Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **4**, September **2015**

Reversible Data Hiding Based on Histogram Modification over Ternary Computers

Yu-Ze Jheng, Chien-Yuan Chen, and Chien-Feng Huang

Department of Computer Science and Information Engineering
National University of Kaohsiung
700 University Road, Kaohsiung, 811, Taiwan
yu.ze.jheng@gmail.com; cychen07@nuk.edu.tw; cfhuang15@nuk.edu.tw

Received June, 2013; revised January, 2015

Abstract. *In this paper, we propose two data hiding methods over ternary computers called Ternary Data Hiding (TDH) and Coded-Ternary Data Hiding (C-TDH). We embed ternary secret data, NAF format, into cover image. According to experiment results, TDH can keep higher PSNR when the number of first peak point is six times than the number of second peak point. Furthermore, we propose C-TDH method to increase the amount of the embedded secret data in TDH method. According to our analysis, as compared with TDH, C-TDH increases the capacity by 50% on the average.*

Keywords: Histogram-based data hiding, Ternary computer, Balanced ternary numbering system

1. **Introduction.** To avoid that important information is taken by an illegal user during transmission, data hiding has drawn more and more attention [2, 3, 4, 5, 6, 9, 10, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22]. The main idea of data hiding is to embed the secret data into cover image and generate quality stego-image with PSNR higher than 40dB. Since the stego-image is imperceptible when PSNR is higher than 40dB [15], the illegal user can not recognize whether the stego-image is embedded the secret data or not. Only the specific user can extract the secret data from the stego-image. An effective data hiding method with high PSNR was proposed in 2006 by Ni *et al.* [15]. Assume that the sender wants to send the secret data to the receiver. The sender computes all the numbers of pixel values in histogram and embeds the secret data by shifting the histogram. The sender thus generates stego-image with PSNR higher than 48dB. Ni *et al.*'s method can keep high quality of stego-image since each difference between original and shifted pixel values is at most one. But the capacity of this method is little because the sum of top two highest numbers of pixel values in histogram is not big. In 2008, Lin *et al.* [13] presented a different method by which the histogram was constructed using a difference image to accomplish the task of reversible data hiding. Lin*et al.*'s method introduces a more centralized histogram, which in turn may generate more height of peak points such that the capacity is improved. To enhance the performance of Lin *et al.*'s histogram-based method, Yang and Tsai [22] proposed a new method using column-based interleaving predictions. In this method, the even-column pixels are predicted by pixels in odd columns; then the odd-column pixels are predicted by pixels in even columns, or vice versa. Improving column-based interleaving predictions, Yang and Tsai [22] presented a chessboard prediction method to reduce the prediction errors. The heights of peak

points are raised in the histogram and the capacity is largely increased. In addition to the increase in capacity, Yang and Tsai [22] showed that their histogram-based method can also yield a good image quality, as compared to other histogram-based methods.

Although many papers [2, 3, 4, 5, 6, 9, 10, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22] are proposed to improve data hiding methods over binary computers, the studies seldom focus on data hiding methods over non-binary computers. Non-binary computers are used for special purpose nowadays like quantum computer [16]. The famous non-binary computer is the ternary computer [7]. Over the ternary computer, the balanced ternary numbering system [8] is described by Knuth as "the prettiest number system of all" [11] for its elegant arithmetic properties. The ternary concept will be found in optical computer [23] or quantum computer [16]. The main notation in the ternary computer is represented by $\{0, 1, \bar{1}\}$. Non-Adjacent Form (NAF) is the famous method of transforming binary data into ternary format. The secret data of NAF format have three characteristics [8, 23]: the data are represented by $\{0, 1, \bar{1}\}$, 1 and $\bar{1}$ are never adjacent, and the probability of appearing 0 is about 66%. Based on the characteristics in NAF format, we propose a data hiding method, Ternary Data Hiding (TDH), with high quality of stego-image over ternary computer.

In Ternary Data Hiding (TDH), the sender embeds the secret data of NAF format into cover image. The sender divides the cover image into white and black segments. The sender computes the predictive errors of white segment and outputs the numbers of predictive errors in histogram. Through the histogram, the sender gets a peak point and its two corresponding zero points. Predictive errors between two zero points are shifted except the peak point to generate the spaces nearby the peak point. The sender embed secret digit "$\bar{1}$" by shifting the predictive errors of peak point to their left space, secret digit "1" by shifting the predictive errors of peak point to their right space and secret digit "0" by remaining the pixel value. After all the predictive errors are scanned, the sender reverses predictive errors into pixel values and generates quality stego-image. If there are remaining secret data, the sender computes the predictive errors of black segment and outputs the numbers of predictive errors in histogram. Then the sender embeds remaining secret data using the same scheme as white segment.

Next, we propose Coded-Ternary Data Hiding (C-TDH) to improve the capacity of TDH. According to the characteristics of NAF, 1 and $\bar{1}$ are never adjacent, secret data are represented by three symbols, 10, $\bar{1}$0, and 0. Thus, the sender can embed at most two digits in one pixel. According to our analysis, C-TDH increases the capacity by 50% on the average, as compared with TDH. According to experiment results, our method can keep higher PSNR when the number of first peak point is six times than the number of second peak point. The outline of this paper is as follows. In Section 2, we introduce Yang and Tsai's method and the NAF format. Our methods are given in Section 3. The experiment results are shown in Section 4. In Section 5, we compare Yang and Tsai's method and ours. Finally, we make a conclusion in Section 6.

