

Medical image steganographic algorithm via modified LSB method and chaotic map

A. A. Karawia

Abstract

Many methods of hiding information in an image are existing now. The least significant bit is the famous method used in steganographic algorithms. Medical image steganography is a technique used to make the transmission of these images secure so that the decision of the Specialist physician based on these images is not affected. In this paper, medical image steganographic algorithm using modified least significant bit and chaotic map is proposed. The main problem is that the selection of embedding pixels within the host image is not protected enough in most existing methods. So, the author used two-dimensional piece-wise smooth chaotic map to select the positions of these pixels randomly. On the other hand, all bits in the secret medical image are transmitted without losing any bit. To do that, the secret medical image is encrypted using one-dimensional piecewise chaotic map (Tent map). Then, the steganographic algorithm is used to hide the bits of the encrypted secret medical image. The bits of each embedded pixel are shuffled before the embedding process randomly. After that, the stego image is created. The host image and stego image are analyzed with the peak signal-to-noise ratio, the mean square error, histogram test, image quality measure and relative entropy test. The stego image displays acceptable result when comparing with the host image. Also, the chi-square attack test is performed and the stego image can resist it. The proposed algorithm can assist the sending of medical images via communication media.

1 INTRODUCTION

Digital image transmission via open networks is one of the important research problems. Medical image is one of the famous digital images. Due to their sensitivity, the medical images must be transmitted by networks securely. Any change in the medical image during the transmission will affect the accuracy of the doctor’s diagnosis. So the problem here is how to send the image safely and without change.

Steganography algorithms are used to hide the secret information within the host image. Then the host image is sent to the receiver. Finally, the secret information can be retrieved from the host image. The hiding data using watermarking is the traditional method where the secret information is embedded into host image [1]. In this case, the host image is called stego image. The host image used two styles for hiding the secret information, frequency domain and spatial domain [2]. In the first one, the pixels of secret information (image) are hidden in the coefficient of the pixel’s values in the host image [3, 4]. In the second one, the secret information is hidden in intensity values of host image [5, 6]. Many algorithms of spatial domain steganography have been presented based on watermarking [7], LSB substitution [8], LSB matching [9], chaos theory and random distortion [10], chaos [11]. In LSB, the LSB set of pixels in the host image are replaced by the secret image’s bits in random or in sequential method. In terms of security, the random method is better than the sequential method [12]. In the current proposed algorithm, the pixels of the host image are chosen randomly via the sequence created by the chaotic map. Chaos implies a condition of turmoil. A dynamical system of discrete type is called map. The main essential features of the chaotic map are sensitive to the initial conditions, ergodicity, and structure complexity. Much information hiding algorithms has been proposed using a chaotic sequence [13]. Medical image steganography has been
receiving considerable attention [14, 15]. The proposed algorithm is developed based on three main steps. The first step is encrypting the secret image via one dimensional-piecewise chaotic map (Tent map). The second is determining the pixel positions of the host image randomly, by the two-dimensional piecewise smooth chaotic map. The third step is embedding the bits of the encrypted medical secret image in the colour channel of the selected pixels of the host image randomly.

The remaining sections of the current paper are organised as follows: the piecewise chaotic maps and their features are given in Section 2. In Section 3, the proposed algorithm is described. The experimental results are presented in Section 4. Finally, the conclusion is presented in Section 5.

2 | THE PIECEWISE CHAOTIC MAPS

In this section, two piecewise chaotic maps are given to encrypt the secret medical image and to select the pixels of the host image randomly.

2.1 | One dimensional-piecewise chaotic map (Tent map)

The Tent map can be described piecewise by

\[ x_{n+1} = \begin{cases} \mu x_n & \text{if } 0 \leq x < \frac{1}{2} \\ \mu(1 - x_n) & \text{if } \frac{1}{2} \leq x \leq 1 \end{cases} \]

(1)

where \( n \geq 0 \) and the control parameter \( \mu \in [0, 2] \). Depending on \( \mu \), the dynamical system behaviours of system (1) going from predictable to chaotic. Figure 1 displays the bifurcation diagram of the Tent chaotic map with respect to the \( \mu \) parameter. Lyapunov exponent of the Tent chaotic map with respect to the \( \mu \) parameter shows in Figure 2. For more details, see for example [16].

