International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:2, 2007

Image Authenticity and Perceptual Optimization via Genetic Algorithm and a Dependence Neighborhood

Imran Usman, Asifullah Khan, Rafiullah Chamlawi, and Abdul Majid

***Abstract*—**Information hiding for authenticating and verifying the content integrity of the multimedia has been exploited extensively in the last decade. We propose the idea of using genetic algorithm and non-deterministic dependence by involving the un-watermarkable coefficients for digital image authentication. Genetic algorithm is used to intelligently select coefficients for watermarking in a DCT based image authentication scheme, which implicitly watermark all the un-watermarkable coefficients also, in order to thwart different attacks. Experimental results show that such intelligent selection results in improvement of imperceptibility of the watermarked image, and implicit watermarking of all the coefficients improves security against attacks such as cover-up, vector quantization and transplantation.

***Keywords*—**Digital watermarking, fragile watermarking, genetic algorithm, Image authentication.

I. Introduction

RECENT developments in computer industry and the prevalence of interconnected networks have prompted the research in the field of digital watermarking, which involves hiding or embedding of information codes in signals such as audio, video, images, text and graphics. These codes, known as *digital watermarks*, may contain different information, based on the application, about the signal they are embedded in. The application range for digital watermarking includes copyright protection, content authentication, media forensics, data binding, broadcast monitoring, and covert communication [1]. In case of content authentication, such a watermarking is called *fragile watermarking*.

Manuscript received February 28, 2007. The authors greatly acknowledge the financial support provided by Higher Education Commission, Government of Pakistan (HEC-Pakistan) under the indigenous PhD scholarship program No.17-5-1(Cu-204) HEC/Sch/2004 and Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, Pakistan.

Imran Usman is with the Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabd, Pakistan (phone: 92-51-2207381 ext.3030; e-mail: Imran.usman@gmail.com).

Asifullah Khan is with the Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan (e-mail: asif@pieas.edu.pk).

Rafiullah Chamlawi is with the Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan (e-mail: chamlawi@pieas.edu.pk).

Abdul Majid is with the Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan (e-mail: ab_majid@pieas.edu.pk).

A fragile watermark is simply a mark, likely to be destroyed after a marked media is modified in any way, so that alarms can be raised when wrong watermark is extracted [1, 2]. There are two potential benefits to using watermarks in content authentication:

1. Watermarks remove any need to store separate, associated metadata, such as cryptographic signatures
2. A watermark undergoes the same transformations as the work in which it is embedded. Unlike appended signature, the watermark itself changes when the work is corrupted.

By comparing the watermark against a known reference, it might be possible to conjecture not just that an alteration occurred but what, when, and where changes happened [1].

To be effective, a fragile watermarking scheme must localize tampering, detect geometric transformations, and alert other image processing operations which may affect contents of the media. In addition, it should signal removal of original objects and addition of foreign objects. It must not leave any security gaps for attacks such as cut-and-paste [3,11] and birthday attack [3]. Block-wise dependence is accepted as an essential requirement for countering the afore-mentioned attacks. Baretto *et al*. [3] has pointed out that dependence with deterministic context is susceptible to transplantation attacks [3] as discussed in [2,3,11].

Several different schemes are proposed in literature to cope with most of the attacks, including Wong's public key scheme [8], and Wu's scheme [9] of inserting a binary watermark sequence into the DCT coefficients via look-up table. But they are block-wise independent and therefore vulnerable to cover-up, vector quantization & transplantation attacks.

Admitting the importance of establishing dependence among neighboring pixels or blocks, several schemes are proposed including Li's scheme that uses a binary feature map extracted from the underlying image as watermark [5]. But, this is again vulnerable to transplantation attack, since contextual dependence is not non-detrerministic.

To thwart transplantation attack, Barreto et al. [3] proposed a scheme, which calculates a hash function that is relatively time-consuming and the accuracy of localizing tampering is limited by the size of the block.

Li *et al.* proposed a transform domain scheme [2] which

copes with most of the attacks, but uses only middle frequency band for watermark embedding, which is not an optimal choice. Also, while establishing block-wise dependence, high frequency coefficients are not taken into consideration, and thus, leaves a wide-open security gap for attacks to be mounted on them.

