Journal of Innovative Technology, Vol. 1, No. 2, pp. 21-25, September 2019
http://doi.org/10.29424/JIT.201909_1(2).0004

ANALYSIS OF OVERFLOW IN DATA HIDING BASED ON EXTRACTION FUNCTION

Wen-Chung Kuo [1*], Jyun-Jia Li [2], and Chun-Cheng Wang [3]

ABSTRACT

When Shen and Huang proposed a steganographic method that increased the hidden data capacity within files, they did so at the cost of increased pixel change. They also solved the problems where stego-pixel values may cross the interval and overflow. Unfortunately, however, the methods proposed by Shen and Huang did not solve the problems perfectly. Using a combined approach of the method proposed by Kuo *et al*. and the method proposed by Shen and Huang, this paper proposes an improved steganographic method which solves the problems of crossing intervals and overflow completely, while generating an improved pixel value.

Keywords: EMD, extraction function, modulus, overflow, steganography.

1. INTRODUCTION

The dramatic increase in data movement through the internet since the turn of the century has also led to an increase in the risk of a data confidentiality issues during the sending and receiving of information. Data in motion may be eavesdropped on by a third party whenever it is transmitted. In order to ensure that information is secure, individuals and businesses may use tactics such as cryptography or data hiding to prevent unauthorized access. Data hiding often embeds some data that is to be hidden, into a cover file (usually an image) that masquerades as a standard file type, to form what is known as a stego-file or stego-image. A stego-image must be similar in size to the original; and to the naked eye, indistinguishable from the original. The recipient will be familiar with the file type and extraction method and will then be able to securely access the secreted data.

There are three criteria used to evaluate data hiding scheme:

1. Security: Security is the most important characteristic. A data hiding technology without security is useless. Only the rightful receiver can extract the secret message. In order to achieve this goal, we can encrypt the secret message before embedding. Although the malicious people may intercept the stego-image, he can not extract the secret data without the decrypted key.
2. Capacity: The capacity means the size of embedded secret message in the stego-image. The larger capacity means that we can use fewer images to embed the same secret message. Transform fewer stego-image can decrease the probability of attacked.
3. Imperceptibility: This criterion is the most basic requirement in data hiding technology. The imperceptibility is the difference of stego-image and cover image, which can not be easily observed from the naked eye. If the stego-image is very similar to the cover image, the stego-image should not cause suspicion and attack. There is an objective standard to measure the imperceptibility shown as Eq. (1):

$$PSNR = 10 \times log_{10} \frac{255^2}{MSE} \tag{1}$$

Where MSE is Mean Square Error, which is defined as (2):

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{M} (g(i,j) - g'(i,j))^2 \tag{2}$$

Where M and N are the size of image. $g(i, j)$ is the pixel value of cover image, and $g'(i, j)$ is the pixel value of stego-image.

If the PSNR is more than 30 dB, the stego-image does not cause suspicion by the malicious people. In general, the more secret message embedded in the stego-image, the less value in the imperceptibility. If we embedded less secret message, we can maintain the stego-image quality good, but we need to use more images to transport data. It needs much time to embed data and increase the probability that be attacked. So how to get the balance between the capacity and the imperceptibility is a very important consideration.

Currently, the most common data hiding method is the Least Significant Bit (LSB) method (Mielikainen 2006), which embeds the secret data in the least significant bit of each pixel of the image; however, as this is the most common approach, it is now the most easily detected. In 2006, Zhang and Wang proposed the Exploiting Modification Direction (EMD) (Zhang and Wang 2006) approach to improve on the shortcomings of LSB. Since then, many new, yet related, methods such as GEMD (General EMD) (Kuo *et al*. 2012), MGEMD (Multi-General EMD) (Kuo *et al*. 2016), Flexible EMD (Sun *et al*. 2013) and BEMD (Binary EMD) (Kuo *et al*. 2015) have been successively investigated. In 2011, (Kieu and Chang 2011) proposed another data hiding tech-

Manuscript received March 3, 2019; revised June 30, 2019; accepted July 5, 2019.
[1*] Professor (corresponding author), D Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan 64002, R.O.C. (e-mail: simonkuo@yuntech.edu.tw).
[2] Engineer, System Development Center, National Chung-Shan Institute of Science Technology, Taoyuan, Taiwan 32547, R.O.C.
[3] Engineer, National Center for High-performance Computing, National Applied Research Laboratories, Tainan, Taiwan, 74147, R.O.C.

nique based on a new extraction function that can increase the hidden data capacity of EMD (1.5bpp ~ 4.5bpp). In the Kieu-Chang method, a look-up table was used to find appropriate stego-pixel values. In order to reduce extraneous information, in 2013, Kuo and Kao (2013) followed a similar method used, but used formula solutions instead of a table lookup.

