Intelligent Watermarking Scheme for image Authentication and Recovery

Rafi Ullah
Department of Computer Science and Information, College of Science at Al-Zulfi, Majmaah University, KSA

Hani Ali Alquhayz
Department of Computer Science and Information, College of Science at Al-Zulfi, Majmaah University, KSA

***Abstract*—Recently, researchers have proposed semi-fragile watermarking techniques with the additional capability of image recovery. However, these approaches have certain limitations with respect to capacity, imperceptibility, and robustness. In this paper, we are proposing two independent watermarks, one for image recovery and the other for authentication. The first watermark (image digest), a highly compressed version of the original image itself, is used to recover the distorted image. Unlike the traditional quantisation matrix, genetic programming based matrices are used for compression purposes. These matrices are based on the local characteristics of the original image. Furthermore, a second watermark, which is a pseudo-random binary matrix, is generated to authenticate the host image precisely. Experimental results show that the semi-fragility of the watermarks makes the proposed scheme tolerant of JPEG lossy compression and it locates the tampered regions accurately.**

***Keywords*—Watermarking; Genetic Programming (GP); Authentication; Quantisation; and Recovery**

I. Introduction

The internet has brought substantial benefits, one of which is the distribution of multimedia content; images, video, audio, text, graphics etc. However, achievements regarding effective development, distribution and storage of multimedia content have also brought concerns about copyright protection, protection from tampering and authentication. One of the prospective solutions to these problems is to watermark the multimedia content [1]. The three different watermarking approaches: (1) fragile, (2) semi-fragile, and (3) robust are applied for securing the digital content.

In a watermarking system, there is an intrinsic relationship between three of its contradicting attributes: (1) robustness, (2) imperceptibility, and (3) capacity. Imperceptibility means that the watermarked data should be perceptually equivalent to the original data. On the other hand, robustness means that the watermark should be undetectable, unless that damages the usefulness of the original data [2]. Capacity refers to the maximum length of the message that can be hidden in the host image. Similarly, the security attribute of a watermarking system has gained appreciable importance. The field of watermarking has great potential in authentication-based applications. The basic requirements of authenticating digital content are: imperceptibility, fragility, security, and efficient computation. A watermarking technique is proposed in [3], where two watermarks are embedded in LL3, HL2 and LH2 sub-bands of the wavelet transform. This scheme accurately authenticates images but at the cost of imperceptibility. In our current work, we increase the imperceptibility of the watermark using the Genetic Programming (GP) based exploitation of the Human Visual System (HVS). Intelligent approaches have been used for enhancing imperceptibility and robustness properties of robust watermarking approaches [4, 5]. However, in authentication related applications, they have rarely been exploited.

Besides authentication and copyright protection of the digital content, the researchers are proposing the techniques that can recover the image as well. These techniques are quite useful for medical images, sequences as medical data are more sensitive and they need to be recovered after manipulation. For example, the rehashing model is proposed in [6] to authenticate and recover both the altered colour and gray-scale images. In addition, this model is able to reduce the failure rate of tamper detection. Wavelet based dual watermarking techniques have been applied to authenticate and recover the image [7]. The authors are using two watermarks: (1) a semi-fragile watermark for authentication, and (2) a robust watermark for recovery purposes. Both watermarks are embedded in the wavelet domain and are able to identify the tampering up to 20% of the original image. By using a quick response (QR) code, a subsampling-based image authentication and recovery has been proposed in [8]. QR is the trademark and is always scanned to acquire the data. The properties of QR have been used to detect the tampered regions and recover the altered images. A self-recovery watermarking method has been proposed for authentication and error concealment [9]. This method can be used for images and videos. The scheme is based on watermarking and half-toning techniques. A quantisation index modulation (QIM) watermarking algorithm is modified to increase and improve the capacity and an inverse half-toning method is used to improve the quality of the recovered area(s). A DCT based effective self-embedding algorithm has been designed for authentication and localisation along with recovery in [10]. In this algorithm, for each 2×2 block, two authentication bits and ten recovery bits are generated from the five most significant bits. Authentication bits are embedded in the block itself while recovery bits are embedded in the corresponding mapped block. This scheme is also effective for high probability tamper detection because the authenticity blocks are based on two levels of hierarchical tamper detection mechanisms.

The rest of the paper is summarised as: Section 2 explains the proposed method and GP module for digest generation. The watermarks generation and embedding are explained in Section 3. In Section 4, we analyse both of the watermarks for authentication, tamper proofing and recovery of the altered image. Experimental results are presented in Section 5. In Section 6, the paper concludes and provides some future directions.