All the above mentioned schemes employ the embedding of the watermark into some of the selected coefficients in their corresponding domains, which might be fixed in a predetermined set of coefficients. One major disadvantage for these schemes is how to choose the predetermined set. In most of the literature, *middle frequency bands* are claimed as a trade-off for watermark embedding in the transform domain [12].

Therefore, instead of using a fixed frequency band for watermark embedding, we propose a scheme which intelligently selects the optimum/near optimum frequency band for embedding and, thus, provides enhancement in visual quality. Another contribution in this work is that it takes into consideration all the rest of the unwatermarkable coefficients, both in the higher and lower frequencies, thus, leaving no security gap at all for attacks to be mounted on. This work extends Li's work [2] with modifications on content integrity verification by involving all the unwatermarkable coefficients from a *9-neighborhood system*, and a modulation rule at the watermark embedding process which decreases the embedding distortions. In addition to this, we enhance the visual quality of the watermarked image through genetic algorithm.

The rest of this work is organized as follows. Section 2 gives a brief description and outline of genetic algorithm. Section 3 proposes the watermark embedding algorithm followed by the extraction algorithm in section 4. Experimental results are summarized in section 5 and section 6 concludes this work.

II. Genetic Algorithm: An Overview

Formal search techniques are mostly incapable of optimizing non-linear functions which comprises of multiple objectives [10]. The Genetic Algorithm (GA) is a type of stochastic search technique which is directed. It is a method for solving optimization problems that are based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals, based on their fitness, from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

In GA, an individual in a population is represented by an encoded binary string, called the *chromosome*. The fitness function, which is composed of multiple variables and/or objectives to be optimized by GA, generates their corresponding fitness values. A population consists of these chromosomes. The elements, or the *genes* (or *genomes*), in the binary strings are adjusted to maximize or minimize the fitness value. For every generation in GA, a pre-determined number of individuals correspondingly produce fitness values associated with the chromosomes.

Fig. 1 demonstrates the flow chart for a typical genetic algorithm. It begins with defining the optimization parameters and the fitness function, and it ends by testing for convergence.

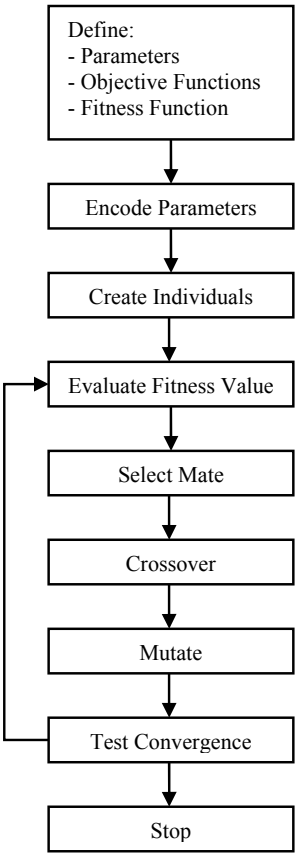

Fig. 1 The Flow chart of a typical Genetic Algorithm

The following outline summarizes how the genetic algorithm works:

1. The algorithm begins by creating a random initial population.
2. The algorithm then creates a sequence of new populations. At each step, the algorithm uses the individuals in the current generation to create the next generation. To create the new generation, the algorithm performs the following steps:
   a. Scores each member of the current population by computing its fitness value.
   b. Scales the raw fitness scores to convert them into a more usable range of values.
   c. Rank and Selects the parents based on their fitness.

d. Produces children from the parents. Children are produced either by making random changes to a single parent (*mutation*), or by combining the vector entries of a pair of parents (*crossover*).
e. Replaces the current bad individuals with the children to form the next generation.

3. The algorithm stops when one of the stopping criteria is met, which may include number of generations, time limit or fitness limit**.**

III. Watermark Embedding Algorithm

Fig. 2 presents block diagram of watermark embedding algorithm of the proposed scheme. First, we initialize a population of chromosomes randomly. We let the original image to be **X** with size $M \times N$ and perform an 8×8 block DCT transformation on **X** to generate **Y**.

$$\mathbf{Y} = DCT(\mathbf{X}) \tag{1}$$

and

$$\mathbf{Y} = \bigcup_{i=1}^{r} \mathbf{Y}_i(j), \tag{2}$$

where $r = (M \times N)/(8 \times 8)$ i.e. the number of blocks, and $j = 0,1,\ldots,63$ representing the coefficient index in zigzag order within each 8×8 block as shown in Fig. 3.