2.2 | Two-dimensional piecewise smooth chaotic map (2DPSCM)

Recently, the authors in [17] described the problem of remanufacturing duopoly game using 2DPSCM. It takes the following form:

\[ x_{n+1} = x_n + k_1 x_n [1 - 2(1 + \epsilon_1) x_n - \theta y], \]
\[ y_{n+1} = \begin{cases} \gamma_1 + k_2 y_n [\theta (1 - x_n - 2 y_n) - 2 \epsilon_2 y_n] & \text{if } x_n \geq f(y_n), \\ x_n & \text{if } x_n < f(y_n), \end{cases} \]

(2)

where \( f(y_n) = y_n [1 + \theta x_n - 2 \epsilon_2 (y_n + \theta y_n)] / (1 + \theta x_n) \), \( x_n \) and \( y_n \) reflect the amount of production received in the market by Company 1 and Company 2. Equation (1) displays 2DPSCM’s chaotic behaviour with seven separate parameters. In the economy, these parameters are significant. The shift cost parameters are \( \epsilon_1 \) and \( \epsilon_2 \) such that \( \epsilon_1 > \epsilon_2 \). The parameter \( \theta \) denotes to a fraction where customers will pay for the purchase of new remanufactured goods such that \( \epsilon_2 < \theta < 1 \). The functions \( k_1 x_n \) and \( k_2 y_n \) are used to capture the speed of a company’s quantity adjustment with the change that can occur in the marginal profit of a company. The chaotic behaviour of 2DPSCM is observed by the values of the parameters: \( x(0) = 0.19, y(0) = 0.15, \epsilon_1 = 0.55, \epsilon_2 = 0.3, \theta = 0.73, k_1 = 2.1 \) and \( k_2 = 8.4 \). Figure 3 displays the bifurcation diagram of 2DPSCM with respect to the \( k_1 \) parameter. Lyapunov exponent of a 2DPSCM with respect to the \( k_1 \) parameter shows in Figure 4.

3 | PROPOSED WORK

The digital image is represented by RGB model which can use triplets in a 24-bit digital image corresponding to red, green, and blue intensities. A triple of values associated with each colour can be represented for each pixel of the image. All these triplets consist of an array corresponding to the digital image. The image is represented by an array of gray-scale values in the case of 8-bit images. The most noticeable and also the most well-known technique for hiding information in images is the LSB insertion process. Changing the least significant bit is expected to cause a hardly noticeable image alteration.
3.1 Cryptography

Cryptography [18] is maybe the first and greatest common technique to protect the secret images. This technique modifies the secret image from original to cipher image which it is shuffled and hard to read. So, the encryption stage should save secrecy and avoid to drive any information of the secret image from the cipher image.

3.1.1 The secret key of the cryptography stage

Assume the secret medical image is denoted by \( S = (s_{ij}) \), \( i = 1, 2, ..., m \), \( j = 1, 2, ..., n \). The key mixing proportion factor can be used to compute \( K_1 \) as follows [19]:

\[
K_1 = \frac{1}{256} \mod \left( \sum_{i=1}^{m} \sum_{j=1}^{n} s_{ij} \times 256 \right).
\]

3.1.2 The encryption algorithm

Now, the encryption algorithm via the Tent chaotic map is given in Algorithm 1.

For the decryption algorithm, use the inverse steps of Algorithm 1 to restore the secret medical image.

3.2 Steganography

In most methods used in steganography, we lost some low significant bits but in medical image it is not acceptable. In the current proposed algorithm, two pixels from the host image

\[
\begin{align*}
\text{ALGORITHM 1} & \quad \text{The encryption algorithm} \\
\text{Input:} & \quad \text{Secret medical image } S \text{ with size } m \times n, \text{ and initial value } x_0 \text{ of the Tent map.} \\
\text{Output:} & \quad \text{Cipher image } C \text{ with size } m \times n. \\
\text{Step 1:} & \quad \text{Generate the secret key using Equations (3) and (4).} \\
\text{Step 2:} & \quad \text{Convert } S \text{ from decimal to binary, } B = \{b_1, b_2, ..., b_m\}. \\
\text{Step 3:} & \quad \text{For } i = 0 : 999, \text{ compute} \\
& \quad \text{if } (0 \leq x_i < \frac{1}{2}) \text{ then } x_{i+1} = \mu \times x_i \text{ else } x_{i+1} = \mu \times (1 - x_i) \\
\text{end} \\
\text{Step 4:} & \quad \text{Set } x_0 = x_{1000}. \\
\text{Step 5:} & \quad \text{For } i = 0 : mn - 1, \text{ compute} \\
& \quad \text{if } (0 \leq x_i < \frac{1}{2}) \text{ then } x_{i+1} = \mu \times x_i \text{ else } x_{i+1} = \mu \times (1 - x_i) \\
\text{end} \\
\text{Step 6:} & \quad \text{Preprocessing; For } i = 1 : mn, \text{ compute} \\
& \quad x_i = \text{floor}(\text{mod}(\text{abs}(x_i) - \text{floor}(\text{abs}(x_i)) \times 10^{14}, 256)). \text{ end} \\
\text{Step 7:} & \quad \text{Convert the } x_i \text{ values from decimal to binary, } Z = \{z_1, z_2, ..., z_{mn}\}. \\
\text{Step 8:} & \quad e = \text{XOR}(B, Z). \\
\text{Step 9:} & \quad \text{Convert } e \text{ from binary to decimal, } E. \\
\text{Step 10:} & \quad \text{Reshape } E \text{ to two-dimensional array of size } m \times n, \text{ say } C. \\
\text{Step 11:} & \quad C \text{ is the cipher image}.
\end{align*}
\]

