

Optimization of Discrete Cosine Transform-Based Image Watermarking by Genetics Algorithm

Iwan Iwut, Gelar Budiman, Ledy Novamizanti

Electrical Engineering Faculty, Telkom University,

Jl. Telekomunikasi 1, telp/fax : 022-7566123, Bandung, Indonesia

Corresponding author, e-mail: iwan.tritoasmoro@gmail.com, gelarbudiman@telkomuniversity.ac.id, ledyaldn@telkomuniversity.ac.id

Abstract

Information hiding of data in an image file is needed by its owner to set his ownership in a logo as a watermark embedded in the image file. Hiding the logo in the image was done in several methods. One of the methods is domain transform using 2D-DCT in which data is embedded in frequency domain of the image. First, the host RGB image is converted to certain color space. The available and chosen color spaces are RGB, YCbCr or NTSC. The layer in which the watermark is embedded also can be selected. The available choices are 1st layer, 2nd layer, 3rd layer, 1st & 2nd layer, 2nd & 3rd layer, 1st & 3rd layer and all layers. After the selected layer of image in certain color space is transformed in block based to frequency domain by DCT, one bit watermark is embedded on the AC coefficient of each block such a way that the bit is represented by specific value called delta in a zigzag and vary length of pixel. The vary parameters optimized by Genetics Algorithm are selected color space, selected layer, block size, length of pixel to be embedded by one bit watermark, and delta. Bit "1" is represented by +delta, and bit "0" is represented by -delta in vary length of pixel after zigzag. The simulation result performs that GA is useful to search the value of parameter that produces controllable the combination between robustness, invisibility and capacity. Thus, GA improves the method by determining the exact value of parameter achieving BER, PSNR and payload.

Keywords: Image Watermarking, 2D-DCT, Genetics Algorithm, AC coefficient, layer, color spaces, BER, PSNR, payload

Copyright © 2016 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Digital watermarking is a way to hide or insert specific information into a data either image, sound, or video, where its presence is unknown to the human senses, and able to deal with signal processing to a certain extent [1]. One of the differences between watermarking and steganography is, the protected thing in watermarking is the host, while the protected thing in steganography is the embedded or hidden message [2]. Digital technology developed in the community raises its own problems. For example in the image data or digital voice which can be easily changed or acquired by others. Many owners of digital data such as image or sound do not want his work altered or recognized by others. Therefore we need a method of verification and digital authentication, and digital watermarking can be used for these two interests.

The host for image watermarking is images or graphical design or photos and the watermark could be binary information, text, audio, or image [3]. The watermark data type in this paper is also an image as logo containing the copyright of the host image. Image watermarking plays important role for minimizing loss of right intellectual property in image production, including photo and graphical design production. There are several parameter that controls the image watermarking performance, such as [4]:

1. Robustness – The watermarks must be having good robustness to face the watermarked image attack. Stirmark benchmark described all image watermarking attacks types which consists of compression, geometric transformation, and enhancement techniques. The performance parameter we use is Bit Error Rate (BER).

2. Imperceptibility – In this paper we address the image watermarking in which the visibility of watermarks is imperceptible. It means that the watermarks embedded in host image

ought to be invisible. The imperceptibility parameter is represented by Peak Signal To Noise Ratio (PSNR).

3. Capacity – The number of bits embedded in the image or bits payload is also taken into account. Although the importance of capacity is not as well as robustness but it is needed for controlling the trade-off between robustness and imperceptibility. The capacity parameter will be represented as C.

4. Security – For any needs, the owner of the content might secure the watermarks. It is executed to prevent the hacker manipulating the watermarks in watermarked image. But it is infrequently case in watermarking. In Steganography, securing the watermarks is mandatory. In this paper, security is not the concern as the watermarking parameter.

Embedding watermarks in an image could be done in spatial domain, transformed domain or hybrid techniques. The most commonly used image watermarking transformed domain is DCT or DFT [5]. DCT (Discrete Cosine Transform) uses only real number for image watermarking in both embedding and extraction, but DFT uses not only real number, but also imaginary number. Famous image compression like JPEG compression used DCT as transformation scheme before manipulating the signal to be compressed. And also in the watermarking, DCT is more interesting because its simplicity for signal manipulation and processing.

Image watermarking by DCT or hybrid with DCT were published already in several papers. Dongyang [6] proposed DCT Image watermarking method based on the mix of time-domain. He generated watermarked image by using both domain, time domain and frequency domain. First, he used an invertible orthogonal matrix to mix the host image in time domain and changed the distribution of image information properly, embedded watermark in DCT domain and transformed the watermarked image into time domain by utilizing the matrix orthogonally. Dongyang used hadamard matrix as orthogonal matrix that can spread out the pixel value to distribute the the noise very evenly making the robustness and invisibility stronger. Several paper combined DWT and DCT to embed the watermark in image host [7-9]. First, DWT is applied into the host image by one or more level DWT, followed by DCT. Watermark is embedded in DCT coefficients. The robustness and invisibility is better than image watermarking with only DCT method.