II. Proposed Method

We use both the Discrete Cosine Transform (DCT) and Integer Wavelet Transform (IWT) domains to generate and embed the watermarks in an image. Parameterised Integer Wavelet Transform has been employed using the lifting scheme, which is the fast approach of Discrete Wavelet Transform (DWT) [11]. We use two watermarks; one is called image digest, while the other is a binary watermark. These two watermarks are embedded in different sub-bands of the IWT. We compress the original image to generate the image digest using the DCT transform like JPEG compression. However, while generating the image digest, instead of using the standard quantisation matrix [12], we use the Genetic Programming (GP) to develop quantisation matrices according to the local characteristics of the host image. GP automatically decides the 64 quanta for an 8×8 DCT block according to the distortion criteria. We use Peak Signal to Noise Ratio (PSNR) as a distortion measure of the watermarked image.

$$PSNR = 20log_{10}\left[\frac{255^2}{\frac{1}{RS}\sum_{i,j}\left(x(i,j)-y(i,j)\right)^2}\right] \qquad (1)$$

where, $1 \leq$ i $\leq$ R and $1 \leq$ j $\leq$ S; R and S represent the size of the image. The image is decomposed up to three levels and the first watermark i.e. image digest is embedded in the LH2, and HL2 sub-bands. The second watermark, i.e. the binary watermark, is embedded in the LL3 sub-band. Our current work is an extension of the technique proposed in [3]. The extension is brought about by enhancing the imperceptibility of the watermark using GP. The Proposed approach develops Genetic Quantisation Matrices (GQMs) as per the watermarking application. The system learns from observation, continuously improves its performance, and hence provides more efficient and accurate results. A test phase is used to evaluate the generalisation of the developed GQMs [4].

A. *Scaling Image Digest using GP*

Watson perceptual models have been used for the JPEG compression [12, 13]. Watson's perceptual model, although good enough to give us imperceptible alterations, is not an optimum one. This is because some effects like the spatial masking in the frequency domain are ignored and many of the constants are set empirically. Additionally, the quantisation matrix used in [3, 4] for scaling image digest is just based on the frequency sensitivity attribute of HVS. It does not exploit the luminance sensitivity or contrast masking attributes of HVS. To overcome this problem, we develop the quantisation matrices using GP. The strengths of the quanta of the GQMs are set according to local frequency content in an image. Thus, instead of using a fixed quantisation matrix, we use an adaptive quantisation matrix. The quanta of the GQMs and the imperceptibility of the watermark are inversely proportional and consequently demand a delicate balance as per watermarking application.

B. *GP Module*

GP is a machine learning technique based on natural selection and genetics. A data structure, such as a tree is used to represent an individual solution. GP is based on the stochastic method, in which randomness plays an important role in searching and learning [14]. Initially, the random population for such solutions is created and then every solution is evaluated using a fitness function according to the application. The best individuals are retained and the rest are deleted and replaced by the offspring of the best individuals. The retained offspring make a new generation. Some offspring may have a higher score than their parents in the previous generation. The process is repeated until the termination criterion is satisfied. Figure 1 shows the block diagram for developing GQMs.

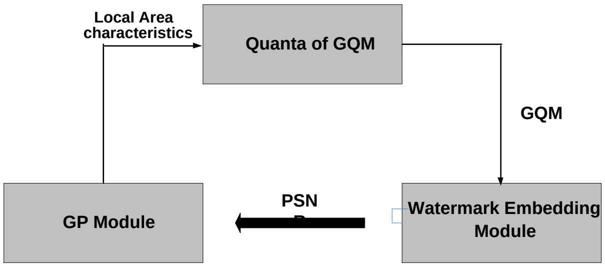

Fig. 1. Basic architecture of developing GQM

Suitable functions, terminals and fitness criteria are defined that represent the possible solutions in the form of a complex numerical function. Different functions of the proposed GP module are as follows:

1) GP Function Set: A GP function set is the collection of mathematical functions available in a GP module. In our simulations, we are using four basic binary functions (+, -, *, /) along with a log and an exponent.

2) Fitness Function: Too grade each individual of the population; a fitness function has to be used. The performance of individuals in the GP population is assessed by the imperceptibility (PSNR) as a fitness function. Each individual of the GP population is scored using $fitness = PSNR_{o,w}$, where 'o' is the original image and 'w' is the watermarked image. This function provides the feedback to the GP module representing the fitness of the individual. A higher individual fitness score indicates a higher performance.

3) Population initialisation: Like other evolutionary algorithms, in GP the individuals in the initial population are randomly generated. Most common methods for initialisation of the population are the $full\, and\, grow$ method and $ramped\ half-and-half$ method. In both methods, the generated initial individual does not exceed the pre-defined maximum tree depth [15]. We are using the ramped half-and-half method for creating the initial population.

4) Termination criteria: The simulation is terminated when one of the following is encountered. The fitness score exceeds i.e. fitness > 50 or the fitness score repeats. A number of generations reach the pre-defined maximum number of generations.

5) GP operators: In the proposed scheme, crossover, mutation and replication GP operators are used for producing the new generation. Crossover creates the offspring by exchanging the genetic material of two individual parents. It tries to mimic recombination and reproduction. Crossover helps in converging on an optimal/near optimal solution. In mutation, the genome is changed in a minor way for the next generation. In replication, the individual is copied to the next generation. In the GP run, we have used a variable ratio of these operators with a high ratio of crossover. All of the necessary settings of the GP module are given in Table 1.

