting wavelet transform (LWT).

Step 2 : Select one subband (SB) among the three following subbands: HH_ℓ, HL_ℓ and LH_ℓ.

Step 3 : Compute the inverse LWT of the selected subband (SB):

$$X = LWT^{-1}(SB) \tag{4}$$

Step 4 : Apply a singular value decomposition (SVD) of matrix X:

$$X = U_X \cdot S_X \cdot V_X^T \tag{5}$$

Step 5 : Encrypt watermark W with the chosen cryptographic function to get the enciphered watermark W_C.

Step 6 : Apply a singular value decomposition to watermark matrix W_C.

$$W = U_{W_C} \cdot S_{W_C} \cdot V_{W_C}^T \tag{6}$$

Step 7 : Compute the one-way hash functions for matrices U_{W_C} and V_{W_C}:

$$\begin{cases} H_U = \text{hash}(U_{W_C}) \\ \\ H_V = \text{hash}(V_{W_C}) \end{cases} \tag{7}$$

Step 8 : Matrices U_{W_C} and V_{W_C}, and hash values H_U and H_V are stored in the private key.

Step 9 : Compute matrix S_Y according to:

$$S_Y = S_X + \alpha \cdot S_{W_C} \tag{8}$$

where α is the watermark strength factor that controls the tradeoff between visual quality and robustness of the watermarking scheme.

Step 10 : Compute matrix Z_W, according to:

$$Z_W = U_X \cdot S_Y \cdot V_X^T \tag{9}$$

Step 11 : Compute the lifting wavelet transform of matrix Z_W,

$$SB_W = LWT(Z_W) \tag{10}$$

Step 12 : The watermarked image I_W is computed by applying the inverse ℓ-level lifting wavelet transform to the modified subband SB_W and the 3ℓ unmodified subbands.

3.2. **Watermark extraction.** The proposed algorithm is a no blind algorithm, i.e. the original image I is required for watermark extraction. The following steps summarizes the extraction process:

Step 1 : A safety test is first done: hash values of matrices U_{W_C} and V_{W_C} (possibly altered by an attacker as $\tilde{U}_W$ and $\tilde{V}_W$). These hash values, $H_{\tilde{U}}$ and $H_{\tilde{V}}$, are compared to the hash values stored during the embedding process:

$$\begin{cases} \text{if} \quad H_U = H_{\tilde{U}} \quad \text{and} \quad H_V = H_{\tilde{V}} \longrightarrow \quad \text{go to Step2} \\ \\ \text{if} \quad H_U \neq H_{\tilde{U}} \quad \text{or} \quad H_V \neq H_{\tilde{V}} \longrightarrow \quad \text{stop (probable} \quad \text{attack)} \end{cases} \tag{11}$$

Step 2 : Decompose the original and watermarked images, I and I_W, by applying the ℓ-level lifting wavelet transform.

Step 3 : Select the same subband (SB) used in Step 2 of the watermark embedding process. SB_I and SB_{I_W} are, respectively, the subbands selected for the original and watermarked images.

Step 4 : Compute the inverse lifting wavelet transform of selected subbands SB_I and SB_{I_W}:

$$\begin{cases} X_I & = & LWT^{-1}(SB_I) \\ \\ X_{I_W} & = & LWT^{-1}(SB_{I_W}) \end{cases} \tag{12}$$

Step 5 : Apply the singular value decomposition on matrices X_I and X_{I_W}:

$$\begin{cases} X_I & = & U_{X_I} \cdot S_{X_I} \cdot V_{X_I}^T \\ \\ X_{I_W} & = & U_{X_{I_W}} \cdot S_{X_{I_W}} \cdot V_{X_{I_W}}^T. \end{cases} \tag{13}$$

Step 6 : Compute matrix $S_{\hat{W}_C}$ as follows:

$$S_{\hat{W}_C} = \frac{S_{X_{I_W}} - S_{X_I}}{\alpha} \tag{14}$$

Step 7 : Determine the estimated enciphered watermark, $\hat{W}_C$, by computing:

$$\hat{W}_C = \tilde{U}_{W_C} \cdot S_{\hat{W}_C} \cdot \tilde{V}_{W_C}^T \tag{15}$$

Step 8 : Decrypt the estimated watermark, $\hat{W}_C$, to get the estimated watermark, $\hat{W}$.

3.3. **Protection against false positive detection.** To protect the proposed image watermarking scheme against false positive detections, two countermeasures have been proposed in [19].

The first countermeasure consists of computing a *one-way hash function* for the matrices U_W and V_W. The hash function algorithm such as *message digest 5* (MD5) or *secure hash algorithm 1* (SHA-1) can be used for this purpose. During embedding process, the hash values of matrices U_W and V_W denoted respectively by H_U and H_V are stored in a private key. In the extracting process, the hash function is first computed from received (and possibly altered by an attacker) matrices $U_{\tilde{W}}$ and $V_{\tilde{W}}$, denoted by $H_{\tilde{U}}$ and $H_{\tilde{V}}$. Thus, if $H_U \neq H_{\tilde{U}}$ or $H_V \neq H_{\tilde{V}}$, the watermark extracting process is stopped because $U_{\tilde{W}} \neq U_W$ or $V_{\tilde{W}} \neq V_W$, otherwise the watermark extracting process is performed. The authors indicate that the hash function test can be also be applied on a combined matrix from U_W and V_W such as: $U_W + V_W$, $U_W \times V_W$, etc.

The second countermeasure consists in encrypting the watermark before the embedding process. The watermark W is encrypted resulting in an encrypted watermark denoted by W_C which will then be embedded in the original image I. Suppose that for watermark extraction process, an attacker uses his own watermark $\tilde{W}$. Then matrices $U_{\tilde{W}}$ and $V_{\tilde{W}}$ will be used instead of proper matrices U_{W_C} and V_{W_C}. The first extracted watermark will be the same as $\tilde{W}$ but the decryption process must be performed since the embedded watermark is encrypted during the watermark embedding process. Therefore, the extracted watermark $\hat{W}$ will be a random-like image. Note that, if there is no attack, the first extracted watermark $\hat{W}_C$ will indeed be the encrypted image but after decryption, the extracted watermark $\hat{W}$ will have a high correlation value with the original watermark W. Figures 4 and 5 illustrate an example of one-way hash function and encryption countermeasures, respectively, using `Lena` as original image I and `JIH-MSP logo` as watermark W.