2. **Related Works.** In this section, we review the algorithm of Yang and Tsai's method and the NAF format. At first, we give the required definitions in this paper. The size of cover image is 512×512. Pixel (i,j) in the cover image has a pixel value $P_{i,j}$, where $1 \leq i$, $j \geq 512$. H_{mn} is the m^{th} peak point of n^{th} segment and Z_{mn} is the m^{th} zero point of n^{th} segment. The length of secret data is L and we assume that L is less than capacity in this paper. B_k is the k^th bit of secret data in binary format and N_t is the t^{th} digit of secret data in NAF format, where $k \leq L$ and $t \leq L + 1$. Every $P_{i,j}$ has its own specific neighbor set $S_{i,j}$. Note that $S_{i,j} = \{P_{i,j} | x, y \in [1, 512], |i - x| + |j - y| = 1\}$. The predictive error

of(i,j) is $D_{i,j}$, where $1 \leq i,\ j \geq 512$. And $B = (B_1B_2...B_L)$ is the secret data in binary format and $N = (N_1N_2...N_{L'})$ is the secret data in NAF format, where $L' = L + 1$.

2.1. **Yang and Tsai's Method** [22]**.** Before transporting secret data B to the receiver, the sender embeds secret data into cover image using Yang and Tsai's method [22] as follows. The sender divides the cover image into white and black segments, as shown in Figure 1.

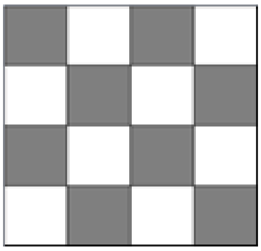


FIGURE 1. Two segments of Yang and Tsai's method

The sender compiles the embedding algorithm with one segment at once. In the first part of the algorithm, the sender computes all the predictive errors in white segment by Formula (1).

$$D_{i,j} = P_{i,j} - \left\lfloor \frac{\Sigma_{P_{i,j} \in S_{i,j}} P_{i,j}}{|S_{i,j}|} \right\rfloor \tag{1}$$

The sender outputs the numbers of predictive errors of white segment and gets histogram of predictive errors. The sender finds $(H_{11}, H_{21}, Z_{11}, Z_{21})$, where $Z_{21} < H_{11} < H_{12} < Z_{21}$. The predictive errors in the range of (Z_{11}, H_{11}) and (H_{21}, Z_{21}) are shifted by Formula (2) to make the spaces nearby peak points.

$$D_{i,j} = \begin{cases} D_{i,j} - 1, \text{if } Z_{11} < D_{i,j} < H_{11} \\ D_{i,j} + 1, \text{if } H_{21} < D_{i,j} < Z_{21} \end{cases} \tag{2}$$

The sender scans predictive errors from top to bottom and left to right until all the predictive errors are scanned. When the scanned predictive error is equal to H_{11} or H_{21}, the sender embeds secret bit B_k into cover image by Formula (3).

$$D_{i,j} = \begin{cases} D_{i,j} + 1, \text{ if } D_{i,j} = H_{21} \text{ and } B_k = 1 \\ D_{i,j} - 1, \text{ if } D_{i,j} = H_{11} \text{ and } B_k = 1 \\ \quad D_{i,j}, \text{ if } D_{i,j} = H_{21} || H_{21} \text{ and } B_k = 0 \end{cases} \tag{3}$$

Predictive errors are scanned again and reversed into pixel values by Formula (4).

