Optics Communications 282 (2009) 769–777

Contents lists available at ScienceDirect

Optics Communications

journal homepage: www.elsevier.com/locate/optcom

An optimal robust digital image watermarking based on SVD using differential evolution algorithm

Veysel Aslantas *

Erciyes University, Faculty of Engineering, Department of Computer Eng., 38039 Melikgazi, Kayseri, Turkey

ARTICLE INFO

Article history:
Received 25 April 2008
Received in revised form 8 November 2008
Accepted 12 November 2008

Keywords:
Robust image watermarking
Singular value decomposition
Differential evolution algorithm

ABSTRACT

The main objective in developing a robust image watermarking technique is to obtain the highest possible robustness without losing the transparency. To achieve this objective, this paper presents a new optimal robust image watermarking technique based on singular value decomposition (SVD) using differential evolution algorithm (DE). The singular values (SV) of the host image are modified by multiple scaling factors to embed a watermark image. The modifications are optimised using DE to achieve maximum robustness and transparency Experimental results show that the proposed approach can effectively improve the quality of the watermarked image and the robustness of the embedded watermark against various attacks.

1. Introduction

Because of the great advances in computer and communication industry, digital multimedia contents such as audio, image and video are being widely and easily transmitted over the Internet. These materials can easily be modified or duplicated for illegal purposes. As a result, copyright protection has become a very important subject. In order to prevent any copyright forgery, misuse or violation, digital watermarking, a new technology for data protection, provides a promising way of protecting multimedia data from illegal manipulation and duplication.

The basic idea in digital watermarking is to embed a watermark signal into the host data for the purpose of copyright protection, access control, broadcast monitoring etc. A host may be a multimedia object such as audio, image or video. A watermark can be a tag, label or digital signal. Watermarking techniques can be categorized into different classes according to domain, visibility and permanency [1–4].

With respect to the domain in which the watermark is embedded, the techniques can be classified into two categories: spatial domain and frequency-domain techniques. In spatial domain schemes, the watermark is embedded by directly modifying the pixel values of the host image [5–10]. On the other hand, transform domain techniques embed the watermark by modulating the coefficients in a transform domain such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT) [11–14]. In general, transform domain methods are more robust than the spatial domain methods against many common attacks.

In terms of the visibility, digital watermarks can be classified into two different groups: Visible and invisible watermarks. Visible watermarks can easily be perceived for example company logos that inserted into or overlaid on some TV channels. Although the owner of the multimedia content can be recognized without any calculation, embedded watermarks can be removed or destroyed easily. On the other hand, invisible watermarks are imperceptible and are embedded on the unknown places in the host data. The watermarked data should look similar to the original one, and should not cause suspicion by others. If it is used illegally, the embedded watermark will be used for showing the ownership. This paper concerns with the invisible watermarking technique.

With respect to permanency, invisible watermarks can be classified as semi-fragile, fragile and robust. Semi-fragile watermarks are capable of tolerating some degree of change of a watermarked image, such as the addition of quantization noise from lossy compression. Fragile watermarks are embedded in such a way that any modification or manipulation of the host image would corrupt the watermark. Therefore, fragile watermarks are mainly used for authentication purposes. Robust watermarks are designed to resist intentional or unintentional image modifications for instance filtering, geometric transformations, noise addition, etc. For copyrights protection, this kind of watermarks has to be utilized.

In general, greater robustness can be achieved if significant modifications are made to the host image either in spatial or transform domain. However, such modifications are distinguishable and thus do not satisfy the requirement of transparency (invisibility). For the optimal watermarking application, a trade-off between these two competing criteria (robustness and transparency) has

* Tel.: +90 352 437 4901x32602; fax: +90 352 437 57 84.
E-mail address: aslantas@erciyes.edu.tr

0030-4018/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.optcom.2008.11.024

to be made. Therefore, image watermarking can be formulated as an optimisation problem. Artificial intelligence (AI) techniques such as genetic algorithm, Fuzzy logic and neural networks have been employed to solve the problem optimally. The AI techniques have been utilised to optimise watermarking requirements in DCT [15–19], DWT [20–25], SVD [26] and spatial domain [9,27–30].

The pixel values or the transform domain coefficients of the host image (I) can be modified to embed a watermark (W) invisibly. Before embedding process, the watermark can be scaled by a scaling factor (k) to control the watermark strength. The larger the scaling factor (SF), the more the distortion of the quality of the host image (transparency) and the stronger the robustness. On the other hand, the smaller the SF, the better the image quality and the weaker the robustness. A single SF may not be applicable for modifying all the values of I because different spectral components may exhibit different tolerance to modification. Therefore, multiple SFs should be employed for adapting to the different spectral components to reduce visual artefacts [11].

Singular value decomposition is one of the most useful techniques of linear algebra with numerous applications in signal processing fields including watermarking [31–40]. Based on SVD, this paper proposes an optimal watermarking scheme for grey-scale images. The singular values of the host image are modified to embed the watermark image by employing multiple SFs. Numerous combinations of scaling factors are possible, and it is difficult to find optimal solutions by trial and error. Therefore, optimisation of the scaling factors is necessary in order to obtain the highest possible robustness without losing the transparency. In this paper, a technique is proposed for scaling factors based on DE that can search for multiple solutions simultaneously over a wide range, and an optimum solution can be gained by combining the obtained results appropriately. Experimental results show both the significant improvement in transparency and the robustness under attacks.

This paper is organized as follows. SVD based watermarking is described in Section 2. Section 3 explains the fundamental concepts of differential evolution algorithm and demonstrates the proposed method for embedding the watermark in the SVD domain with DE. Section 4 gives the simulation results. Section 5 concludes the paper.