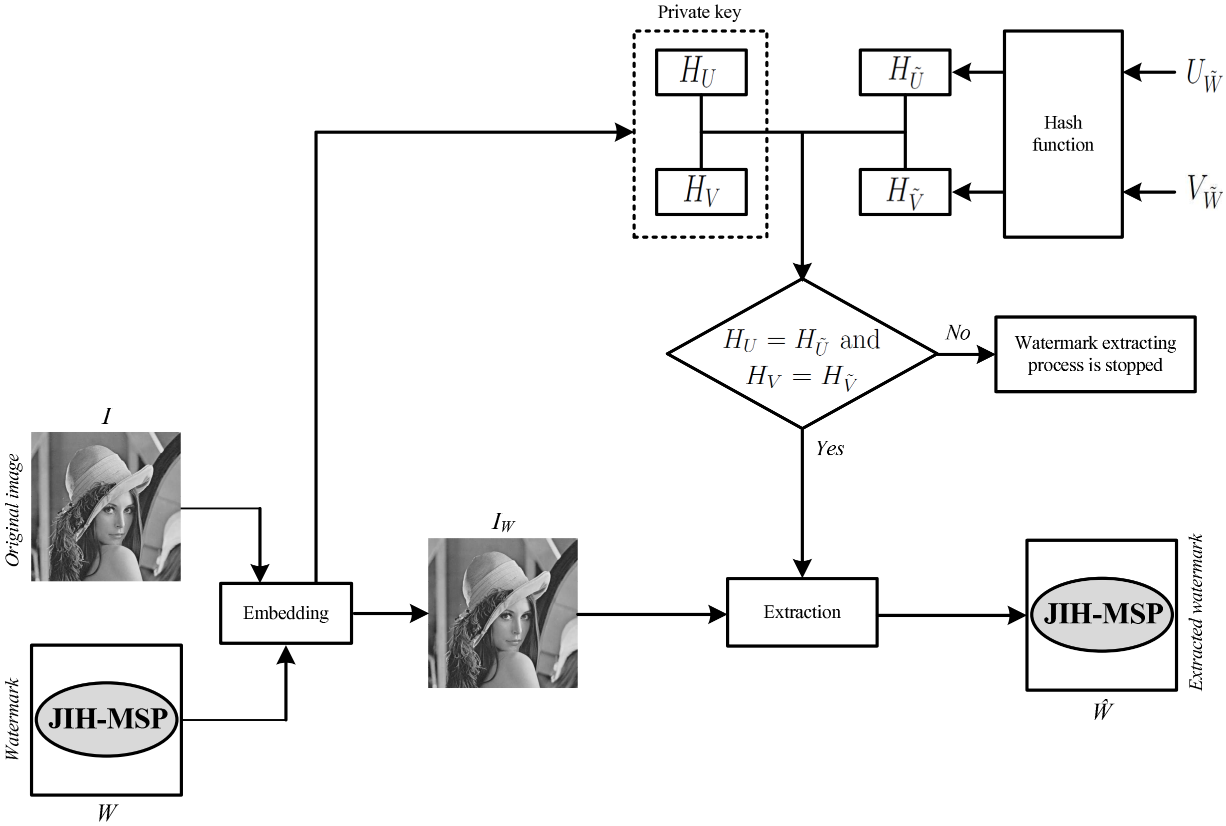


FIGURE 4. Example of the first countermeasure based on one-way hash function.

4. **Watermarking algorithm using multi-objective ant colony optimization.** In general, watermarking schemes are either based on an additive or a multiplicative rule. The embedding rules themselves are usually of the form:

$$\begin{cases} I_W = I + \alpha \cdot W & \longrightarrow \text{additive rule} \\ I_W = I \cdot (1 + \alpha \cdot W) & \longrightarrow \text{multiplicative rule} \end{cases} \tag{16}$$

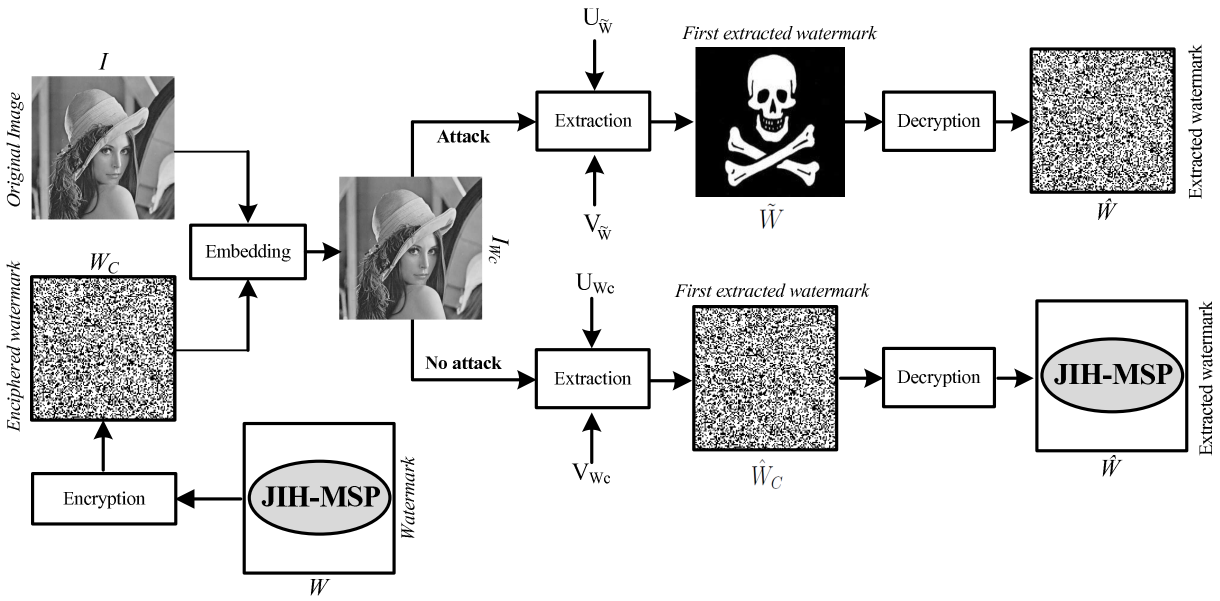


Figure 5. Example of the second countermeasure based on encryption.