TABLE I. GP PARAMETERS SETTING

Objectives	**To evolve optimum result**
Function Set	+, -, *, /, log, exponent
Operands	Wat_St (Watson's standard matrix), DCT_AC (AC component of DCT matrix), constants
Fitness	PSNR
Expected offspring	rank89
Selection	Generational
Population and Generations	120 and 50 respectively
Initial population	Ramped half-and-half
Termination criteria	The fitness score exceeds or repeats OR Number of generations reaches the pre-defined maximum number of generations
Sampling	Tournament
Survival mechanism	Keep the best
GP operators	Crossover, mutation, replication

III. WATERMARKS GENERATION AND EMBEDDING

Two watermarks, the image digest (w_1) and the binary watermark (w_2), are generated and embedded in the wavelet sub-bands. We will discuss the generation of these watermarks individually in Section 3.1 and Section 3.2. Before embedding these watermarks, we generate w_1 and w_2.

A. Generation of w_1 (Recovery Watermark)

The original image of size $\mathrm{N} \times \mathrm{N}$ is decomposed up to level one. The approximation (LL1) sub-band is selected for the image digest i.e. w_1. The full-frame DCT is applied on LL1 to get the DCT transformed image. The DCT coefficients are then quantised using the GQMs. A vector form is generated from the DCT values through zigzag scanning. First $\mathrm{S} = \mathrm{N}^2/32$ coefficients are selected. A key-based scaling is applied to the sequence/vector S and is further scaled-down for reducing the strength of the watermark. Equation 2 is used to scale-down the sequence.

$$S_{Scaled}(i) = S(i) \cdot \alpha \cdot ln(i + 2 + r(i)) \qquad (2)$$

where α is the strength factor depending upon the image quality and r is the shift parameter ranging from -0.5 to 0.5. The DC component is discarded because of its high energy. The embedding area for w_1 is LH2 and HL2, which is the $\mathrm{N}^2/8$ sizes so $\mathrm{S}_{\mathrm{Scaled}}$ should be quadrupled as given in Equation 3.

$$w_1 = C_1, C_2, \ldots, C_S, C_1, C_2, \ldots, C_S, C_1, C_2, \ldots, C_S, C_1, C_2, \ldots, C_S \qquad (3)$$

B. Generation of w_2 (Authentication Watermark)

Let W be the binary image of size X × Y and $\mathrm{P}_{\mathrm{rand}}$ is the Pseudo Random binary matrix of the same size generated by using the secret key, then the second watermark, w_2 is generated in Equation 4.

$$w_2 = W \oplus P_{rand} \qquad (4)$$

where, $\oplus$ is the exclusive OR operator.

1) Embedding Process: After completion of both the watermarks generation, we embed these watermarks in different sub-bands. The embedding process is shown in Figure 2. The original image is decomposed up to level three. The sub-bands selection for watermark embedding is based on the application. If the approximation of the wavelet-transformed image is used for embedding, then the robustness will be enhanced with the cost of tamper localisation. On the other hand, if the watermark is embedded in the details of the wavelet-transformed image, then the accuracy in localising the tampered regions will be increased, but at the cost of robustness. Before embedding, random permutation keeps the watermark bits safe [16].

The LL3, LH2 and HL2, are selected for embedding both the watermarks. We simply replace the LH2 and HL2 sub-bands by the first watermark w_1. Before embedding the first watermark, we scramble it by using the secret key to enhancing its security. The block diagram for the embedding process of the proposed scheme is given in Figure 2.

The second watermark $\mathrm{w}_{2,}$ is embedded in the LL3 sub-bands by using the following procedure [17].

Let "LSFB (a)" denote the least significant five bits of 'a' and "LSFB (a, b)" represent the substitution of"b" for the five least significant bits of 'a'. We select two choices, "11000" and "01000" representing "1" and "0" respectively. These are the best choices selected from the distance diagram based on the quality of the watermarked image. Modifying the coefficients by using other choices, as given in Figure 3, may cause a severe effect on the imperceptibility. The distance diagram is shown in Figure 3.

The second watermark, $\mathrm{w}_{2,}$ is embedded in the LL3 sub-bands by using the following procedure [17].

Let "LSFB (a)" denote the least significant five bits of 'a' and "LSFB (a, b)" represent the substitution of"b" for the five least significant bits of 'a'. We select two choices, "11000" and "01000" representing "1" and "0" respectively. These are the best choices selected from the distance diagram based on the quality of the watermarked image. Modifying the coefficients by using other choices, as given in Figure 2, may cause a severe effect on the imperceptibility. The distance diagram is shown in Figure 2.