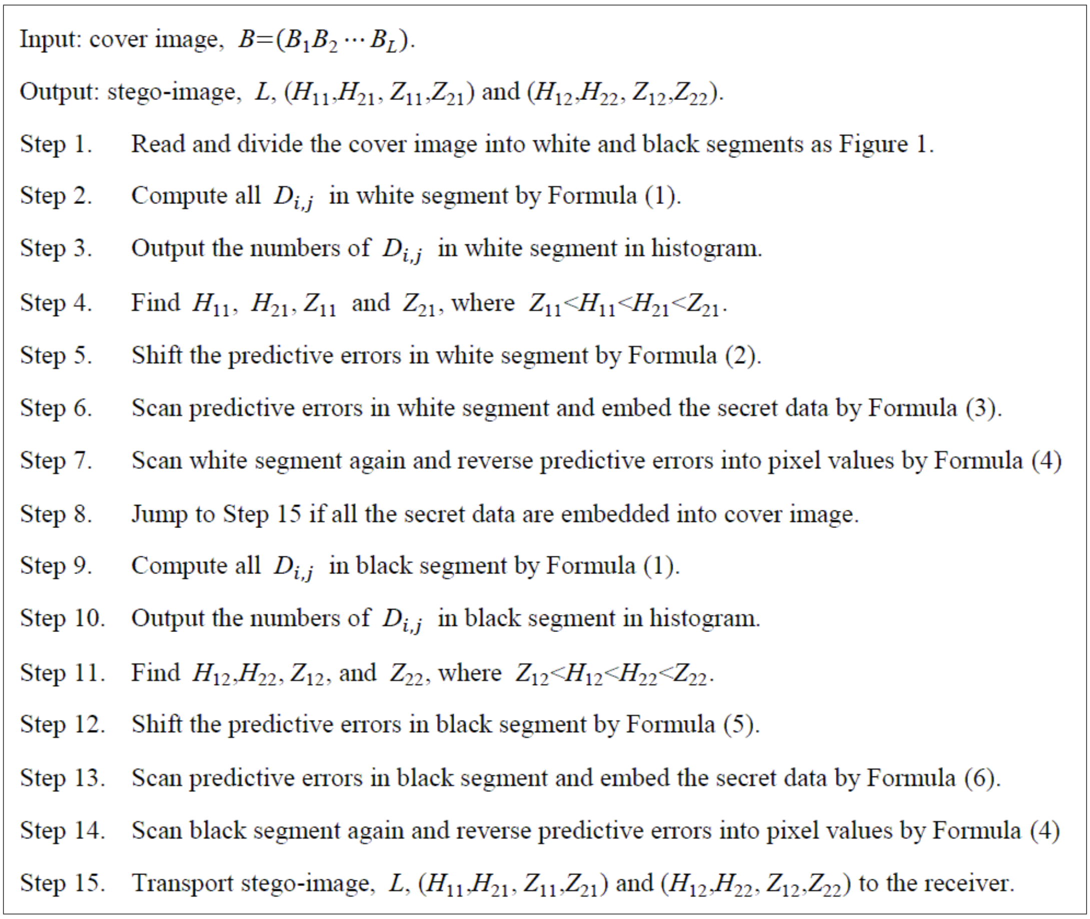
$$P_{i,j} = D_{i,j} + \left\lfloor \frac{\Sigma_{P_{i,j} \in S_{i,j}} P_{i,j}}{|S_{i,j}|} \right\rfloor \tag{4}$$

If there are remaining secret data, the sender compiles the second part of embedding algorithm for black segment as follows. In the second part of the algorithm, the sender computes all the predictive errors in black segment by Formula (1). The sender outputs the numbers of predictive errors of black segment and gets the histogram of predictive errors. The sender finds $(H_{12}, H_{22}, Z_{12}, Z_{22})$, where $Z_{12} < H_{12} < H_{22} < Z_{22}$. The predictive errors in the range of (Z_{12}, H_{12}) and (H_{22}, Z_{22}) are shifted by Formula (5) to generate the spaces nearby peak points.

$$D_{i,j} = \begin{cases} D_{i,j} - 1, \text{if } Z_{12} < D_{i,j} < H_{12} \\ D_{i,j} + 1, \text{if } H_{22} < D_{i,j} < Z_{22} \end{cases} \tag{5}$$

The sender scans predictive errors from top to bottom and left to right until all the predictive errors are scanned. When scanned predictive error is equal to H_{12} or H_{22}, the sender embeds secret bit B_k into cover image by Formula (6).

Predictive errors are scanned again and reversed into pixel values by Formula (4). At last, the sender gets stego-image, L, $(H_{11}, H_{21}, Z_{11}, Z_{21})$ and $(H_{12}, H_{22}, Z_{12}, Z_{22})$, and transports them to the receiver. When receiving the information, the receiver can use the extracting algorithm to extract secret data from stego-image. The embedding algorithm of Yang and Tsai's method is shown in Figure 2.



```
Input: cover image, $B=(B_1B_2\cdots B_L)$.
Output: stego-image, $L$, $(H_{11},H_{21}, Z_{11},Z_{21})$ and $(H_{12},H_{22}, Z_{12},Z_{22})$.
Step 1.   Read and divide the cover image into white and black segments as Figure 1.
Step 2.   Compute all $D_{i,j}$ in white segment by Formula (1).
Step 3.   Output the numbers of $D_{i,j}$ in white segment in histogram.
Step 4.   Find $H_{11}$, $H_{21}$, $Z_{11}$ and $Z_{21}$, where $Z_{11}<H_{11}<H_{21}<Z_{21}$.
Step 5.   Shift the predictive errors in white segment by Formula (2).
Step 6.   Scan predictive errors in white segment and embed the secret data by Formula (3).
Step 7.   Scan white segment again and reverse predictive errors into pixel values by Formula (4)
Step 8.   Jump to Step 15 if all the secret data are embedded into cover image.
Step 9.   Compute all $D_{i,j}$ in black segment by Formula (1).
Step 10.  Output the numbers of $D_{i,j}$ in black segment in histogram.
Step 11.  Find $H_{12}$,$H_{22}$, $Z_{12}$, and $Z_{22}$, where $Z_{12}<H_{12}<H_{22}<Z_{22}$.
Step 12.  Shift the predictive errors in black segment by Formula (5).
Step 13.  Scan predictive errors in black segment and embed the secret data by Formula (6).
Step 14.  Scan black segment again and reverse predictive errors into pixel values by Formula (4)
Step 15.  Transport stego-image, $L$, $(H_{11},H_{21}, Z_{11},Z_{21})$ and $(H_{12},H_{22}, Z_{12},Z_{22})$ to the receiver.
```