In 2015, Shen and Huang (2015) directly embedded binary data without transforming it, and maintained the original capacity of the Kieu-Chang method. At the same time, they also solved the problem of pixel value overflow/underflow. After some analysis, this laboratory found Shen-Huang's solution to overflow is similar to a brute force attack, and thus the stego-pixels could be optimized. For this reason, this paper looks closely at the Shen-Huang method and raises some points that it perceives as possible shortcomings. Then, using the method proposed by Kuo *et al*. this paper attempts to improve the data hiding model, by solving the problem of crossing the interval and overflow.

Section 2 briefly reviews related data hiding methods. Section 3 describes the Shen-Huang method. Section 4 describes the proposed data hiding scheme and an analysis and contrast follows in Section 5. Conclusion are presented in Section 6.

2. RELATED WORK

2.1 EMD (Zhang and Wang 2006)

In 2006, Zhang and Wang (2006) proposed the data hiding scheme based on EMD. They used n adjacent pixels to form one pixel group and set an extraction function. Each pixel group can embed $(2n + 1)$-ary secret message and only one pixel in group is modified by ±1. The extraction function of EMD scheme is shown as Eq. (3):

$$f(g_1, g_2, ..., g_n) = \left[\sum_{i=1}^{n} (g_1 \times i)\right] \bmod (2n \times 1) \quad (3)$$

Where g_i is the i-th pixel value in group and n is the number of pixels in each group. Since the pixel value of the image is an integer, the pixel group can be expressed as a vector $[g_1, g_2, \ldots, g_n]$ in the n-dimensional space. All function values computed by vectors form a Hyper-Cube. For example, Figure 1 is the Hyper-Cube in the 2-dimensional space. Each function value is different with its neighbors in the Hyper-Cube. If we want to embed secret message, we can find the suitable function value in the vector or its neighbors.

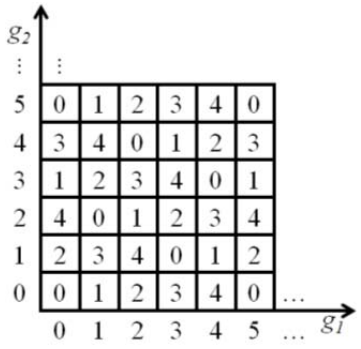

Fig. 1 The Hyper-Cube in 2-dimensional

Input: cover image and secret data s
Output: stego-image

Step 1. Divide cover image into non-overlapping n pixel groups and transform s into $(2n + 1)$-ary secret data stream s'.

Step 2. Use the cover pixel group to compute f by Eq. (3).

Step 3. Obtain $(2n + 1)$-ary secret data s'_i from s' and compute $d = (s'_i - f_{EMD}) \bmod (2n + 1)$.

Step 4. If $d \le n$, then $g'_d = g_d + 1$ and $g'_i = g_i, \forall i \in \{1, 2, \ldots, n \mid i \ne d\}$, else $g'_{2n+1-d} = g_{2n+1-d} - 1$ and $g'_i = g_i, \forall_i \in \{1, 2, \ldots, n \mid i \ne 2n + 1 - d\}$.

Step 5. Repeat from step 2 until all secret data is embedded.

The EMD data hiding scheme is very advantageous to the receiver. When the receiver gets the stego-image, he just computes the Eq. (3) for each pixel group and converts function value to secret message.

The EMD data hiding scheme has very good stego-image quality. However, there are two shortcomings for EMD data hiding scheme. First is that the secret message should be converted to (2n + 1)-ary before embedding. It needs much time to convert whole secret message at once to achieve maximum capacity.

Second is that the capacity is fewer when n is larger. When n is 2, the modulus is 5 and we can embed log_2 5 ≈ 2.32 bits in each group. The capacity is $\frac{log_2 5}{2}$ ≈ 1.16 bpp. When n is 3, the modulus is 7 and we can embed $\log_2 7 \approx 2.81$ bits in each group. The capacity is $\frac{log_2 7}{2}$ ≈ 0.93 bpp. The capacity of EMD data hiding scheme is too less to transform large secret message.