2. Singular value decomposition based watermarking

This section describes the overall scheme of the singular value decomposition based watermarking. First, it briefly introduces SVD transformation and its properties. Next, it explains the stages of embedding process. Then, the extraction procedure is illustrated.

2.1. Singular value decomposition

SVD decomposes an $N \times N$ real matrix A into a product of 3 matrices:

$$A = USV^T = [u_1, u_2, \ldots, u_n] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} [v_1, v_2, \ldots, v_n]^T \quad (1)$$

where S is an $N \times N$ diagonal matrix. U and V^T are $N \times N$ orthogonal matrices, whose column vectors are u_i's and v_i's, respectively. The elements of S are only non-zero on the diagonal that are arranged in decreasing order and are called the SVs of A. When the rank of A is r, $S = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ satisfies $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r \geqslant \sigma_{r+1} = \sigma_{r+2} \cdots = \sigma_n = 0$. Let A be a matrix whose elements are pixel values of an image. The image can be written as:

$$A = \sum_{i=0}^{r} \sigma_i u_i v_i^T \quad (2)$$

2.2. Properties of SVD

SVD has several attractive properties from the viewpoint of image processing applications. The behaviors of singular values of an image under common geometric perturbations have been theoretically analyzed in [35]. SVD is able to efficiently represent the intrinsic algebraic properties of an image, where singular values specify the brightness of the image and corresponding pair of singular vectors reflect the geometry of the image. The properties of the singular values are explained as follows:

- *Stability*: Let A, $B \in R^{m \times n}$ and their corresponding SVs are σ_1, $\sigma_2, \ldots, \sigma_n$ and τ_1, $\tau_2, \ldots, \tau_n$, respectively and then a relation can be established between them as $|\sigma_i - \tau_i| \leqslant \|A - B\|_2$. This indicates that the singular values of an image have very good stability, i.e., when there is a little disturbance with a matrix, the variation of its SV is not grater than 2-norm of disturbance matrix.
- *Proportionality*: The singular values of A (σ_1, $\sigma_2, \ldots, \sigma_n$) and the singular values of kA (σ_1^*, $\sigma_2^*, \ldots, \sigma_n^*$) are related as $|k|(\sigma_1, \sigma_2, \ldots, \sigma_n) = (\sigma_1^*, \sigma_2^*, \ldots, \sigma_n^*)$ which indicates that proportion invariance of singular value must depend on standardization of singular value.
- *Transpose*: A and its transposed counterpart A^T have the same non-zero singular values.
- *Flip*: A and its flipped versions (about vertical axis A_{vf} and about horizontal axis A_{hf}) give the same non-zero singular values.
- *Rotation*: A and its rotated version (A_r) obtained through rotating A by an arbitrary angle have the same non-zero singular values.
- *Scaling*: Suppose $A \in R^{m \times n}$ has the singular values σ_1, $\sigma_2, \ldots, \sigma_n$, then its scaled counterpart A^s has the singular values equal to $\sigma_i^* \sqrt{L_r L_c}$ where L_r and L_c are the scaling factors of rows and columns, respectively. If the scaling function affects only rows or columns, A^s has the singular values equal to $\sigma_i^* \sqrt{L_r}$ or $\sigma_i^* \sqrt{L_c}$, respectively.
- *Translation*: Both the matrix A and its translated counterpart A^t have the same non-zero singular values. A^t is obtained from A by adding rows and columns of black pixels.

The aforementioned properties of SVD are very much desirable for designing watermarking algorithms that are particularly robust to geometric attacks. When the watermarked image is corrupted by attacks such as translation, rotation, noise addition and scaling, the watermark can be extracted effectively from the attacked watermarked image because of these properties.

2.3. Watermark embedding and extracting procedures

The SVD based watermark embedding and extraction procedures can be described as follows [31]:

2.3.1. Watermark embedding.

Without loss of generality, let the size of the host image (I) and watermark (W) is $N \times N$

1. Apply SVD to the host image:

$$I = USV^T \quad (3)$$

2. Modify the S with the W:

$$S_M = S + kW \quad (4)$$

3. Apply SVD to the SM:

$$S_M = U_W S_W V_W^T \tag{5}$$

4. Compute watermarked image:

$$I_W = U S_W V^T \tag{6}$$

2.3.2. Watermark extracting

In general, the extraction process is the inverse of the embedding procedure. In watermark extraction, a possibly distorted watermark W^* is extracted from the possibly distorted watermarked image I_W^* by essentially reversing the above watermark embedding steps. The watermark extraction can be described as follows:

1. Apply SVD to the watermarked (possibly distorted) image:

$$I_W^* = U^* S_W^* V^{*T} \tag{7}$$

2. Compute possibly corrupted S_M^*:

$$S_M^* = U_W S_W^* V_W^T \tag{8}$$

3. Extract the watermark (possibly distorted) image:

$$W^* = (S_M^* - S)/k \tag{9}$$

3. DE-based watermarking

The rest of this section explains how to use differential evolution in order to optimise watermark embedding process with respect to the two conflicting requirements: transparency and robustness to different attacks.

An $N \times N$ host image can have N singular values that may reveal different tolerance to modification. Since we may have no idea of how sensitive the image is to various values of the scaling factor, an algorithm is required to obtain the optimum scaling factors that generate maximum robustness and transparency. Hence, a powerful optimisation algorithm is needed for this objective. For this purpose, DE which is a novel stochastic nonlinear optimisation algorithm is employed in this work.