Combination DCT and DWT was not enough for researcher, Sneha Jose [10] proposed image watermarking by combination 3 method: DWT, DCT and SVD. First, DCT is applied to the image host in block based, then DCT coefficients are decomposed into sub bands using DWT, and at last the singular value of middle value of sub bands are detected and the watermarks are embedded. Sneha claimed that his propose method is better than DWT-SVD, but he didn't compare his method with DCT-DWT method. Neha Bansal [11] published paper that described comparison between LSB, DCT and DWT method in image watermarking. And he found that DCT method is best method between those three methods. Jianghua Li [12] proposed non blind DCT-based image watermarking that combined spatial technique and DCT for embedding the watermark. In his paper, the watermark is not only embedded in AC coefficients, but it is also embedded in DC coefficients. He compared the performance of image watermarking in different watermark position and found that watermark in intermediate frequency of the host image has the best similarity with host image. But he still found difficulty to have good extracted watermark when the watermarked image is attacked by JPG compression. In [13] KE Yun used DWT and SVD as transformation method for embedding the watermark. The attack result of JPG compression also did not reach satisfy performance.

In this paper we propose image watermarking by DCT method in certain color space and layer, and optimized by Genetics Algorithm. First, the host RGB image is converted to certain color space. The available and chosen color spaces are RGB, YCbCr or NTSC. The layer in which the watermark is embedded also can be selected. The available choices are 1st layer, 2nd layer, 3rd layer, 1st & 2nd layer, 2nd & 3rd layer, 1st & 3rd layer and all layers. After the selected layer of image in certain color space is transformed in block based to frequency domain by DCT, one bit watermark is embedded on the AC component of each block such a way that the bit is represented by specific value called delta in a zigzag and vary length of pixel. The vary parameters optimized by Genetics Algorithm are selected color space, selected layer, block size, length of pixel to be embedded by one bit watermark, and delta. Bit "1" is represented by +delta, and bit "0" is represented by -delta in vary length of pixel after zigzag.

Published similar paper about image steganography which used DCT method and optimized by Genetics Algorithm was already proposed by Samir Kumar [14] and Amrita [15]. The difference between their paper with this paper is their paper used LSB method as embedding scheme for the hidden data. And their paper used fix block based when DCT is applied to the image host. And certainly they accentuated security feature rather than robustness, because what they made was steganography.

This paper is organized as follow: section 1 describes introduction of Image Watermarking and its paper review of similar method with the propose method, section 2 describes the color space and layer being used in this paper, section 3 presents DCT and AC coefficient selection, section 4 describes image watermarking model for embedding and extraction phase, section 5 describes genetics algorithm optimization, section 6 describes evaluation of performance, discussion is presented in section 7 and the conclusion is presented in section 8.

2. DCT Based Image Watermarking Optimization by Genetics Algorithm Model

In this section, image watermarking at embedding and extraction stage with the preprocessing of the image is described. Preprocessing before embedding stage consists of color space and layer choice for embedding watermark, DCT conversion and AC coefficient selection. The rest of this section describes Genetics Algorithm as image watermarking optimizer.

2.1. Color Spaces and Layer

In the proposed method, there are 3 color spaces which is selected in the image watermarking, such as: RGB color space, YCbCr color space, and NTSC color space. The optimization algorithm will select which color space gives the optimal performance for image watermarking. And also there are 7 possibilities layer which is selected for locating the embedded watermark. Three color spaces and 7 possibilities layer will be the parameter to be optimized by Genetics Algorithm producing the optimal performance described in section 5.

As displayed in Table 1, respectively the color index of NTSC/YIQ, YCbCr and RGB is 1, 2, and 3. The NTSC/YIQ conversion from RGB and vice versa is as follow [8]:

$$Y=0.299*R + 0.587*G + 0.114*B \quad (1)$$

$$I=0.596 *R - 0.274 *G - 0.322 *B \quad (2)$$

$$Q=0.211 *R - 0.522 *G + 0.311 *B \quad (3)$$

$$R=Y + 0.956*I + 0.621*Q \quad (4)$$

$$G=Y - 0.272*I - 0.647*Q \quad (5)$$

$$B=Y - 1.106*I + 1.702*Q \quad (6)$$

The YCbCr conversion from RGB and vise versa is as follow [16]:

$$Y=16 + 65.481*R + 128.553*G + 24.966*B \quad (7)$$

$$Cb=128 - 37.797*R - 74.203*G + 112*B \quad (8)$$

$$Cr=128 + 112*R - 93.786*G - 18.214 *B \quad (9)$$

$$R=1.1644 *Y + 0 *Cb + 1.596 *Cr - 222.921 \quad (10)$$

$$G=1.1644 *Y - 0.3918 *Cb - 0.7856 *Cr + 135.576 \quad (11)$$

$$B=1.1644 *Y + 2.0172 *Cb + 0 *Cr - 276.836 \quad (12)$$

Table 1 Index of Color Space

Color Index	Color Space	Layer		
1	NTSC	Y	I	Q
2	YCbCr	Y	Cb	Cr
3	RGB	R	G	B

The watermark embedded in host image will use selected layer from 7 layer choices. One selected layer will be the place where the watermark is embedded. And Genetics Algorithm will determine which layer produces optimal performance. The layer and its position to be embedded by the watermark is displayed in Table 2. The tick in the cell means that layer is used for embedding the watermarks.

