Blind Image Watermarking Based on Chaotic Maps

Antonius Cahya Prihandoko[1]
[1] Dept.of Information System
University of Jember
Jember, Indonesia
antonius.cahyaprihandoko@my.jcu.edu.au

Hossein Ghodosi, Bruce Litow[2]
[1,2] Dept. of Information Technology
James Cook University
Townsville, Australia
[2] (hossein.ghodosi/bruce.litow)@jcu.edu.au

***Abstract*—Security of a watermark refers to its resistance to unauthorized detecting and decoding, while watermark robustness refers to the watermark's resistance against common processing. Many watermarking schemes emphasize robustness more than security. However, a robust watermark is not enough to accomplish protection because the range of hostile attacks is not limited to common processing and distortions. In this paper, we give consideration to watermark security. To achieve this, we employ chaotic maps due to their extreme sensitivity to the initial values. If one fails to provide these values, the watermark will be wrongly extracted. While the chaotic maps provide perfect watermarking security, the proposed scheme is also intended to achieve robustness.**

***Index Terms*—Blind Image Watermarking, Chaotic Map, Frequency Domain, Robustness, Security.**

I. INTRODUCTION

Digital watermarking is viewed as a potential tool for preserving protection of digital content. A watermark, which is embedded into a host content will remain associated with the content in its subsequent duplication and distribution. The watermark can later be detected or extracted to provide ownership evidence or copyright notification. Research on digital watermarking was initiated in the early 1990's and the interest in this topic has started to increase since 1995 [1]. The increasing amount of watermarking research has been principally motivated by its essential applications in digital copyright management and protection.

Typically, research on digital image watermarking attempts to achieve three main features: imperceptibility, robustness and security, whereas the evaluation of the last two features relies on watermarking detectability. Based on the method used to detect watermark, a watermarking scheme can be categorized into non-blind or blind watermarking [2]. A non-blind watermarking requires the orginal image to detect or extract the watermark, while a blind watermarking does not. Imperceptibility of a watermark aims at preserving the quality of the host image. While the watermark should act as a distinctive identifier, it must not degrade the aesthetic value of the image.

Watermark security refers to its resistance to unauthorized detecting and decoding, while robustness refers to a watermark's resistance against common processing, such as filtering, geometrical transforms and compression. A survey on watermarking research shows that many watermarking schemes give consideration to robustness more than security [3]. However, a robust watermark is not enough to accomplish protection because the range of hostile attacks is not limited to common processing and distortions. Therefore, robustness and security should be proportionally considered in a watermarking system.

A popular approach to achieve the robust and secure watermarking for multimedia content is the spread-spectrum technique. A general spread-spectrum system encodes data in a chosen binary sequence that appears like noise to an outsider but can be recognized by a legitimate receiver with the aid of an appropriate key [4]. The most cited secure spread-spectrum watermarking method was one presented by Cox *et al.* [5]. In this method, the watermark is placed into the perceptually most significant components of the content spectrum, since many common signal and geometrical processes affect the insignificant regions of content. Cox *et al*'s watermarking scheme yields a watermark which is invisble, difficult to remove and robust to common signal and geometric distorsions. However, this method is a non-blind watermarking and uses a sequence of meaningless random numbers as the watermark. A non-blind watermarking might be less applicable, because the original image may not be available when watermark detection is required. In this paper, we consider using a meaningful watermark, i.e. a binary image, in a blind watermarking. To achieve security, the watermarking scheme is constructed based on chaotic maps.

II. CHAOTIC MAP AND WATERMARKING

A chaotic map is an evaluation function that exhibits some sorts of chaotic behavior. This map can be parameterized by discrete-time or continuous-time. Discrete maps usually take the form of iterated functions. The most attractive features of chaos in the information hiding are its extreme sensitivity to initial conditions and the outspreading of orbits over the entire space. These special characteristics make chaotic maps excellent candidates for securing watermarks.

One of the simplest chaotic maps is the logistic map. The logistic map is discrete and mathematically written as

$$z_{n+1} = rz_n(1-z_n) \tag{1}$$

where $z_n \in (0,1]$ and $0 \le r \le 4$. When $r > 3.57$, the map is in the chaotic state [6]. The sequences generated by the logistic map are sensitive to the initial value, meaning that two logistic

  ISSN: 2203-1731

sequences generated from different initial values are statistically uncorrelated.