where I_W is (transformed) watermarked image, I is the (transformed) original image and α is used to control watermarking strength: watermark W is scaled by factor α during the embedding process. Cox et al. [14] suggest the use of multiple scaling factors instead of one. They state that a single scaling factor may not be applicable for altering all the values of original image I. Therefore, multiple scaling factors are required instead. Determining the optimal values of these multiple factors, in order to achieve the highest robustness and transparency under various attacks, is a difficult problem. To reduce the computational complexity, we use a multi-objective ant colony optimization, based on the generic ant colony optimization (Section 2), to find the optimal multiple scaling factors of the watermarking algorithm presented in Section 3. The steps for applying MOACO into SVD-LWT watermarking scheme are enumerated below:

Step 1 : Define the colony size, the initial pheromone trail, the dissolving rate (σ), the objective function, and a generation number as the algorithm stopping criterion.

Step 2 : Using the equation (17), generate randomly an initial population of ants, which constitute a set of potential solutions. Each ant is denoted by $X = \{x_1, x_2, ..., x_n\}$;

$$f_{\text{pheromone}}(x_i) = e^{\frac{(x_i - x_i^*)^2}{2\sigma_i^2}} \tag{17}$$

where x_i^* is the i^{th} coordinate of the best point found by optimization within the design space at the current iteration and σ_i is the ants aggregation index for the i^{th} coordinate of the design space. At the first iteration, x_i^* is chosen according a uniform distribution with σ_i taken as at least 3 times larger than the length of the search interval.

Step 3 : For each ant X of the population:

- Produce the watermarking image I_W using embedding process previously described in Section 3.1 using the ant X as the watermark strength factor. Note that equation 8 in watermark embedding process is transformed into [1]:

$$S_Y = S_X + diag(\alpha) \cdot S_{W_C} \tag{18}$$

[1] α is changed into multiple scaling factors instead single scaling factor.

with $diag(\alpha)$ is diagonal matrix create from the vector X, i.e. the ant X.

- Compute the normalized correlation $NC(I, I_W)$ between original and watermarked images I and I_W.
- Apply a watermark attack out of a set of T selected attacks upon the watermarked image I_W. This leads to T different attacked watermarked images $\left\{\hat{I}_W\right\}$ for each original watermarked image I_W.
- Extract the watermarks $\hat{W}_i$ from the attacked watermarked images $\hat{I}_W$ using the extraction process, as described in Section 3.2, where $i = \{1, 2, \ldots, T\}$.
- Compute the normalized correlation coefficients between the original watermark W and the set of extracted watermarks $\left\{\hat{W}_i\right\}$, i.e. $\left\{NC(W, \hat{W}_i)\right\}$.
- Construct the vector of objective values, $F(X)$, defined as:

$$F(X) = \begin{pmatrix} \frac{1}{NC(I,I_W)} \\ \frac{1}{NC(W,\hat{W})} \\ \frac{1}{NC(W,\hat{W}_1)} \\ \frac{1}{NC(W,\hat{W}_2)} \\ \vdots \\ \frac{1}{NC(W,\hat{W}_T)} \end{pmatrix} \tag{19}$$

- Evaluate the vector of objective values according to the *exponential weighted method* for multi-objective optimization [20]:

$$F_{\text{obj}}(X) = \sum_{i=1}^{T+2} (e^{p\cdot w} - 1) \cdot e^{p\cdot(F(X) - F_0)} \tag{20}$$

where: p, w and F_0 are positive constants. In experiments, we take $p = 2$, $w = 5$ et $F_0 = 10$.

Step 4 : Find the best ant $X_{\text{best}} = \{x_1^*, x_2^*, ..., x_n^*\}$ as the one having the smallest objective value F_{obj}.

Step 5 : Update the pheromone trail distribution using the formula (17), in this step the aggregation index for the i^{th} dimension σ_i is given by :

$$\sigma_i = \sqrt{\frac{1}{m} \sum_{i=1}^{m} y_i - \bar{y}} \tag{21}$$

where $\mathbf{y}$ is the i^{th} column of the colony matrix C, $\bar{y}$ is the mean value of the vector $\mathbf{y}$ and m is the number of colony size.

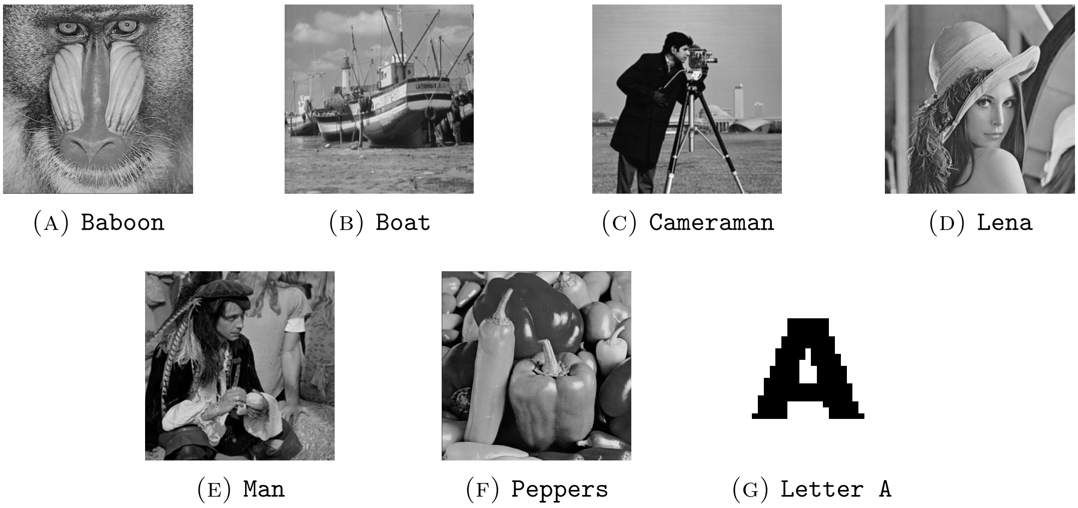
Step 6 : Save the best ant X_{best} between this generation and the old one generation.