3.1. Basic structure of differential evolution algorithm

DE [41] is a population-based stochastic optimisation algorithm that can be used to minimize nonlinear and non-differentiable continuous space functions with real-valued parameters. It resembles the structure of an evolutionary algorithm, but differs from traditional evolutionary algorithms in its generation of new candidate solutions and by its use of a 'greedy' selection scheme. Like other evolutionary algorithms, DE also employs similar operators such as the initial population generation, crossover, mutation and selection. The main steps of a basic DE algorithm are shown in Fig. 1. For each generation, the operators are repeated until a predefined stopping criterion is satisfied, for example maximum generation number. The basic operators of DE are briefly explained in the following sections.

3.1.1. Initial population

An optimisation task consisting of D parameters can be represented as a D-dimensional vector. As a member of the evolutionary algorithm family, DE also starts with a population of NP solution vectors. If there is no preliminary knowledge about the problem to be optimised, the vectors can be generated randomly as follows:

$$X_{i,G} = x_{i(L)} + \mathrm{rand}_i[0,1](x_{i(H)} - x_{i(L)}) \tag{10}$$

where $x_{i,(L)}$ and $x_{i,(H)}$ are the lower and higher boundaries of D-dimensional vector $x_i = \{x_{j,i}\} = \{x_{1,i}, x_{2,i}, \ldots, x_{D,i}\}^T$.

3.1.2. Mutation

At generation G, a mutant vector $v_{i,G+1}$ is generated for each target vector $x_{i,G}$ by the combination of vectors randomly chosen from the current population as:

$$v_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}) \tag{11}$$

where i, r_1, r_2, and r_3 are mutually different random integer indices selected from {1, 2,...,NP}. As a result, the population size NP must be greater than 3. $F \in [0, 2]$ is a real constant positive weighting factor which controls the scale of the added differential variation of $(x_{r2,G} - x_{r3,G})$. Larger values for F result in higher diversity in the generated population and the lower values in faster convergence. Mutation expands the search space.

This way of generating a mutant vector is known as the DE1 scheme. Another frequently used scheme is called DE2 in which mutant vectors are produced by including the impact from the best vector $x_{best,G}$ of the current generation as:

$$v_{i,G+1} = x_{r_3,G} + K(x_{best,G-x} - x_{r_2,G}) + F(x_{r_1,G} - x_{r_2,G}) \tag{12}$$

where $r_1 \neq r_2 \neq r_3 \neq i$ are the randomly chosen indices, and r_1, r_2, $r_3 \in \{1, 2, \ldots, NP\}$. K is the combination factor that provides increased greediness of the search process.

3.1.3. Crossover

To increase diversity of the population, DE employs crossover operation that incorporates successful solutions from the previous generation. The trial vector $u_{i,G+1}$ is produced from the elements of the mutated vector, $v_{i,G+1}$, and the elements of the target vector, $x_{i,G}$.

$$U_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } rand_{j,i} \leq CR \text{ or } j = I_{rand} \\ X_{j,i,G} & \text{if } rand_{j,i} > CR \text{ and } j \neq I_{rand} \end{cases} \tag{13}$$

where $j = 1,2,\ldots,D$; $rand_{j,i} \in [0; 1]$ is the random number; $CR \in [0; 1]$ is predefined crossover constant and $I_{rand} \in [1, 2, \ldots, D]$ is a randomly chosen index. I_{rand} ensures that $v_{i,G+1} \neq x_{i,G}$

3.1.4. Selection

If the value returned by the objective function for the trial vector $u_{i,G+1}$ is better than or equal to the value obtained for the target vector $x_{i,G}$, the latter is replaced by the former otherwise the latter is retained in the population of the next generation.

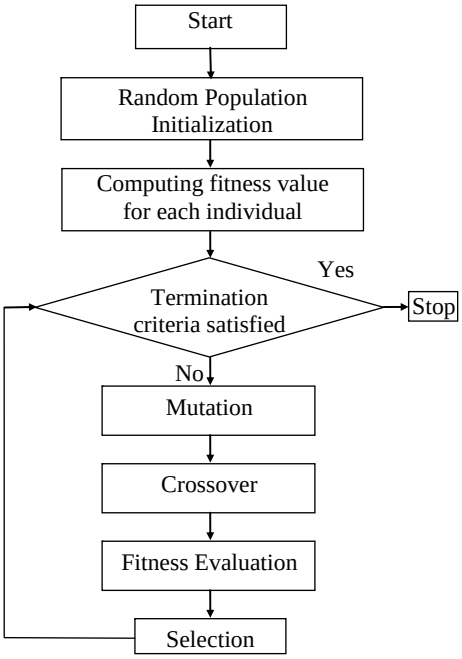

Fig. 1. Flowchart of a basic DE Algorithm.

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(u_{i,G+1}) \le f(x_{i,G}) \; i = 1,2,\ldots,NP \\ X_{i,G} & \text{otherwise} \end{cases} \quad (14)$$

3.2. Optimisation of SFs

In order to achieve the optimal performance of a digital image watermarking algorithm, the developed technique in this paper employs the DE algorithm to search for optimal parameters. The parameters that are subject to the optimisation process for this work are the scaling factors. By applying DE algorithm, the SFs (k) are obtained for optimal watermarking depending on both the transparency and the robustness factors. Fig. 2 diagrammatically illustrates the flowchart of the watermark embedding technique based on DE algorithm developed in this study.

In the application of DE, each member vector in the population represents a possible solution to the problem and hence consists of a set of SFs. To start the optimisation, DE use randomly produced initial solutions (the values of SFs) generated by random number generator. In the proposed scheme, k in Eq. (4) is in the form of diagonal matrix whose elements are only non-zero on the diagonal and they represent the SFs. After modifying the SVs of the host image by employing the SFs, the watermarked images of the current generation are calculated according to the watermark embedding procedure explained in Section 2. Modification of the SVs of the host image may cause the pixel values of the watermarked image to change. These may, in turn, result a watermarked image with the intensity values that are out of the eight bit range. For this reason, the watermarked images are, thus, rescaled according to the host image pixel values.