By keeping the performance of imperceptibility and robustness in mind the following formulae are used to embed the second watermark in LL3:

When $w_2(i,j) = 0$ then equation 5 is adopted.

$$f'(i,j) = \begin{cases} \mathrm{LSFB}(f(i,j) - 01000, 11000), \text{if } \mathrm{LSFB}(f(i,j)) \leq 01000 \\ \mathrm{LSFB}(f(i,j), 11000), \quad \text{otherwise} \end{cases} \quad (5)$$

where, $f(i,j)$ is the wavelet coefficient in the LL3 sub-band before embedding, and $f'(i,j)$ is the wavelet coefficient in the LL3 sub-band after embedding

When, $w_2(i,j) = 1$ then equation 6 is adopted.

$$f'(i,j) = \begin{cases} \mathrm{LSFB}(f(i,j) + 10000, 01000), \text{if } \mathrm{LSFB}(f(i,j)) \leq 11000 \\ \mathrm{LSFB}(f(i,j), 01000), \quad \text{otherwise} \end{cases} \quad (6)$$

By simply replacing the two choices, the amplitude of the coefficients changes from -23 to 24, while applying the above conditional substitutions; it may change from -15 to 16 [17]. After embedding both the watermarks, applying the inverse wavelet transform (Inverse integer wavelet transform) gets the watermarked image.

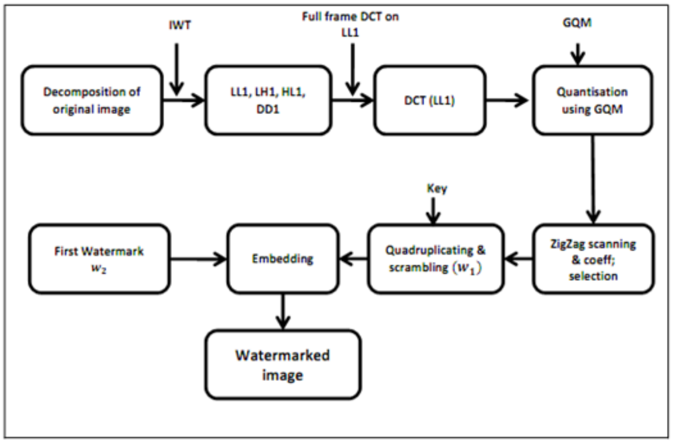

Fig. 2. Embedding Process

IV. Watermarks Extraction and Analysis

A. Authentication Process

On the authentication side, the integrity of both the watermarks w_1 and w_2 is verified. For integrity verification, the authentication watermark is generated in the same way as discussed in Section 3.2. Now, we extract the authentication watermark from the watermarked image and compare it with the generated one. If they match then the image is authentic otherwise, it has been tampered with. The authentication process is shown in Figure 4. Decompose the watermarked image and select the sub-bands, where the watermarks were embedded, i.e. LL3, LH2 and HL2. From LH2 and HL2, extraction of w_1 is the reverse of the embedding process.

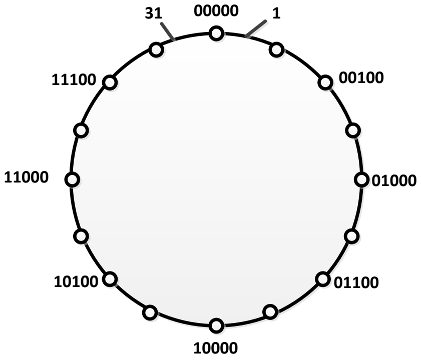

Fig. 3. Distance Diagram

The data are inversely scrambled using the same key and the average is taken from the four copies of the selected data to get the S=$N^2/32$ number of coefficients. These coefficients are then replaced in their correct positions by means of anti-zigzag scanning. All the missing elements are set to zero and the DC component is replaced by 128. The resultant values are weighed by using the same GQMs. The GQMs are generated in the same way by using the same best-evolved GP expression. The inverse DCT is applied to obtain the approximation of the original image of size N/2 × N/2.

Let $\mathrm{LSFB}_{5th}(a)$ denote the 5th least significant bit of a then:

$$W'(i,j) = \begin{cases} 1, & \text{if } \mathrm{LSFB}_{5th}(f'(i,j)) = 0 \\ 0, & \text{if } \mathrm{LSFB}_{5th}(f'(i,j)) = 1 \end{cases} \quad (7)$$

where $(1 \leq i \leq X, 1 \leq j \leq Y)$

As in the embedding phase, the watermark has been pre-processed. Thus, on the verification side, the extracted bits are again processed by using the same sequence. This is done by using equation 8.

$$w_2'(i,j) = W'(i,j) \oplus P_{rand} \quad (8)$$

where, P_{rand} is the same pseudo random matrix as used in Section 3.2.