Chaotic maps have been proposed to undertake various tasks in watermarking schemes. For example, a watermarking scheme employs logistic function to modify Cox's watermarking by adding a watermark encryption feature [7]. Instead of using a sequence of meaningless random numbers, this scheme uses a binary image as the watermark. The binary watermark is firstly encrypted using two generated logistic sequences and then spread into the host image spectrum using Cox's technique [5]. To extract the watermark, the modified Cox's scheme also requires the original host image. Thus, it is a non-blind watermarking. However, though the scheme is robust against JPEG compression, it is susceptible to cropping, resizing and adding noise.

Chaotic maps are often used to modify a watermark before the embedding process. The modification could be mutation, permutation or mixture. Watermark mutation is a process to randomly change the value of watermark pixels according to a chaotic sequence [8]. The mutated watermark pixels are then randomly embedded into the host image spectrum according to another generated chaotic sequence. This mechanism can preserve the hidden information against geometric and non geometric attacks. Next, watermark permutation is a process to randomly change the watermark bits' positions. In a scheme proposed by Wang *et al.* [9], after the watermark is permuted using the first chaotic sequence, a small number of reference points are randomly selected in the middle frequency bands of the host image spectrum according to the second chaotic sequence. The permuted watermark bits are then embedded into the neighborhood of each reference point according to the third sequence. This mechanism was proposed to achieve watermark imperceptibility and robustness. Another watermarking scheme utilize chaotic map iterations to accomplish watermark mixture [10]. The initial state of the iteration is constituted by the watermark, which is considered as a boolean vector. The subsequent chaotic boolean vectors are generated by a number of iterations of a logistic map. The mixed watermark is the last boolean vector generated by the chaotic iterations.

Some chaotic-based watermarking schemes rely on securing selection of embedding locations. To firmly insert a watermark signal, some schemes randomly select only several local spectrums as the embedding location [6], [11-13]. In these schemes, a sub image is constructed from the host image according to a logistic sequence. The spectrum of the sub image is then used as the embedding location. Another scheme uses a 2-D invertible chaotic map to determine watermark insertion location in the spatial domain [14], [15]. Basically, the 2-D chaotic map is a one to one map. Utilizing this map is intended to improve the success of watermark extraction.

III. Proposed Scheme

We proposed a frequency domain watermarking scheme. This scheme employs two 1-D logistic maps (equation (1)) and one 2-D chaotic map (equation (3)). The first and second 1-D logistic maps are used to encrypt the watermark and construct a sub image from the host image, respectively, while the 2-D chaotic map is utilized to select the location of inserting watermark elements.

A. Watermark Insertion

The watermark insertion process (see Fig. 1) begins with the watermark encryption. Firstly, a logistic sequence L_1 is generated using the first logistic map and then converted into a binary sequence L_{bin}. Finally, watermark W is bitxor-ed with L_{bin} to obtain the encrypted watermark W_{enc}. The detail of this encryption process is presented in Algorithm 1.

Algorithm 1 Encrypt a binary watermark

Input: watermark W which is an $M \times M$ binary image, and initial value of logistic map z_{0_1}
Output: encrypted watermark W_{enc}
BEGIN
- Generate a logistic sequence L_1 of length M^2 using equation 1 with the initial value z_{0_1}
- Compute m = mean of L_1
- Convert L_1 into a binary sequence L_{bin} using:

for $i = 1$ to M^2 **do**
  if $L_{1_i} \geq m$ **then**
    $L_{bin_i} = 1$
  else
    $L_{bin_i} = 0$
  end if
end for

- Reshape L_{bin} to an $M \times M$ binary matrix
- Bitwise exclusive OR (bitxor) the reshaped L_{bin} and W to obtain W_{enc}

END

The next stage is generating sub image I_{sub} from the host image I with the aid of the second logistic map. Suppose I is of size $N \times N$ and m is the divisor of N, the construction of I_{sub} is performed by Algorithm 2.