Table 2. Index of Layer Used for Embedding

	NTSC	Y	I	Q
	YCbCr	Y	Cb	Cr
	RGB	R	G	B
Layer Index	1 st Layer	2 nd Layer	3 rd Layer	
1	√			
2			√	
3				√
4	√		√	
5	√			√
6			√	√
7	√		√	√

2.2. DCT and AC Coefficient Selection

Image watermarking proposed in this paper use frequency domain for embedding the watermarks. Transforming spatial domain to frequency domain is using 2 Dimensions DCT. Before transforming the image to frequency domain, the image is segmented in $M \times M$ block based. M is the integer number that is vary from 2 until 10. Figure 1 and 2 respectively shows the segmentation of the image in $P \times Q$ resolution to 2×2 and 4×4 block based. The 2D DCT equation from the block based image $f(x,y)$ and the inverse is as follow 15:

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} f(x, y) \cos \left[\frac{\pi(2x+1)u}{2M} \right] \cos \left[\frac{\pi(2y+1)v}{2M} \right] \quad (13)$$

$$f(x, y) = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \alpha(u)\alpha(v) C(u, v) \cos \left[\frac{\pi(2x+1)u}{2M} \right] \cos \left[\frac{\pi(2y+1)v}{2M} \right] \quad (14)$$

Where :

$$u \ \& \ v = 0, 1, \dots, M-1$$

x and y is respectively vertical and horizontal pixel position of the image.

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{M}}, & \text{for } u = 0 \\ \sqrt{\frac{2}{M}}, & \text{for } u \neq 0 \end{cases} \quad (15)$$

After transforming the block based image, the frequency domain of the block based image is zigzag processed. As example the zigzag processed to get the vector scheme excluding DC coefficient in 3×3 block based is displayed in Figure 3. Then there is selection of sample or pixel range from the vector scheme of zigzag result to be embedded by watermarks. For this selection two parameter are needed, such as : POS1 and POS2. POS1 is the beginning of the selection range, while POS2 is the end of selection range. Figure 4 displays the vector scheme and the position of POS1 and POS2 as index of the vector. Figure 5 shows the example of selected range if POS1=2 and POS2=5, it means the vector $[c \ d \ e \ f]$ will be replaced by the watermarks. The remaining of vector will be left as it is.