B. Tamper Proofing

Differentiate the original binary watermark and extracted binary watermark using Equation 9.

$$\text{Difference} = |w_2 - w_2'| \quad (9)$$

Black pixel i.e. “0” corresponds to the correctness in Difference image while white pixel, i.e. “1”, corresponds to the error pixel. Therefore, we can accurately locate the tampered areas and differentiate the malicious and accidental attacks. Dense and sparse pixels are defined as: an error pixel in Difference image is dense pixel if one of its eight neighbour pixels is also an error pixel; otherwise, it is a sparse pixel. These erroneous pixels can be detected by using the following parameters.

$$\text{Dense Area (DA)} = \text{Number of dense pixels in an appoximation sub} - \text{band}$$

$$\text{Sparse Area (SA)} = \text{Number of sparse pixels in an appoximation sub} - \text{band}$$

$$\text{Total Area (TA)} = \text{DA} + \text{SA}$$

$$\Delta = \frac{\text{DA}}{\text{SA}}$$

$$\rho = \frac{\text{TA}}{\text{LL1}}$$

$$\text{If } \rho = 0, \text{then watermarked image has not been tampered with.}$$

$$\text{If } \rho > 0, \text{and } \Delta < \beta, \text{then the image has been tampered with accidently, where } 0.5 \leq \beta \leq 1.$$

$$\text{If } \Delta \geq \beta, \text{then the image has been tampered with maliciously.}$$

The above parameters depict that if the Difference image has sparse pixels then the watermarked image has been attacked accidently i.e. JPEG Compression, file format change etc. Otherwise, in the case of dense pixels, the image has been attacked maliciously i.e. cut/copy-paste.

C. Image Recovery

The image can be recovered in two ways: the first is to recover the tampered areas and the second is to recover the whole image, whether the watermarked image has been tampered with or not. Our proposed scheme employs the second approach in which we embed the compressed version of the host document itself and such an approach is usually referred to as a self-recovery technique [18]. The original image is decomposed and then its low level is highly compressed like a JPEG compression, using GP based quantisation matrices. On the authentication/verification side, the reverse procedure of a digest generation process is applied to get the recovered image. As we will see in the experimental results, we can recover the image after manipulations, either malicious (cut/copy-paste) or accidental (Lossy Compression). The degradation of the recovered image increases while increasing the strength of the manipulation/compression. In the case of lossy compression, the recovered image is acceptable up to a 70% compression factor for which the detail is shown in the figure later below.

V. Experimental Results

We tested our scheme on a LENA image in bmp format of size 512×512. MATLAB environment was used for our experiments. GP-Lab was used to carry out the GP simulations [19, 20]. PSNR values of the watermarked images were up to 44db, which is quite good as compared to [3]. Figure 5 shows the original image of Lena and the watermarked image with PSNR = 43.7db. As we were embedding two watermarks, the imperceptibility increased. We used the printed name of the first two authors as a binary watermark. The proposed approach effectively authenticated the data. Due to the second (binary) watermark, it localized the manipulation accurately. Figure 6 shows the authenticity of our scheme. The watermarked image was tampered with invisibly on the hairs of Lena. As the system is semi-fragile, it survived the JPEG lossy compression to some extent. Figure 7 shows the recovered images after JPEG compression using different quality factors. When the quality factor was 70 or above, the difference image contained the sparse pixels and below 70, the number of dense pixels increased which shows that the image was tampered with maliciously. The recovered image and the difference images were not affected while using the quality factor = 100.

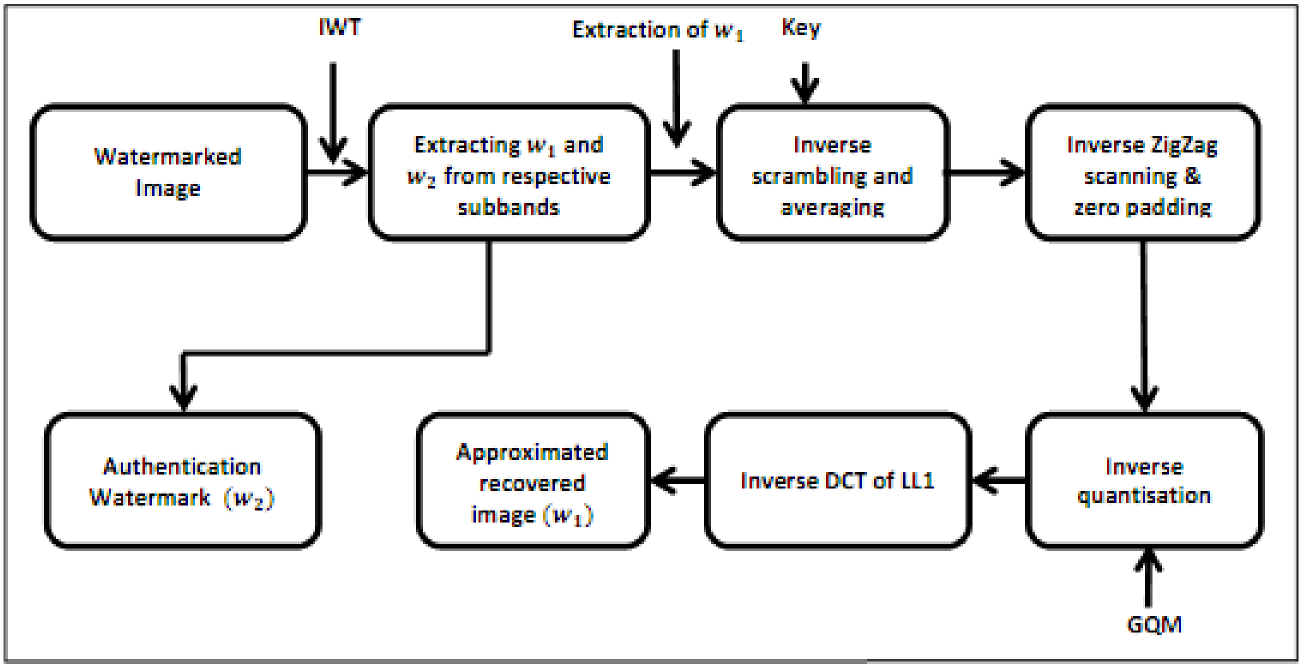
IWT
Extraction of w_1
Key
Watermarked Image
Extracting w_1 and w_2 from respective subbands
Inverse scrambling and averaging
Inverse ZigZag scanning & zero padding
Authentication Watermark (w_2)
Approximated recovered image (w_1)
Inverse DCT of LL1
Inverse quantisation
GQM