Figure 2. The embedding algorithm of Yang and Tsai's method

$$D_{i,j} = \begin{cases} D_{i,j} + 1, \text{ if } D_{i,j} = H_{22} \text{ and } B_k = 1 \\ D_{i,j} - 1, \text{ if } D_{i,j} = H_{12} \text{ and } B_k = 1 \\ \quad D_{i,j}, \text{ if } D_{i,j} = H_{12} || H_{22} \text{ and } B_k = 0 \end{cases} \tag{6}$$

To understand Yang and Tsai's method, we give the following example. Assume that the sender wants to embed the secret data $B = (011011110101)$ into the cover image in Figure 3. The sender divides the cover image into white and black segments and computes all the $D_{i,j}$ of white segment by Formula (1). The results are shown in Figure 4. Obviously, the sender gets $H_{11} = 0, H_{21} = 1, Z_{11} = -1$ and $Z_{21} = 2$. The sender embeds secret data (0110111101) into the cover image by Formula (3). After shifting predictive errors, the sender gets the embedded histogram and shifted predictive errors as Figure 5. The sender scans predictive errors again and reverses the predictive errors into pixel values by Formula (4). The results are shown in Figure 6. To embed remaining secret data (01), the sender computes all predictive errors of black segment by Formula (1). The results are shown in Figure 7. Obviously, the sender gets $H_{12} = 0, H_{22} = 1, Z_{12} = -2$ and $Z_2 2 = 3$. Then the sender shifts $D_{i,j}$ by Formula (5) and gets Figure 8. The sender embeds secret data (01) into the image by Formula (6). The sender gets the embedded histogram as Figure 9. The sender scans predictive errors again and reverses the predictive errors into pixel values by Formula (4). The results are shown as Figure 10. The secret data are completely embedded into cover image so far. The sender transports stego-image, $L = 12, H_{11} = 0, H_{21} = 1, Z_{11} = -1, Z_{21} = 2, H_{12} = 0, H_{22} = 1, Z_{12} = -2$ and $Z_{22} = 3$ to the receiver. The receiver uses this information with extracting algorithm to get the secret data and cover image.

2.2. **Non-Adjacent Form(NAF) Format** [1, 7, 8]**.** NAF format is the ternary string satisfying three important characteristics:

1. The data are represented by $\{0, 1, \bar{1}\}$;
2. The probability of appearing 0 is about 66%;
3. $\bar{1}$ and 1 are never adjacent.

We use these characteristics to hide the secret data for keeping high quality of stego-image. Before we introduce the NAF algorithm, which transfers the binary string $B = (B_1 B_2 \cdots B_L) = (011011110101)$ into the ternary string N by scanning bits of B from right to left. Let the corresponding NAF format of B be $N = (N_1 N_2 \cdots N_{L'})$, where $L' = L + 1$. We list the NAF algorithm in Figure 11. Note that R is the carrier whose initial value is zero.

To understand the NAF algorithm, we use the secret data $B = (011011110101) = (B_1 B_2 \cdots B_{12})$ in Section 2.1. We set $R = 0$ and show the process of running NAF algorithm in Figure 12.

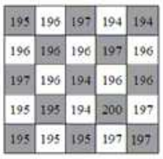
According to Figure 12, we get the NAF format of secret data $N = (0100\bar{1}000\bar{1}0101)$.

3. **Our Method.**

In this section, we propose TDH method and give an example to understand our method easily. Furthermore, by improving the amount of the embedded secret data in TDH method, we propose C-TDH method.

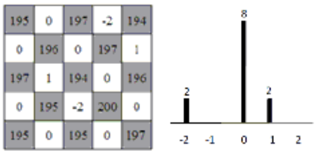
3.1. **Ternary Data Hiding (TDH).** Assume that the sender wants to embed ternary secret data $N = (N_1 N_2 \cdots N_{L'})$ into cover image. The sender divides the cover image into white and black segments as shown in Figure 1. The sender compiles the algorithm with one segment at once. In the first part of the algorithm, the sender computes all predictive errors in white segment by Formula (1). The sender outputs the numbers of predictive errors of white segment and gets the histogram of predictive errors. Unlike Yang and Tsai's method, the sender only finds (H_{11}, Z_{11}, Z_{21}) where $Z_{11} < H_{11} < Z_{21}$. The predictive errors in the range of (Z_{11}, H_{11}) and (H_{11}, Z_{21}) are shifted by Formula (7) to make the spaces nearby the peak point H_{11}.

$$D_{i,j} = \begin{cases} D_{i,j} - 1, \text{if } Z_{11} < D_{i,j} < H_{11} \\ D_{i,j} + 1, \text{if } H_{21} < D_{i,j} < Z_{21} \end{cases} \tag{7}$$



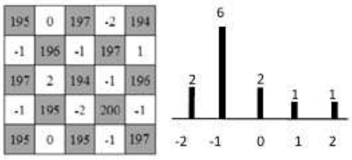
195	196	197	194	194
196	196	196	197	196
197	196	194	196	196
195	195	194	200	197
195	195	195	197	197

Figure 3. The cover image



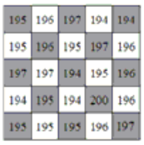
195	0	197	-2	194
0	196	0	197	1
197	1	194	0	196
0	195	-2	200	0
195	0	195	0	197