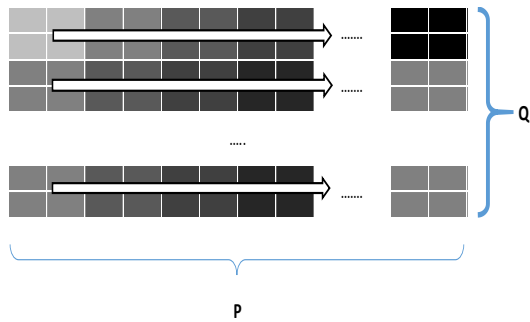


Figure 1. PxQ Image segmented in 2x2 block based

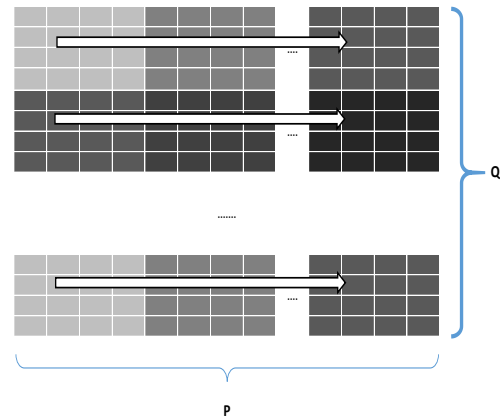


Figure 2. PxQ Image segmented in 4x4 block based

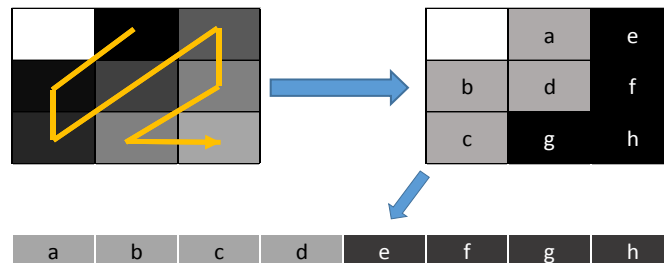


Figure 3. Zigzag in 3x3 block based produces 1x8 vector excluding DC coefficient

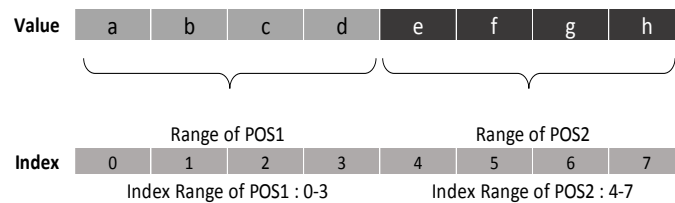


Figure 4. Vector value and index with POS1 and POS2 as the beginning and end of selected range to be embedded by watermarks

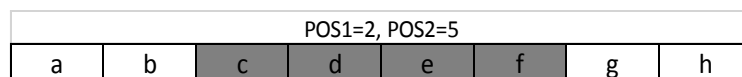


Figure 5. Selected Range if POS1=2 and POS2=5, [c d e f] will be replaced by watermarks

Parameter M , POS1, and POS2 are parameters which optimized by Genetics Algorithm to produce the optimal performance. POS1 and POS2 value range depends on the parameter M . The POS1 and POS2 value will be same for all blocks segmented in the host image. In the watermark extraction stage M , POS1 and POS2 are assumed to be known.

2.3. Embedding and Extraction Stage

Embedding process of image watermarking consists of several steps in which the subprocess are already described in previous section. The detailed steps of embedding stage is as follow:

1. The host image is read and converted to selected color space from RGB as described in Table 1. There are 3 choices of selected color space. Select only one color space. The selected color space is put in variable $C/$ (Color Index).
2. Select the layer used for embedding as described in Table 2. There are 7 choices for selecting the layer. The selected layer is put in variable L .
3. The selected layer is block based segmented into $M \times M$.
4. Each segment from 3 is transformed by DCT to be frequency domain.
5. Each segment in frequency domain is zigzag processed excluding DC component produces the vector $s(n)$ which its size is $(1 \times (M^2-1))$.
6. The selected vector from zigzag vector scheme is assumed as $s(n)$ and the watermark bit is $w(n)$. $s(n)$ length for embedded is POS2-POS1+1 as described in previous section. In embedding stage, the selected vector of zigzag vector scheme or $s(n)$ is replaced by one bit watermarks or $w(n)$ in such way as follow:

$$\begin{aligned}
 &\text{If } w(n) \text{ is "0" then} \\
 & s(n) = -\delta \\
 &\text{if } w(n) \text{ is "1" then} \\
 & s(n) = +\delta
 \end{aligned} \tag{16}$$

δ is a rational number parameter which can be changed, and it is also optimized by Genetics Algorithm to find the optimal trade off performance between robustness and imperceptibility. The range of δ is from 0.1 to 127. Different bit will be embedded in different block. So, the maximum payload of watermarks will be the maximum number of blocks in the host image multiplied by the number of the layer used for embedding process.

7. After all bits are embedded, inverse DCT is applied to each block, and we get the watermarked image.

Extraction process of image watermarking consists of several steps. The detailed steps of extraction stage is as follow:

1. The watermarked image is read and converted to color space same as the color space used in embedding process.
2. Select the layer used for extraction same as the layer used in embedding process.
3. The selected layer is block based segmented into $M \times M$.
4. Each segment from 3 is transformed by DCT to be frequency domain.
5. Each segment in frequency domain is zigzag processed excluding DC component produces the vector $s(n)$ which its size is $(1 \times (M^2-1))$.
6. Select the $s(n)$ in the range from POS1 until POS2 and do the extraction stage below:

$$\begin{aligned}
 &\text{If } \sum_{n=POS1}^{POS2} s(n) < 0 \text{ then} \\
 & \hat{w}(n) = 0 \\
 &\text{if } \sum_{n=POS1}^{POS2} s(n) \geq 0 \text{ then} \\
 & \hat{w}(n) = 1
 \end{aligned} \tag{17}$$

7. After all segments or blocks are extracted as point 6, recover the bits into a vector of bit, then we get the watermarks back.