Fig. 4. Extraction Process

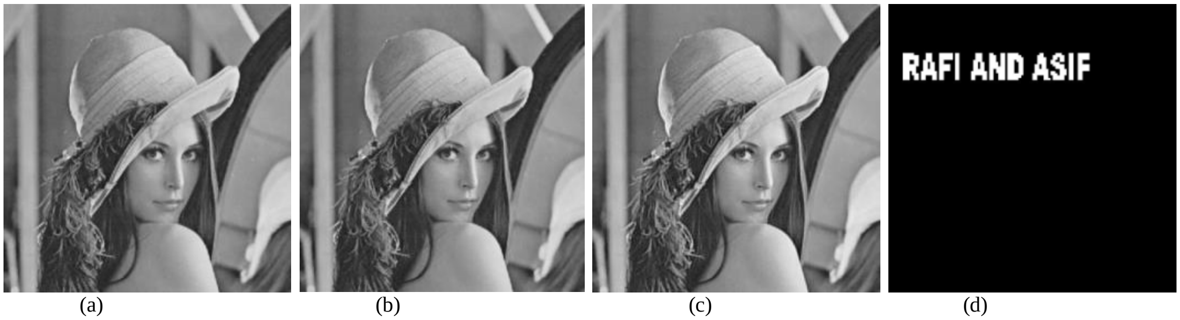

Fig. 5. (a) Original image (b) Watermarked image (c) Recovered image (d) Extracted binary watermark

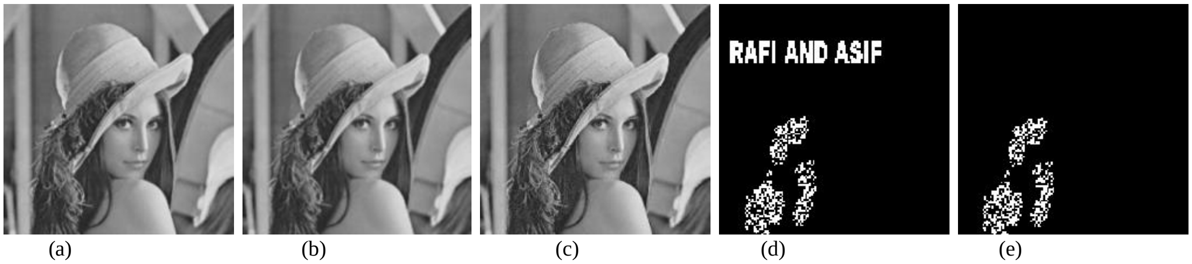

Fig. 6. (a) Original image (b) Watermarked image (c) Tampered with maliciously on hairs; invisible tampering (d) Tamper detection on extracted binary watermark (e) Difference in original and extracted binary watermarks