Step 7 : If the generation number is reached the optimization process of the multiple scaling factors (MSF) is terminated, else it goes to the next step.

Step 8 : Using the distribution of (17), generate a new population of ants and then go back to the Step 3.

5. **Experimental results.** In this section, we present a series of tests that were conducted to assess the performance of the multi-objective ant colony optimization algorithm

for the search of the best multiple scaling factors for the proposed watermarking algorithm. Computer simulations were run using the six 256×256 gray-scale test images and a 32×32 binary (black-and-white) watermark depicted in Figure 6.



(A) Baboon (B) Boat (C) Cameraman (D) Lena

(E) Man (F) Peppers (G) Letter A

Figure 6. Original (Baboon, Boat, Cameraman, Lena, Man and Peppers) grey-scale test images and binary watermark (Letter A) used for the MOACO algorithm simulations.

To evaluate the effectiveness of multi-objective ant colony optimization for determining the multiple scaling factors for the watermarking algorithm, in terms of imperceptibility and robustness, it is compared with the results obtained with the same watermarking algorithm but using a single scaling factor. For the case of multiple scaling factors, parameter α in equation (8) is replaced by a vector of real elements of length 32, since the watermark size is 32×32. Table 1 shows comparative the simulated results for imperceptibility tests of the proposed algorithm along with those obtained with three other watermarking algorithms. The first algorithm, proposed by Xianghong et al. [7], is based on discrete wavelet transform and vector transform. The second one, presented by Liu [21], is also based on the discrete wavelet transform. The last one is based on the discrete cosine transform and was introduced by Pai and Ruan [22].

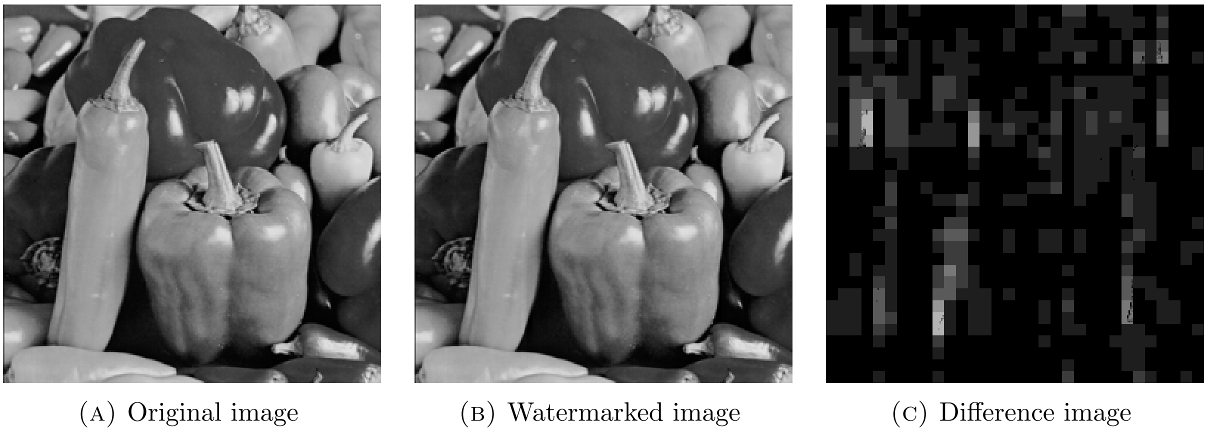
From Table 1, one can see that the peak signal to noise ratios between the original and the watermarked images, $PSNR(I, I_W)$, using either a single scaling factor or multiple scaling factors are close to each other. The peak signal to noise ratio values of the proposed algorithm using MSFs is greater than that obtained for Xianghong and Liu watermarking schemes [7, 21] but slightly less than that obtained with Pai algorithm [22]. The normalized correlation between watermark W and extracted watermark $\hat{W}$, $NC(W, \hat{W})$, obtained with our algorithm using either single or multiple scaling factors is very close to unity. Thus, in terms of imperceptibility, the proposed algorithm leads to better results than Xianghong and Liu watermarking schemes. Note that the embedding process is done in the LH_3 wavelet subband with the number of lifting wavelet transform levels ℓ equal to 3.

Figure 7 depicts a watermarked image (Figure 7b) having a PSNR value of 48.097 dB relative to original image (Figure 7a) as well as the absolute difference between original and watermarked images, magnified by factor of 30 (Figure 7c).

For the robustness tests, eight different attacks were selected in conjunction to multi-objective optimization (i.e. $T = 8$). These attacks are: salt & peppers noise (with a

TABLE 1. Comparative results of the imperceptibility tests.

Image	**Algorithm**	**PSNR($\mathbf{I}, \mathbf{I_W}$)(dB)**	**NC($\mathbf{W}, \hat{\mathbf{W}}$)**
Baboon	using MSF	52.379	1.000
	using SSF	51.124	1
	Xianghong [7]	49.075	0.999
	Liu [21]	45.320	1.000
	Pai [22]	54.844	1.000
Boat	using MSF	54.810	1.000
	using SSF	53.716	1.000
	Xianghong [7]	49.075	1.000
	Liu [21]	45.428	1.000
	Pai [22]	54.829	1.000
Cameraman	using MSF	48.902	1.000
	using SSF	49.341	1.000
	Xianghong [7]	49.075	1.000
	Liu [21]	45.529	0.998
	Pai [22]	55.289	1.000
Lena	using MSF	47.718	1.000
	using SSF	48.899	1
	Xianghong [7]	49.075	1.000
	Liu [21]	45.472	0.997
	Pai [22]	55.186	1.000
Man	using MSF	50.181	1.000
	using SSF	50.735	1.000
	Xianghong [7]	49.075	1.000
	Liu [21]	45.323	1.000
	Pai [22]	55.186	0.998
Peppers	using MSF	48.097	1.000
	using SSF	48.943	1.000
	Xianghong [7]	49.075	0.999
	Liu [21]	45.318	0.995
	Pai [22]	55.038	1.000