2.2 Fully EMD (Kieu and Chang 2011)

Since Zhang and Wang proposed the EMD method, many related models such as GEMD, MGEMD, Flexibile EMD and BEMD have emerged. In 2011, Kieu and Chang (2011) proposed a new state, FEMD (Fully EMD) to improve capacity further. Their principal development was the new extraction function shown below in equation (4):

$$F(g_1, g_2) = (s-1) \times g_1 + s \times g_2 \; mod(s^2) \quad (4)$$

Where g_1, g_2 are group pixel values, and $(s-1)$ and s are coefficients. At the same time, Kieu and Chang designed a 256 × 256 matrix function table, to record the function value calculated by equation (4). The value corresponding to the g_1 row and the g_2 column is the extraction function value. Figure 1 shows the function table for $s = 4$.

g_1 \ g_2	0	1	2	3	4	5	6	7	8	…	255
0	0	4	8	12	0	4	8	12	0	…	12
1	3	7	11	15	3	7	11	15	3	…	15
2	6	10	14	2	6	10	14	2	6	…	2
3	9	13	1	5	9	13	1	5	9	…	5
4	12	0	4	8	12	0	4	8	12	…	8
5	15	3	7	11	15	3	7	11	15	…	11
6	2	6	10	14	2	6	10	14	2	…	14
7	5	9	13	1	5	9	13	1	5	…	1
8	8	12	0	4	8	12	0	4	8	…	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
255	13	1	5	9	13	1	5	9	13	…	9

Fig. 2 Matrix function table (s = 4)

This method increased the modified range of pixels from +1 or −1 to +s or −s. The hiding steps for the Kieu-Chang method are described below:

Input: cover image with size $H \times W$, binary secret data $M = m_1\ m_2\ m_3 \ldots m_n$.

Output: stego-image with size $H \times W$.

Step 1. Divide all the pixels of cover image into two non-overlapping pixels (p_1, p_2) for a group, and then use each pixel group sequentially to compute the extraction function value by Eq. (4).

Step 2. Compute $k = \lfloor log_2 s^2 \rfloor$, $r = \lfloor s/2 \rfloor$.

Step 3. Get k bits secret data and transform it to s^2-ary sequentially.

Step 4. Look up (g_1, g_2) corresponding to s^2-ary secret data in matrix function table.

Step 5. If the function value of a pixel group equals the value from the secret data, then do not modify the pixel. Else, search all possible pixel groups in range $(g_1 \pm s, g_2 \pm s)$ and choose the suitable pixels (g'_1, g'_2) for stego-pixels according to Eq. (5).

$$D\ min = \left\{ \left| g'_1 - p_1 \right| + \left| g'_2 - p_2 \right| \right\} \tag{5}$$

The FEMD data hiding method has three characteristics: first, the capacity is more than 1.5bpp (when s = 3); second, it has a high secret data hiding capacity (when s = 23, it can reach 4.5bpp); third, stego-pixels can be directly obtained by use of a look-up function in the matrix table.

2.3 Formula Fully EMD (Kuo and Kao 2013)

Although the Kieu-Chang method can greatly increase capacity, it is necessary to generate a corresponding 256 × 256 matrix function table according to the various s values. A user needs to record this large amount of extra information, which consumes considerable storage space. Therefore, in 2013, Kuo and Kao (2013) proposed the KG-Theorem which could directly calculate the stego-pixel values. Assuming that the function values $F(g_1, g_2)$ and the modulus value s are obtained, the adjusted pixel values can be obtained by simply substituting Eq. (6) and (7):

$$g_1 = (s-1) \times F(g_1, g_2)\ mod\ s, \tag{6}$$

$$g_2 = \left[\frac{(F(g_1, g_2) - (s-1) \times g_1}{s} \right] \ mod\ s. \tag{7}$$

The embedding steps are as follows:

Step 1. Get (g_1, g_2) and s. Compute $F(g_1, g_2) = (s-1) \times g_1 + s \times g_2\ mod\ s^2$.

Step 2. Extract secret data m.