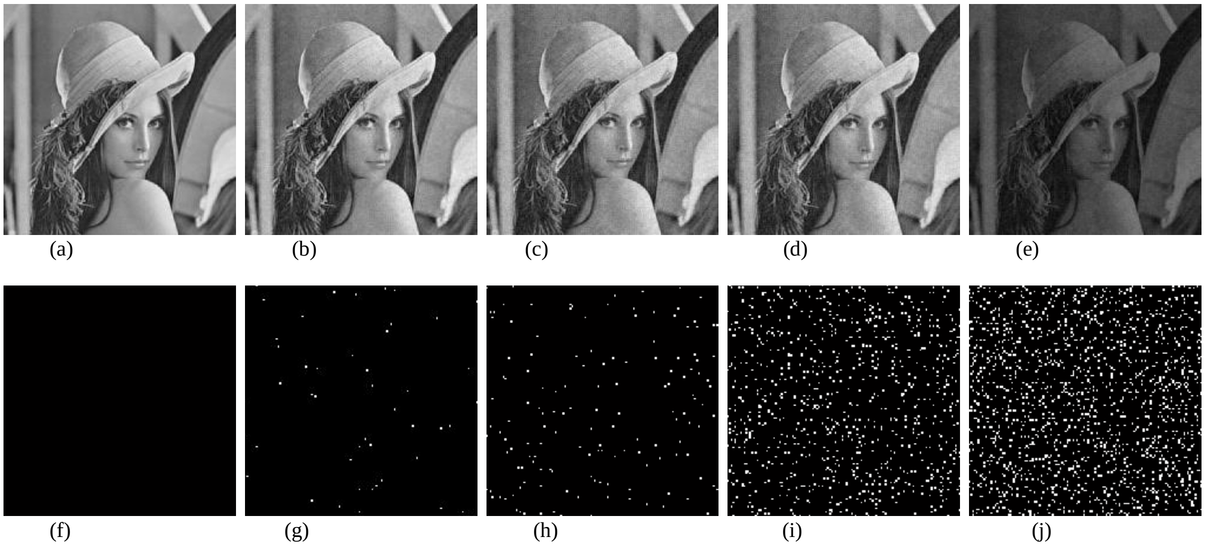

Fig. 7. The first row contains the recovered images (a ~ e) and the second row shows the differences in extracted binary watermarks after applying JPEG compression of the quality factors (f ~ j), 100, 90, 80, 70 and 60, respectively.

Figure 8 shows the number of dense and sparse pixels for Lena, Cameraman and Baboon images. These images have different textures, especially the Baboon image which is a highly textured image compared to the other two images.

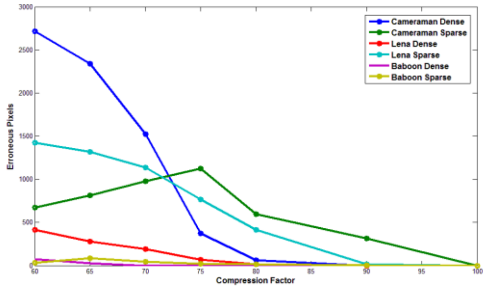

Fig. 8. Erroneous (Dense and Sparse) pixels versus compression factors

Table 2 shows the comparison of PSNR for our GQMs and fixed quantisation matrix (FQM). The performance of GQMs used in our proposed approach is better compared to the FQM used in other related research. The Lena image is used as a test image. Other approaches are using FQM.

TABLE II. PSNR OF THE PROPOSED APPROACH

Features	**Proposed Scheme**
PSNR	41db~43db
Type of Quantisation matrix	Application Dependent

Table 3 shows the performance comparison between the proposed method with previous methods [6-10]. The comparison is made with respect to imperceptibility (PSNR), robustness, recovery, recovery after compression and recovery after malicious manipulations. We use gray-scale test images (Lena) for comparison in our experimental results. In the proposed approach, the genetic/dynamic quantisation table has been used for compressing the original image to generate an image digest. The compression of the image is based on local features of the image and the result, in terms of imperceptibility, varies accordingly. The performance given in Table 2 is based on the images having normal textured regions. Increase in the textures in the input images will increase the performance of our algorithm as we are using genetic quantisation for compressing the image based on local features of the image.

TABLE III. PERFORMANCE COMPARISON OF OUR PROPOSED APPROACH WITH [6 – 10].

Parameters	**Ref [6]**	**Ref [7]**	**Ref [8]**	**Ref [9]**	**Ref [10]**	**Proposed Scheme**	**Supporting Results**
Average PSNR	38dB	38dB-41dB	40dB but if Quanta is 12 then PSNR is below 37dB	41db~43db	37dB-38dB	41db~43db	Table 1
Recovery	Yes (Self-Embedding)	Yes (Self-Embedding)	Yes (Self-recovery) using FQM	Yes (Self-recovery) using FQM	Yes	Yes (Self-Embedding using GQM)	Section 7
Robustness/Fragility	Semi-Fragile	Semi-Fragile	Semi-Fragile	Sub-Sampling based	Fragile	Semi-Fragile	Section 3
Recovery after Compression	NO	YES	YES (After compression, the restored image is highly degraded)	NO	NO	Can Recover after 70% QF	Section 8 (Para 1)
Recovery after Malicious Manipulations	Even after 5% tampering, the region is visible	Can recover after 5% - 25% Tampering	Can recover after malicious manipulations	Can recover after malicious manipulations	Can recover but visible for grayscale images	Can Recover and determine the strength of tampering by using sparse and dense error pixels	Section 6

An exemplary numerical expression in prefix form, developed by GP, is given as:

fitness='plus(cos(mylog(mylog(times(DCT_AC,Wat_st)))),times(kozadivide(Wat_st,Wat_st),plus(plus(times(DCT_AC,DCT_AC),mylog(0.84729)),kozadivide(DCT_AC,plus(0.99372,DCT_AC)))))'

VI. CONCLUSION AND FUTURE DIRECTIONS

The proposed authentication strategy using GP has successfully improved the imperceptibility of the watermark. As compared to the approach proposed in [3], the PSNR is improved from 40db to 44db. Our scheme is able to maintain security and accurate authenticity without sacrificing imperceptibility. The scheme is secure by using two secret keys: one is used in pre-processing the binary watermark and the other one is used in scrambling before embedding the image digest. Our scheme tolerates the JPEG lossy compression with a quality factor as low as 70%. The recovered image is still readable/recognisable while using the quality factor = 60. If the GP evolved expressions are not made public, the security of the proposed system would be further enhanced as it would be extremely difficult for an attacker to know the exact watermarking strength for each selected coefficient.