Then the initial condition \( x_0 \) and the parameter of the Tent map are modified via the following formulas:

\[
\begin{align*}
\left\{ \begin{array}{l}
x_0 = \frac{(x_0 + K_1)}{2}, \\
\mu = 1.8 + 0.2 \times K_1
\end{array} \right. \\
\end{align*}
\]
are needed to hide one pixel from the cipher image (encrypted secret medical image). Those pixels are selected randomly using 2DPSCM. LSB embedding is modified by shuffling the 8-bits of each pixel in the cipher image randomly. This modification is shown in Figure 5. Suppose \( X \) and \( Y \) are two random pixels from the host image and the arrows indicate the new positions of the random bits of the cipher image’s pixel. In [20], the authors decided that the R and G channels are more sensitive to embedding bits and make more distortion in stego image than the B channel. So, two bits are embedded in each B channel of \( X \) and \( Y \).

3.2.1 The secret key of the steganography stage

Assume the secret medical image is denoted by \( C = (c_{ij}), i = 1, 2, ..., m, j = 1, 2, ..., n \). The key mixing proportion factor can be used to compute \( K_2 \) and \( K_3 \) as follows:

\[
K_2 = \frac{1}{256} \mod \left( \sum_{i=1}^{m/2} \sum_{j=1}^{n/2} c_{ij}, 256 \right),
\]

\[
K_3 = \frac{1}{256} \mod \left( \sum_{i=m/2+1}^{m} \sum_{j=n/2+1}^{n} c_{ij}, 256 \right). \tag{5}
\]

Then the initial condition \( x(0) \) and \( y(0) \) of 2DPSCM are modified according to the following formulas:

\[
\begin{align*}
x(0) &= \frac{\sum_{i=1}^{m/2} c_{ij} + K_2}{2}, \\
y(0) &= \frac{\sum_{i=m/2+1}^{m} c_{ij} + K_3}{2}. \tag{6}
\end{align*}
\]

Also, the parameters of 2DPSCM can be calculated by the following formulas:

\[
\begin{align*}
c_1 &= 0.5 + 0.05 \times K_2, \\
c_2 &= 0.29 + 0.01 \times K_3, \\
\theta &= 0.7 + 0.01 \times K_3, \\
K_1 &= 2.0 + 0.01 \times K_3, \\
K_2 &= 8.0 + 0.4 \times K_3. \tag{7}
\end{align*}
\]

3.2.2 Modified LSB embedding algorithm

In the current proposed algorithm, two pixels from the host image are needed to hide one pixel from the cipher image (encrypted secret medical image). Those pixels are selected randomly using 2DPSCM. LSB embedding is modified by shuffling the 8-bits of each pixel in the cipher image randomly. This modification is shown in Figure 5. Suppose \( X \) and \( Y \) are two random pixels from the host image and the arrows indicate the new positions of the random bits of the cipher image’s pixel. In [20], the authors decided that the R and G channels are more sensitive to embedding bits and make more distortion in stego image than the B channel. So, two bits are embedded in each B channel of \( X \) and \( Y \). The proposed embedding algorithm based on modified LSB and 2DPSCM may be given as: where the Matlab function bitset(A,bit,V) returns A with position bit set to the value of V.

3.2.3 Modified LSB extracting algorithm

In this stage, the embedded secret medical image \( S \) can be extracted from the given stego image \( O \) without referring to the original host image \( H \). The set of pixels that store the cipher image bits in the stego image is chosen using the same sequence as in the modified embedding algorithm. Extract and unshuffle the LSBs of the chosen pixels to reconstruct the cipher image. Then, the cipher image is decrypted using the inverse of Algorithm 1 to get the secret medical image \( S \). The proposed modified extracting algorithm based on modified LSB and 2DPSCM may be given as: where the Matlab function bitand(A,B) returns the bit-wise AND of \( A \) and \( B \).