Step 3. If $m = s$, do not modify any pixels; else,

1. Compute $t = (s-1) \times m\ mod\ s$.
2. Compute $t_{1,1} = t - g_1\ mod\ s$; if $t_{1,1} > 0$, then $t_{1,1} = t_{1,1} - s$, $y_{1,1} = y_{2,1} = g_1 + t_{1,1}$.
3. Compute $t_{1,2} = [m - (s-1) \times y_{1,1} / s]\ mod\ s$.
4. Compute $t_{1,2} = t_{1,2} - (g_2\ mod\ s)$; if $t_{1,2} > 0$, then $t_{2,2} = t_{1,2} - s$; else $t_{2,2} = t_{1,2} + s$.
5. Compute $y_{1,2} = g_2 + t_{1,2}$, $y_{2,2} = g_2 + t_{2,2}$.
6. Compute $t_{2,1} = t_{1,1} + s$.
7. Compute $y_{3,1} = y_{4,1} = g_1 + t_{2,1}$.
8. Compute $t_{3,2} = [m_1 - (s-1) \times y_{3,1} / s]\ mod\ s$.
9. Compute $t_{3,2} = t_{3,2} - (g_2\ mod\ s)$, if $t_{3,2} > 0$, then $t_{4,2} = t_{3,2} - s$; else $t_{4,2} = t_{3,2} + s$.
10. Compute $y_{3,2} = g_2 + t_{3,2}$, $y_{4,2} = g_2 + t_{4,2}$.

Step 4. Compute $D = \{(|g_1 - x| + |g_2 - y|) | (x, y) \in \{(y_{1,1}, y_{1,2}), (y_{2,1}, y_{2,2}), (y_{3,1}, y_{3,2}), (y_{4,1}, y_{4,2})\}\}$.

Step 5. Select (x, y) with the smallest distortion as the stego-pixel pair.

The method proposed by Kuo and Kao allows a user to calculate the adjusted pixel value simply and quickly. Without generating a 256 × 256 matrix function table. This makes the method extremely convenient, especially for hiding small amounts of secret data on something like a mobile device.

2.4 Improving Fully EMD (Shen and Huang 2015)

In 2015, Shen and Huang used the concept of pixel difference to improve the number of embedded secret information bits in FEMD. The hidden steps are as follows:

Step 1. Divide the difference between two pixels to $W = \{w_j = [l_j, u_j]\} = \{[0,7], [8,15], [16,31], [32,63], [64,127], [128,255]\}$.

Step 2. Divide the $M \times N$ cover image into non-overlapping blocks using a Hilbert curve. Each block then contains two pixels (g_{2i}, g_{2i+1}).

Step 3. Compute $d_i = |g_{2i} - g_{2i+1}|$, $d_i \in W$.

Step 4. Compute $s_i = \lfloor log_2 w_j \rfloor$, $w_j = u_j - l_j + 1$, $k_i = \lfloor log_2 s_i^2 \rfloor$.

Step 5. Get k_i bits secret data m_i and transform it to s_i^2-ary.

Step 6. Compute $t_i = F_f[(g_{2i} \times (s_i - 1) + g_{2i+1} \times s_i)]\ mod\ (s_i^2)$.

Step 7. If $m_i = t_i$, then $(g'_{2i}, g'_{2i+1})) = (g_{2i}, g_{2i+1})$; else,

(i) if $m_i > t_i$, then $g'_{2i} = g_{2i} - [m_i - F_f(g_{2i}, g_{2i+1})]\ mod\ s_i$; $g'_{2i+1} = g_{2i+1} + \lfloor m_i - F_f(g_{2i}, g_{2i+1})/s_i \rfloor + [m_i - F_f(g_{2i}, g_{2i+1})]\ mod\ s_i$.

(ii) if $m_i < t_i$, then $g'_{2i} = g_{2i} + [m_i - F_f(g_{2i}, g_{2i+1})]\ mod\ s_i$; $g'_{2i+1} = g_{2i+1} - \lfloor m_i - F_f(g_{2i}, g_{2i+1})/s_i \rfloor - [m_i - F_f(g_{2i}, g_{2i+1})]\ mod\ s_i$.