Figure 4. The predictive errors of white segment



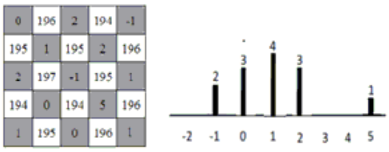
195	0	197	-2	194
-1	196	-1	197	1
197	2	194	-1	196
-1	195	-2	200	-1
195	0	195	-1	197

Figure 5. Embedded histogram and shifted predictive errors of white segment



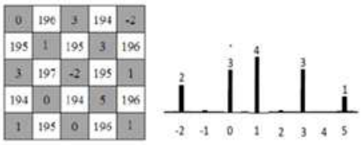
195	196	197	194	194
195	196	195	197	196
197	197	194	195	196
194	195	194	200	196
195	195	195	196	197

Figure 6. The stego-image after first algorithm



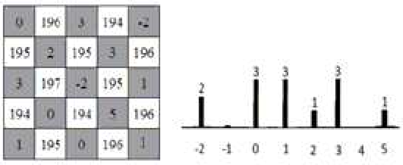
0	196	2	194	-1
195	1	195	2	196
2	197	-1	195	1
194	0	194	5	196
1	195	0	196	1

Figure 7. The predictive errors of black segment



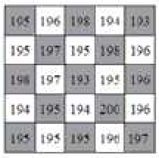
0	196	3	194	-2
195	1	195	3	196
3	197	-2	195	1
194	0	194	5	196
1	195	0	196	1

Figure 8. The shifted predictive errors of black segment



0	196	3	194	-2
195	2	195	3	196
3	197	-2	195	1
194	0	194	5	196
1	195	0	196	1

Figure 9. Embedded histogram of black segment



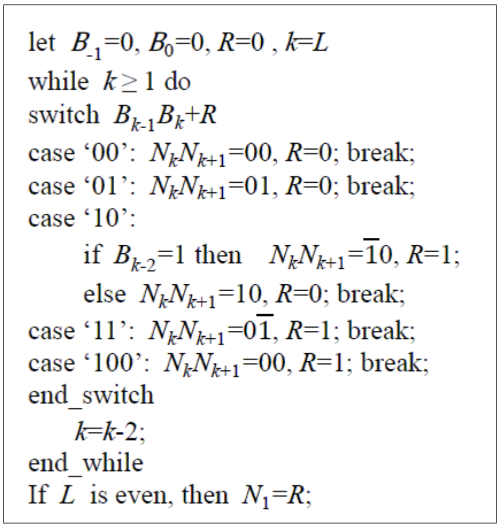
195	196	198	194	193
195	197	195	198	196
198	197	193	195	196
194	195	194	200	196
195	195	195	196	197

Figure 10. The stego-image

The sender scans all predictive errors. When the scanned predictive error is equal to the peak point H_{11}, the sender shifts the predictive error and embeds secret digit into cover image by Formula (8).

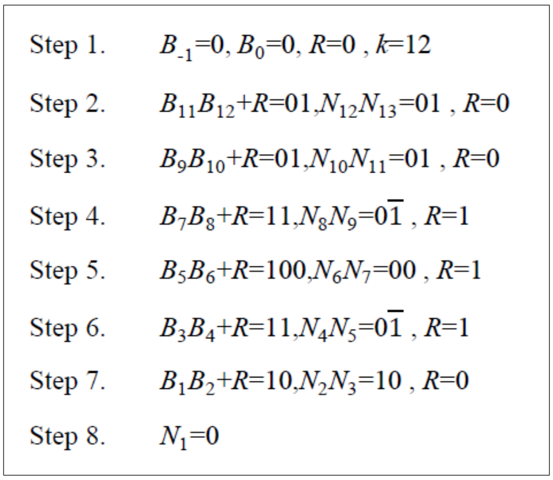
$$D_{i,j} = \begin{cases} D_{i,j}+1, \text{ if } D_{i,j}=H_{11} \text{ and } N_k=1 \\ D_{i,j}-1, \text{ if } D_{i,j}=H_{11} \text{ and } N_k=\bar{1} \\ D_{i,j}, \text{ if } D_{i,j}=H_{11} \text{ and } N_k=0 \end{cases} \tag{8}$$

Predictive errors are scanned again and reversed into pixel values by Formula (4). If there are remaining secret data, the sender compiles the second part of the algorithm for black segment as follows. The sender computes all predictive errors in black segment by Formula (1). The sender outputs the numbers of predictive errors of black segment and



```
let $B_{-1}=0$, $B_0=0$, $R=0$ , $k=L$
while $k \geq 1$ do
switch $B_{k-1}B_k+R$
case '00': $N_kN_{k+1}=00$, $R=0$; break;
case '01': $N_kN_{k+1}=01$, $R=0$; break;
case '10':
        if $B_{k-2}=1$ then   $N_kN_{k+1}=\overline{1}0$, $R=1$;
        else $N_kN_{k+1}=10$, $R=0$; break;
case '11': $N_kN_{k+1}=0\overline{1}$, $R=1$; break;
case '100': $N_kN_{k+1}=00$, $R=1$; break;
end_switch
      $k=k-2$;
end_while
If $L$ is even, then $N_1=R$;
```

