A New Watermarking Method Based on Chaotic Maps

Jiashu Zhang and Lei Tian
Sichuan Province Key Lab of Signal & Information Processing, Southwest Jiaotong University
Chengdu, 610031, P. R. China

Heng-Ming Tai
Department of Electrical Engineering
University of Tulsa, Tulsa, OK 74104, USA
tai@utulsa.edu

Abstract
This paper presents a new digital watermarking method based on chaotic maps. Different from most of the existing chaotic watermarking methods, two chaotic maps are incorporated into the watermarking system to resolve the finite word length effect and to improve the system’s resistance to attacks. One map is used for watermark generation, another as the private key. Simulation results show that the proposed digital watermarking system is feasible and robust to common signal processing procedures. In addition, it performs better than existing watermarking schemes and exhibits better protection than watermarking systems using only one chaotic map.

1. INTRODUCTION
Digital watermarking of images is a technology with applications in copyright protection. It embeds certain information into digital content in a way not immediately discernible. Most of the existing watermarking techniques for still images are based on the construction of a pseudo-random sequence of real numbers that is threshold to provide a binary map as watermark [1,2]. Recently, chaotic systems have been used in watermarking schemes [3,4]. In [3], several chaotic maps are employed for watermark embedding and detection. Their methods generate watermarks that provide good correlation properties, and are robust against JPEG compression and lowpass filtering. Another chaotic watermarking scheme that embeds the chaotic sequence in the frequency domain was reported [4]. It shows good robustness to JPEG compression.

However, these chaotic watermarking methods are based on single chaotic system and still exhibit some drawbacks: (1) chaotic sequence does not have infinite period due to the finite word length effect and (2) they are not secure enough in that the watermark key used is fixed and can be easily figured out, and (3) although they are robust, but possibly can be predicted [7].

In this paper, we present a new watermarking method based on two chaotic maps. One chaotic sequence is used as the private key. The proposed approach is different from the methods reported in [3,4]. It provides the following features: (1) resistance to the finite word length effect of the chaotic sequence, (2) very unpredictable, (3) the robustness against attacks other than lowpass filtering and JPEG compression, and (4) resistance to repeated group attack. In addition, we examine whether or not the chaotic watermarking system may be more secure than other watermarking techniques by exploiting the chaotic features such as sensitivity to the initial condition and control parameters [5,6].

The rest of the paper is organized as follows. Section 2 describes the proposed chaotic technique in watermark generation and detection. Examples are shown in Section 3 to demonstrate the effectiveness of the proposed watermarking method. Conclusions are given in Section 4.

2. CHAOTIC WATERMARKING
A general watermarking technique consists of three steps: watermark generation, embedding, and detection. Watermark generation involves using a secret key and the host image to produce a binary watermark sequence. And the watermark sequence can be generated from successive iterations of a chaotic map applied to an initial condition [3,4]. The proposed chaotic watermarking system based on the secret key is shown in Fig.1. Fig. 1(a) depicts the watermark generation and embedding process, and Fig. 1(b) the detection process.

2.1 Chaotic Maps
A one-dimensional chaotic map is used to generate 1-D sequences of real numbers:

\[ x_{n+1} = f(x_n, \lambda) \]  

(1)

where \( n = 1, 2, \ldots \) is the map iteration index and \( \lambda \) a system parameter. Two chaotic maps are employed in our watermark generation and embedding. One is the Henon map [5]:

\[ x_n = 1 + b(x_{n-2} - x_{n-3}) + cx_{n-2}^2 \]  

(2)
where the parameter values are selected as $b = 0.3$ and $1.07 < c < 1.09$. The initial values are chosen in the range of (-1.5, 1.5) to make sure that the chaotic system contains chaotic attractors. Another chaotic sequence used is the logistic map:

$$y_{n+1} = g(y_n) = \lambda y_n (1 - y_n)$$  \hspace{1cm} (3)

For $3.57 < \lambda < 4$, the sequence is non-periodic, non-convergent, and very sensitive to the initial value. The symmetric chaos attractors of these two chaotic maps are shown in Fig 2. It can be seen from Fig 2 that the two chaos attractors are fixed trajectories whether the iteration index is short or long. These trajectories have been successfully predicted [7].