2.4. Genetics Algorithm Optimization Procedure

Here we have several parameters to be optimized by Genetics Algorithm such as: color index ($C/$), layer index (L), $M \times M$ block segment (M), beginning of selected zigzag vector (POS1), end of selected zigzag vector (POS2), and δ . All above parameters will be the inputs of optimization algorithm for finding their optimum values to get optimum performance of the image watermarking. As described in the section 1, the outputs of the optimization algorithm will be: robustness which represented by BER , imperceptibility or watermarked image quality represented by $PSNR$, and the capacity or payload of watermarks represented by C . The host image and watermarked image will have the same resolution $P \times Q$. Formulation of BER and $PSNR$ is as follow:

$$BER = \frac{\text{Number of Bit Error from extraction}}{\text{Watermark Bit Number}} \quad (18)$$

$$PSNR = 10 * \log \left(\frac{255^2}{MSE} \right) \quad (19)$$

$$MSE = \frac{1}{PQ} \sum_{x=0}^{P-1} \sum_{y=0}^{Q-1} (f(x, y) - \hat{f}(x, y))^2 \quad (20)$$

All output parameters are combined into one output parameter called Fitness Function (*FF*). Finding the optimum trade off of the watermarking performance depends on which parameters of *BER*, *PSNR* and *C* we need to be selected for calculating *FF*. As example, if we need only robustness for finding the optimum watermarking performance, then *FF* will be function of *BER* only. But, if not only robustness but also imperceptibility we need for finding the optimum performance, *FF* will be function of *BER* and *PSNR*.

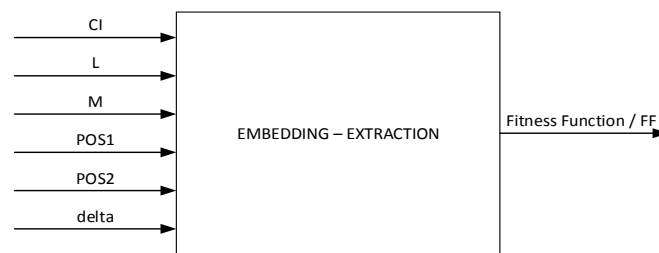


Figure 6. Input and Output Parameters of Embedding and Extraction Process

To determine the equation of *FF*, first we must assume that the range of *FF* is limited from minimum value to maximum value. We can define the minimum value as 0, and the maximum value is 3. Maximum value 3 means each output parameter will have maximum performance in value 1. So, if all the output parameters reach maximum performance or 1, the total performance will be sum of all parameters to be 3. If only *BER* and *PSNR* we need for calculating *FF*, the maximum value of *FF* is certainly 2. We also have to determine the optimum target value of each parameter. *BER* optimum value will be 0. *PSNR* optimum value will be infinite, but we will limit it to 60 dB. And *C* optimum value also depends on the size of the host image, but we can limit it to *N* bit as example. Then, the *FF* formula will be as follow:

$$FF = 1 - BER + PSNR/60 + C/N \quad (21)$$

Where:

$$N = \text{floor}(LPQ / M^2) \quad (22)$$

L = number of layer used for embedding

P = height of host image

Q = width of host image

M = segmented *MxM* block based in host image

The formula uses floor because *P* and *Q* is not necessarily multiple of *M*.

The Genetic Algorithm procedure for finding the optimum performance of the image watermarking displayed in figure 7 is described as follow:

1. State the population number or *PN*, generation number or *GN*, and mutation probability or *Pm*.

2. Initialization of all parameter inputs and then execute embedding and extraction process to get *FF*. This stage is executed *PN* times. Thus we get *FF* and 6 input parameters of *PN* rows, or we get outset population matrix *PNx7*.

3. Sort the outset population matrix in descending order at *FF*, we can name this matrix *PM*.

4. Execute GN times for the tasks below:
 - a. Select the top ten rows of PM and save it to matrix EL at size 10×7 . This matrix is also called as elite matrix.
 - b. Do crossover 3 times producing 2 children from 2 parents in each iteration. Parents are selected by roulette wheel rule. Thus we get 6 children consist of 6 new value of input parameters. Then execute embedding and extraction process for those 6 input parameters producing 6 new FF . We get new population matrix at size 6×7 , save it to matrix CO as result of crossover.