Not all watermarking applications require robustness to all possible signal processing operations. Therefore, the level of robustness of the watermark varies from application to application. The robustness of the watermarking scheme proposed in this paper is evaluated using the attacks that are commonly employed in literature. Four major attacking schemes are used in DE optimisation procedure. They are average filtering (AV), rotation (RT), sharpening (SH), and resizing (RS). Because of the flexibility of the developed system, the other attacking methods can easily be included to the system or replaced with those used in the DE optimisation process.

Using the extraction procedure given in Section 2, the watermarks are computed from the attacked watermarked images. The two dimensional correlation values are calculated between the host and watermarked images ($corr_I = corr(I, I_W)$) and between the original watermark and the extracted watermarks ($corr_W = corr(W, W^*)$). The correlation values are then employed to compute the fitness value of a solution in the population of DE. At each generation of optimisation process, the fitness of a solution is calculated depending on both the transparency ($corr_I$) and the robustness ($corr_W$) under certain attacks as mentioned above. The objective function to be minimized is then formulated as:

$$F_i = \left[1 / \left(\frac{1}{t} \sum_{i=1}^{t} corr_w(W, W_i^*) \right) - corr_I(I, I_W) \right] \quad (15)$$

where f_i and t are the objective value of the ith solution and the number of attacking methods, respectively, $corr_I()$ and $corr_W()$ are related to transparency and robustness measure, respectively. In order to calculate fitness values, the following expression is used.

$$fitness_i = [1/f_i] \quad (16)$$

where $fitness_i$ is the fitness value of the ith solution. Once values of fitness are assigned for all vectors in the population, the vectors that produced the better fitness values are selected for the next generation. The population of the next generation is then produced from these vectors by using DE operators (mutation, crossover, and selection). The above iterative process on the population will continue until there is no appreciable improvement in the maximum fitness value or predefined maximum number of iterations reached.

The following are the outline of DE-based watermarking algorithm:

1. Define the fitness function, numbers of variables (D) and the values for population size (NP), control parameters (F and CR), and number of generations (or any other terminating criteria).
2. Generate an initial population of potential solutions at random.
3. Calculate watermarked images using the solutions in the population by means of embedding process.
4. Apply the attack functions upon the watermarked images one by one.
5. Extract out the watermarks from the attacked images using the extraction procedure.
6. Compute the NC values between the host image and each watermarked images.
7. Compute the NC values between the watermark and the extracted ones.
8. Evaluate the fitness value for each corresponding solution.
9. Apply mutation and crossover operators.
10. Apply the selection process.
11. Repeat Steps 3–11 until a predefined condition is satisfied, or a constant number of generations is reached.

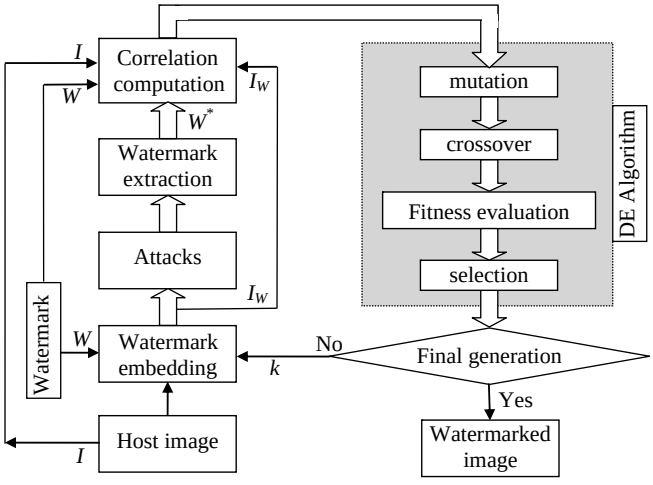

Fig. 2. Block diagram of DE-based watermark embedding scheme.

4. Results

In this section, several experimental results are demonstrated to show the effectiveness and the robustness of the proposed watermarking algorithm. Experiments were conducted using the 256 × 256 Lena (host) and a 32 × 32 grey level watermark images that are illustrated in Fig. 3a and b, respectively. The performance measures are the invisibility of the inserted watermark and the robustness of the method against various types of attacks. The attacks that were used in the fitness evaluation process were: AV (3 × 3), SH (3 × 3), RS (bicubic: 256 → 128 → 256) and RT (30°). Rotation was made in a counterclockwise direction around image center point using the bilinear interpolation method and the output image was cropped to be the same size as the input image.

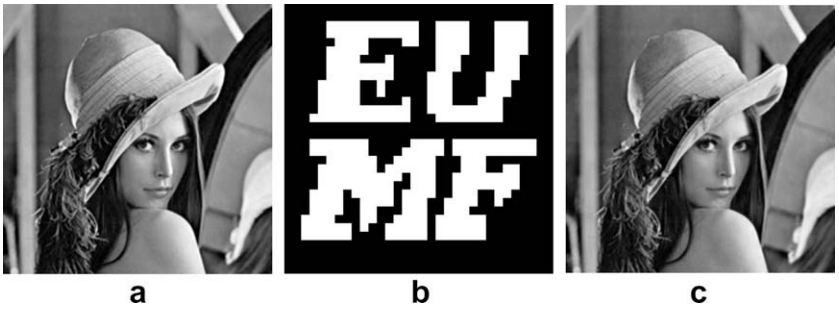

Fig. 3. (a) Host image Lena, (b) watermark, and (c) watermarked Lena.

