An Improved Watermarking Algorithm using Variable Block Image Features[1]

I. Dagher* and P. Hanna

University of Balamand, POBox 100 Elkoura, Lebanon

**e-mail: dagheri@balamand.edu.lb*

Abstract– The ease in digital imaging has led to a decrease in image fidelity where illegal reproduction of multimedia information has become difficult to detect. The most challenging problem is to protect image copyright against illegal copies. Therefore a watermark detection process is required to verify the owner of the image. This paper proposes a blind algorithm to extract the watermark. This algorithm is robust against noise and geometric attacks for grayscale images. Robust feature points are detected using the Harris Detector. In the embedding stage, sequences are placed into regions located around feature points. The sequences used are PN, Gold and decimal. In the extraction process, image features are re-allocated using the same detector. Each feature point is used as a center of an N × N region. This region is moved horizontally and vertically within its neighboring pixels. Each move should be registered in a matrix as a correlation value of this region with the initial sequence. This procedure is repeated for all feature points till we find all the watermarked regions. The maximum correlation obtained will determine the center of the watermarked region. The proposed algorithm is robust against a wide variety of tests and is compared to other schemes.

*Keywords: w*atermarking, variable block, Harris detector, PN sequence, GOLD sequence, Decimal Sequence

DOI: 10.1134/S105466181702002X

1. INTRODUCTION

Today, content owners are eagerly seeking technologies that promise to protect their rights and secure their content from piracy and unauthorized usage. The two main kinds [2] of watermark embedding are: **Transform-based** embedding and **Feature-Based** embedding.

It should be noted that our paper is based on Feature-Based embedding with different **pseudo random sequences**. In [7] Code division multiple access technique was used with sequence spread spectrum modulation to ensure robustness due to redundancy. In the receiver side, the image is normalized, and the same **PN** sequence is regenerated. Finally the message is decoded from the correlation values.

Transform-based embedding:

In [1] digital watermarking was inserted in the **DCT** domain. Literature Review of inserting the watermark in the **wavelet** transform is shown in [3]. In [6], the image is transformed in the **Fourier** domain. Watermark is added to the log polar Fourier transform. At the destination, inverse Fourier Transform was used to obtain the watermarked image. In [10], the watermark is embedded into the image in the frequency domain using DWT, SVD and torus automorphism techniques.

Feature-Based embedding:

In [4], an algorithm has been proposed that inserts watermark by building an image feature description table (IFDT). This table includes **Harris corner detection** and Delaunay trigonometry. Watermark extraction was done by regenerating Delaunay trigonometry to the image. Then another table was used to match the same features as in the embedding. After geometric correction watermark was extracted. In [5], watermarking is embedded and extracted using the **scale-invariant feature transform** (SIFT). However this method takes a lot of computation to search all the features in the image. Principle axis alignment of embedding circles is extracted to match the feature points. In [12], the **modified Harris-Laplace** detector is used to extract steady feature points. These points are used for image correction. The feature points are saved as secret key for the watermark detection. However, the extracting process is complex as watermarked image needs to be corrected before extracted. This correction was done using linear equation, basically affine transformation to see if the image has suffered rotation or scaling attacks. In [13], feature points are obtained from the normalized host image. Watermark is embedded into the most stable feature points. They have created a mask image that takes the bits from the normalized image. This was done to know the rotation and scale factors. The watermark is added in the middle frequency range of the DCT block. In the

[1] The article is published in the original.

Received May 18, 2016

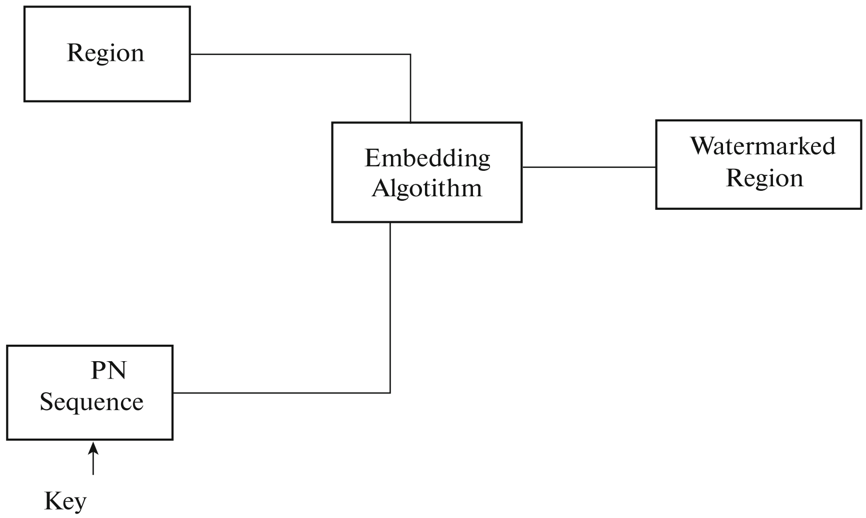

Fig. 1. PN Sequence Added to a Region.

extraction process, same normalization procedure was done to the attacked image. DCT blocks are recalculated to detect the watermark. In [8], watermark is embedded in the **Delaunay triangles pattern**, and the watermarked image is rotated using affine transformation. In the detection procedure inverse affine transform was used to recall the angle of the rotated image, and using pattern recognition to detect the embedded region. Xinguo et al, [9] have also used **SIFT** feature points on the host image. To decide the embedding location and capacity, the watermark sequence length is adjusted by Neighbor interpolation. Watermark is embedded into DCT low-frequency coefficients of some qualified sub-blocks for the host image. Akarpol et al. [11] proposed a better algorithm which uses both time domain and frequency domain for embedding. This method was used because in time domain the image can be damaged by scaling and cropping, and in frequency domain, image can be damaged by geometric attack or noise attack.