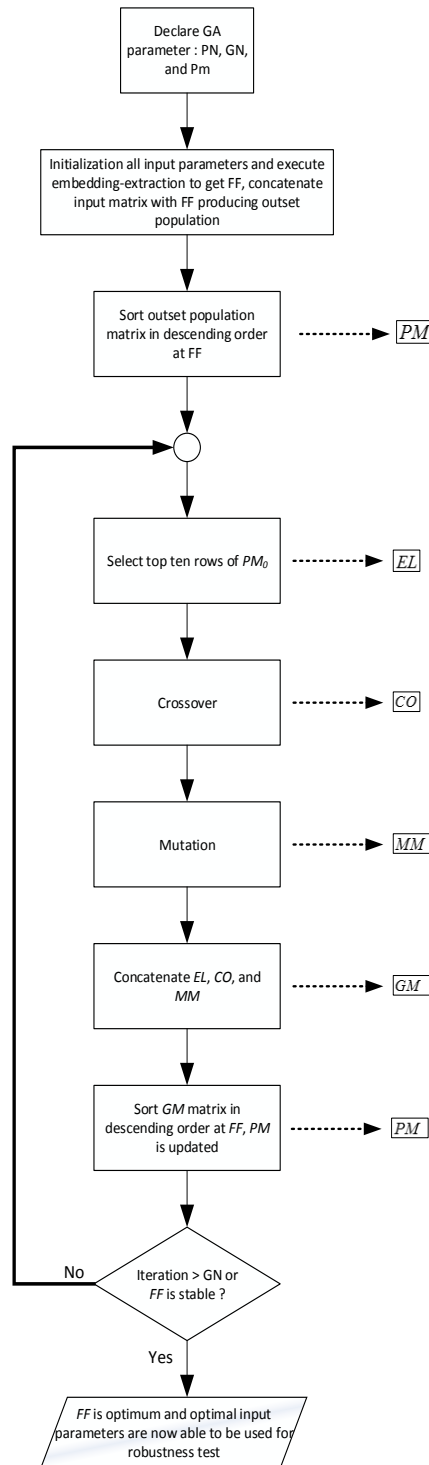


Figure 7. Genetics Algorithm Procedure

- c. Do mutation with mutation probability Pm of elite matrix. There will be 10 iterations to make 10x7 new mutant as new input parameters. Run all new input parameters in embedding and extraction process to get 10 new FF . Then we get mutation matrix MM at size 10x7.
- d. Concatenate horizontally the matrix EL , CO , and MM to get new generation matrix GM .
- e. Sort GM at FF in descending order, and clear matrix EL , CO , and MM . We get new matrix of PM at size 26x7.
- f. Plot top of FF value from PM to get the convergence plot of Genetics Algorithm. We could see at which generation when the graphics is steady state or no change anymore. This means we can stop the iteration of this 4th stage if there is no change any more for very long generation. Usually if there is no change for more than 100 generations, the parameters are already optimum. Thus we get the optimum parameters at top of last PM matrix.

3. Evaluation of Performance

In this evaluation, number of image to be tested as host images are 5 images with free copyright, such as: tulips.png, airplane.png, baboon.png, fruits.png, and peppers.png. The binary logo image inserted into host image consists of 96 bits which is displayed in Figure 8. Pixel “1” shows white pixel, and pixel “0” shows black pixel. In the extraction process, length of watermark is assumed to be known. The outset population number are 20 populations, maximum generation is set to 100, and mutation probability is 50%.

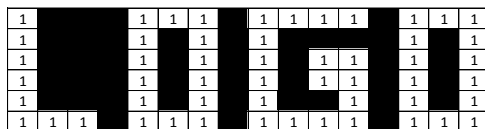


Figure 8. Watermark embedded into host image

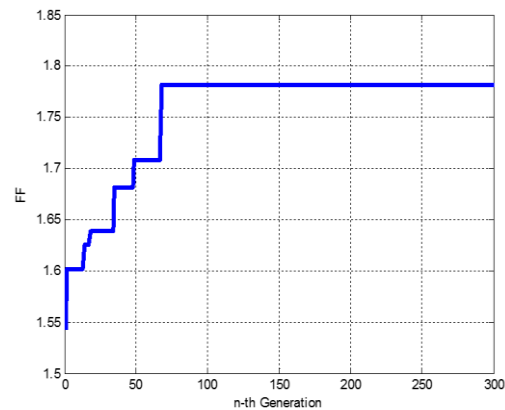


Figure 9. GA Convergence Plot for “airplane.png” as host image

In the first experiment, we run the image watermarking simulation to find the optimum parameters for 5 host image with same watermark as displayed in Figure 8. It is assumed that there is no attack to the watermarked image. And the value of FF only depends on BER and PSNR. It means that the performance we consider are only robustness and invisibility of watermark or watermarked image quality. The simulation result is displayed in Table 3. Figure 9 displays the convergence plot of FF for GA simulation in “airplane.bmp” host image. FF is 2 at its maximum value because only two performance indicator used for GA simulation: robustness (BER) and invisibility of watermarks (PSNR). We see that FF at generation for more than 70 is steady state. The optimum parameter for that FF is displayed in Table 3 at “airplane” row. Optimum color space and layer used for airplane is 3rd layer (layer=3) in YCbCr color space (color=2). Block size is 6 and delta is 4, while POS1=16 and POS2=18. This parameters combination produces result: PSNR=46.91 and BER=0, but payload is 185 bits. In Table 3 we can see that we get optimum parameter when BER=0 or no error at the extracted logo and PSNR>40 dB or the watermark is invisible at all. For achieving that performance the value of

parameters are different depending on the host image. Each host image has unique statistics character.