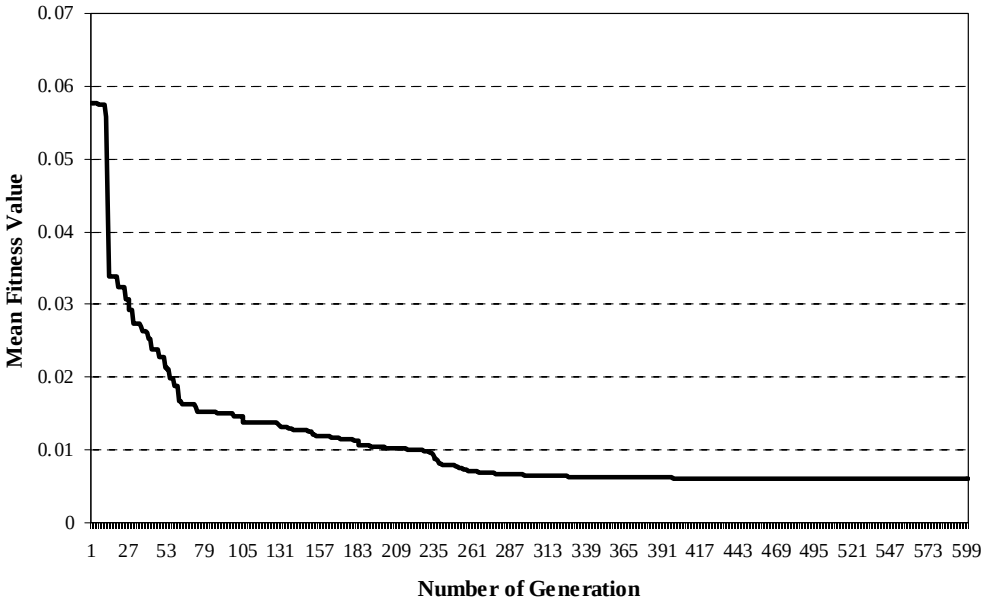

Fig. 4. Evolution curve of DE.

The number of the modified SVs is directly related to the size of the watermark. There is a trade-off with watermark size: the larger the size of watermark (the larger the number of modified SVs) is, the more the distortion of watermarked image quality and vice versa. It is worth noting that watermark which is too small tend to have little value in real applications. The size of the watermark can also be optimised in accordance with the host image and intended attacks but it is out of the scope of this paper. In many watermarking applications, the size of the watermark image is chosen as 32×32. The same size of watermark is also employed in this work. Therefore, only 32 SVs of the host image are needed to be modified. Due to the large SVs of the host image are more stable than the smaller ones against common image processing attacks, the maximum 32 SVs of the host image (σ_1, σ_2,...,σ_{32}) are employed. Therefore, each vector in the population corresponding to a solution in the problem domain consisted of 32 parameters. Every parameter represented a possible value for SF.

For obtaining an optimal result, the parameters of DE should be carefully selected. The observed parameters of DE are mutation factor (weighting factor), crossover rate, and population size. Initial values of each DE parameter are empirically chosen. After that, while one parameter is altered, the others are kept fixed. Each experiment was repeated 30 times with different initial populations and settings.

An important question for any heuristic is when an optimisation should be halted. It is almost impossible to guarantee that an optimal solution has been reached after a number of generations because of the stochastic behavior of DE. However, in practice, several termination criteria have been employed within the DE. In this study, the termination criteria employed is the "maximum number of generations" that was obtained experimentally. First, optimisation process was carried out 30 times for a maximum of 600 evaluations. Than, the fitness values of independent runs were averaged and plotted with respect to number of generations (Fig. 4). As can be seen from the figure, it is observed that there is no significant improvement after 400 generations. Therefore, "the maximum number of generations" as the termination criteria was set to 400 for the other experiments conducted in this study.

In addition to the DE mutation defined in Eqs. (11) and (12), different objective vectors can be chosen to form the mutation vector as defined in [42]. The mutation methods used in this work are given as follows:

$$v_{i,G+1} = x_{best,G} + F(x_{r1,G} - x_{r2,G}), \tag{17}$$

$$v_{i,G+1} = x_{best,G} + F(x_{r1,G} - x_{r2,G} + x_{r3,G} - x_{r4,G}) \tag{18}$$

These alteration and selection procedures given above are known as the 'DE/best/1/exp' (DE1) operator and 'DE/best/2/exp' (DE2) operator, respectively. Experiments were carried out using both of the operators. DE1 and DE2 were run 30 times with different initial populations and settings as summarised in Table 1 that illustrates the results of correlation values ($corr_I$ and $corr_W$) obtained by averaging the results of independent runs. The standard deviation (Std-

Table 1
Average results of DE1, DE2 and GA. Global optimum solution is highlighted.