2. IMAGE WATERMARKING USING RANDOM SEQUENCES

Many random sequences that are generated by processors are seemed to be random. In fact random sequence can be known if the generator algorithm is known or its initial polynomial state. This scheme of data hiding by adding random sequences uses the concept of code division multiple access. Our message vector to be sent through an image is spread over a wide frequency band. This band is wider than the actual bandwidth required, spreading signal. This concept is robust against several attacks. Portion of the data can be lost due to attacks, but the total watermark information will be extracted.

2.1. Random Sequences

2.1.1. PN Sequence. PN sequence is a robust watermark because of its resistance against malicious attack on an image. Here are some properties concerning PN sequence used in image watermarking:

• *Balance property*: the sequence generated have equal number of ones and zeros.

• *Shift Property*: For any maximal length sequence and its cyclically shifted sequences, the agreements and disagreements among them will be approximately equal.

• *Run property*: A run in random sequences is a sequence that contains a single type of a digit. In general, a PN sequence of length n has a frequency $\frac{1}{2^n}$.

• *Autocorrelation Property*: The autocorrelation properties of pseudo noise sequences are similar to the correlation properties of random noise, i.e. there is a single autocorrelation peak.

Cross-correlation property: The cross correlation function compares two different sequences rather than shifting a sequence itself. Let $a = (a_0 \dots a_{N-1})$ denoted by the first generated step of a PN sequence. $b = (b_0 \dots b_{N-1})$ second generated step. The cross correlation between a and b is equal to zero because a and b are orthogonal and 1 if $a = \pm b$.

2.1.2. Image Watermarking Using PN Sequence. To simulate the effect of a PN over a region of an image, the following chart describes the PN sequence addition to an arbitrary region. Figure 1 shows that a region in the image is added to the PN sequence in order to get the watermarked PN sequence is generated using

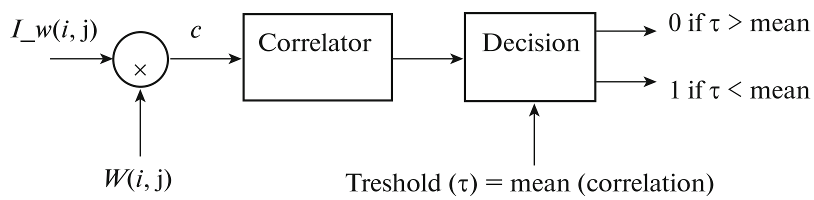

Fig. 2. Decision Rule when Extracting the Watermark.

MATLAB. Every watermark is added to an image using the following expression:

$$I_w(i,j) = I(i,j) + k \times W(I,i), \tag{2.1}$$

where,

I_w is the watermarked region.

I is the main region.

k is the gain factor

W is the random sequence (in our case the PN sequence)

In the extraction phase,

$$C = I_w(i,j) \times b(i,j), \tag{2.2}$$

where b (i, j) is the newly generated sequence.

$$C = (I(i,j) + kW(I,j) \times b(i,j) = (I(i,j)b(i,j) + k \times W(i,j) \times b(i,j),\ W(i,j) \times b(i,j) = b^2(i,j).$$

This shows that in the extraction process, the watermarked image is multiplied twice with the noise signal. So b^2 (i, j) becomes 1 and $I(i, j)b(i, j)$ is the unwanted term that can be filtered out during the correlation threshold.

The detection process of the added sequence can be resumed as follow (in Fig. 2).

2.2. Gold Sequence

• Gold sequences are heavily used in watermarking due to their low values of correlation. The values are same as for PN sequence.

• Gold sequences have a period of $2^n - 1$, therefore when n increases, the number of gold sequence will increase as the power of 2.

• Gold sequences are well balanced. The balanced Gold-sequence is a sequence that the number of 1 is the same with or only one more than the number of 0.

2.3. Decimal Sequence

Decimal sequences are obtained when a list of numbers are represented in a decimal form in a base r [15]. Decimal sequence can repeat in a watermark. D-sequences of the form $\frac{1}{q}$, q being prime, exhibits the property where the digits spaced half a period apart add up to exactly $r - 1$, where the base is in which the number has been expressed and for the same class of d-sequences, all the subsequences of length m, where $r^m > q$, are distinct. It was recently shown that d-sequences can be generated using shift registers that allow carry [16]. This can be done by using Tirtha algorithm, which can be used whenever $q = tr - 1$, where r is the radix. We can apply this algorithm to an arbitrary q which let us generate d-sequences efficiently.