FIGURE 11. The algorithm of NAF format



Step 1. $B_{-1}=0, B_0=0, R=0\ , k=12$

Step 2. $B_{11}B_{12}+R=01, N_{12}N_{13}=01\ , R=0$

Step 3. $B_9B_{10}+R=01, N_{10}N_{11}=01\ , R=0$

Step 4. $B_7B_8+R=11, N_8N_9=0\overline{1}\ , R=1$

Step 5. $B_5B_6+R=100, N_6N_7=00\ , R=1$

Step 6. $B_3B_4+R=11, N_4N_5=0\overline{1}\ , R=1$

Step 7. $B_1B_2+R=10, N_2N_3=10\ , R=0$

Step 8. $N_1=0$

FIGURE 12. An example of NAF format

gets histogram of predictive errors. The sender finds (H_{12}, Z_{12}, Z_{22}), where $Z_{12} < H_{12} < Z_{22}$. The predictive errors in the range of (Z_{12}, H_{12}) and (H_{12}, Z_{22}) are shifted by Formula (9) to generate the spaces nearby the peak point H_{12}.

$$D_{i,j} = \begin{cases} D_{i,j} - 1, \text{if } Z_{12} < D_{i,j} < H_{12} \\ D_{i,j} + 1, \text{if } H_{12} < D_{i,j} < Z_{22} \end{cases} \tag{9}$$

The sender scans all predictive errors. When the scanned predictive error is equal to peak point H_{12}, the sender shifts the predictive error and embeds secret digit N_k into cover image by Formula (10).

$$D_{i,j} = \begin{cases} D_{i,j} + 1, \text{ if } D_{i,j} = H_{12} \text{ and } N_k = 1 \\ D_{i,j} - 1, \text{ if } D_{i,j} = H_{12} \text{ and } N_k = \bar{1} \\ D_{i,j}, \text{ if } D_{i,j} = H_{12} \text{ and } N_k = 0 \end{cases} \tag{10}$$

Predictive errors are scanned again and reversed into pixel values by Formula (4). At last, the sender transports stego-image, L', (H_{11}, Z_{11}, Z_{21}) and (H_{12}, Z_{12}, Z_{22}) to the receiver.

When receiving the information, the receiver divides stego-image into white and black segments as Figure 1. The receiver compiles the extracting algorithm with one segment at once. In the first part of the extracting algorithm, the receiver computes all predictive errors in black segment by Formula (1). The receiver scans all predictive errors. When the scanned predictive error is equal or next to the peak point H_{12}, the receiver extracts the embedded secret digit N_k by Formula (11).

$$N_k = \begin{cases} 1, \text{ if } D_{i,j} = H_{12} + 1 \\ \bar{1}, \text{ if } D_{i,j} = H_{12} - 1 \\ 0, \text{ if } D_{i,j} = H_{12} \end{cases} \tag{11}$$

After extracting embedded secret data, the receiver shifts the predictive errors by Formula (12).

$$D_{i,j} = \begin{cases} D_{i,j} - 1, \text{if } Z_{12} < D_{i,j} < H_{12} \\ D_{i,j} + 1, \text{if } H_{12} < D_{i,j} < Z_{22} \end{cases} \tag{12}$$

To get the cover image, the receiver scans the predictive errors again and reverses the predictive errors into pixel values by Formula (4). The receiver then swaps black segment for white segment and computes the second part of the extracting algorithm as follows. The receiver computes all predictive errors in white segment by Formula (1). The receiver scans all predictive errors. When the scanned predictive error is equal or next to peak point H_{11}, the receiver extracts the embedded secret digit N_k by Formula (13).

$$N_k = \begin{cases} 1, \text{ if } D_{i,j} = H_{11} + 1 \\ \bar{1}, \text{ if } D_{i,j} = H_{11} - 1 \\ 0, \text{ if } D_{i,j} = H_{11} \end{cases} \tag{13}$$