A binary sequence, ***S***, of the same size as the image, is generated using a secret key, key_o. Another binary sequence map, ***T***, is generated of the same size as of original image such that all its pixels corresponding to the non-zero-valued coefficients are set to 1, and the others set to 0. We generate the watermark ***W*** such that,

$$\boldsymbol{W} = \boldsymbol{S} \oplus \boldsymbol{T}. \tag{3}$$

For each DCT block, five coefficients are selected as watermarkable according to the following rule:

"The coefficients correspond to the five on-bits (the only on-bits) in the chromosome, ***x***, of the current population".

For each of the watermarkable coefficient, $Y_i(j)$, a secret sum $SUM_i(j)$ is calculated by adding up coefficients picked from a *9-neighbourhood* system, as shown in Fig. 4, according to their corresponding watermark bits in ***W***.

$$SUM_i(j) = \sum_{m=1}^{9} \sum_{n \in \mathfrak{h}} (W_m(n) \oplus W_i(j)) \cdot Y_m(n), \tag{4}$$

where, $n = 0,1 \cdots, 63$,and m is the dependence neighborhood including the block itself as shown in figure 4, whereas, $\mathfrak{h}$ is the set of indices of the five watermarkable coefficients.

Let $Cat_i(j)$ be the concatenation of $SUM_i(j)$ and $Y_i(j)$ in one's complement format. And, let $Cat_i'(j)$ be the concatenation of $Cat_i(j)$ with $Cat_i(j)$. The five watermarkable coefficients are modulated so that,

$$Alarm\left(Cat_i'(j)\right) = W_i(j) \tag{5}$$

Where *Alarm* is a function which returns 1 as output when the number of "1" bits is odd, and 0 when it is even.

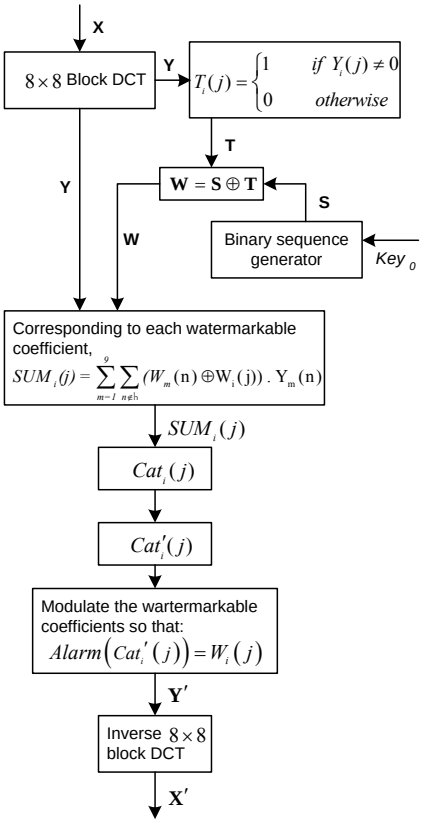

Fig. 2 Block diagram of the watermarking embedding algorithm

This modulation implicitly performs the embedding of the watermark, to generate the watermarked image Y'. Next, we apply the inverse DCT transformation to generate a watermarked image, X', in the spatial domain.

$$\mathbf{X}' = inverse\ DCT(\mathbf{Y}') \tag{6}$$

To perform the fitness evaluation, we take into account the signal to noise ratio (SNR) and structural similarity index (SSIM) of the original image and the watermarked image using the fitness function

$$f_c = SNR + (SSIM_c \quad \alpha_c), \qquad \cdot \tag{7}$$

where f_c is the fitness value of the current chromosome, and α_c is a scaling factor normally in the range of 30 to 50, to magnify the SSIM values in order to balance the influences caused by both SNR and SSIM. SNR and SSIM are quality measures and are added to provide a generic visual quality analysis in any given image.