3. HARRIS FEATURE POINTS DETECTOR

3.1. Feature Points Using Harris Detector

Among various detectors, Harris detector shows the best performance. The Harris points are stable points, resist against noises and geometric attack. These points are robust against rotation, noise addition and different attacks.

The Harris feature points can reduce the amount of computation compared to the processing of the whole pixels of an image. It is rotation and noise invariant. We also can find the strong Harris points where higher resistivity against noise compared to the rest pixels in the image. Each time Harris is applied to an image, the results are the location of each point, the metric and the count. Metric values are calculated with a descending order with all values near zero. The bigger these values are, higher resistivity against noise. Some Harris points in noisy image might have higher strength, but can be filtered by setting a threshold. This threshold is chosen by the following equation:

$$Thres = \alpha \times max \times metric, \tag{3.1}$$

where $\alpha \in \{0, ..., 1\}$ and $max \times metric$ is the maximum value of metric.

3.2. Harris Corner Measure

Harris detector uses the second moment matrix as the basis of its feature decision. The matrix A has values closely related to the derivatives of image intensity.

$$A(x) = \sum_{p,q} w(p,q) \begin{bmatrix} I^2x(x) & IxIy(x) \\ IxIy(x) & I^2y(x) \end{bmatrix}, \tag{3.2}$$

where I_x and I_y are the respective derivatives of pixel intensities in the x and y direction at point x and p, q are the values of the weighting function. The off-diagonal entries are the product of I_x and I_y, The diagonal

Fig. 3. Feature Points Detected before Threshold.

Fig. 4. Feature Points Detected after Threshold.

entries are squares of the respective derivatives. The weighting function $w(x, y)$ can be uniform, but is more typically an isotropic, circular Gaussian.

As it turns out, this A matrix describes the shape of the autocorrelation measure as due to shifts in window location. Thus, if we let λ_1 and λ_2 be the eigenvalues of A, then these values will provide a quantitative description of how the autocorrelation measure changes in space: its principal curvatures. As Harris and Stephens (1988) point out, the A matrix centered on corner points will have two large, positive eigenvalues. Rather than extracting these eigenvalues using methods like singular value decomposition, the Harris measure based on the trace and determinant is used:

$$\begin{aligned} R &= det(A) - \alpha \times trace^2(A) \\ &= \lambda_1 \times \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 \end{aligned} \tag{3.3}$$

Where α is a constant.

Thus feature points are expressed as a local maxima threshold:

$$\begin{aligned} \{x_c\} = \{x_c | R(x_c) > R(x_i), \\ x_i \in W(x_c)\},\ R(x_c) > t_{threshold}, \end{aligned} \tag{3.4}$$

where $R(x)$ are the Harris points calculated at x.

Example:

Harris detector is applied to the 'Lena' image. The detector encountered 454 points. Adding sequence to all these points will take a lot of computation.

Using equation (3.1) will limit the number of detected feature points. The following figure shows the strongest feature points in the image.

4. IMPROVED WATERMARKING ALGORITHM

4.1. Algorithm

Our algorithm can be summarized as follows.

4.1.1. Watermark Embedding Process.

Step 1. Read M × N grayscale image.

Step 2. Apply the Harris detector to allocate image feature points. Calculate the first N points with highest metric values.

Step 3. Choose N regions each centered by a feature point.

Step 4. A sequence is added to each region. Each region can have a different sequence.

Watermark is embedded according to the following equation:

$$R_{wi} = \begin{cases} Ri + \mathrm{k} * PNi & if\, message vector = 1 \\ Ri & if\, message vector = 0, \end{cases} \tag{4.1}$$
$$i = \{1, 2, ..., N\},$$

where R is the region, R_w is the watermarked region

The block diagram of the watermark embedding is shown below:

Notice that PN1, PN2 ... PNN could be PN sequences or different sequences like Gold sequences, decimal sequences ...

4.1.2. Watermark Extraction Process.

Step 1. Read the watermarked image. Noise will be added to account for attacks.

Step 2. M feature points will be detected by applying the Harris detector.

Step 3. For each feature point do the following:

- Move the region centered around it left, right, up and down. Correlate each region obtained with different PN sequences.
- The maximum correlation will determine the region and its sequence embedded.

Step 4. Correlate each region obtained with its sequence.

Step 5. Extract message vector.

The block diagram of the watermark extraction process is shown below.

4.2. Experiments

In our experiments, we have used 50 different images taken from the image processing database [20] including the 4 popular images (Cameraman, Lena, Baboon, and Peppers) shown in the following figure.

Many experiments were implemented to evaluate our proposed watermarking algorithm. We have used 256 × 256 gray-scale images, each with different textures. Our proposed watermarking algorithm is robust against noise and geometric attacks. Increasing the number of regions will result in a better extraction scheme. Comparisons are performed with different algorithms. Extraction process is evaluated by using the 2-D correlation coefficient, which is given by the following equation:

$$R = \frac{\Sigma_m \Sigma_n [(A_{mn} - \bar{A})(B_{mn} - \bar{B})]}{\sqrt{\Sigma_m \Sigma_n (A_{mn} - \bar{A})^2 \Sigma_m \Sigma_n (A_{mn} - \bar{A})^2}}. \quad (4.2)$$

where $\bar{A}$ = mean (A), $\bar{B}$ = mean (B)

A and B refer to the original and extracted message vector respectively.