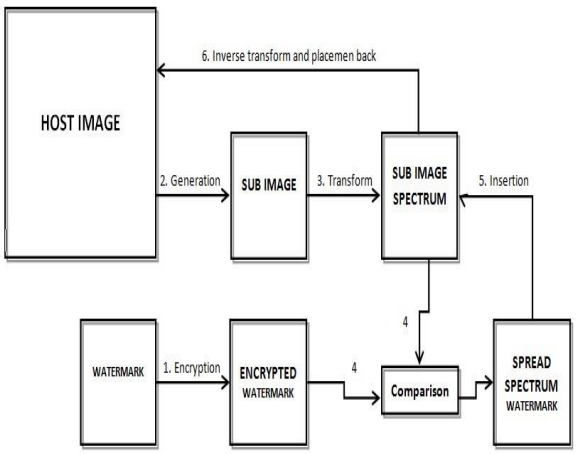

Fig. 1. Watermark Insertion Process

Algorithm 2 Construct a sub image

Input: image I which is an $N \times N$ gray scale image, and initial value of the logistic map z_{0_2}
Output: sub image I_{sub}
BEGIN
- Split I into some disjoint blocks of $m \times m$ pixels each, where m is a divisor of N
- Label all blocks in a scan-line order by positive integers from 1 to $(N/m)^2$
- Generate a logistic sequence L of length $(N/m)^2$ using equation 1 with the initial value z_{0_2}
- Sort L in ascendance
- Construct a sequence S containing the original indexes of the elements of the sorted L. Note that the elements of S must be the first $(N/m)^2$ positive integers
- Select blocks from split I according to the first $1/4(N/m)^2$ elements of S
- Construct I_{sub} by arranging the selected blocks in a scan-line order. Note that I_{sub} is an $\frac{N}{2m} \times \frac{N}{2m}$ blocks or $\frac{N}{2} \times \frac{N}{2}$ pixels image.

END

Once I_{sub} has been generated, it is transformed into its spectrum I_{spec}. The information of I_{spec} and the encrypted watermark W_{enc} are then used to generate a spread-spectrum watermark W_{spec}. First of all, each element of I_{spec} is compared to its neighbors. An element may have three, five or eight neighbors depending on its position. Suppose $I_{spec_{i,j}}$ is the element of I_{spec} at the coordinate (i, j) and $t_{i,j}$ is the number of $I_{spec_{i,j}}$'s neighbors having less values than $I_{spec_{i,j}}$. The construction of W_{spec} is undertaken according to equation 2.

$$W_{spec_{i,j}} = \begin{cases} 1, & if \quad (t_{i,j} \geq 2 \wedge W_{enc_{i,j}} = 1) \\ & \quad \vee (t_{i,j} < 2 \wedge W_{enc_{i,j}} = 0) \\ -1, & otherwise \end{cases} \qquad (2)$$

The spread-spectrum watermark W_{spec} is then inserted into sub image spectrum I_{spec} according to the 2-D chaotic map. This map is a generalized form of the 2-D Arnold cat map [16] and is described by:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} = A \begin{bmatrix} x_t \\ y_t \end{bmatrix} \bmod N_S \qquad (3)$$

where a, b, c and d are positive integers such that $|A| = ad - bc = 1$. Thus, three out of these four parameters are independent. N_S in our scheme is the number of rows or columns of the constructed sub image.

To select the insertion position for each elemen of W_{spec}, we adopt the technique proposed by Wu *et al.* [14], [15] with some minor modifications. To determine the insertion location of a watermark bit, Wu *et al.* initially chose an initial value (x_0, y_0) and iterated it using equation (3) for n times. The output of this iteration round, (x_n, y_n), is determined as the insertion location of the watermark bit and used as the initial value of the next iteration round for the next watermark bit. The chain of these iteration rounds will end after all watermark bits are placed. However, according to our simulation using the same initial values as theirs ($a = 1$, $b = 2$, $c = 3$, $(x_0, y_0) = (2,3)$, $n = 20$), this mechanism results in failure to extract watermark (see Fig. 2). This is because of placement conflicts. Some watermark bits are mapped into the same insertion location (see Table I), so that the inverse function failes to reveal some original watermark bits.

In our scheme, therefore, instead of using a chain of dependent iteration rounds, the insertion location of each element of W_{spec} is computed independently. This mechanism is intended to keep the natural property of the 2-D chaotic map, which is one to one map, so that the watermark bits of different coordinates will be mapped into different insertion positions. For each element of W_{spec}, its coordinate (i, j) is served as the initial value and iterated using equation (3). After n iterations, the result (x_n, y_n) will be served as the insertion position of the watermark bit $W_{spec_{i,j}}$. Note that each pixel of W_{spec} has only one bit.