After calculating the fitness values corresponding to all individuals in a given population, individuals with larger fitness values are selected for reproduction. Replication, mutation and crossover operators are then applied to create the

next generation, thus, achieving diversity as well as convergence in the solution space generation by generation.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Fig. 3 Indices of DCT coefficients in zigzag order of the i^{th} block. Shaded blocks are the selected watermarkable coefficients

After several generations, the algorithm would stop when one of the stopping criteria is met. The scheme may converge to an optimal choice of coefficients for watermark embedding, which would provide enhanced imperceptibility. Figure 5 illustrates a block diagram for watermark embedding process in a genetic watermarking system. The optimum chromosome, representing the locations of the watermarkable DCT coefficients, is saved as a key, key_1, and is transmitted over an open network by integrating cryptography with our watermarking technology.

IV. Watermark Extraction Algorithm

For the extraction of watermark, the original image is not required. The watermarked image X' is 8×8 block DCT transformed to generate Y'. Binary sequences $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{W}$ are generated following the same procedures as in the embedding process. key_1 is used to identify the five watermarked coefficients, and their indices are saved in a set $\natural$. The secret sum, $SUM_i(j)$, corresponding to marked coefficients, is calculated using Eq.(4). $SUM_i(j)$ is then concatenated with $Y_i(j)$ in one's complement format to produce $Cat_i(j)$ and $Cat_i'(j)$ as explained in section 3.

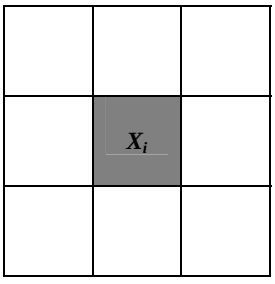

Fig. 4 9-neighbourhood of the shaded 8×8 block X_i including the block itself

Next, the selected coefficients are authenticated by verifying whether Eq. (5) holds or not. If the coefficient fails the authentication, i.e. Eq. (5) does not hold, the block to which the coefficient belongs to, is shaded to indicate the occurrence of tempering. To turn off the false alarms, any blocks marked as inauthentic surrounded by less than z inauthentic blocks are treated as authentic.

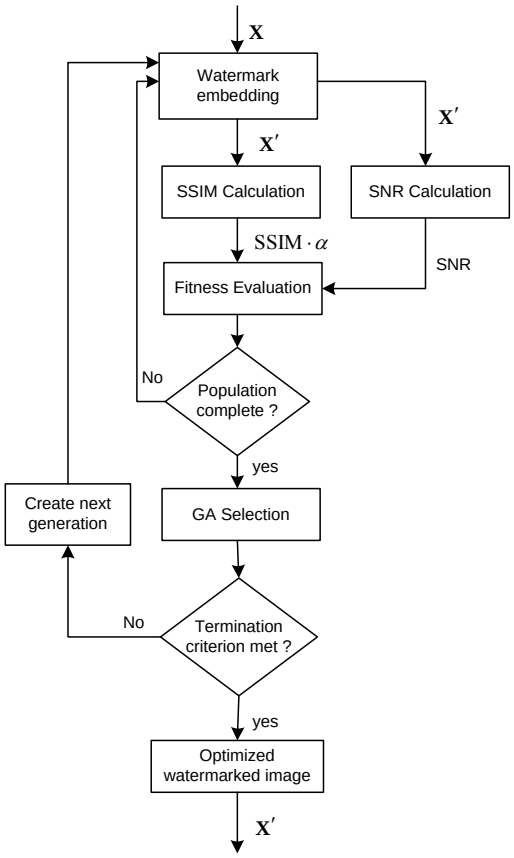

Fig. 5 The block diagram for watermark embedding in genetic algorithm based watermarking system

V. Experimental Results

In our experiments, *Lena* image of size 256×256 has been used as shown in Fig. 6(a). Fig. 6(b) shows the watermarked *Lena* image with our proposed technique. We can see, subjectively as well as objectively, that the distortion after adding the watermark is invisible.

Table I shows the experimental results in terms of SNR and SSIM after different generations. Improvements in the watermarked image quality can be observed by looking at the SNR and scaled SSIM values in Table I. We scale SSIM values in the current experiments by multiplying it with a scaling factor, α=50.

Fig. 6 (a) Original *Lena* image (b) Watermarked version of the same image. Note that the two images are perceptually alike

To test effectiveness of our proposed technique, experiments are conducted by mounting local tampering and low-pass filtering attacks on the watermarked image. The value selected for z is 6. Fig. 7(a) shows *Lena* image tampered by placing a mole over the upper lip. Fig. 7(b) and 7(c) shows the results obtained from our proposed technique to localize tampering before and after the false alarms are turned off respectively.