The proposed steganographic algorithm embeds the bits of the secret medical image according to the modified least significant bit technique as shown in Figure 6.

4 EXPERIMENTAL RESULTS

In the current section, the result of our proposed algorithm will be investigated. All programs are designed using Matlab R2016b and the features of Matebook are Intel(R) Core(TM) m5-6Y54 CPU @ 1.51 GHz and 8 GB RAM. The stego-key of our proposed algorithm is \( x_0 = 0.0001, \mu = 1.9999, x(0) = 0.19, y(0) = 0.15 \).
**Algorithm 2**  Modified embedding algorithm

**Input:** Secret medical image S with size $m \times n$, host image H with size $2m \times 2n \times 3$, and the initial values of 2DPSCM, $x(0)$ and $y(0)$.

**Output:** Stego image O with size $2m \times 2n \times 3$.

**Step 1:** Reshape H to two-dimensional array, say h with size $4mn \times 3$.

**Step 2:** Encrypt the secret medical image S using Algorithm 1 to get the cipher image C.

**Step 3:** Generate the secret key using Equations (5)–(7).

**Step 4:** Reshape C to two-dimensional array, say msg with size $mn \times 1$.

**Step 5:** Generate two different random vectors of size $mn \times 1$ using 2D-PSGM, say X and Y such that for all $i \in X, Y$ then $i$ (\& 1, 2, 3, ..., 4mn).

**Step 6:** Convert msg from decimal to binary, say out.

**Step 7:** Shuffle each pixel in out, say the new positions of the extracting bits

\[ P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\} \]

**Step 8:** for $j = 1$ to $mn$ do

\[ h(X(j),1) = \text{bitand}(h(X(j),1), \text{out}(p_1)); \]
\[ h(Y(j),1) = \text{bitand}(h(Y(j),1), \text{out}(p_2)); \]
\[ h(X(j),2) = \text{bitand}(h(X(j),2), \text{out}(p_3)); \]
\[ h(Y(j),2) = \text{bitand}(h(Y(j),2), \text{out}(p_4)); \]
\[ h(X(j),3) = \text{bitand}(h(X(j),3), \text{out}(p_5)); \]
\[ h(Y(j),3) = \text{bitand}(h(Y(j),3), \text{out}(p_6)); \]
\[ h(Y(j),3) = \text{bitand}(h(Y(j),3), \text{out}(p_7)); \]
\[ h(Y(j),3) = \text{bitand}(h(Y(j),3), \text{out}(p_8)); \]

End

**Step 9:** Reshape h to three-dimensional array of size $2m \times 2n \times 3$, say O.

**Step 10:** O is the stego image.

\[ \theta = 0.73, c_1 = 0.55, c_2 = 0.3, k_1 = 2.1, k_2 = 8.4, m = 256, \]
and $n = 256$. Figure 7 shows four images which are used as host images. Figure 8 shows the host image (baboon), the secret medical image, the stego-image and the retrieved secret medical image.

**Algorithm 3**  Modified extracting algorithm

**Input:** Stego image O with size $2m \times 2n$.

**Output:** Secret medical image S with size $m \times n$.

**Step 1:** Reshape O to two-dimensional array, say steg with size $4mn \times 3$.

**Step 2:** Declare an one-dimensional array of size 8, say W, to store the extracting LSBs for each pixel.

**Step 3:** Generate two random vectors of size $mn \times 1$ using 2D-PSGM, say X and Y such that for all $i \in X, Y$ then $i$ (\& 1, 2, 3, ..., 4mn).

**Step 4:** Generate the shuffled positions of the extracting bits

\[ P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\} \]

**Step 5:** for $j = 1$ to $mn$ do

\[ W(p_1) = \text{bitand}(\text{steg}(X(j),1), 1); \]
\[ W(p_2) = \text{bitand}(\text{steg}(Y(j),1), 1); \]
\[ W(p_3) = \text{bitand}(\text{steg}(X(j),2), 1); \]
\[ W(p_4) = \text{bitand}(\text{steg}(Y(j),2), 1); \]
\[ W(p_5) = \text{bitand}(\text{steg}(X(j),3), 1); \]
\[ W(p_6) = \text{bitand}(\text{steg}(Y(j),3), 1); \]
\[ \text{sbx} = \text{dec2bin}(\text{steg}(X(j),3), 8); \]
\[ W(p_7) = \text{bitand}(\text{str2double}(\text{sbx}(7)), 1); \]
\[ W(p_8) = \text{bitand}(\text{steg}(Y(j),3), 1); \]
\[ \text{sbx} = \text{dec2bin}(\text{steg}(Y(j),3), 8); \]
\[ W(p_9) = \text{bitand}(\text{str2double}(\text{sbx}(7)), 1); \]
\[ \text{msg}(i) = \text{W}(8) + \text{W}(7) + 2 \times \text{W}(6) + 4 \times \text{W}(5) + 8 + \text{W}(4) + 16 \]
\[ + W(3) + 32 + W(2) + 64 + W(1) + 128; \]