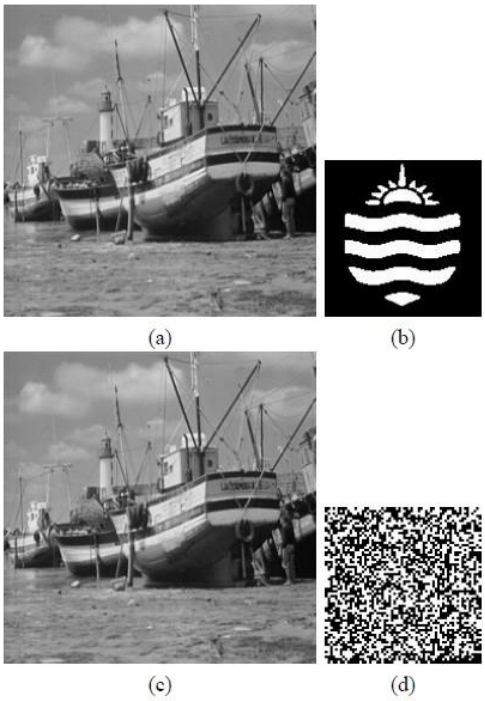

Fig. 2. Failure of extraction due to placement conflicts: (a) original host image; (b) watermark; (c) watermarked image; (d) wrongly extracted watermark

TABLE I. EXAMPLE OF THE PLACEMENT CONFLICTS

Original coordinate	Insertion location	Original coordinate	Insertion location
(60 , 62)	(2 , 99)	(60 , 64)	(2 , 3)
(60 , 46)	(2 , 99)	(60 , 48)	(2 , 3)
(60 , 14)	(2 , 99)	(60 , 39)	(2 , 179)
(59 , 62)	(2 , 99)	(60 , 55)	(2 , 179)

Once the insertion position has been selected, $W_{spec_{i,j}}$ is then combined with the element of I_{spec} of the coordinate (x_n, y_n) using equation (4).

$$I'_{spec_{x,y}} = I_{spec_{x,y}} + \beta W_{spec_{i,j}} \mid I_{spec_{x,y}} \mid \quad (4)$$

where β is an intensity parameter of the watermark. After all elements of W_{spec} have been inserted, the watermarked sub image spectrum I'_{spec} is finally transformed back into its spatial domain and then placed back into the host image to obtain a watermarked image.

.A summary of the watermark insertion process is as follows. The insertion process requires six components of the secret keys: initial values of the first and second logistic map (z_{0_1} and z_{0_2} , respectively); three independent parameters (*a*, *b* and *c*) for the 2-D chaotic map; and the number of iteration (*n*). Suppose *I* and *W* are the host and the watermark, respectively, the insertion process is undertaken through the following stages (refer to Fig. 1).

1) *W* is encrypted according to the first logistic map by Algorithm 1 to obtain W_{enc}.
2) A sub image I_{sub} is constructed from *I* according to the second logistic map by Algorithm 2.
3) I_{sub} is transformed into its spectrum I_{spec}.
4) The information of I_{spec} and W_{enc} are then used to construct a spread-spectrum watermark W_{spec}.
5) W_{spec} is embedded into I_{spec} using equation (3) (for the insertion location) and equation (4) to obtain a watermarked sub image spectrum I'_{spec}.
6) Finally, I'_{spec} is inverse-transformed to its spatial domain and is then placed back into the host image *I* to obtain a watermarked image I_{wat}.

The spread-spectrum watermark W_{spec} generated in this insertion process has to be saved and will be used in the watermark extraction.