It should be noted that we have used Matlab 2014 in all of our simulations.

4.2.1. Attack Free Experiment.

4.2.1.1. Embedding process. After reading the image, Harris detector gave the following result which is shown below:

We spotted 184 points by Harris feature detector. Watermark embedding with such number of regions will take a lot of computation and delay in the extraction domain. Therefore, we will select the 7 most robust feature points according to the metric value obtained from the Harris detector.

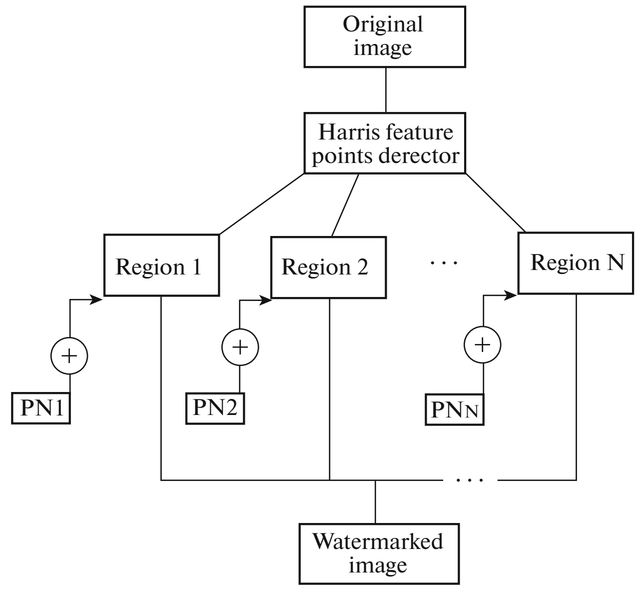

Fig. 5. Block Diagram for Watermark Embedding Process.

Feature points with the highest metric values are shown in the figure below:

We chose 5 out of 7 points randomly for watermark embedding. Each point will be the center of a 15 × 15 region. PN sequence is added to this region as shown in the figure below.

4.2.1.2. Extraction process. Each feature point is moved in all direction as shown in Fig. 7. Each time the feature point is moved a pixel, a value that corresponds to the correlation of this region with its PN sequence. The highest value will lead us to figure out the center of each watermarked region.

The following figure contains the feature points detected in our example.

Table 1. Watermarked Region Detected after no Attacks

		No attack		
		3 regions	4 regions	5 regions
Cameraman	PN	3/3, 100%	4/4, 100%	5/5, 100%
	Gold	3/3, 100%	4/4, 100%	5/5, 100%
	Decimal	3/3, 100%	4/4, 100%	5/5, 100%
Street	PN	3/3, 100%	4/4, 100%	4/5, 100%
	Gold	3/3, 100%	3/4, 100%	5/5, 100%
	Decimal	3/3, 100%	4/4, 100%	5/5, 100%
Pepper	PN	3/3, 100%	4/4, 100%	4/5, 100%
	Gold	3/3, 100%	4/4, 100%	3/5, 100%
	Decimal	2/3, 100%	2/4, 100%	2/5, 100%