Table 3. No Attack Parameter Optimization

FF	Delta	Block Size	Color Space	Layer	POS1	POS2	BER	C	PSNR	File
1.71685031	0.01563	3	1	3	3	5	0	740	43.011	Baboon
1.7818967	4	6	2	3	16	18	0	185	46.914	Airplane
1.80761898	2	7	2	3	23	26	0	196	48.457	Fruits
1.71146325	2	7	3	7	23	25	0	460	42.688	Peppers
1.64879785	1	5	3	0	10	13	0	438	38.928	Tulips

Table 4. Robustness Result From Attack for "Fruits.png"

File	Attack	Delta	Block Size	Color Space	Layer	POS1	POS2	BER	C	PSNR	FF
	JPEG Compression 75%							0.448			1.3597
	LPF 3x3							0.219			1.5889
Fruits	Additive Noise 0.001	2	7	2	3	23	26	0.313	196	48.4571	1.3389
	Resizing 50%							0.313			1.4951
	Histogram Equalization							0.073			1.7347

In the second experiments we use parameter from Table 3 for image watermarking attack testbed. The attacked file is only "fruits.png". This simulation should be executed to know how robust the watermark with optimum parameters from GA when there is an attack. The attacks consist of compression, additive noise, low pass filtering, resizing, and histogram equalization. The result is displayed in Table 4. The robustness of the watermarks facing attacks using optimum parameter in no attack GA simulation is very weak. The only one attack which the watermarks still robust is histogram equalization with 7% BER. This case also happens to other file. Thus, in order to have strong robustness facing the attacks, the GA simulation must apply not only embedding-extraction but also embedding-attacking-extraction to find optimum parameter when the attack is applied.



Figure 10. "Fruits.png" as host image (left-side) and watermarked image (right-side) with the optimized parameter as displayed in Table 5

In the third experiments GA will apply to embedding-attacking-extraction for host image "Fruits.png". The attack type for optimizing is JPG compression. And the optimized parameter will be used for other attacking test: LPF, Additive Noise, Resizing and histogram equalization. Figure 11 displays the convergence plot of FF for GA simulation in "Fruits.png" host image with JPG attack. The result of attacking test with this new optimized parameter is displayed in Table 5. The host and watermarked image benchmark for "Fruits.png" are displayed in Figure 10. There are 3 attacks which the watermark is robust. Watermark is still not too robust but still acceptable facing resizing attack at BER 12.5% and additive Gaussian noise at BER 2%.

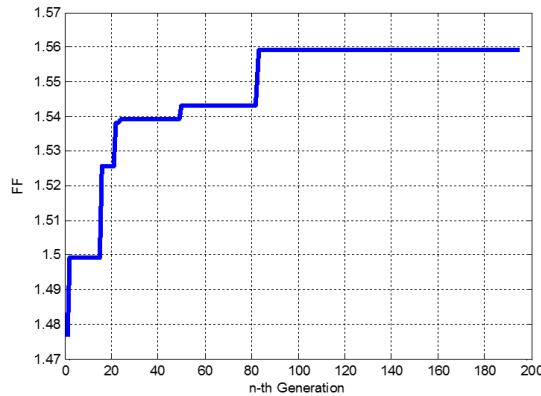


Figure 11. GA Convergence Plot for "Fruits.png" as host image with JPG attack

Table 5. Robustness Result From Attack for "Fruits.png" with optimized parameter from JPG attack

File	Attack	Delta	Block Size	Color Space	Layer	POS1	POS2	BER	C	PSNR	FF
Fruits	JPEG Compression 75%							0			1.5257
	Salt & Pepper 0.01							0			1.5257
	Additive Noise 0.01	28	5	3	2	11	16	0.02	113	31.5395	1.505
	Resizing 50%							0.125			1.4
	Histogram Equalization							0			1.5257

Table 6. Robustness Result From Attack for all image files with optimized parameter from No attack





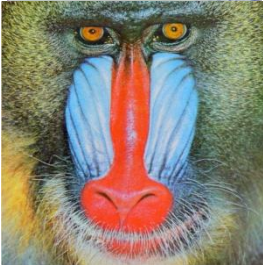



Image	BER				
	JPEG Compression	Salt & Pepper	Additive Noise	Resizing	Histogram Equalization
Tulips	0.5	0.14	0.47	0.4	0.28
Airplane	0.46	0.14	0.38	0.29	0
Baboon	0.47	0.031	0.34	0.41	0.031
Peppers	0.57	0.14	0.46	0.44	0.22
Fruits-	0.45	0.22	0.46	0.31	0.07

Table 7. Robustness Result From Attack for all image files with optimized parameter from JPG attack

Image	BER				
	JPEG Compression	Salt & Pepper	Additive Noise	Resizing	Histogram Equalization
Tulips	0	0.02	0.01	0.18	0
Airplane	0	0	0.02	0.02	0
Baboon	0	0	0.02	0.22	0
Peppers	0.01	0	0.01	0.22	0
Fruits	0	0	0.02	0.125	0

Table 6 shows the attack result when using same optimized parameters from No Attack (Table 3) for all image files. For same parameters from Table 5 we run the other files for attack testing. Table 7 shows the robustness result of attack testing to the image files: tulips.png, airplane.png, baboon.png, and peppers.png. Generally, by the above parameter the watermark is robust eventhough it is embedded in different file, except at the resizing attack. The robustness of resizing attack depends on the statistical character of each image. If the image is rich of the pixel low intensity variation, then the robustness of resizing will be weaker. The visually host and watermarked image is displayed in Table 8. The PSNR in Table 8 is trade off PSNR in order the watermark to be robust facing JPG attack.