B. Watermark Extraction

The watermark extraction process (see Fig. 3) requires six components of the secret keys that were used in the insertion process, the size of watermark and the spread-spectrum watermark W_{spec}. The extraction process begins with the construction of sub image I'_{sub} from the test image according

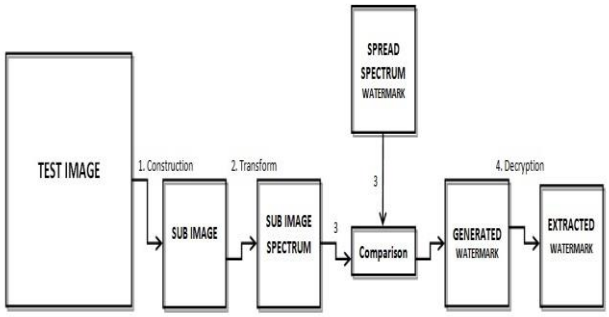

Fig. 3. Watermark Extraction Process

to the second logistic map by Algorithm 2. The test image could be the watermarked image or its modifications. I'_{sub} is then transformed into its spectrum I'_{spec}. The information of I'_{spec} and W_{spec} are used to generate a binary image W'_{enc}. First of all, each element of I'_{spec} is compared to its neighbors. Suppose $t_{i,j}$ is the number of $I'_{spec_{i,j}}$'s neighbors having less values than $I'_{spec_{i,j}}$. W'_{enc} is generated according to equation 5.

$$W'_{enc_{i,j}} = \begin{cases} 1, & if \quad (t_{i,j} \geq 2 \wedge W_{spec_{i,j}} = 1) \\ & \quad \vee (t_{i,j} < 2 \wedge W_{spec_{i,j}} = -1) \\ 0, & otherwise \end{cases} \quad (5)$$

Once W'_{enc} has been generated, it is then decrypted using the first logistic map through the same way as Algorithm 1 to obtain an extracted watermark.

IV. EXPERIMENTAL RESULTS

To implement the proposed algorithm, MATLAB simulations are performed by using 256×256 gray scale images ("boat", "chilli", "house", "lena" and "mandril") as the host images and a 128×128 binary "JCU Logo" as the watermark. The secret keys consist of $z_{0_1} = 0.642$, $z_{0_2} = 0.537$, $a = 1$, $b = 2$, $c = 3$ and $n = 20$.

Figure 4 shows a perfect imperceptibility watermark achieved by this watermarking scheme. There is no significant difference between the original and the watermarked image. This scheme also achieves a perfect security. The watermark can only be recovered using the correct values of all secret key components (z_{0_1} , z_{0_2} , *a*, *b*, *c*, *n*), the size of the watermark and the spread-spectrum watermark matrix.

  ISSN: 2203-1731

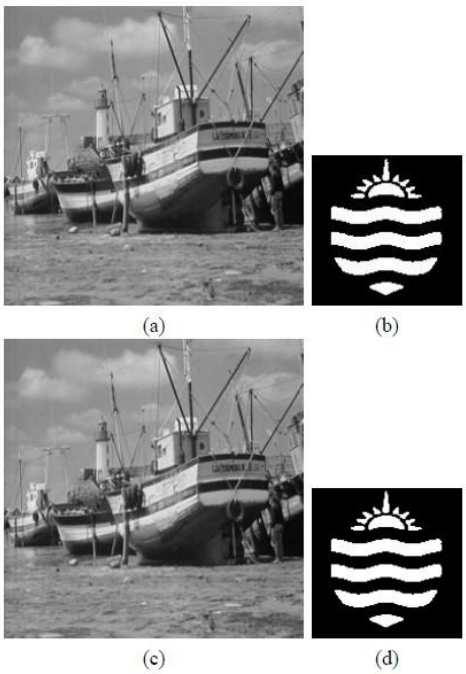

Fig. 4. Demonstration of imperceptibility: (a) original image; (b) watermark; (c) watermarked image; (d) extracted watermark.

To test the robustness of the watermarking scheme, some image processing are applied to the watermarked image including image enhancement (contrast and brightness), JPEG compression (with quality 20 % and 80 %), "salt & pepper" noise, cropping, rotation (degree 2, 10 and 90), horizontal and vertical flipping. After each processing, a watermark signal is extracted from the modified watermarked image. The extracted signal is then compared to the original watermark to measure their structural similarity index (SSIM) [17]:

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (6)$$

where μ, σ and σ_{xy} are mean, variance and covariance of the images, respectively, and c_1, c_2 are the stabilizing constants. *SSIM* has a value from 0 to 1. Similar images have *SSIM* close to 1.