(A) Original image (B) Watermarked image (C) Difference image

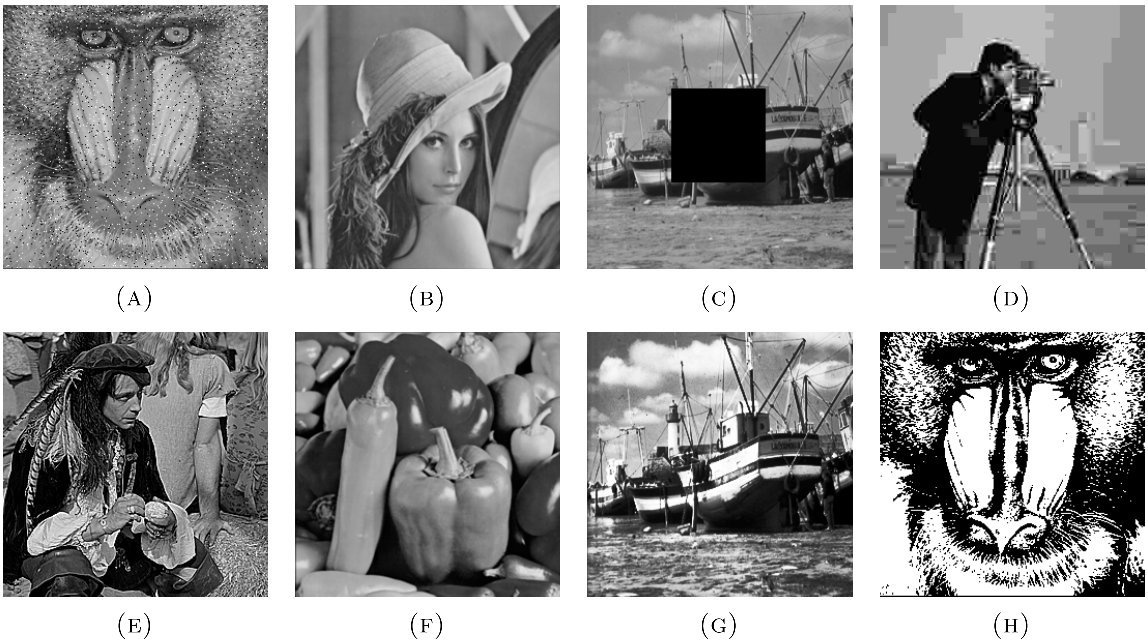
FIGURE 7. Comparison between original and watermarked images.

density of 0.05), gaussian filtering (3 × 3), cropping (1/8 of the image center), JPEG compression ($Q = 5$), sharpening, scaling (256 → 512 → 256), histogram equalization, and gray-scale quantization (1 bit): these attacks are identified respectively as **SP**, **GF**, **CR**, **CM**, **SH**, **SC**, **HE** and **QN**. The level of robustness of the proposed watermarking scheme differs according to the nature of the watermarking application. Table 2 gives the normalized correlation values between the embedded and the extracted watermarks, $NC(W, \hat{W}_i)$, $i = \{1, 2, \ldots, 8\}$) under the 8 different attacks for the MOACO-based watermarking scheme.

Table 2. Robustness tests (first series).

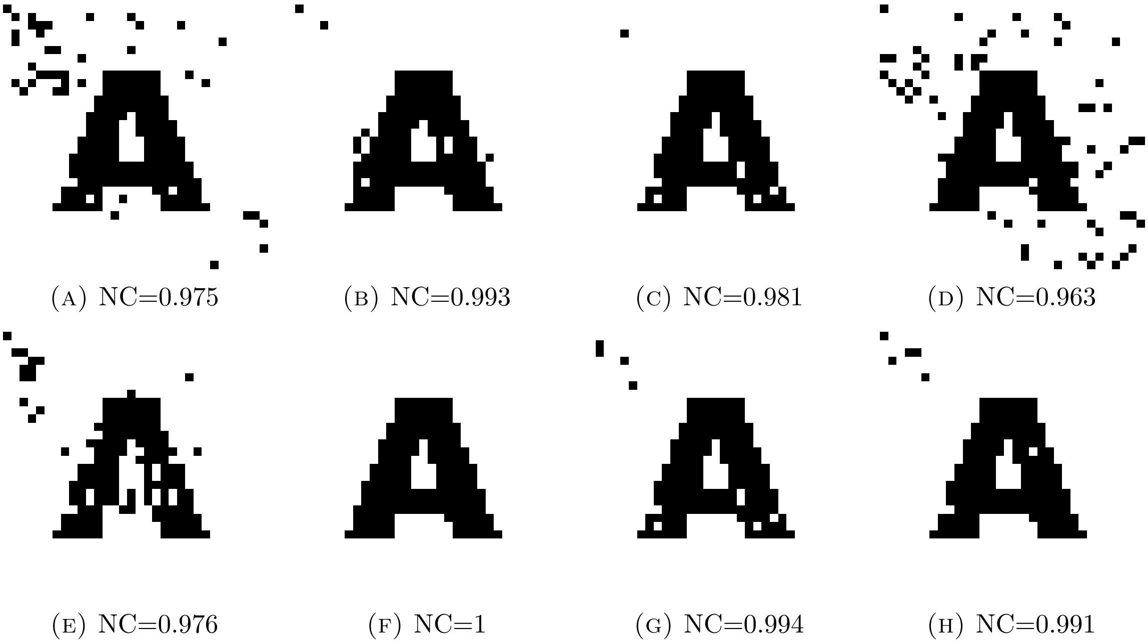
Image	**Algorithm**	**SP**	**GF**	**CR**	**CM**	**SH**	**SC**	**HE**	**QN**
Baboon	using MSF	0.975	0.985	0.987	0.986	0.985	1	0.995	0.995
	using SSF	0.885	0.936	0.998	0.930	0.950	1.000	0.977	0.965
	Xianghong [7]	0.694	0.858	0.983	0.633	0.713	0.986	0.440	0.570
	Liu [21]	0.947	0.906	0.980	0.545	0.951	0.870	0.974	0.714
	Pai [22]	0.374	0.827	0.976	0.293	0.370	0.926	0.313	0.220
Boat	using MSF	0.987	0.975	0.981	0.994	0.996	1	0.994	0.995
	using SSF	0.777	0.847	0.855	0.990	0.980	0.992	0.915	0.928
	Xianghong [7]	0.555	0.843	0.983	0.575	0.723	0.992	0.486	0.814
	Liu [21]	0.938	0.888	0.958	0.538	0.946	0.860	0.966	0.582
	Pai [22]	0.343	0.858	0.982	0.373	0.486	0.982	0.354	0.342
Cameraman	using MSF	0.968	0.970	0.941	0.963	0.982	1	0.981	0.978
	using SSF	0.726	0.907	0.894	0.729	0.976	1.000	0.959	0.958
	Xianghong [7]	0.495	0.847	0.983	0.621	0.580	0.993	0.621	0.750
	Liu [21]	0.936	0.872	0.983	0.470	0.940	0.846	0.954	0.743
	Pai [22]	0.407	0.898	0.975	0.537	0.697	0.989	0.474	0.504
Lena	using MSF	0.963	0.993	0.946	0.976	0.986	1	0.991	0.983
	using SSF	0.774	0.943	0.867	0.944	0.982	1.000	0.988	0.974
	Xianghong [7]	0.616	0.866	0.983	0.640	0.667	0.994	0.587	0.625
	Liu [21]	0.926	0.845	0.953	0.541	0.952	0.826	0.987	0.716
	Pai [22]	0.383	0.920	0.980	0.442	0.681	0.994	0.514	0.393
Man	using MSF	0.977	0.948	0.973	0.973	0.977	1	0.988	0.986
	using SSF	0.588	0.845	0.930	0.905	0.949	0.999	0.977	0.963
	Xianghong [7]	0.738	0.876	0.983	0.641	0.692	0.993	0.828	0.456
	Liu [21]	0.944	0.876	0.994	0.532	0.966	0.845	0.983	0.803
	Pai [22]	0.412	0.866	0.981	0.324	0.432	0.983	0.468	0.282
Peppers	using MSF	0.968	0.986	0.964	0.975	0.991	1	0.994	0.980
	using SSF	0.848	0.943	0.886	0.859	0.984	1.000	0.984	0.960
	Xianghong [7]	0.713	0.891	0.983	0.610	0.699	0.997	0.749	0.534
	Liu [21]	0.942	0.796	0.970	0.593	0.942	0.795	0.977	0.622
	Pai [22]	0.383	0.920	0.980	0.442	0.681	0.994	0.514	0.393