### 2.2 Watermark generation and embedding

In this paper, the watermark generation model is described by

$$x_{n+1} = f(x_n, y_n, s_n)$$  \hspace{1cm} (4)

where $y_n$ is used as the secret key and $s(n)$ represents the mid-frequency block-DCT coefficients. To prevent the chaotic sequence $f$ from divergence, the modulo operation is applied to $x_n$ to confine the value of the watermark sequence to the range of $(0, a)$ via

$$x_{n+1} = [f(x_n, y_n) + s(n)](\text{mod} \ a)$$  \hspace{1cm} (5)

Based on equation (1),

$$x_{n+1} = [1 + b(x_{n-1} - c) + 379y_n^2](\text{mod} \ 3)$$  \hspace{1cm} (6)

where the initial values of $x_n$ are chosen from the range of $(0, 3)$. Then $x_n$ is normalized to a binary watermark sequence $w_n \in \{-1, 1\}$ and embedded in the image. Fig. 3(a) depicts the symmetric chaos attractor of the generated watermark sequence using the above chaotic maps. It can be seen that the watermark is uniformly distributed so that it is very difficult to be predicted and
detected. The length of the watermark sequence is adapted to the characteristics of the original image. Fig 3(b) is the autocorrelation test of the generated watermark. Note that the watermark sequence has infinite period.

The tradeoff between watermark strength and invisibility can be carried out by the following procedure [6]. First, the original image is partitioned into square blocks of size 8×8, and these blocks are divided into two categories: $S_1$ with weak texture and $S_2$ with strong texture, based on the edge information of each block. Each category is embedded with watermark of the corresponding strength; then apply DCT to each block and embed the watermark into the mid-frequency DCT coefficients according to (7), where $a_1$ and $a_2$ denote the watermark strength of each category. Finally, the watermarked image is obtained by performing IDCT on each 8×8 block.

$$Block^*_w = \begin{cases} 
(1 + a_1x_v)\times Block^*_w(u,v), & Block^*_w \in S_1 \\
(1 + a_2x_v)\times Block^*_w(u,v), & Block^*_w \in S_2 \\
Block^*_w(u,v), & \text{otherwise}
\end{cases} \quad (7)$$

### 2.3 Watermark detection

Watermark detection is generally formulated by the hypothesis testing [2],

$$H_0: I' = I + N$$

$$H_1: I' = I + W' + N$$

$H_0$ indicates that the detected image does not contain watermark, while $H_1$ indicates the opposite. $I$ and $I'$ represent the original and to-be-tested images, respectively. $W'$ is the extracted watermark, and $N$ the Gaussian noise. Correct hypothesis can be obtained by use of the correlation measure between $W'$ and the original watermark $W$. The watermark is extracted based on the procedure of [6], but our scheme does not need the original image. It can be carried out by computing the block-DCT on $I'$ and apply the same chaotic map at the embedding side. The watermark detection procedure is described below:

Step 1: $I'_w(u,v) = DCT[I'_w(x,y)]$

Step 2: $x'_w = I'_w(u,v)(\text{mod } 3), 0 \leq u, v \leq 8$

Step 3: $w'_w = \{x'_w | x'_w = (1+b(x'_{u-1} - c) + 379)y^2_{x_{u-1}}(\text{mod } 3)\}$

Step 4: $W' = \sum_{n=1}^{M} w'_w$

In step 2, $n = 1,2,\ldots,M$ and $M$ is the watermark sequence length.

To determine whether $W'$ and $W$ match, we employ the following detector response (DR) measure,

$$\rho(W,W') = \frac{\sum_{n=1}^{M} w_n w'_n}{\sqrt{\sum_{n=1}^{M} w_n^2} \sqrt{\sum_{n=1}^{M} w'_n^2}} \quad (8)$$

If $\rho(W,W') > T$, where $T$ is a pre-defined threshold, then the detected image contains a watermark. According to [1], the detection threshold can be set according to the tradeoff between the probability of false alarm and probability of missed detection. When $T$ decreases, the miss-detection probability decreases and the false-alarm probability increases, and vice verse.