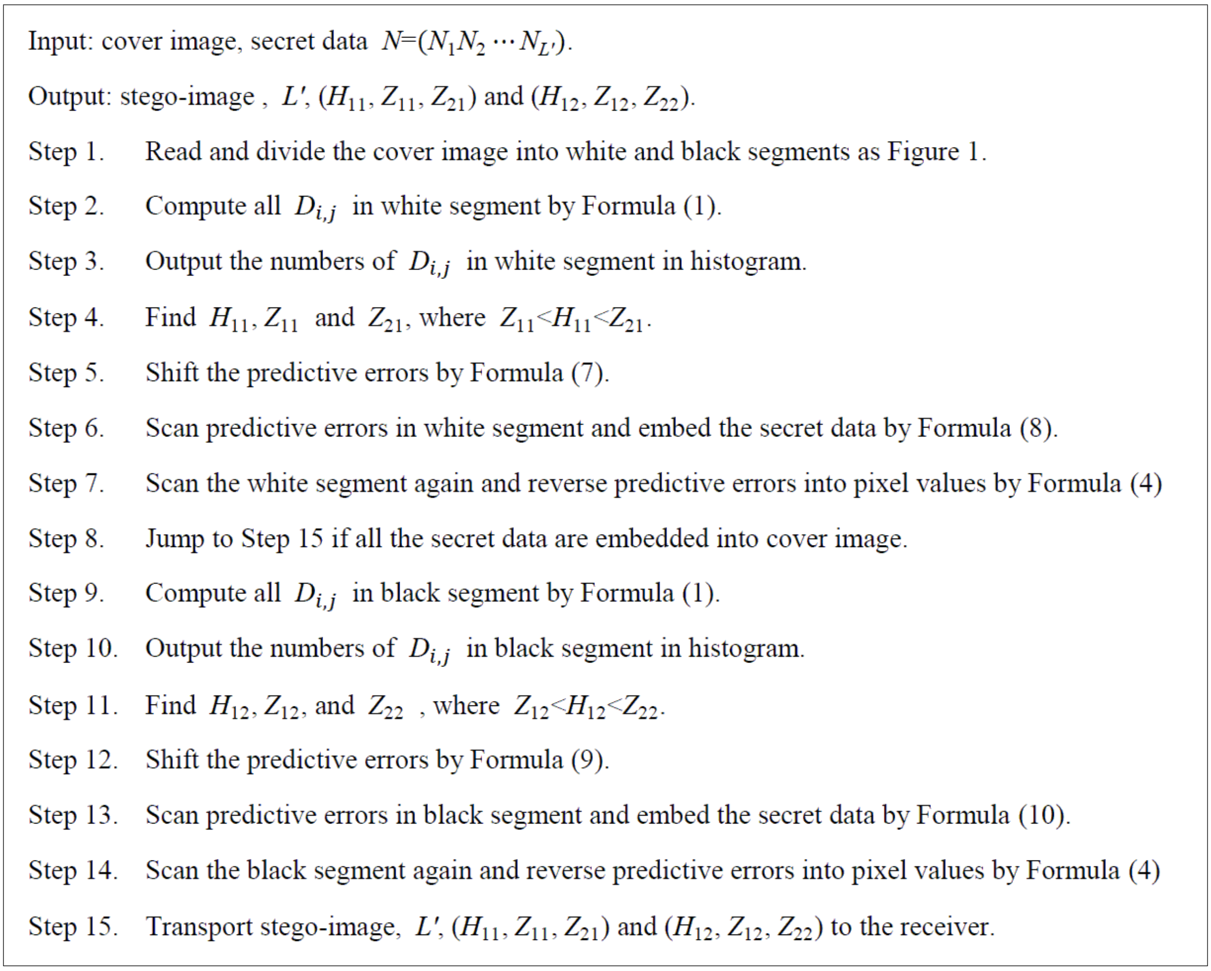
After extracting embedded secret data, the receiver shifts the predictive errors by Formula (14).

$$D_{i,j} = \begin{cases} D_{i,j} - 1, \text{if } Z_{11} < D_{i,j} < H_{11} \\ D_{i,j} + 1, \text{if } H_{11} < D_{i,j} < Z_{21} \end{cases} \tag{14}$$

To get the cover image, the receiver scans predictive errors again and reverses predictive errors into pixel values by Formula (4). When $L' \leq |H_{11}| + |H_{12}|$, the receiver has to combine the extracted secret data of two segments and delete redundant N_k where $k > L'$ to get real secret data.

The embedding algorithm and the extracting algorithm of TDH are shown in Figure 13 and Figure 14, respectively.

Like the example in Section 2.1, the sender wants to embed the secret data $N = (0100\bar{1}000\bar{1}0101)$ into the cover image in Figure 3. The sender divides the cover image into white and black segments and computes all the predictive errors of white segment by Formula (1). The sender outputs the numbers of predictive errors in white segment and gets the histogram of predictive errors as Figure 15 shows. As opposed to Yang and Tsais method, our method only gets $H_{11} = 0$, $Z_{11} = -1$ and $Z_{21} = 2$. Then, the sender shifts the predictive errors in the range (-1,0) and (0,2) to generate the spaces nearby