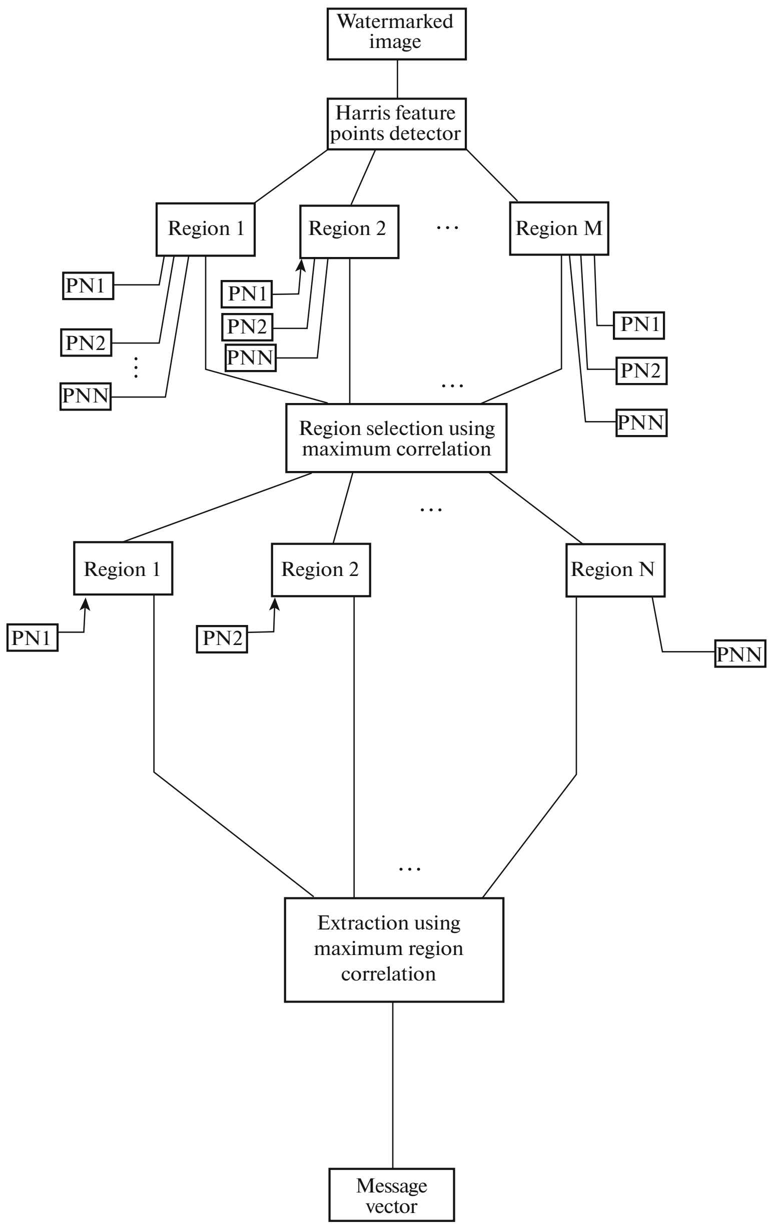

Fig. 6. Extraction Process Threshold Based-Correlation.

4.2.2. Noise Free Images Results. Table 1 shows the results of the proposed watermarking algorithm. The watermarked image was not affected by any attack like noise or geometric attacks. It is clearly shown that the algorithm was able to detect the watermarked regions. In each of the cases shown in the table, regions were embedded with the same sequence. PN and GOLD sequences have similar properties for random sequences. Results were the same for both sequences.

The bases used for decimal sequence are 3, 6 and 7. And the prime numbers are 79 and 137 respectively. The choice of these values is random.

For example in Table 1, in the case where five regions were embedded using the cameraman image, the algorithm detected 5 out of 5 regions. In this case the correlation of the message vector extracted with the original vector is 100%, where no errors occurred in the extraction process. For example for the Pepper image and using the Decimal Sequence, 2 out of 3 regions were correctly detected and 100% of the watermark was detected.

4.2.3. Noise Attacks.

4.2.3.1. Noise addition experiments. We have used Gaussian noise with different variances and Uniform noise with different scales.

For example, when Gaussian noise is added to the watermarked image with mean = 0, and variance = 0.1, the following image is obtained.

Fig. 7. Some of the images used.

Fig. 8. Feature Points Location.

Fig. 9. Strongest Feature Points Location.

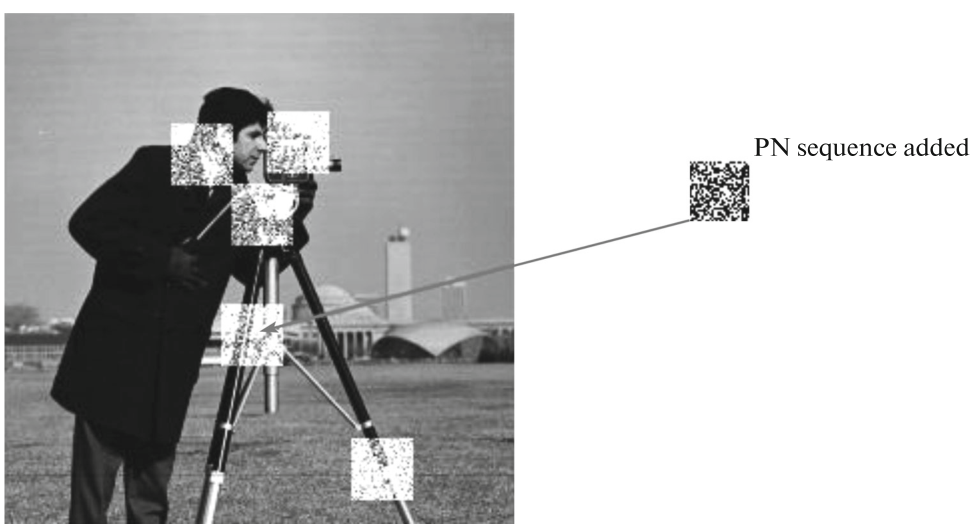

Fig. 10. Watermarked Image.