Experimental results under low-pass filtering attack are shown in Fig. 8. The white shaded blocks indicate that the image has been tampered with.

TABLE I
THE SNR AND SSIM VALUES FOR WATERMARKED LENA IMAGE UNDER DIFFERENT GENETIC ALGORITHM GENERATIONS

Iteration	SNR	SSIM * α
0	37.51	43
50	38.61	44.5
100	39.19	45.7
150	39.92	46.4
200	40.03	46.9

Fig. 7 (a) Tampered Lena image (b) Authentication result when the false alarms are turned on (c) Authentication result after the false alarms are turned off - The shaded square indicates the location of tampering

VI. CONCLUSION

An authentication method for images with enhancement in visual quality based on genetic algorithm is presented in this work. Our authentication algorithm improves Li's work [2], by taking into account all the unwatermarkable coefficients, thus, creating a stronger block-wise dependence to enforce security against attacks. In addition, we propose a modulation rule at the watermark embedding stage which decreases the embedding distortions due to the modulation of coefficients by using one's complement format and calculating $Cat_i'(j)$. Furthermore, watermarked image is enhanced in visual quality

by the intelligent selection of the right coefficients for embedding purpose. Experimental results prove that by selecting the embedding coefficient band intelligently, the watermarked image imperceptibility and visual quality increases.

Fig. 8 Authentication result after a low-pass filtering attack. The shaded blocks indicate that the image has been tampered with

REFERENCES

[1] I. Cox, M. Miller, and J. A. Bloom, "Digital Watermarking", *Morgan Kaufmann,* 2002.
[2] C.-T.Li, "Digital Fragile Watermarking Scheme for Authentication of JPEG Images", *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 151, no. 6, pp. 460 - 466, 2004.
[3] P. S. L. M. Barreto, H. Y. Kim, and V. Rijmen, "Toward secure public-key blockwise fragile authentication watermarking," in *IEE Proceedings - Vision, Image and Signal Processing*, vol. 148, no. 2, pp. 57 – 62, April 2002.
[4] P. W. Wong and N. Memom, "Secret and public key authentication watermarking schemes that resist vector quantization attack," in *Proc. SPIE Security and Watermarking of Multimedia Contents II*, vol. 3971, no. 40, Jan. 2000.
[5] C.-T. Li, D. C. Lou, and T. H. Chen, "Image Authenticity and Integrity Verification via Content-based Watermarks and a Public Key Cryptosystem," in *Proc. IEEE Int. Conf. Image Processing,* vol. III, Vancouver, Canada, Sept. 2000, pp. 694-697.
[6] C.-T. Li and F.-M. Yang, "One-dimensional Neighbourhood Forming Strategy for Fragile Watermarking", *Journal of Electronic Imaging*, vol. 12, no 2, pp. 284-291, 2003.
[7] C.-T. Li, F. M. Yang, and C. S. Lee, "Oblivious Fragile Watermarking Scheme For Image Authentication", in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. VI, Orlando, FL, USA, May 2002, pp. 3445-3448.
[8] P. W. Wong, "A Public Key Watermark for Image Verification and Authentication," in *Proc. IEEE Intl. Conf. Image Processing,* vol. I, Chicago, USA, October 1998, pp.455-459.
[9] M. Wu and B. Liu, "Watermarking for Image Authentication", in *Proc. IEEE Intl. Conf Image Processing,* vol. II, Chicago, USA, October 1998, pp. 437-441.
[10] C. S. Shieh, H.C. Huang, F. H. Wang, and J. S. Pan, "Genetic Wartermarking based on transform domain techniques", *Pattern Recognition* 37 ,pp. 555-565, 2004.
[11] C. -T. Li, H. Si, "Wavelet-based Fragile Watermarking Scheme for Image Authentication" *Journal of Electronic Imaging*, vol. 16, no. 1, 2007.
[12] C. T. Hsu, J. L. Wu, "Hidden Digital Watermarks in Images", *IEEE Transactions on Image Processing.* 8 (1) pp. 58-68, 1999.