Table II shows that all *SSIM* indexes of watermark signals extracted from modified images are more than 0.2. The visualization of these results exhibits that the watermark can be recovered after the application of all processing. Though the extracted signals are noisy, the watermark pattern can be visibly distinguished from the noise. Nevertheless, the qualities of the extracted signals vary. These results exhibit that the watermarking scheme is robust to withstand image enhancement, compression, noise addition and cropping, but is vulnerable to some geometric operations, such as rotating and flipping. The watermark signals extracted from the test images after these geometric operations suffer a significant degradation. This condition could be a challenge for the watermark at preserving protection of the image.

TABLE II. SSIM OF EXTRACTED WATERMARKS

Processing	Boat	Chilli	House	Lena	Mandril
Contrast	0.5176	0.5174	0.5097	0.5971	0.5214
Brightness	0.6645	0.6490	0.6318	0.6303	0.6437
JPEG Q.20%	0.4775	0.5580	0.4605	0.5329	0.4468
JPEG Q.80%	0.6134	0.6167	0.5613	0.6182	0.5918
Adding noise	0.3729	0.3680	0.3481	0.3679	0.4041
Cropping 25%	0.4508	0.3774	0.3434	0.3863	0.3986
Rotate 2^o	0.2879	0.3094	0.2836	0.3202	0.2998
Rotate 10^o	0.2481	0.2606	0.2562	0.2605	0.2848
Rotate 90^o	0.2219	0.2293	0.2249	0.2350	0.2269
Horizontal flip	0.2412	0.2519	0.2324	0.2378	0.2657
Vertical flip	0.2372	0.2198	0.2364	0.2375	0.2369

To improve the capability of the watermark as a protection preserver, especially after geometric operations, the extraction mechanism can be undertaken in another way. Assume that the geometric operation that was used to modify the image can be identified, the inverse of the operation can be applied to revert the modified image back to its original. Therefore, instead of directly extracting from the modified image, the watermark could be extracted from the recovered one. This mechanism relies on the reverse function that is used to make the recovered image is as close as possible to the original. The closer the recovered image is to the original one, the higher the similarity of the extracted signal to the original watermark.

Apparently, the mechanism to recover an image depends on the operation that was used to modify the image. For example, a cropped image can be recovered by firstly identifying the cropping coordinates and then replacing the missing parts with the corresponding pixels of the watermarked image. A rotated image can be recovered by applying the same degree of rotation with the reverse direction. A flipped image can be reverted back to its origin by implementing the same kind of flipping. In addition to previous recovery mechanisms, an image that was modified by adding noise can be recovered by applying some filtering.

The recovery mechanisms significantly improve the quality of the extracted watermark signals. Table III shows that signals extracted from each recovered image have higher *SSIM* indexes. Images recovered from cropping, flipping and 90 degree rotation reveal perfect watermark signals (see Fig. 5f, 7 and 6f). However, the signal that is extracted from the image recovered from 2 degree rotation still has noise in it (Fig. 6c). This is because the recovered image is not as perfect as one that is recovered from the 90 degree rotation. We believe that this problem could be solved by modifying the computation. The recovery process is not undertaken just by applying the same degree rotation with the reverse direction, but also by finding the matching features between the rotated and the original watermarked images. A different story comes from the image recovered from adding noise. Though the recovery mechanism succeeds in removing noise, the recovered image is a bit blurry because of the filtering process. Thus, it cannot reveal a perfect watermark signal (Fig. 5c). Nevertheless, the recovery mechanisms can be used to improve the capability of the watermark at preserving image protection.

  ISSN: 2203-1731

TABLE III. SSIM OF EXTRACTED SIGNALS FROM RECOVERED IMAGE

Processing	Boat	Chilli	House	Lena	Mandril
Adding noise	0.4799	0.5788	0.4808	0.5741	0.4347
Cropping 25%	0.9355	0.9355	0.9355	0.9355	0.9355
Rotate 2^o	0.4739	0.4401	0.3911	0.44383	0.4754
Rotate 10^o	0.4133	0.3729	0.3479	0.3632	0.4161
Rotate 90^o	0.9355	0.9355	0.9355	0.9355	0.9355
Horizontal flip	0.9355	0.9355	0.9355	0.9355	0.9355
Vertical flip	0.9355	0.9355	0.9355	0.9355	0.9355

V. REMARKS

Chaotic maps can be utilized to achieve a perfect watermarking security because of their natural sensitivity to initial values. However, this characteristic is also a challenge for chaotic based watermarking scheme to achieve robustness.