From Table 2, one can conclude that the proposed algorithm provides better robustness compared to Xianghong [7], Pai [22] and Liu [21] algorithms. Our watermarking algorithm is robust against the following attacks: additive noise, gaussian filter, cropping, JPEG compression, sharpening, scaling, histogram equalization and gray-scale quantization. Furthermore, the robustness results are improved with the use of multiple scaling factors. Figure 8 shows different watermarked images under the eight different attacks, while Figure 9 depicts their corresponding extracted watermarks.


(A) (B) (C) (D)

(E) (F) (G) (H)

Figure 8. Watermarked images under different attacks: (A) salt & peppers noise (5%), (B) gaussian filter (3 × 3), (C) cropping (1/8 center), (D) JPEG compression (Q=5), (E) sharpening, (F) scaling (256→512→256), (G) histogram equalization and (H) gray-scale quantization (1 bit).


(A) NC=0.975 (B) NC=0.993 (C) NC=0.981 (D) NC=0.963

(E) NC=0.976 (F) NC=1 (G) NC=0.994 (H) NC=0.991

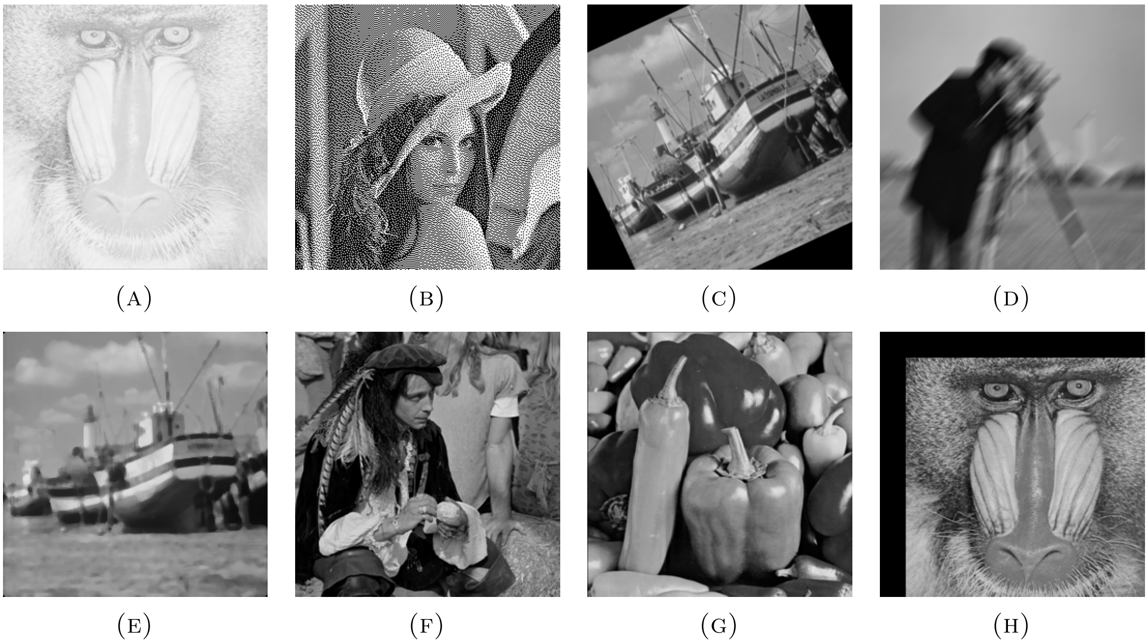
Figure 9. Extracted watermarks under different attacks: (A) salt & peppers noise (5%), (B) gaussian filter (3 × 3), (C) cropping (1/8 center), (D) JPEG compression (Q=5), (E) sharpening, (F) scaling (256→512→256), (G) histogram equalization and (H) gray-scale quantization (1 bit).