End

**Step 6:** Reshape msg to two-dimensional array, say C with size $m \times n$.

**Step 7:** C is the cipher image.

**Step 8:** Decrypt C using the inverse of Algorithm 1.

**Step 9:** The result of Step 8 is the secret medical image, S.

4.1  Statistical analysis

The efficiency of the proposed algorithm was investigated using various types of statistical tests.
4.1.1 Histogram test

An image histogram displays a graph of the number of pixels at each different value of intensity contained in that image. So, it is considered a statistical attack. This attack permits human eye to discover the difference between the host and stego images if there is an image inserted in the channels of the host image. Figure 9 shows four samples of secret medical images. The average histograms of the host image and stego images with the four tested images are displayed in Figures 10–12. By the human eye, it may be noted that the average histograms of the stego images are almost identical to that of the host images.

4.1.2 PSNR test

The distortion degree of a stego image regarding the host image plays an essential role. It is usually evaluated by the peak signal-to-noise ratio (PSNR) that is given by

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \quad (8)$$

where

$$\text{MSE} = \frac{1}{3MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} (H(i, j, k) - O(i, j, k))^2 \quad (9)$$

$H$ and $O$ are the host and stego images, respectively.

Table 1 shows MSE and PSNR of different host images with the same secret medical image. Furthermore, Table 2 presents a comparison result between the proposed algorithm, algorithm [21] and algorithm [22]. In this comparison, lena (512 × 512) is

| Host image     | Secret image | MSE  | PSNR (dB) |
|----------------|--------------|------|-----------|
| Baboon (512 × 512) | Image_1     | 0.5856 | 50.4545   |
|                | Image_2     | 0.5846 | 50.4619   |
|                | Image_3     | 0.5830 | 50.4739   |
|                | Image_4     | 0.5853 | 50.4567   |
| Lena (512 × 512)  | Image_1     | 0.5397 | 50.8091   |
|                | Image_2     | 0.5363 | 50.8370   |
|                | Image_3     | 0.5394 | 50.8118   |
|                | Image_4     | 0.5408 | 50.8007   |
| Airplane (512 × 512) | Image_1    | 0.5904 | 50.4196   |
|                | Image_2     | 0.5884 | 50.4339   |
|                | Image_3     | 0.5883 | 50.4347   |
|                | Image_4     | 0.5861 | 50.4510   |
| Peppers (512 × 512) | Image_1    | 0.5968 | 50.3725   |
|                 | Image_2     | 0.5938 | 50.3947   |
|                 | Image_3     | 0.5952 | 50.3843   |
|                 | Image_4     | 0.5957 | 50.3804   |

Table 1 | PSNR between different host images and stego images with different secret medical images
FIGURE 9  Secret medical images

FIGURE 10  Histograms of host and stego images: (a) Host image, (b) the histogram of host image, (c) stego image with secret image_1, (d) the histogram of Stego_1

FIGURE 11  Histograms of stego images: (a) stego image with secret image_4, (b) the histogram of Stego_4, (c) stego image with secret image_3, and (d) the histogram of Stego_3

TABLE 2  PSNR between the proposed algorithm, algorithm [21] and algorithm [22]

| Algorithm         | MSE   | PSNR(dB) |
|-------------------|-------|----------|
| Proposed algorithm| 0.5397| 50.8091  |
| [21]              | 10.5458| 37.9000  |
| [22]              | 12.1082| 37.3000  |

4.1.3  Embedding payload (capacity)

The percentage of the embedded hidden bits in the entire pixels of the cover image is represented by ER, which stands for embedding rate. It is given by the following formula:

\[
ER = \frac{T}{M \times N} \text{ pbb} \tag{10}
\]

where the total number of embedded secret bits is \( T \), and \( M \times N \) is the size of the host image. A large value of ER, according to the embedding payload evaluation, indicates that
the steganographic algorithm performs better in terms of the embedding payload, i.e. a host pixel in the host image will hold more hidden bits. Table 3 presents the embedding rate of host images and secret images.