	CR	F	NP	Average/Stdev	$corr_I$	$corr_W$				fval
						RT	RS	AV	SH	
DE1	0.7	0.6	50	Average	0.9993	0.9938	0.9879	0.9894	0.9804	77.1
				Stdev	0.00013	0.00014	0.00024	0.00022	0.00036	
	0.8	0.6	50	Average	0.9992	0.9920	0.9860	0.9877	0.9848	75.0
				Stdev	0.00022	0.00149	0.00239	0.00216	0.00255	
	0.9	0.6	50	Average	0.9993	0.9967	0.9918	0.9896	0.9743	78.5
				Stdev	0.00012	0.00134	0.00276	0.00239	0.00105	
	0.9	0.7	50	Average	0.9911	0.9745	0.9706	0.9710	0.9756	27.2
				Stdev	0.00172	0.02434	0.02495	0.02473	0.01933	
	0.9	0.8	50	Average	0.9949	0.9824	0.9807	0.9812	0.9845	43.1
				Stdev	0.02122	0.03410	0.03143	0.03112	0.02625	
	0.9	0.6	100	Average	0.9994	0.9962	0.9918	0.9895	0.9870	104.7
				Stdev	0.00343	0.00917	0.00677	0.00681	0.00363	
	0.9	0.6	150	Average	0.9994	0.9969	0.9918	0.9920	0.9851	108.4
				Stdev	0.00021	0.00184	0.00283	0.00267	0.00431	
DE2	0.3	0.5	50	Average	0.9992	0.9954	0.9955	0.9973	0.9837	127.0
				Stdev	7.44E-05	0.00028	0.00035	0.00018	0.00081	
	0.4	0.5	50	Average	0.9997	0.9966	0.9946	0.9961	0.9896	163.7
				Stdev	9.72E-05	0.00052	0.00081	0.00067	0.00316	
	0.5	0.5	50	Average	0.9994	0.9961	0.9950	0.9966	0.9845	131.5
				Stdev	0.00016	0.00072	0.00101	0.00074	0.00394	
	0.6	0.5	50	Average	0.9995	0.9963	0.9943	0.9960	0.9863	136.6
				Stdev	0.00018	0.00092	0.00175	0.00126	0.00508	
	0.4	0.5	100	Average	0.9997	0.9974	0.9962	0.9972	0.9930	228.8
				Stdev	4.47E-05	0.00028	0.00113	0.00114	0.00074	
	0.4	0.5	150	Average	0.9997	0.9974	0.9964	0.9976	0.9931	238.7
				Stdev	1.19E-16	0.00028	0.00044	0.00031	0.00075	
GA	0.7	0.04	150	Average	0.9996	0.9965	0.9965	0.9976	0.9896	186.1
				Stdev	0.00021	0.00068	0.00058	0.00042	0.00287	

Table 2
One of the best SFs obtained by DE2.

Number of Variables	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	SFs
	1862.7	392.08	2.28	0.62	0.87	-3.33	-1.63	-1.15	-2.63	1.23	1.46	1.13	-1.05	-1.06	-72.47	-58.97	
	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
	-1.23	-1.17	1.31	1.37	-1.68	-1.54	-1.62	-1.74	1.26	1.27	1.24	-1.72	57.03	32.69	177.97	204.26	

Table 3
Correlation values obtained by the SFs given in Table 2.

	$corr_I$	$corr_W$											
		RT	RS	AV	SH	GN	TR	CP1	JPEG	CP2	GC	MF	HE
DE2 SFs	0.9997	0.9978	0.9967	0.9976	0.9939	0.9970	0.9854	0.9862	0.9996	0.9948	0.9894	0.9989	0.9946
Constant SFs													
0.1	1.0000	0.9790	0.9827	0.9829	0.9673	0.9678	0.9596	0.9625	0.9763	0.9676	0.9825	0.9843	0.9712
0.3	0.9996	0.9799	0.9820	0.9829	0.9728	0.9744	0.9708	0.9708	0.9922	0.9732	0.9809	0.9839	0.9717
0.5	0.9989	0.9783	0.9820	0.9829	0.9713	0.9749	0.9773	0.9710	0.9973	0.9763	0.9815	0.9839	0.9737
0.7	0.9986	0.9703	0.9838	0.9846	0.9710	0.9774	0.9776	0.9751	0.9975	0.9794	0.9823	0.9856	0.9798
0.9	0.9981	0.9741	0.9840	0.9847	0.9733	0.9837	0.9804	0.9784	0.9988	0.9818	0.9828	0.9858	0.9858

Table 4
PSNR values obtained by the SFs given in Table 2.

	$PSNR_I$	$PSNR_W$											
		RT	RS	AV	SH	GN	TR	CP1	JPEG	CP2	GC	MF	HE
DE2 SFs	38.023	22.596	20.158	19.212	21.029	19.874	13.141	15.515	34.602	15.716	17.982	20.279	23.821
Constant SFs													
0.1	50.768	17.956	19.170	18.812	16.058	16.070	12.778	14.567	17.440	14.960	17.207	19.199	16.405
0.3	37.770	18.045	18.919	18.602	16.811	16.800	12.845	15.237	22.065	15.235	17.244	18.933	16.461
0.5	33.605	17.563	18.409	18.143	16.485	16.605	12.320	15.175	26.697	14.821	16.339	18.409	16.772
0.7	32.412	16.092	17.952	17.719	16.245	16.389	11.563	14.722	26.994	14.024	15.728	17.989	17.946
0.9	30.855	16.557	17.175	16.888	16.272	16.534	11.214	13.811	30.088	13.408	15.080	17.202	19.537

dev) for the correlation values of the extracted watermarks and watermarked images were also computed showing the variations of the correlation values with respect to the average correlation values. Table 1 also lists the computed Stddev values of the results. From the standard deviations calculated for all the experiments, it can be viewed that the variability between different runs is quite low. This indicates that the algorithm is working similarly with different runs and is not dependent on the initial population.