In addition to the eight attacks used in the multi-objective ant colony optimization process, the effectiveness of the proposed algorithm using MSFs was also tested against various others attacks: gamma correction($\gamma = 0.2$), dithering, rotation (25 °), motion blur (45 °), median filter (3×3), rewatermarked using other watermark that differ from the watermark (`Letter A`), collusion attack using five watermarks and translation (25 pixels × 25 pixels). These attacks are respectively denoted by **GC**, **DI**, **RT**, **MB**, **MF**, **RW**, **CA** and **TR**. Table 3 gives the normalized correlation values, $NC(W, \hat{W}_i)$, under eight other attacks. It is important to note here that the MSFs were not further optimized by the MOACO process.

Table 3. Robustness tests (second series).

Image	**Algorithm**	**GC**	**DI**	**RT**	**MB**	**MF**	**RW**	**CA**	**TR**
`Baboon`	using MSF	0.991	0.992	0.886	0.989	0.884	0.999	1.000	0.989
	Xianghong [7]	0.915	0.001	0.493	0.516	0.604	0.850	0.914	0.444
	Liu [21]	0.997	0.676	0.702	0.643	0.627	0.867	0.918	0.752
	Pai [22]	0.313	0.267	0.271	0.304	0.414	0.867	0.912	0.278
`Boat`	using MSF	0.984	0.986	0.848	0.986	0.949	1.000	1.000	0.934
	Xianghong [7]	0.879	0.001	0.372	0.491	0.621	0.850	0.914	0.350
	Liu [21]	0.994	0.681	0.679	0.670	0.631	0.866	0.922	0.689
	Pai [22]	0.410	0.358	0.277	0.357	0.550	0.867	0.917	0.297
`Cameraman`	using MSF	0.964	0.966	0.783	0.964	0.870	1.000	1.000	0.726
	Xianghong [7]	0.847	0.001	0.368	0.428	0.665	0.850	0.914	0.411
	Liu [21]	0.988	0.688	0.665	0.652	0.643	0.865	0.907	0.659
	Pai [22]	0.616	0.532	0.421	0.546	0.758	0.867	0.924	0.462
`Lena`	using MSF	0.982	0.982	0.396	0.975	0.418	1.000	1.000	0.866
	Xianghong [7]	0.875	0.001	0.446	0.584	0.748	0.850	0.914	0.419
	Liu [21]	0.986	0.686	0.641	0.666	0.623	0.863	0.912	0.683
	Pai [22]	0.535	0.435	0.332	0.527	0.766	0.867	0.918	0.374
`Man`	using MSF	0.975	0.983	0.880	0.981	0.918	0.997	1.000	0.886
	Xianghong [7]	0.912	0.001	0.518	0.518	0.594	0.850	0.914	0.492
	Liu [21]	0.990	0.683	0.689	0.636	0.626	0.865	0.915	0.766
	Pai [22]	0.374	0.316	0.287	0.334	0.482	0.867	0.920	0.292
`Peppers`	using MSF	0.449	0.975	0.906	0.973	0.656	1.000	1.000	0.902
	Xianghong [7]	0.897	0.001	0.448	0.509	0.644	0.850	0.914	0.409
	Liu [21]	0.989	0.698	0.698	0.653	0.651	0.864	0.910	0.709
	Pai [22]	0.497	0.386	0.265	0.385	0.664	0.867	0.923	0.291

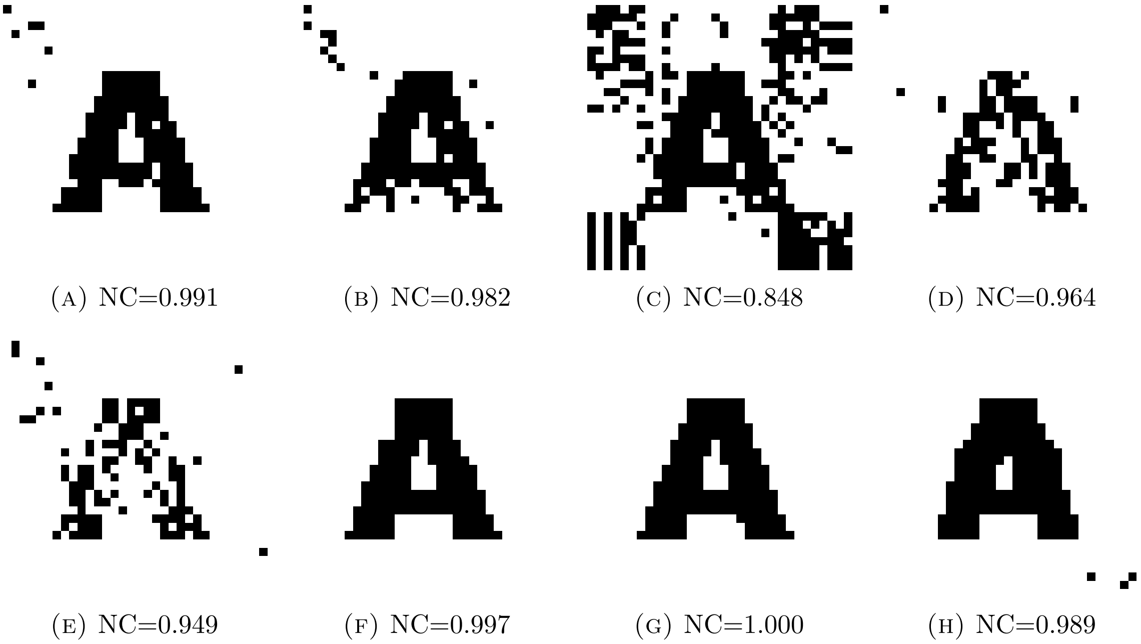
From Table 3, one observes that our algorithm still give better results in term of robustness when compared to Xianghong, Pai, and Liu algorithms. The proposed watermarking algorithm is thus robust against gamma correction, dithering, rotation, motion blur, median filter, rewatermarked, collusion attack and translation. Figure 10 shows different watermarked images under these eight different attacks with Figure 11 depicting the resulting watermarks.