### 4.1.4 Image quality measures (IQMs)

The human visual system and the display have a relationship can be measured by mathematical equations of IQMs. Eventually, steganographic algorithms leave statistical indicators which can be used to quantify the stego image’s hidden content relative to the host image [24]. After inserting the secret image, the measurements calculate the similarity between the host image and the stego image, summing up all bytes and computed by [25]:

\[
\begin{align*}
\text{CQ} &= \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} H(i, j, k) O(i, j, k)}{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} H^2(i, j, k)}, \\
\text{SC} &= \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} O^2(i, j, k)}{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} H^2(i, j, k)}, \\
\text{IF} &= 1 - \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} (H(i, j, k) - O(i, j, k))^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} H^2(i, j, k)}, \\
\text{AD} &= \frac{1}{3MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} |H(i, j, k) - O(i, j, k)|.
\end{align*}
\]

(11)

where CQ, SC, IF, and AD denote the correlation quality, the structure content, the image fidelity and the average absolute difference respectively. If these measurements are close to one then the high level of similarity is obtained. Moreover, if AD is close to zero then the global distortion of the stego image related to the host image is very low. Table 4 shows that CQ, SC, and IF are close to 1 and AD are close to zero. So, we can conclude that no significant differences between the host image and the stego image are detected. Moreover, the comparison between the proposed algorithm and algorithm [25] is given in Table 5. It shows that the proposed algorithm gives result better than algorithm [25].
4.1.5 Relative entropy (RE)

One of the important measurements is relative entropy which is defined by [26]:

$$\text{RE}(P_H||P_O) = \sum P_H \log \left( \frac{P_H}{P_O} \right),$$

(12)

where $P_H$ and $P_O$ denote the distribution of host and stego image, respectively. If the relative entropy between host image and stego image is close to zero then the stego image is similar to the host image [26]. Table 6 shows the relative entropy between different host images and stego images with different secret medical images. Based on the result of Table 6, the relative entropies are close to zero approximately. The comparison between the proposed algorithm and the algorithm in [25] is given in Table 7. The proposed algorithm gives results better than algorithm in [25]. From Tables 6 and 7, we can conclude that the host image and the stego image are similar approximately.

### Table 5: Image quality comparison between the proposed algorithm and algorithm [25] with a baboon (512×512) as a host image and a Lena (256×256) as secret image

| Host image | Secret image | CQ | SC | IF | AD |
|------------|--------------|----|----|----|----|
| Algorithm  | Algorithm [25] |
| Baboon     | Lena         | 1.0000 | 1.0000 | 1.0000 | 0.3754 |
| Lena       | Lena         | -8.22×10^{-6} | - | -5.9×10^{-5} | 0.0067 |

### Table 6: Relative entropy between different host images and stego images with different secret medical images

| Host image | Secret image | Red | Green | Blue | Gray |
|------------|--------------|-----|-------|------|------|
| Baboon     | Image_1      | 0.00051 | 0.00028 | 0.00110 | 0.00018 |
|            | Image_2      | 0.00065 | 0.00027 | 0.00110 | 0.00017 |
|            | Image_3      | 0.00047 | 0.00025 | 0.00110 | 0.00015 |
|            | Image_4      | 0.00052 | 0.00024 | 0.00100 | 0.00018 |
| Lena       | Lena         | 0.00054 | 0.00029 | 0.00100 | 0.00017 |
| Airplane   | Image_1      | 0.29030 | 0.28760 | 0.47580 | 0.08260 |
|            | Image_2      | 0.28810 | 0.28810 | 0.47330 | 0.08300 |
|            | Image_3      | 0.28970 | 0.28810 | 0.47450 | 0.08330 |
|            | Image_4      | 0.28660 | 0.28810 | 0.47580 | 0.08260 |
| Airplane   | Lena         | 0.00037 | 0.00013 | 0.00013 | 0.00013 |
| Peppers    | Image_1      | 0.00042 | 0.00088 | 0.01390 | 0.00021 |
|            | Image_2      | 0.00038 | 0.00089 | 0.01400 | 0.00029 |
|            | Image_3      | 0.00043 | 0.00087 | 0.01410 | 0.00027 |
|            | Image_4      | 0.00043 | 0.00091 | 0.01410 | 0.00029 |
| Peppers    | Lena         | 0.00037 | 0.00009 | 0.00140 | 0.00024 |

### Table 7: Relative entropy: Comparison between the proposed algorithm and algorithm in [25]

| Host image | Secret image | Proposed algorithm | Algorithm [25] |
|------------|--------------|--------------------|----------------|
| Airplane   | Lena         | 0.00034            | 0.002249       |
| (512×512)  | (256×256)    |                    |                |
| Lena       | Lena         | 0.00024            | 0.004113       |
| (256×256)  | (256×256)    |                    |                |