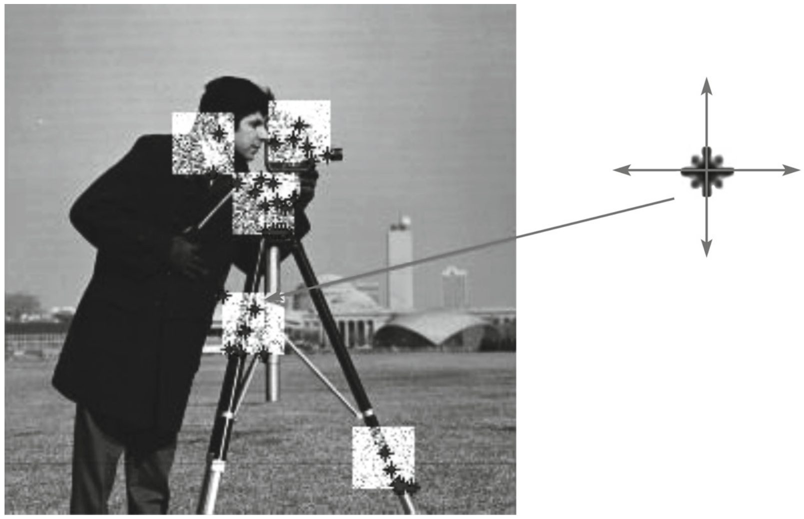

Fig. 11. Each Feature Point is Moved in these Directions.

Fig. 12. Regions Center Points Detected.

Fig. 13. AWGN Embedded to the Watermarked Image.

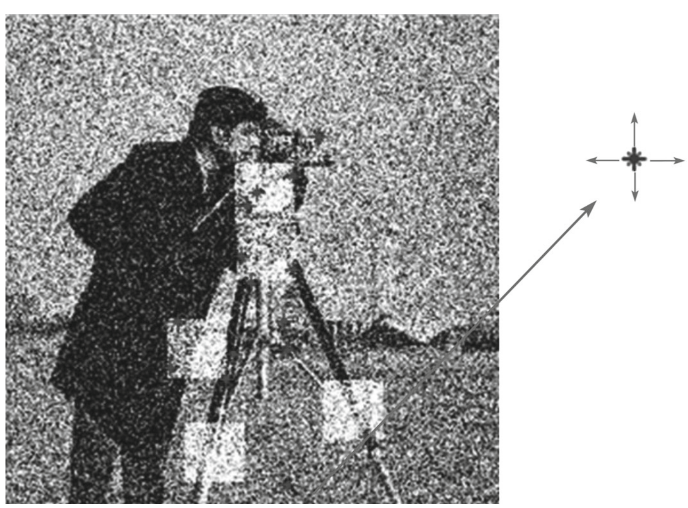

Fig. 14. Each Feature Point is Moved in these Directions.

Table 2. Watermarked Region Detected after Noise Attacks

		Additive Uniform noise with scale		
		0.1	0.15	0.2
Lena	Scheme [17]	5/8	4/8	1/8
	Our scheme	4/4	3/4	2/4
Baboon	Scheme [17]	6/11	4/11	5/11
	Our scheme	4/4	3/4	2/4
Pepper	Scheme [17]	4/4	2/4	1/4
	Our scheme	4/4	3/4	2/4

Harris detector is applied to the previous image. Not all the feature points obtained are useful in the extraction domain.

Each feature point is moved in all directions, and then correlated to its PN sequences as shown in Fig. 14.

Each move should be saved in a matrix which contains the value of the correlation of this region with its PN sequence. The highest value will lead us to figure out the center of each watermarked region.

The following figure contains the feature points detected in our example. 4 out of 5 regions have been detected. In this case, the extraction will be 100%.

4.2.3.2. Noise addition results. Table 2 shows the same extraction scheme shown in table 1, but with noise attack. Uniform noise was added with different scales (0.1, 0.15, and 0.2). Our results are compared with scheme [17].

Table 3 shows Bit Error Rate results for the Baboon image compared with scheme [18] that uses DFT-SVD embedding and extracting process. Gaussian noise was added with different variances (10, 15, and 20). Average and Median filter attacks were also applied.

4.2.4. Watermark Extraction with Geometric Attacks. The embedding process will be the same as in the previous example. The process can be resumed as follows:

• Reading the grayscale image.

• Detecting feature points and selecting the robust among all the points detected.

• Adding sequences to each region centered on the chosen feature points.

• Watermarked image is rotated by an angle. In our case the angle will be 50°.

• The extraction process is summarized as follows

• Each region centered by a feature point is moved in all directions.

• Within each move, the region is correlated with its PN sequence.

• Maximum correlation will determine the angle of the image and the center of the watermarked regions.

We calculate the feature points by the Harris detector as shown in Fig. 4. 12.

4.2.4.1. Rotation attack results. Table 4 shows the result after geometric attack. In geometric attack, all regions will be detected by the algorithm, but the message vector obtained will contain some error.

Compared to scheme [19], our results are better. They have used the watermark edge detection technique. Table 5 shows the average detection results for all the 50 images under different attacks. Extraction process is evaluated by using the 2-D correlation coefficient.