Shen and Huang solve the cross-interval problem for the pixel pair after embedding. When the difference of a stego-pixel pair is not the same as the difference of a cover pixel pair, the solution is to choose the pixel pair with the same interval as the cover pixel pair from one of six adjustment methods $(g'_{2i} - 3s_i, g'_{2i+1} - 3)$, $(g'_{2i} - 2s_i, g'_{2i+1} - 2)$, $(g'_{2i} - s_i, g'_{2i+1} - 1)$, $(g'_{2i} + s_i, g'_{2i+1} + 1)$, $(g'_{2i} + 2s_i, g'_{2i+1} + 2)$ and $(g'_{2i} + 3s_i, g'_{2i+1} + 3)$. If the stego-pixel generates an overflow problem, a new pixel value (g^*_{2i}, g^*_{2i+1}) is found around (g'_{2i}, g'_{2i+1}) for the new stego-pixel pair, where $0 \le g^*_{2i}, g^*_{2i+1} \le 255$ and $F_s(g'_{2i}, g'_{2i+1}) = F(g^*_{2i}, g^*_{2i+1})$.

3. ANALYSIS OF THE SHEN-HUANG METHOD

Although the Shen-Huang data hiding method has solutions to the situation of cross-interval and overflow, it can be shown that the most suitable pixels cannot be selected for processing of the interval (as shown in EX 1). Furthermore, for the overflow problem, the procedure tries all the possibilities, which takes time and constant testing (as shown in EX 2).

EX 1: If cover pixel pair (g_1, g_2) = (20,80) and secret data $m_i = 1110_2$. The embedding steps are as follows:

Step 1. Divide the difference into $W = \{w_j = [l_j, u_j]\} = \{[0,7],$

[8,15], [16,31], [32,63], [64,127], [128,255]}.
Step 2. Compute $d_i = 60$, $w_j \in [32,63]$, $w_j = 63 - 32 + 1 = 32$.
Step 3. Compute $s_i = \lfloor log_2 32 \rfloor = 5$, $k_i = \lfloor log_2 5^2 \rfloor = 4$.
Step 4. Transform m_i to 25-ary. $m_i = 1110_2 = 14_{25}$ ◦
Step 5. Compute $t_i = F[(20 \times 4 + 80 \times 5)] \ mod\ 25 = 5$.
Step 6. $m_i = 14 \neq 5$ & $m_i > t_i$, $g'_{2i} = 20 - (14 - 5) \ mod\ 5 = 16$;
$g'_{2i+1} = 80 + \lfloor 14 - 5/5 \rfloor + (14 - 5) \ mod\ 5 = 85$.

The difference of the stego-pixel pair (16,85) is in [64,127]. We compute (16 – 3 × 5,85 – 3), (16 – 2 × 5,85 – 2), (16 – 1 × 5, 85 – 1), (16 + 1 × 5,85 + 1), (16 + 2 × 5,85 + 2) and (16 + 3 × 5, 85 + 3) six pixel pair, and choose (16 + 2 × 5,85 + 2) = (26,87) to derive the stego-pixel pair.

EX 2: If cover pixel pair $(g_1, g_2) = (2,253)$ and secret data mi = 110012. The embedding steps are as follows:
Step 1. Divide the difference into $W = \{w_j = [l_j, u_j]\} = \{[0,7], [8,15], [16,31], [32,63], [64,127], [128,255]\}$.
Step 2. Compute $d_i = 251$, $w_j \in [128,255]$, $w_j = 255 - 128 + 1 = 128$.
Step 3. Compute $s_i \lfloor log_2 128 \rfloor = 7$, $k_i = \lfloor log_2 7^2 \rfloor = 5$.
Step 4. Transform mi to 49-ary. $m_i = 11001_2 = 25_{49}$ ◦
Step 5. Compute $t_i = F[(2 \times 6 + 253 \times 7)] \ mod\ 49 = 19$.
Step 6. $m_i = 25 \neq 19$, $m_i > t_i$, $g_{2i} = 2 - (25 - 19) \ mod\ 7 = -4$;
$g'_{2i+1} = 253 + \lfloor 25 - 19/7 \rfloor + (25 - 19) \ mod\ 7 = 259$.

The stego-pixel pair produces an overflow problem for (–4,259). According to the Shen-Huang adjustment method, it is necessary to search for pixel pairs that satisfy similar pixels and satisfy the same function value, and eventually find (3,253) as the adjusted camouflage pixel pair. When Shen-Huang's method produces an overflow problem, it must repeatedly test similar pairs of pixels until a suitable solution is found.

4. PROPOSED METHOD

In order to solve the above problems effectively, a combination of the Shen-Huang (2015) concept and the Kuo-Kao (2013) method is proposed here. Its main hiding algorithm is shown in the Algorithm 1.

Algorithm 1.