The experimental results given in Table 1 indicate that the overall performance of DE2 is better than those of DE1. The results also show that the performance of DE2 is slightly better than that of GA presented in Ref. [26]. In addition, DE is easy to implement and re-

Fig. 5. Robustness test results: distorted images and corresponding extracted watermarks.

quires a negligible amount of parameter tuning to achieve considerably good search results. The most suitable mutation factor for DE2 that gives the best fitness value is 0.5. This is the value for F that is mentioned as the standard choice for DE2 in DE home page [43]. The results also show that the best crossover rate is obtained at 0.4. As can be observed from the results, the fitness value increases with the increase in population size of DE. However, the larger the population size, the higher the computation time it requires. Therefore, it is important to achieve a good or acceptable result within a reasonable computation time. As can be seen from Table 1, the difference between the fitness values obtained for different NP is not very high. Thus, the experiments were not carried out with the NP beyond 150. Hence, these three values ($CR = 0.4$, $F = 0.5$ and $NP = 150$) are finally chosen for DE2. One of the best SFs found by DE2 is given in Table 2. The watermarked image obtained by employing these SFs is illustrated in Fig. 3c. It is clear from Fig. 3c that the watermarked image is not distinguishable from the original one.

The effectiveness of the algorithm was also tested to cope with several other attacks in addition to the attacks used in the optimisation process, such as Gaussian noise (GN) (mean = 0, variance = 0.001), 50 pixel $\times$ 50 pixel translation (TR), cropping (on left half (CR1) and 20 pixels on each side (CR2)), JPEG compression (quality = 5), gamma correction (0.6), histogram equalization (automatic) and median filtering (3×3) attacks. For all of the attacks employed in this study, Table 3 shows the correlation values obtained by the SFs given in Table 2. Moreover, the peak signal-to-noise ratio ($PSNR$) was also utilized to evaluate the quality of the watermarked image ($PSNR_I$) and extracted watermarks ($PSNR_W$) by using the same SFs. The PSNR results obtained are given in Table 4.

Fig. 5 illustrates the distorted images after attacks and the corresponding extracted watermarks. To better view the extracted watermarks, the images are enlarged. The same experiments were also carried out with constant SFs [31] ranging from 0.1 to 0.9 with the interval of 0.2. The results of constant SFs are also illustrated in Table 3 and 4 for the comparison of the correlation and PSNR values, respectively. As can be seen from Table 3 and 4, the smaller the constant SF, the better the image quality and the weaker the robustness. In contrary, the larger the constant SF, the more the distortion of the quality of the host image (transparency) and the stronger the robustness. On the other hand, results of the multiple SFs obtained by DE2 show both the significant improvement in transparency and the robustness under attacks. The experimental results obtained confirm the effectiveness of proposed watermarking technique and its superiority over the use of a constant SF.

5. Conclusion

This paper proposes a new optimal method of robust image watermarking based upon SVD using differential evolution. DE was employed to optimise the fitness function that was defined based on the two conflicting requirements of image watermarking, that is, visibility and robustness. The watermark image is embedded into the host image by modifying the singular values of it. In order to achieve highest possible robustness without loosing transparency, modifications were made using multiple scaling factors that were obtained by employing DE2. Experimental results show the efficiency of multiple SFs estimated by DE2 and the superiority of the algorithm with respect to the use of a constant SF in terms of robustness and transparency. The results achieved with the DE2 are also better than the results of GA.

Different watermarking applications may require robustness to some specific attacks. Due to the flexibility of the developed system, the attacking modules used in this work for training DE can easily be replaced with those attacking schemes. Further research can be carried out with the U and V components of the SVD [40]. The developed system can also be applied to watermarking in DWT, DFT, DCT and the spatial domains, by changing the SVD embedding algorithm into the other domain embedding algorithms.

As with the other published intelligent based watermarking techniques [15–30], one of the disadvantages of the developed technique is that it is a time consuming process to find an optimal solution for watermark embedding. However, watermark embedding into a multimedia media data is mostly a static and an off-line process, i.e., it is done either during or just after the creation of data by the content owner. Once the data is watermarked, the copies of it are distributed by the owner, that is, the host data does not need to be watermarked again and again for each distribution. The developed technique has no problem with watermark extracting stage. It extracts the watermark similar to the classical SVD. The proposed watermarking algorithm can be preferred to the classical one because it not only improves the transparency and robustness, but it also introduces more security to the watermarking by employing multiple scaling factors.

Although the preliminary results of the experiments conducted in this study seem to be satisfactory when they are compared with those obtained by a single SF, further study can still be carried out in order to assess the performance of other types of fitness functions that may either employ different image fidelity metrics in Eq. (15) or be designed in a different way. It is also possible to define a fitness function that combines different fidelity metrics. If the fitness function (Eq. (15)) is employed using another fidelity metric (such as MSE, PSNR, WPSNR, SSIM [44], etc.) other than the correlation values, a big difference might be occurred between the fidelity metrics of the watermarked image and the extracted watermarks as is the case in PSNR values shown in Table 4. In these instances, weighting factors should be introduced in the fitness function in order to balance the transparency and robustness of the embedded watermark, i.e., between the fidelity metrics [16,18,25]. Therefore, since there is no universal image quality metric, special attention has to be devoted to the selection of the fitness function which plays an important role in successful searches using DE.

Acknowledgment

The author would like to thank Research Foundation of the Erciyes University, Kayseri, Turkey for supporting this work under the Grant No. FBA-05-04.