4.3. Comparisons of the Results Obtained with Other Papers

The results of the proposed algorithm are compared with Tang et al. method [17], Hu et al. method [18] and Gao et al method [19]. These methods are chosen because all of them belong to the feature-based

Table 3. Bit Error Rate

Baboon	Gaussian noise	10	15	20
	Scheme [18]	1.6	3.1	4.7
	Our scheme	1.1	2.8	4.1
	Average Filter	2 × 2	3 × 3	4 × 4
	Scheme [18]	1.6	3.1	11.1
	Our scheme	1.1	2.8	9.8
	Median Filter	2 × 2	3 × 3	4 × 4
	Scheme [18]	1.6	3.1	15.9
	Our scheme	1.1	2.8	13.5

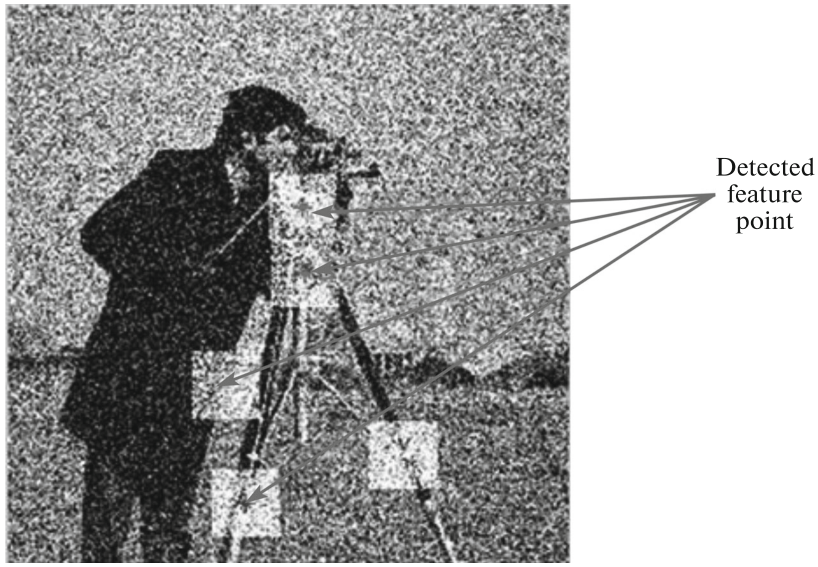

Fig. 15. Regions Center Points Detected.

Table 4. Watermark Extraction after Geometric Attacks

		Rotation		Scaling	
		5 degrees	30 degrees	1.5 × 1.5	0.7 × 0.9
Lena	Scheme [19]	4/10	4/10	6/10	4/10
	Our scheme	4/4	3/4	3/4	2/4
Baboon	Scheme [19]	5/15	4/15	7/15	2/15
	Our scheme	4/4	3/4	3/4	2/4
Pepper	Scheme [19]	9/23	7/23	16/23	6/23
	Our scheme	4/4	3/4	3/4	2/4

Table 5. Watermark Extraction under Different Attacks for the 50 images

Attacks	Average Extraction
Uniform noise scale 0.1	3.83/4, 0.61
Uniform noise scale 0.2	2.25/4, 0.42
Gaussian noise variance = 10	3.21/4, 0.58
Gaussian noise variance = 20	1.84/4, 0.52
Median Filter 3 × 3	2.89/4, 0.47
Average Filter 3 × 3	2.93/4, 0.49
Rotation: 5 degrees	3.92/4, 0.63
Rotation: 30 degrees	2.88/4, 0.45
Scaling: 1.5 × 1.5	3.11/4, 0.56
Scaling:0.7 × 0.9	2.21/4, 0.42

watermarking group. Table 2 compares our method to Tang et al method [17]. As shown in the table we compared our results using the Pepper, Lena, and Baboon images. Uniform noise with different scales was added. In [17] Feature points were extracted by a method called Mexican Hat. The watermark was embedded into regions in the image using DFT (Discrete Fourier Transform). Different values for the embedded noise were tested. Our algorithm detected all the four watermarked regions. In [17], the detection was not reliable when the noise was added to the image. Our proposed algorithm extracts more stable feature points and the extraction is more reliable.

In Table 3, we compared our results with scheme [18] where they used DFT-SVD transformation. The results were also better. Our algorithm gave lower Bit Error Rate for Gaussian noise with different variances and for Median and Average filtering attacks.

Fig. 16. Feature Points of the Watermarked Image Rotated 50°.

Table 4 summarizes the experimental results after geometric attack. Rotation and Scaling degrade the quality of the image. Our algorithm detected the watermark with less loss compared to method [19].

CONCLUSION

In this paper, we proposed a robust noise and geometric resilient digital watermarking scheme. The major contributions consist of:

• Block movement to account for noise attacks.

• Angle movement to account for geometric attacks.

The proposed algorithm is robust against a wide variety of tests as experimented in the previous sections. The extraction is better in noise attacks than in geometric attacks.

• 100% extraction of the message when the image was not attacked.

• Increasing the intensity of the noise will lead to some errors in the message vector.

Our scheme showed better results compared to other schemes. Different noise scales were added to three different images. When the scale of the noise increases above 0.1, our algorithm detected all watermarked regions, whereas in [17], the detection missed 2 and 3 watermarked regions in each case. Our algorithm detects the strongest watermarked regions. These regions have the highest resistivity against noise.

• Rotation attacks will always occur an error in the extraction process.

In rotation attack, our extracted watermark has a better precision than scheme [19]. Although some errors are present in our extracted message vector, our scheme showed better correlation results.

When noise is added, two scaling parameters are required to detect the message vector with less error:

• Increasing the dimension of each region, but note that not all sequences can be generated as the dimension of the region.