Input: cover image with size $H \times W$, binary secret data $M = m_1 \ m_2 \ m_3 \ldots m_n$.
Output: stego-image with size $H \times W$.
Step 1. Divide the difference in to $W = \{wj = [lj, uj]\} = \{[0,7], [8,15], [16,31], [32,63], [64,147], [128,255]\}$.
Step 2. Divide the $M \times N$ cover image into non-overlapping blocks using a Hilbert curve. Each block contains two pixels (g_{2i}, g_{2i+1}).
Step 3. Choose the next block sequentially.
Step 4. Compute $d_i = |g_{2i} - g_{2i+1}|$, $d_i \in W$.
Step 5. Compute $s_i = \lfloor log_2 w_j \rfloor$, $ki = \lfloor log_2 s_i^2 \rfloor$.
Step 6. Get k bits secret data and transform it to s^2-ary sequentially.
Step 7. Compute $F(g_{i1}, g_{i2})$. If $F(g_{i1}, g_{i2}) = mi$, then $(y_1, y_2) = (g_1, g_2)$, goto step 9.
Step 8. If $F(g_{i1}, g_{i2}) \neq m_i$, then
1. Compute $t = (s - 1) \times m_i \ mod\ s$.
2. Compute $t_{1,1} = t - (g_1 \ mod\ s)$, if $t_{1,1} > 0$, compute $t_{1,1} = t_{1,1} - s$, $y_{1,1} = y_{2,1} = g_1 + t_{1,1}$.
3. Compute $t_{1,2} = [(m_i - (s - 1) \times y_{1,1})/s] \ mod\ s$.
4. Compute $t_{1,2} = t_{1,2} - (g_2 \ mod\ s)$, if $t_{1,2} > 0$, then $t_{2,2} = t_{1,2} - s$; else $t_{2,2} = t_{1,2} + s$.
5. Compute $y_{1,2} = g_2 + t_{1,2}$, $y_{2,2} = g_2 + t_{2,2}$.
6. Compute $t_{2,1} = t_{1,1} + s$.
7. Compute $y_{3,1} = y_{4,1} = g_1 + t_{2,1}$.
8. Compute $t_{3,2} = [(m_i - (s - 1) \times y_{3,1})/s] \ mod\ s$.
9. Compute $t_{3,2} = t_{3,2} - (g_2 \ mod\ s)$, if $t_{3,2} > 0$, then $t_{4,2} = t_{3,2} - s$; else $t_{4,2} = t_{3,2} + s$.
10. Compute $y_{3,2} = g_2 + t_{3,2}$, $y_{4,2} = g_2 + t_{4,2}$.
11. Choose the most suitable pixel pair from (y_{11}, y_{12}), (y_{21}, y_{22}), (y_{31}, y_{32}) and (y_{41}, y_{42}) for the stego-pixel pair.

Step 9. When the block is the last block, it ends; otherwise, go to step 3.

Then, we use EX 3 and EX 4 to compare the proposed method with the existing Shen-Huang method to check it solves the cross-interval and overflow problems.

EX 3: If cover pixel pair $(g_1, g_2) = (20,80)$ and secret data $m_i = 1110_2$. The embedding steps are as follows:
Step 1. Compute $d_i = |20 - 80| = 60 \in [32,63]$.
Step 2. Compute $s_i = \lfloor \log_2 60 \rfloor = 5$, $k_i = \lfloor \log_2 5^2 \rfloor = 4$.
Step 3. Get 4 bits secret data mi and transform it to 25-ary. $m_i = 1110_2 = 14_{25}$.
Step 4. Compute $F(20,80) = [20 \times (5 - 1) + 80 \times 5] \ mod\ 25 = 5$.
Step 5. $5 \neq 14_{25}$.
Step 6. Compute $t = (5 - 1) \times 14 \ mod\ 5 = 1$.
Step 7. Compute $t_{1,1} = 1 - (20 \ mod\ 5) = 1$, $t_{1,1} = 1 - 5 = -4$, $y_{1,1} = y_{2,1} = 20 - 4 = 16$.
Step 8. Compute $t_{1,2} = [14 - (5 - 1) \times 16 / 5] \ mod\ 5 = 0$.
Step 9. Compute $t_{1,2} = 0 - (80 \bmod 5) = 0$, $t_{2,2} = 0 + 5 = 5$.
Step 10. Compute $y_{1,2} = 80 + 0 = 80$, $y_{2,2} = 80 + 5 = 85$.
Step 11. Compute $t_{2,1} = -4 + 5 = 1$.
Step 12. Compute $y_{3,1} = y_{4,1} = 20 + 1 = 21$.
Step 13. Compute $t_{3,2} = [14 - (5 - 1) \times 21 / 5] \ mod\ 5 = 1$.
Step 14. Compute $t_{3,2} = 1 - (80 \ mod\ 5) = 1$, $t_{4,2} = 1 - 5 = -4$.
Step 15. Compute $y_{3,2} = 80 + 1 = 81$, $y_{4,2} = 80 - 4 = 76$.