### 3. EXPERIMENTAL RESULTS

The characteristics and effectiveness of the proposed chaotic watermarking system are demonstrated by a gray-level image (size 256×256) as shown in Fig. 4. Watermark parameters used in the simulation are $a_1 = 0.008$, $a_2 = 0.02$, and $T = 0.2$. The embedded watermark is imperceptible (Fig. 4(b)). The robustness of the proposed method is evaluated by subjecting the watermarked image to a series of signal processing attacks. Table 1 shows the detector response and PSNR of the watermarked image under various attacks. It can be seen from Table 1 that the values of DR under various attacks are all larger than the threshold, even when the PSNR is as low as 10 dB for the geometric distortion. In addition, DR values of about 0.55 against rotation and cropping (Fig. 4(d) and 4(e)) are better than that of 0.3 by the RST-invariant watermarking scheme reported in [8]. Fig. 5 shows the robustness of the watermark generated by the proposed method and Huang’s algorithm [6] against additive Gaussian noise corruption. The results demonstrate that the proposed method performs well on watermark detection under severe conditions.

The proposed watermarking method also exhibits better protection than those with only one chaotic map. In our scheme, the sequence generated by equation (6) is not sensitive to the initial value. Hence it cannot be used as the private key. Fig. 6 shows the effect of different initial value of chaotic maps on the performance; def represents the deviation from the initial value. The detector response and bit error rate (BER) of extracted watermarks are nearly not affected even when the initial value of the input $x_0$ varies a lot ($|x_0 - x_0| < 2$). On the other hand, because of low detector response ($\rho < 0.2$) and high BER, the watermark cannot be detected even for a tiny change ($|y_0 - y_0| \approx 10^{-17}$) of the initial key ($y$). This implies that that the proposed digital watermarking system is more secure and can provide better protection.
4. CONCLUSIONS

We have presented a new digital watermarking system using two chaotic maps. One chaotic map is used as a private key and another one for watermark generation. Results show that the proposed watermark is robust to more common signal procedures and more secure than the watermarking scheme using only one chaotic map.

ACKNOWLEDGMENT

This work is supported in part by Sichuan Youth Science and Technology Foundation (Grant No. 03ZQ026-33)

Fig. 4. (a) Original Lena image. (b) Watermarked image. (c) Corrupted by noise, PSNR = 19.62. (d) Cropped watermarked image. (e) Rotated watermarked image. (f) Median filtered, PSNR = 28.52.

Table I

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Detector Response</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No attack</td>
<td>1.00</td>
<td>55.62</td>
</tr>
<tr>
<td>JPEG (10:1)</td>
<td>0.91</td>
<td>29.12</td>
</tr>
<tr>
<td>Median filter (5×5)</td>
<td>0.98</td>
<td>28.52</td>
</tr>
<tr>
<td>Wiener filter (6×6)</td>
<td>0.96</td>
<td>30.15</td>
</tr>
<tr>
<td>Resizing (50%)</td>
<td>0.95</td>
<td>25.57</td>
</tr>
<tr>
<td>Gaussian noise (0.01,0)</td>
<td>0.93</td>
<td>19.62</td>
</tr>
<tr>
<td>16 gray level equalizer</td>
<td>0.88</td>
<td>17.07</td>
</tr>
<tr>
<td>Cropping</td>
<td>0.52</td>
<td>11.34</td>
</tr>
<tr>
<td>Rotation (45°)</td>
<td>0.57</td>
<td>10.38</td>
</tr>
</tbody>
</table>

Fig. 5. Detector response to watermarked image corrupted by Gaussian noise versus image strength. With watermark: - - - - proposed, - - - - Huang [6]; no watermark: - - - - proposed, - - - - Huang [6].

Fig. 6. Effects of different initial values, \( def = |x_0' - y_0'| \) or \( |x_0' - x_0| \) (- o -): DR vs. \( def_{x0} \); + + + +: DR vs. \( def_{y0} \); - - - -: BER vs. \( def_{x0} \); * : BER vs. \( def_{y0} \).

References