The proposed approach can be used for colour image authentication as well. All of the RGB (Red, Green and Blue) channels can be used for generating image digest, but this may affect the visual perception of the watermark. If one of the RGB channels is considered for image digest and correlated to the considered channel with others before embedding, then this could be an interesting future work.

REFERENCES

[1] N, Ishihara, A.B.E. Koki, “A semi fragile watermarking scheme using weighted vote with sieve and emphasis for image authentication”, IEICE Transaction Fundamentals, vol. E90(5), pp. 1045-1054, 2007.

[2] A. Khan, A. M. Mirza, “Genetic perceptual shaping utilizing cover image and conceivable attack information during watermark embedding”, Information Fusion, Elsevier, vol. 8(4), pp. 354-365, 2007.

[3] R. Chamlawi, A. Khan, and A. Idris, “Wavelet based image authentication and recovery”, Journal of Computer Science and Technology, Springer, vol. 22(6), pp. 795-804, 2007.

[4] A. Khan, “Intelligent perceptual shaping of a digital watermark”, PhD dissertation, Faculty of Computer Science, GIK Institute of Engineering Sciences and Technology, Pakistan, 2006.

[5] A. Khan, S. F. Tahir, A. Majid, and T-S. Choi, “Machine learning based adaptive watermark decoding in view of an anticipated attack”, Pattern Recognition, Elsevier Science, vol. 41, pp. 2594-2610, 2008.

[6] W. L. Lyu, C-C. Chang, F. Wang, “Image authentication and self-recovery scheme based on the rehashing model”, Journal of Information Hiding and Multimedia Signal Processing, vol. 7(3), pp. 460-474, 2016.

[7] P. G. Freitas, R. Rigoni, M. C. Q. Farias, “Secure self-recovery watermarking scheme for error concealment and tampering detection”, Brazilian Computer Society, Springer, vol. 22(5), pp. 01-13, 2016.

[8] R. O. Preda, I. Marcu, A. Ciobanu, Image authentication and recovery using wavelet-based dual watermarking. UPB Scientific Bulletin (C), 77(4) (2015) 199-212.

[9] W. W. Chuan, "Subsampling-based image tamper detection and recovery using quick response code", International Journal of Security and Its Applications, vol. 9(7), pp. 201-216, 2015.

[10] D. Singh and S. K. Singh, "Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability", Journal of Visual Communication and image representation, Elsevier, vol. 38, pp. 775-789, 2016.

[11] W. Xiaoyun, J. Hu, Z. Gu, and J. Huang, "A secure semi-fragile watermarking for image authentication based on integer wavelet transform with parameters" In Proc. of Australian Information Security Workshop, New Castle, Australia, 2005, pp. 75-80.

[12] A. B. Watson, "Visual optimization of DCT quantization matrices for individual images", In Proceedings of AIAA Computing in Aerospace 9, San Diego, CA, 1993, pp. 286–291.

[13] A. B. Watson, G. Y. Yang, and J. A. Solomo, "Villasenor J. Visibility of wavelet quantization noise", IEEE Transactions on Image Processing, vol. 6(8), pp. 1164–1175, 1997.

[14] R. O. Duda, P.E. Hart, D. G. Stork, "Pattern Classification", 2nd Edition, New York: John Wiley & Sons, Inc. 2001.

[15] J. R. Koza, "Genetic programming: on the programming of computers by means of natural selection Cambridge", USA: MIT Press. 1992.

[16] R. Chamlawi, A. Khan, "Digital image authentication and recovery: Employing integer transform based information embedding and extraction", Information Sciences, Elsevier Sciences, vol. 180(24), pp. 4909-4928, 2010.

[17] H. Liu, J. Liu, and J. Huang, "A robust DWT based blind data hiding algorithm", In Proceedings of IEEE on circuits and systems, Phoenix Scottsdale's, USA, 2002, pp. 672-675.

[18] A. Piva, F. Bartolini, and R. Caldelli, "Self-recovery authentication of images in the DWT domain", International Journal of Image and Graphics, vol. 5(1), pp. 149-165, 2005.

[19] S. Silva, J. Almeida, "Dynamic maximum tree depth - a simple technique for avoiding bloat in tree-based GP", In Lecture Notes in Computer Science, Proceedings on Genetic Evolution. Computation (GECCO-2003), Springer. 2003, pp. 1776–1787.

[20] Silva S, "GPLAB - a Genetic Programming toolbox for MATLAB", Version 2015.