This produces four pixel pairs (16,80), (16,85), (21,81) and (21,76), from which the nearest is chosen as a stego-pixel pair (21,81). Compared to EX 1, the stego-pixel pair from the new algorithm is closer to the cover pixel pair.

EX 4: If cover pixel pair $(g_1, g_2) = (2,253)$ and secret data $m_i = 11001_2$. The embedding steps are as follows:
Step 1. Compute $d_i = |2 - 253| = 251 \in [128,255]$.
Step 2. Compute $s_i = \lfloor \log_2 251 \rfloor = 7$, $k_i = \lfloor \log_2 7^2 \rfloor = 5$.
Step 3. Get 5 bits secret data m_i and transform it to 49-ary. $m_i = 11001_2 = 25_{49}$.
Step 4. Compute $F(2,253) = [2 \times (7 - 1) + 253 \times 7] \ mod\ 49 = 19$.
Step 5. $19 \neq 25_{49}$.
Step 6. Compute $t = (7 - 1) \times 25 \ mod\ 7 = 3$.
Step 7. Compute $t_{1,1} = 3 - (2 \ mod\ 7) = 1$, $t_{1,1} = 1 - 7 = -6$, $y_{1,1} = y_{2,1} = 2 - 6 = -4$.
Step 8. Compute $t_{1,2} = [25 - (7 - 1) \times (-4) / 7] \ mod\ 7 = 0$.
Step 9. Compute $t_{1,2} = 0 - (253 \ mod\ 7) = -1$, Compute $t_{2,2} = (-1) + 7 = 6$.
Step 10. Compute $y_{1,2} = 253 - 1 = 252$, $y_{2,2} = 253 + 6 = 259$.
Step 11. Compute $t_{2,1} = -6 + 7 = 1$.

Step 12. Compute $y_{3,1} = y_{4,1} = 2 + 1 = 3$.
Step 13. Compute $t_{3,2} = [25 - (7 - 1) \times 3 / 7]$ *mod* $7 = 1$.
Step 14. Compute $t_{3,2} = 1 - (253$ *mod* $7) = 0$, $t_{4,2} = 0 + 7 = 7$.
Step 15. Compute $y_{3,2} = 253 + 0 = 253$, $y_{4,2} = 253 + 7 = 260$.

Following the above steps derives four pixel pairs (–4, 252), (–4, 259), (3, 253) and (3, 260), from which the nearest (21, 81) is chosen as a stego-pixel pair. Compared to EX 2, this new stego-pixel pair is computed directly, there is no need to expend additional effort to calculate all possible solutions.

5. ANALYSIS AND CONTRAST

From the examples provided here, it can be seen the while Shen-Huang's method was a major leap forward, it does not solve the data-hiding problem perfectly. In the revised method proposed here, at least one of the four pixel pairs meets the same interval difference and there is no overflow. Table 1, is a comparison of the discussed methods, Kieu and Chang (2011), Shen and Huang (2015) and Kuo-Li.

Table 1 Comparison table

Item \ Method	Kieu and Chang (2011)	Shen and Huang (2015)	Kuo-Li
Get non-overflow solution directly	Yes	No	Yes
Get nearest solution	Yes	No	Yes
Adaptive modulus	No	Yes	Yes
Need extra memory storage	Yes	No	No
capacity	1 ~ 4.5bpp	1.5 ~ 1.69bpp	1.5 ~ 1.69bpp
PSNR	52.4 ~ 31.6dB (s = 2 ~ 23)	38.9 ~ 42.4dB	48.2 ~ 49.3dB

From Table 1 it can be seen that the following three characteristics emerge:

1. The method proposed here can find the most suitable solution, with no overflow or cross-interval problem.
2. The proposed method does not require additional storage of a matrix function table(s).
3. The proposed method can achieve a balance between capacity and image quality, with regard to the original image characteristics. By comparison, for the same carrier image, the image quality using the proposed method is superior to that using the Shen-Huang method.