Table 8 Host and Watermaked Image Visual with PSNR

Image Filename	Host Image	Watermaked Image	PSNR (dB)
Tulips			30.8934
Airplane			30.9459
Baboon			31.1132
Peppers			31.0832
Fruits			31.5359

2D DCT based image watermarking with vary length of AC coefficient, block size, color space, amplitude of delta and layer of the color space is presented in this paper by optimizing them to reach strong robustness, enough payload and good invisibility. To know those optimal parameter to reach that target is not easy, genetics algorithm (GA) made it simpler to analyze. And also GA can determine any value of the parameter to reach any target we decide in the watermarking design planning. Trade off between contrary parameters of the image watermarking can be easily found by GA. But GA is made not to improve the method, GA is only searching at which value we want for achieving the destination. In many papers, the performance described is not yet optimum. We still can find the value of watermarking parameters to set the optimum performance. Anyway, this paper improves the robustness as described in Table 6 and 7, and the consequence is the lower invisibility based on PSNR formula. But visually there is no difference between host image and watermarked image.

4. Conclusion

GA in 2D DCT based image watermarking plays an important role for determining the exactly optimum parameter value. The robustness, invisibility, and capacity problem in 2D-DCT method is solved by GA. There is a new information which describes in which value and what parameter for achieving strong robustness and good invisibility. Anyway, increasing the robustness will decrease the invisibility. And GA improves the robustness while maintains the invisibility to an acceptable PSNR.

References

- [1] Cox IJ, Kilian J, Leighton FT, Shamoon T. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*. 1997; 6(12): 1673-1687.
- [2] Gunjal BL, Mali DSN. Unseen to Seen with Cryptography, Steganography and Watermarking. *CSI Communications*. 2014; February: 24-26..
- [3] Huang J, Shi YQ, Shi Y. Embedding image watermarks in DC components. *IEEE Transactions on Circuits and Systems for Video Technology*. 2000; 10(6): 974-979.
- [4] Lu CS, Huang SK, Sze CJ, Liao HYM, Huang SK, Lu CS. *A New Watermarking Technique for Multimedia Protection*. In *Multimedia Image and Video Processing, Image Processing Series*. 2000: 1-32.
- [5] Piva AA. DCT-Domain Watermarking System for Copyright Protection of Digital Images. 1998.
- [6] Teng D, Shi R, Zhao X. *DCT image watermarking technique based on the mix of time-domain*. In *Proceedings 2010 IEEE International Conference on Information Theory and Information Security, ICITIS 2010*. 2010: 826-830.
- [7] Al-Hajj A. Combined DWT-DCT Digital Image Watermarking. *Journal of Computer Science*. 2007; 3(9): 740-746.
- [8] Gunjal BL, Mali SN. Secured Color Image Watermarking Technique in DWT-DCT Domain. *International Journal of Computer Science, Engineering and Information Technology (IJCSSEIT)*. 2011; 1(3): 36-44.
- [9] Akter A, Ullah MA. *Digital Image Watermarking Based on DWT-DCT: Evaluate for a New Embedding Algorithm*. In *International Conference on Informatics, Electronics & Vision*. 2014.
- [10] Jose S, Roy RC, Shashidharan S. Robust Image Watermarking based on DCT-DWT- SVD Method. *International Journal of Computer Applications*. 2012; 58(21): 12-16.
- [11] Bansal N, Bansal A, Deolia VK, Pathak P. *Comparative Analysis of LSB, DCT and DWT for Digital Watermarking*. In *International Conference on Computing for Sustainable Global Development*. Mathura, India. 2015: 40-45.
- [12] Li J, Cao Q. DSDWA: A DCT-based Spatial Domain Digital Watermarking Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(1): 693-702.
- [13] Yun K. In the Network Communication an Improved Algorithm of Image Watermarking based on DWT. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(11): 6304-6308.
- [14] Bandyopadhyay SK, Paul TU, Raychoudhury A. Genetic Algorithm Based Substitution Technique of Image Steganography. *Journal of Global Research in Computer Science*. 2010; 1(5): 62-69.
- [15] Khamrui A, Mandal JK. *A Genetic Algorithm based Steganography Using Discrete Cosine Transformation (GASDCT)*. *Procedia Technology*. 2013; 10: 105-111.
- [16] Prathibha E, Yellampalli S, Manjunath PA. Design and Implementation of Color Conversion Module RGB to YCbCr and Vice Versa. *International Journal of Computer Science Issues*. 2011; 1(1): 13-18.