With the aid of chaotic maps, we construct a blind image watermarking scheme in the frequency domain. The experimental results show that the watermarking is robust against image enhancement, compression, adding noise and cropping. However, rotating and flipping are still challenging. The signals extracted from the rotated and flipped images suffer a significant degradation. In this case, a watermark can be extracted from the recovered image to improve the capability of the watermark to preserve image protection. In any future work, an improvement is needed to make this scheme more robust to withstand such geometric operations.

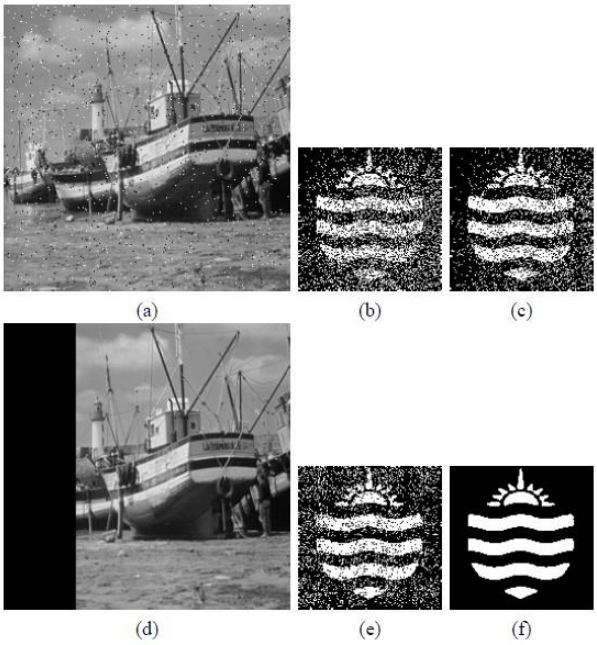

Fig. 5. The simulation results: (a) noisy image; (d) cropped image; (b)&(e) the corresponding direct extracted watermarks; (c)&(f) watermarks extracted from the corresponding recovered images.

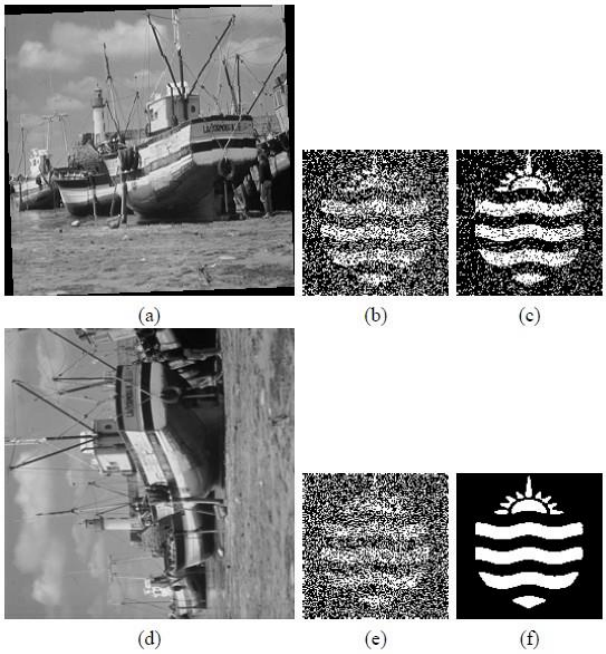

Fig. 6. The simulation results: (a) 2 degree rotated image; (d) 90 degree rotated image; (b)&(e) the corresponding direct extracted watermarks; (c)&(f) watermarks extracted from the corresponding recovered images.