4.1.6 Chi-square attack

It is one of the basic tests which aim to detect whether the sent image transmits (or not) any unseen image (secret image) [27]. In this test, we calculate the probability of embedding information as the following:

$$\rho = 1 - \frac{1}{2} \sum_{\frac{k-1}{2}}^{k-1} \frac{1}{\Gamma \left( \frac{k-1}{2} \right)} \int_0^{\chi^2_{k-1}} e^{-\frac{t}{2}} \frac{t^{\frac{k-1}{2}-1}}{2^{\frac{k-1}{2}}} dt$$

(13)

where $\chi^2_{k-1} = \sum_{i=0}^{k-1} \frac{(X_i - \mu)^2}{\sigma^2}$ is the $\chi^2$ static with $k-1$ degrees of freedom, $X_i, i = 0, 1, ..., k-1$ is the total pixels in the sample image of the gray-scale value $2i$, $\mu = \frac{\text{total pixels with gray value}2i}{\text{total pixels with gray value}2i+1}$, $i = 0, 1, ..., k-1$ is the expected value of the total pixels in the sample image, $k$ is the category which has the values 1, 2, ..., 128.

The probability of embedding is the probability of $\chi^2_{k-1}$ under the condition that $X_i = \mu, i = 0, 1, ..., k-1$. The density function, $1 - \rho$, tends to 1 as $\chi^2_{k-1}$ approaches $\infty$, so $\rho$ approaches 0 as $\chi^2_{k-1}$ approaches $\infty$. So, the probability of embedding is close to 0 for large $\chi^2_{k-1}$. The Matlab code in [28] is used to calculating the probability of embedding information. Figure 13 shows that the probability of embedding information is nearly equal to zero. So the proposed algorithm can resist the chi-square attacks.
4.1.7 | Noise attack

The proposed algorithm has been tested against noise attack in this section. Salt & pepper noise has corrupted the stego image with densities 0.01, 0.05 and 0.1. The effect of noise attack on the proposed algorithm is measured using root mean squared error (RMSE) and peak signal-to-noise ratio (PSNR). Figure 14 displays the extracted secret medical image after adding the salt & pepper noise to the stego image with densities 0.01, 0.05 and 0.1. The RMSE and PSNR are evaluated between the secret medical image and the extracted images in Table 8. Based on Table 8, extracting images after adding salt & pepper noise with small density produces better results than extracting images after adding salt & pepper noise with large density.

5 | CONCLUSION

In this paper, a new image steganographic algorithm using modified LSB and chaotic map is proposed. The author used two piecewise chaotic maps. The first is Tent chaotic map which is used to encrypting the secret medical image. The second is two-dimensional piecewise smooth chaotic map, which is used to selecting the positions of the host pixels randomly. Those pixels are used to embed the bits of the cipher images. In this algorithm, no bits of the secret medical image are lost. The experimental results show that the histogram of the stego image is almost identical to that of the host image. The comparison result, based on PSNR and MSE between the proposed algorithm and other algorithms in literature, shows that the proposed algorithm has the best result. The image quality measures display acceptable results based on the proposed algorithm. The relative entropies between different host images and stego images with different secret medical images are close to zero approximately. Also, the probability of embedding information is nearly equal to zero. So, the new steganographic algorithm can resist chi-square attacks. Therefore, the proposed algorithm can assist with the transmission of secret medical images via the communication media.

ACKNOWLEDGMENTS

The author deeply thank Dr. Abdel-Mohsen Elsherif for his proof reading of the paper.