6. CONCLUSION

This paper has proposed a direct data hiding method using an embedded binary secret message. This method can provide suitable modulus according to image block complexity, and achieve a balance between image quality and capacity. The method set out in this paper is presently able to find the most suitable stego-pixel pair, and solve the cross-interval and overflow problems.

REFERENCES

Chan, C.K. and Cheng, L.M. (2004). "Hiding Data in Images by Simple LSB Substitution." *Pattern Recognition*, **37**(3), 469-474.

Chao, R.M., Wu, H.C., Lee, C.C., and Chu, Y.P. (2009). "A Novel Image Data Hiding Scheme with Diamond Encoding." *EURASIP Journal on Information Security*, 1-9.

Kieu, D. and Chang, C.C. (2011). "A steganographic scheme by fully exploiting modification directions." *Expert Systems with Applications*, **38**(8), 10648–10657.

Kuo, W.C., Wuu, L.C., and Kuo, S.H. (2012). "The high embedding steganographic method based on general multi-EMD." *2012 International Conference on Information Security and Intelligence Control (ISIC)*, pp. 286-289.

Kuo, W.C. and Kao, M.C. (2013). "A steganographic scheme based on formula fully exploiting modification directions." *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E96-A*, **11**, 2235-2243.

Kuo, W.C. and Wang, C.C. (2013). "Data hiding based on generalized exploiting modification direction method." *Image Science Journal*, **61**(6), 484-490.

Kuo, W.C. (2013). "A data hiding scheme based on square formula fully exploiting modification directions." *Journal of Information Hiding and Multimedia Signal Processing*, **4**(3), 127-136.

Kuo, W.C., Kuo, S.H., and Huang, Y.C. (2013). "Data hiding schemes based on the formal improved exploiting modification direction method." *Applied Mathematics & Information Sciences Letters*, **1**(3), 1-8.

Kuo, W.C., Lai, P.Y., Wang, C.C., and Wuu, L.C. (2015). "A Formula Diamond Encoding Data Hiding Scheme." *Journal of Information Hiding and Multimedia Signal Processing*, **6**(6), 1167-1176.

Kuo, W.C., Wang, C.C., and Huang Y.C. (2015). "Binary power data hiding scheme." *AEU - International Journal of Electronics and Communications*, **69**(11), 1574-1581.

Kuo, W.C., Kao, M.C., and Chang, C.C. (2015). "A generalization of fully exploiting modification directions data hiding scheme." *Journal of Information Hiding and Multimedia Signal Processing*, **6**(4), 718-727.

Kuo, W.C., Kuo, S.H., Wang C.C., and Wuu, L.C. (2016). "High capacity data hiding scheme based on multi-bit encoding function." *Optik*, **127**, 1762-1769.

Lee, C.F., Wang, Y.R., and Chang, C.C. (2007). "A Steganographic Method with High Embedding Capacity by Improving Exploiting Modification Direction." *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, **1**, 497-500.

Mielikainen, J. (2006). "LSB matching revisited." IEEE Signal Processing Letters, **13**(5), 285-287.

Petitcolas, F.A.P., Anderson, R.J., and Kuhn, M.G. (1999). "Information Hiding – A Survey. Proc." *IEEE*, **87**(7), 1062-1078.

Shen S.Y. and Huang, L.H. (2015). "A data hiding scheme using pixel value differencing and improving exploiting modification directions." *Computers & Security*, **48**, 131–141.

Sun, H.M., Weng, C.Y., Wang, S.J., and Yang, C.H. (2013). "Data embedding in image-media using weight-function on modulo operations." *ACM Transactions on Embedded Computing Systems*, **12**(2), 21.

Wang, C.C., Kuo, W.C., Huang, Y.C., and Wuu, L.C. (2017). "A High Capacity Data Hiding Scheme Based on Re-adjusted GEMD." *Multimedia Tools and Applications*, 1-15.

Wu, K.S., Liao, W.D., Lin, C.N., and Chen, T.S. (2015). "A High Payload Hybrid Data Hiding Scheme with LSB, EMD and MPE." *The Imaging Science Journal*, **63**(3), 174-181.

Zhang, X. and Wang, S. (2006). "Efficient steganographic embedding by exploiting modification direction." *IEEE Communications Letters*, **10**(11), 1-3.