References

[1] F. Hartung, M. Kutter, Proc. IEEE 87 (1999) 1079.
[2] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Proc. IEEE 87 (1999) 1062.
[3] S.J. Lee, S.H. Jung, A survey of watermarking techniques applied to multimedia, in: Proceedings of IEEE International Symposium on Industrial Electronics, Pusan, Korea, 2001, pp. 272–277.
[4] V.M. Potdar, S. Han, E. Chang, A survey of digital image watermarking techniques, in: Proceedings of IEEE Third International Conference on Industrial Informatics, INDIN'05, 2005, pp. 709–716.
[5] C.C. Chang, T.S. Chen, L.Z. Chung, Inf. Sci. 141 (2002) 123.
[6] D.C. Wu, W.H. Tsai, Pattern Recogn. Lett. 24 (2003) 1613.
[7] S.C. Chu, J.F. Roddick, Z.M. Lu, J.S. Pan, IEICE Trans. Fund Electron. Commun. Comput. Sci. E87-A1 (2004) 282.
[8] Y.H. Yu, C.C. Chang, Y.C. Hu, Pattern Recogn. 38 (2005) 691.
[9] C.H. Chang, Z. Ye, M. Zhang, IEEE Trans. Circuits Sys. Video Technol. 15 (2005) 65.
[10] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, IEEE Trans. Image Process. 14 (2005) 253.
[11] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, IEEE Trans. Image Process. 6 (1997) 1673.

[12] C.T. Hsu, J.L. Wu, IEEE Trans. Circuits Sys. II: Analog Digital Signal Process. 45 (1998) 1097.
[13] Y. Wang, J.F. Doherty, R.E. Van Dyck, IEEE Trans. Image Process. 11 (2002) 77.
[14] J.L. Liu, D.C. Lou, M.C. Chang, H.K. Tso, Comput. Stand. Interf. 28 (2006) 356.
[15] D.C. Lou, T.L. Yin, Opt Eng. 41 (2002) 2675.
[16] C.S. Shieh, H.C. Huang, F.H. Wang, J.S. Pan, Pattern Recogn. 37 (2004) 555.
[17] F.Y. Shih, Y.T. Wu, J. Vis. Commun. Image Represent. 16 (2005) 115.
[18] Y.L. Chang, K.T. Sun, Y.H. Chen, ART2-based genetic watermarking, in: 19th international conference on advanced information networking and applications, 2006, pp. 729–734.
[19] Y.T. Wu, F.Y. Shih, IEEE Trans. Sys. Man Cybernet – Part B: Cybernet 36 (2006) 24.
[20] D.C. Lou, T.L. Yin, IEICE Trans. Fund Electron Commun. Comput. Sci. E84-A 8 (2001) 2052.
[21] G.J. Yu, C.S. Lu, Pattern Recogn. 36 (2003) 957. MLHY.
[22] P. Kumsawat, K. Attakitmongcol, A. Srikaew, IEEE Trans. Signal Process. 53 (2005) 4707.
[23] T.E. Areef, H.S. Heniedy, O.M.O. Mansour, Optimal transform domain watermark embedding via genetic algorithms, in: ITI third international conference on information and communications technology, enabling technologies for the new knowledge society, 2005, pp. 607–617.
[24] D. Lee, T. Kim, S. Lee, J. Paik, Genetic Algorithm-Based Watermarking in Discrete Wavelet Transform Domain, vol. 4113, LNCS, 2006. pp. 709–716.
[25] Y. Lu, J. Han, J. Kong, G. Hou, W. Wang, A Novel Color Image Watermarking Method Based on Genetic Algorithm, vol. 345, LNCIS, 2006. pp. 72–80.
[26] V. Aslantas, Int. J. Electron Commun. (AEU) 62 (2008) 386.
[27] P.T. Yu, H.H. Tsai, J.S. Lin, Signal Process. 81 (2001) 663.
[28] Y.G. Fu, R.M. Shen, H.T. Lu, Electron Lett. 40 (2004) 986.
[29] F. Zhang, H. Zhang, Neurocomputing 67 (2005) 345.
[30] R.M. Shen, Y.G. Fu, H.T. Lu, J. Sys. Software 78 (2005) 1.
[31] R. Liu, T. Tan, IEEE Trans. Multimedia 4 (2002) 121.
[32] D.V.S. Chandra, Digital image watermarking using singular value decomposition, in: Proceedings of 45th IEEE Midwest Symposium on Circuits and Systems, Tulsa, OK, 2002, pp. 264–267.
[33] E. Ganic, N. Zubair, A.M. Eskicioglu, An optimal watermarking scheme based on singular value decomposition, in: International Conference on Commutation Network and Information Security, Uniondale, NY, 2003, pp. 85–90.
[34] E. Ganic, A.M. Eskicioglu, Secure DWT-SVD domain image watermarking: embedding data in all frequencies, in: ACM Multimedia and Security Workshop, Magdeburg, Germany, 2004, pp. 166–174.
[35] B. Zhou, J. Chen, Chin J. Image Graphics 9 (2004) 506.
[36] F. Huang, Z.H. Guan, Pattern Recogn. Lett. 25 (2004) 1769.
[37] C.C. Chang, P. Tsai, C.C. Lin, Pattern Recogn. Lett. 26 (2005) 1577.
[38] P. Bao, X. Ma, IEEE Trans. Circuits Sys. Video Technol. 15 (2005) 96.
[39] J.M. Shieh, D.C. Lou, M.C. Chang, Comput. Stand. Interf. 28 (2006) 428.
[40] K.L. Chung, W.N. Yang, Y.H. Huang, S.T. Wu, Y.C. Hsu, Appl. Mathe. Comput. 188 (2007) 54.
[41] R. Storn, K. Price, J. Global Optim. 11 (1997) 341.
[42] D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, M.N. Vrahatis, Parallel differential evolution, in: Proc. Congress on Evolutionary Computation (CEC2004), Portland, Oregon, 2004, pp. 2023–2029.
[43] DE home page: <http://www.icsi.berkeley.edu/~storn/code.html>.
[44] S. Dandapat, O. Chutatape, S.M. Krishnan, Perceptual model based data embedding in medical images, in: International Conference on Image Processing, Singapore, 2004, pp. 2315–2318.