ORCID

A. A. Karawia https://orcid.org/0000-0001-9819-9860

REFERENCES

1. Chen, H., et al.: Optical color image hiding scheme by using gerchberg-saxton algorithm in fractional Fourier domain. Opt. Lasers Eng. 66, 144–151 (2015)
2. Amirtharajan, R., Rayappan, J.: An intelligent chaotic embedding approach to enhance stego-image quality. Inf. Sci. 193, 115–124 (2012)
3. Tsai, Y., Chen, J.T., Chan, C.S.: Exploring LSB substitution and pixel-value differencing for block-based adaptive data hiding. International Journal of Network Security 16(5), 363–368 (2014)
4. Vigila, S.M.C., Muneeswaran, K.: Hiding of confidential data in spatial domain images using image interpolation. International Journal of Network Security 17(6), 722–727 (2015)
5. Seyyedi, S., Saday, V., Ivanov, N.: A secure steganography method based on integer lifting wavelet transform. International Journal of Network Security 18(1), 124–132 (2016)
6. Wahballa, O., et al.: A secure and robust certificateless public key steganography based on SVD-DDWT. International Journal of Network Security 18(5), 888–899 (2016)
7. Essaidani, D., Seddik, H., Braiek, E.B.: Asynchronous invariant digital image watermarking in radon field for resistant encrypted watermark. International Journal of Network Security 18(3), 19–32 (2016)
8. Amirtharajan, R., et al.: Pixel indicated triple layer: A way for random image steganography. Research Journal of Information Technology 5(2), 87–99 (2013)
9. Lu, T.C., Tseng, C.Y., Wu, J.H.: Dual imaging-based reversible hiding technique using LSB matching. Signal Process. 108(3), 77–89 (2015)
10. Tutuncu, K., Demirci, B.: Adaptive LSB steganography based on chaos theory and random distortion. Advances in Electrical and Computer Engineering 18, 15–22 (2018)
11. Gambhir, G., Mandal, J.: Multicore implementation and performance analysis of a chaos based LSB steganography technique. Microsyst. Technol. (2020) https://doi.org/10.1007/s00542-020-04762-4. Published online 12 February 2020
12. Pisarchik, A.N., Zarin, M.: Chaotic Map Cryptography and Security. In Encryption: Methods, Software and Security, pp. 1–28. Nova Science Publishers, New York (2010)
13. Liu, Z., Xi, L.: Image information hiding encryption using chaotic sequence. In: Knowledge-Based Intelligent Information and Engineering Systems and the XVII Italian Workshop on Neural Networks on Proceedings of the 11th International Conference, pp. 202–208. Springer, Berlin, Heidelberg (2007)

14. Liao, X., et al.: Medical jpeg image steganography based on preserving inter-block dependencies. Comput. Electr. Eng. 67, 320–329 (2018)

15. Karakus, S., Awei, E.: A new image steganography method with optimum pixel similarity for data hiding in medical images. Med. Hypotheses 139, 109691 (2020)

16. Li, C., et al.: An image encryption scheme based on chaotic tent map. Nonlinear Dyn. 87, 127–133 (2017)

17. Askar, S., Al-khedhairi, A.: The dynamics of a business game: A 2d-piecewise smooth nonlinear map. Physica A 537, 122766 (2020)

18. Malina, L., et al.: On feasibility of post-quantum cryptography on small devices. IFAC-Papers OnLine 51(6), 462–467 (2018)

19. Karawia, A.A.: Encryption algorithm of multiple-image using mixed image elements and two dimensional chaotic economic map. Entropy 20(10), 801 (2018). doi: 10.3390/e20100801

20. Khan, S. et al.: 5LSB steganography using monotonic RGB color image as cover medium. In: 2016 Sixth Int. Conf. on Innovative Computing Technology, pp. 275–278. IEEE, Piscataway, NJ (2016). doi: 10.1109/INTECH.2016.7845023

21. Yadav, G., Ojha, A.: Chaotic system-based secure data hiding scheme with high embedding capacity. Comput. Electr. Eng. 69, 447–460 (2018)

22. Wu, Q., et al.: A magic cube based information hiding scheme of large payload. J. Inf. Secur. Appl. 26, 1–7 (2016)

23. Amirulhaqi, A., Purboyo, T., Nugrahaeni, R.: Comparison of steganography in the gif image using LSB and spread spectrum method. Journal of Engineering and Applied Sciences 15(1), 247–251 (2020)

24. Kutter, M., Petitcolas, A.: Fair benchmark for image watermarking systems. Proc. SPIE 3657, Security and Watermarking of Multimedia Contents (9 April 1999). doi: 10.1117/12.344672

25. Soria-Lorente, A., Berres, S.: A secure steganographic algorithm based on frequency domain for the transmission of hidden information. Secur. Commun. Netw. 2017, 5397082 (2017)

26. Eggers, J., Baeuml, R., Girod, B.: Communications approach to image steganography. Security and watermarking of Multimedia Contents IV, pp. 1–12. SPIE, Bellingham, WA (2002)

27. Westfeld, A., Pfitzmann, A.: Attacks on steganographic systems. In: Pfitzmann A. (eds) Information Hiding. IH 1999. Lecture Notes in Computer Science, pp. 61–76. Springer, Berlin, Heidelberg (2000)

28. Toumazis, A.: Steganography (2009). https://www.cl.cam.ac.uk/teaching/0910/R08/work/essay-at443-steganography.pdf. Accessed 3 Dec 2009

How to cite this article: Karawia, A.A.: Medical image steganographic algorithm via modified LSB method and chaotic map. IET Image Process. 1–11 (2021). https://doi.org/10.1049/ipr2.12246