• Increasing the number of embedded regions.

It should be noted that different sequences can be embedded in different regions. For example, in the same image we can embed the PN, Gold and decimal sequence into different regions. And decimal sequence can lead to better extraction results.

REFERENCES

1. W. Y. Chen and S. Y. Huang, "Digital watermarking using DCT transformation," Nat. Chin-Yi Inst. Technol. **18**, 173–183 (2008).
2. M. Vidyasagar, H. Song, and C. Elizabeth, "A survey of digital image watermarking techniques," in *Proc. 3rd IEEE Int. Conf. on Industrial Informatics (INDIN)* (Curtin Univ. of Tech. Perth, 2005), pp. 709–713.
3. S. J. Vaishalim and R. G. Sachin, "Literature review of wavelet based digital image watermarking techniques," Int. J. Comput. Appl. **31** (1), 975–987 (2011).
4. Z. Chuzhong, Z. Li, T. Xiaolin, and X. Shaowei, "A novel anti-geometric-attacking watermarking algorithm based on Harris feature points," in *Proc. 4th Int. Congress on Image and Signal Processing* (Shanghai, 2011), pp. 1003–1007.
5. Y. T. Lin, C. Y. Huang, and G. C. Lee, "Rotation, scaling, and translation resilient watermarking for images," IET Image Processing **5** (4), 328–340 (2009).
6. C. Y. Lin, W. Min, A. B. Jeffrey, C. Ingemar, L. M. Matt, and M. L. Yui, "Rotation, scale, and translation resilient watermarking for images," IEEE Trans. Image Processing **10** (5) (2001).
7. A. Sangeetha, B. Gomathy, and K. Anusudha, "A watermarking approach to combat geometric attacks," in *Proc. Int. Conf. on Digital Image Processing* (Cairo, 2009), pp. 381–385.
8. B. Patrick, C. Jean-Marc, and B. Macq, "Robust watermarking based on the warping of pre-defined triangular patterns," in *Proc. SPIE Electronic Imaging, Security and Watermarking of Multimedia Content II* (2000), pp. 99–109.
9. Z. Xinguo, L. Na, and J. Nawei, "Robust watermarking algorithm for digital image based on SIFT feature points," in *Proc. 5th IEEE Int. Conf. on Software Engineering and Service Science (ICSESS)* (Beijing, June 27–29, 2014), pp. 996–999.
10. R. K. Krishna and R. C. Srinivasa, "Block based robust blind image watermarking using discrete wavelet transform," in *Proc. IEEE Int. Colloquium on Signal Processing & Its Applications* (Kuala Lumpur, 2009), pp. 58–61.
11. P. Akarapol, S. Sippawit, and T. Panwit, "Image watermarking using joined wavelet and time domain," in *Proc. 3rd ICT Int. Student Project Conf.* (Bangkok, 2014), pp. 47–50.
12. Z. Jun, S. El-Na, G. Mei-Jing, Y. Zheng-Ling, X. Jun-Tao, and W. Liang, "Robust watermarking of digital

image based on feature and DWT," Proc. 2012 Int. Conf. on Machine Learning and Cybernetics **4** (2), 1356–1361 (2012).

13. N. Wei, Y. Kazuhiko, C. H. Manuel, N. M. Mariko, and P. M. Héctor, "Robust image watermarking using feature points and image normalization," Proc. Electronics, Robotics and Automotive Mechanics Conf. **4** (1), 313–318 (2010).
14. Afaq Ahmad, Sayyid Samir Al-Busaidi, and Mufeed Juma Al-Musharafi, "On properties of PN sequences generated," Indian J. Sci. Technol. **21** (4), 1290–1303 (2013).
15. S. C. Kak and A. Chatterjee, "On decimal sequences," IEEE Trans. Inf. Theory **27**, 647–652 (1981).
16. S. C. Kak, "Generating d-sequences," Electron. **23** (1), 202–203 (1987).
17. W. T. Chih and M. Hsueh, "A feature-based robust digital image watermarking scheme," IEEE Trans. Signal Processing **51** (4), 1–5 (2003).
18. Yuping Hu, Zhijian Wang, Hui Liu, and Guangjun Guo, "A geometric distortion resilient image watermark algorithm based on DWT-DFT," J. Software **6** (9), 1805–1812 (2011).
19. G. Xinbo, D. Cheng, L. Xuelong, and T. Dacheng, "Geometric distortion insensitive image watermarking in affine covariant regions," IEEE Trans. Syst. **40** (3), 278–286 (2010).
20. http://www.imageprocessingplace.com/root_-files_V3/image_databases.htm.

Issam Dagher finished his MS in electrical engineering degree in 1994 from Florida International University, Miami, USA. He finished his PhD in 1997 from University of Central Florida, Orlando USA. He is now an associate professor at the University of Balamand, Lebanon. His area of interests are pattern recognition, neural networks, artificial intelligence, and computer vision. He published many papers on these topics.

Patrick Hanna finished his MS in electrical engineering degree in 2014 from University of Balamand. His area of interests are image processing and computer vision.