(A) (B) (C) (D)

(E) (F) (G) (H)

Figure 10. Watermarked images under different attacks: (A) gamma correction ($\gamma = 0.2$), (B) dithering, (C) rotation (25 °), (D) motion blur (45 °), (E) median filter (5×5), (F) rewatermarked, (G) collusion attack using five different watermarks and (H) translation (25 pixels × 25 pixels).



(A) NC=0.991 (B) NC=0.982 (C) NC=0.848 (D) NC=0.964

(E) NC=0.949 (F) NC=0.997 (G) NC=1.000 (H) NC=0.989

Figure 11. Extracted watermarks under different attacks: (A) gamma correction($\gamma = 0.2$), (B) dithering, (C) rotation (25 °), (D) motion blur (45 °), (E) median filter (5×5), (F) rewatermarked, (G) collusion attack using five watermarks and (H) translation (25 pixels × 25 pixels).

6. Conclusion. In this paper, a novel watermarking algorithm based on lifting wavelet transforms and singular value decomposition, using multi-objective ant colony optimization is presented. To achieve the highest possible robustness without losing watermark transparency, multiple scaling factors are used. Determination of the optimal values for the multiple scaling factors is however a prohibitively complex problem. A multi-objective ant colony optimization based algorithm is used to find the set of watermarking scaling factors. Experimental results demonstrate that the proposed MOACO-based MSF watermarking scheme outperforms single scaling factor watermarking schemes in terms of imperceptibility and robustness. Furthermore, our algorithm showed better imperceptibility and excellent resiliency against a wide range of watermarking attacks. Also the problem of false positive detections which affects most SVD-watermarking algorithms is resolved using one-way hash functions and watermark encryption.

REFERENCES

[1] A. Bors and I. Pitas, Image watermarking using DCT domain constraints, *Proc. of IEEE International Conference on Image Processing*, vol. 3, pp. 231-234, 1996.

[2] J. Hernndez, M. Amado, and F. Prez-Gonzlez, DCT-Domain Watermarking Techniques for Still Images: Detector Performance Analysis and a New Structure, *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 55-68, 2000.

[3] H.-C. Huang, Y.-H. Chen, and A. Abraham, Optimized Watermarking Using Swarm-Based Bacterial Foraging, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 1, pp. 51-58, january 2010.

[4] M. Ramkumar, A. Akansu, and A. Alatan, A robust data hiding scheme for images using DFT, *Proc. of IEEE International Conference on Image Processing*, vol. 2, pp. 211-215, 1999.

[5] J. Ruanaidh, W. Dowling, and F. Boland, Phase watermarking of digital images, *Proc. of IEEE International Conference on Image Processing*, vol. 3, pp. 239-342, 1996.

[6] H. Al-Qaheri, A. Mustafi, and S. Banerjee, Digital Watermarking using Ant Colony Optimization in Fractional Fourier Domain, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 3, pp. 179-189, 2010.

[7] T. Xianghong, L. Lu, Y. Lianjie, and N. Yamei, A digital watermarking scheme based on DWT and vector transform, *Proc. of International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 635-638, 2004.

[8] X. G. Xia, C. Boncelet, and G. Arce, A multiresolution watermark for digital images, *Proc. of IEEE International Conference on Image Processing*, vol. 1, pp. 548-551, 1997.

[9] L. Xiao, H. Z. Wu, Z. H. Wei, and Y. Bao, Perceptual digital watermark of images using ridgelet transform, *Proc. of IEEE International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3937-3942, 2004.

[10] P. Campisi, D. Kundur, and A. Neri, Robust digital watermarking in the ridgelet domain, *IEEE Signal Processing Letters*, vol. 11, no. 10, pp. 826-830, 2004.

[11] W. Lu, W. Sun, and H. Lu, Robust watermarking based on DWT and nonnegative matrix factorization, *Computers and Electrical Engineering*, vol. 35, no. 1, pp. 183-188, 2009.

[12] A. A. Mohammad, A. Al-Haj, and S. Shaltaf, An improved SVD-based watermarking scheme for protecting rightful ownership, *Signal Processing*, vol. 88, no. 9, pp. 2158-2180, 2008.

[13] K. Loukhaoukha and J. Y. Chouinard, Hybrid watermarking algorithm based on SVD and lifting wavelet transform for ownership verification, *Proc. of the 1th Canadian Workshop on Information Theory*, pp. 177-182, may 2009.

[14] I. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, Secure Spread Spectrum Watermarking for Multimedia, *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1673-1687, 1997.

[15] M. Dorigo, *Optimization, Learning and Natural Algorithms*, Ph. D Thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, Italian, 1992.

[16] K. Socha and M. Dorigo, Ant colony optimization for continuous domains, *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155-1173, 2008.

[17] F. Viana, G. Kotinda, D. Rade, and V. S. Jr, Tuning dynamic vibration absorbers by using ant colony optimization, *Computers and Structures*, vol. 86, no. 13-14, pp. 1539-1549, 2008.

[18] S. Pourtakdoust and H. Nobahari, An extension of ant colony system to continuous optimization problems, *Proc. of International Workshop of Ant Colony Optimization and Swarm Intelligence*, pp. 294-301, 2004.

[19] K. Loukhaoukha and J. Y. Chouinard, On the security of ownership watermarking of digital images based on SVD decomposition, *Journal of Electronic Imaging*, vol. 19, no. 1, pp. 013007, 2010.

[20] R. Marler and J. Arora, Survey of multi-objective optimization methods for engineering, *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369-395, 2004.

[21] J. L. Liu, D. C. Lou, M. C. Chang, and H.-K. Tso, A robust watermarking scheme using self-reference image, *Computer Standards & Interfaces*, vol. 28, no. 3, pp. 356-367, 2006.

[22] Y. T. Pai and S. J. Ruan, A high quality robust digital watermarking by smart distribution technique and effective embedded scheme, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90A, no. 3, pp. 597-605, 2007.