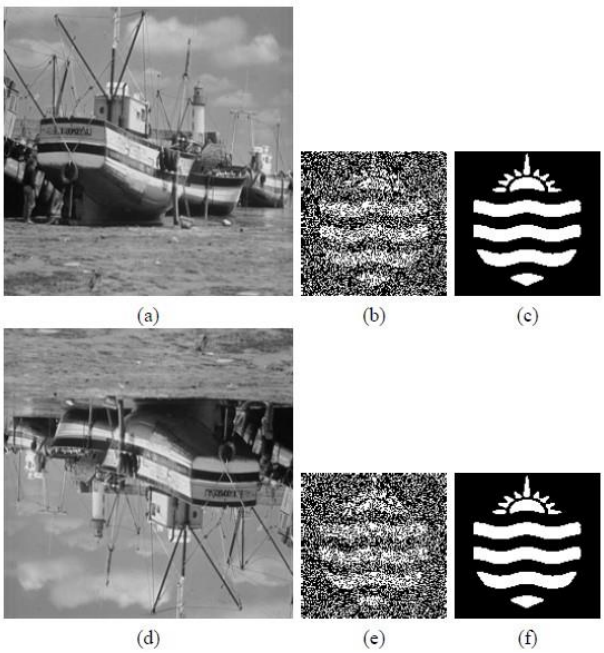

Fig. 7. The simulation results: (a) horizontally flipped image; (d) vertically flipped image; (b)&(e) the corresponding direct extracted watermarks; (c)&(f) watermarks extracted from the corresponding recovered images.

REFERENCES

[1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. Burlington, USA: Morgan Kaufmann Publisher, 2008.

  ISSN: 2203-1731

[2] S. E. I. Baba, L. Z. Krikor, T. Arif, and Z. Shaaban, "Watermarking of Digital Images in Freqeuncy Domain," *International Journal of Automation and Computing*, vol. 7, pp. 17--22, 2010.
[3] H. Nyeem, W. Boles, and C. Boyd, "On the Robustness and Security of Digital Image Watermarking," in *International Conference on Informatics and Vision*, Dhaka, Bangladesh, 2012.
[4] F. A. P. Petitcolas, "Components of DRM Systems: Digital Watermarking," in *Proceeding of Digital Rights Management: Technological, Economic, Legal and Political Aspect (LNCS 2770)*, 2003, pp. 81-92.
[5] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking," *IEEE Trans. on Image Processing*, vol. 6, pp. 1673--1687, 1997.
[6] Z. Dawei, C. Guanrong, and L. Wenbo, "A Chaos-based Robust Wavelet-domain Watermarking Algorithm," *Chaos, Solitons and Fractals*, vol. 22, pp. 47-54, 2004.
[7] R. Munir, B. Riyanto, S. Sutikno, and W. P. Agung. (2007). *Modifikasi Spread Spectrum Watermarking dari Cox Berbasiskan pada Enkripsi Chaotic*. Available: http://www.academia.edu/1357660/Modifikasi_Spread_Spectrum_Watermarking_dari_Cox_Berbasiskan_pada_Enkripsi_Chaotic
[8] J. Ayubi, S. Mohanna, F. Mohanna, and M. Rezaei, "A Chaos Based Blind Digital Image Watermarking in the Wavelet Transform Domain," *IJCSI - International Journal of Computer Science Issues*, vol. 8, pp. 192-199, 2011.
[9] H. Wang, C. He, and K. Ding, "Public Watermarking Based on Chaotic Map," *IEICE Trans. Fundamentals*, vol. E87-A, pp. 2045-2047, 2004.
[10] C. Guyeux and J. M. Bahi, "A New Chaos-based Watermarking Algorithm," presented at the International Conference on Security and Cryptography, 2010.
[11] M. Ghebleh, A. Kanso, and H. S. Own, "A Blind Chaos-based Watermarking Technique," *Security and Communication Networks*, vol. 2013, 2013.
[12] S. Mabtoul, E. Ibn-Elhaf, and D. Aboutajdine, "A Blind Chaos-based Complex Wavelet-domain Image Watermarking Technique," *IJCSNS - International Journal of Computer Science and Network Security*, vol. 6, pp. 134-139, 2006.
[13] R. Munir, B. Riyanto, S. Sutikno, and W. P. Agung, "Metode Blind Image Watermarking Berbasis Chaos dalam Ranah Discrete Cosine Transform (DCT)," presented at the National Conference on Computer Science and Information Technology 7, 2007.
[14] X. Wu, Z.-H. Guan, and Z. Wu, "A Chaos Based Robust Spatial Domain Watermarking Algorithm," presented at the ISNN 2007, Part II, 2007.
[15] X. Wu and Z.-H. Guan, "A Novel Digital Watermark Algorithm Based On Chaotic Maps," *Physics Letters A*, vol. 365, pp. 403-406, 2007.
[16] T. Kohda and K. Aihira, "Chaos in Discrete Systems and Diagnosis of Experimental Chaos," *IEICE Transactions*, vol. E 73, pp. 772-783, 1990.
[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600--612, 2004.

  ISSN: 2203-1731