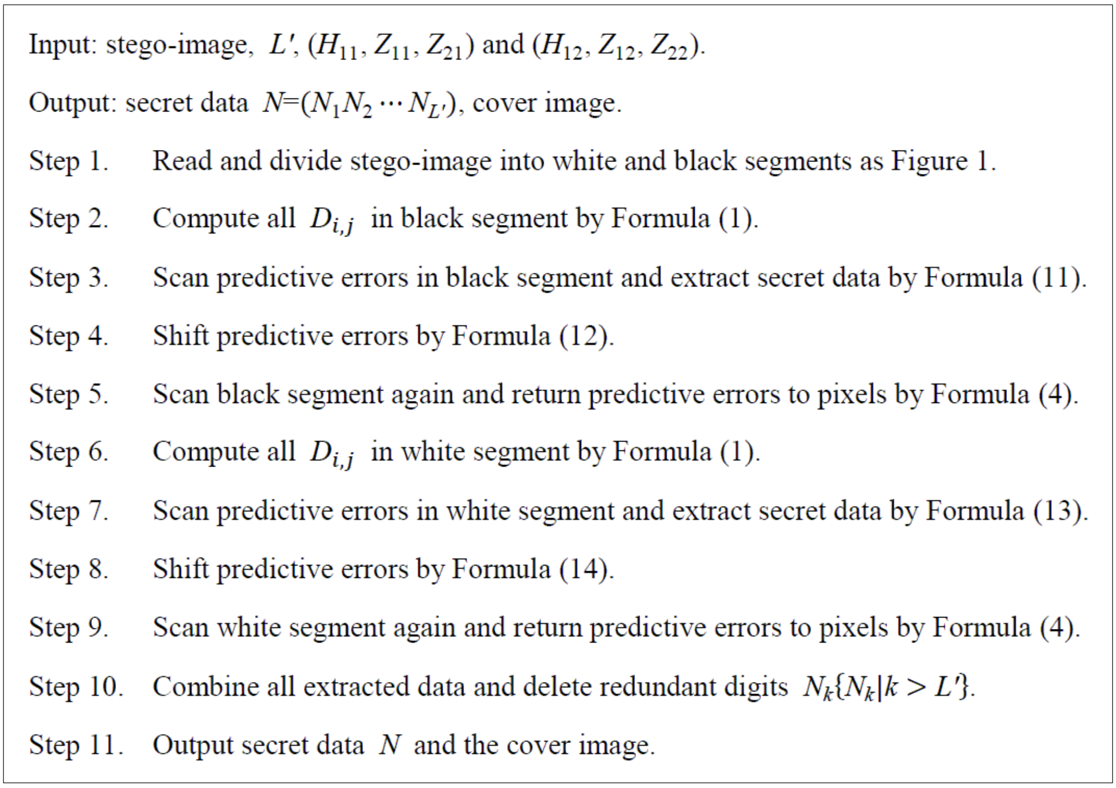
```
Input: cover image, secret data $N$=($N_1N_2 \cdots N_{L'}$).
Output: stego-image , $L'$, ($H_{11}$, $Z_{11}$, $Z_{21}$) and ($H_{12}$, $Z_{12}$, $Z_{22}$).
Step 1.   Read and divide the cover image into white and black segments as Figure 1.
Step 2.   Compute all $D_{i,j}$ in white segment by Formula (1).
Step 3.   Output the numbers of $D_{i,j}$ in white segment in histogram.
Step 4.   Find $H_{11}$, $Z_{11}$ and $Z_{21}$, where $Z_{11}<H_{11}<Z_{21}$.
Step 5.   Shift the predictive errors by Formula (7).
Step 6.   Scan predictive errors in white segment and embed the secret data by Formula (8).
Step 7.   Scan the white segment again and reverse predictive errors into pixel values by Formula (4)
Step 8.   Jump to Step 15 if all the secret data are embedded into cover image.
Step 9.   Compute all $D_{i,j}$ in black segment by Formula (1).
Step 10.  Output the numbers of $D_{i,j}$ in black segment in histogram.
Step 11.  Find $H_{12}$, $Z_{12}$, and $Z_{22}$ , where $Z_{12}<H_{12}<Z_{22}$.
Step 12.  Shift the predictive errors by Formula (9).
Step 13.  Scan predictive errors in black segment and embed the secret data by Formula (10).
Step 14.  Scan the black segment again and reverse predictive errors into pixel values by Formula (4)
Step 15.  Transport stego-image, $L'$, ($H_{11}$, $Z_{11}$, $Z_{21}$) and ($H_{12}$, $Z_{12}$, $Z_{22}$) to the receiver.
```

Figure 13. The embedding algorithm of TDH

peak point H_{11} by Formula (7). The results are shown in Figure 16. The sender scans the predictive errors and embeds secret data $(0100\bar{1}000)$ by Formula (8). After embedding secret data $(0100\bar{1}000)$, the sender gets the embedded histogram as Figure 17 shows. The sender reverses the predictive errors into pixel values by Formula (4). The results are shown in Figure 18.

To embed the remaining secret data $(\bar{1}0101)$, the sender swaps white segment for black segment and computes all the predictive errors of black segment by Formula (1). Given the numbers of predictive errors in black segment, the sender gets the histogram of predictive errors as Figure 19 shows. Obviously, the sender gets $H_{12} = 0$, $Z_{12} = -2$ and $Z_{22} = 3$. The sender then shifts the predictive errors in the range of (-2,0) and (0,3) by Formula (9) as Figure 20 shows. The sender scans the predictive errors and embeds secret data $N = (\bar{1}0101)$ by Formula (10). After embedding secret data $(\bar{1}0101)$, the sender computes the embedded histogram and shifted predictive errors as Figure 21 shows. The sender reverses the predictive errors into pixel values by Formula (4). The results are shown in Figure 22. Now the secret data are completely embedded into the cover image. The sender transports stego-image, $L' = 13$, $H_{11} = 0$, $Z_{11} = -1$, $Z_{21} = 2$, $H_{12} = 0$, $Z_{12} = -2$ and $Z_{22} = 3$ to the receiver.

After receiving the information, the receiver divides the stego-image into white and black segments as Figure 22 shows. Then, the receiver computes all the predictive errors of black segment by Formula (1). The results are shown in Figure 21. Then the receiver scans the predictive errors and extracts secret data $(\bar{1}0101)$ by Formula (11). After extracting



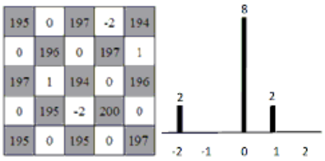
```
Input: stego-image, $L'$, $(H_{11}, Z_{11}, Z_{21})$ and $(H_{12}, Z_{12}, Z_{22})$.
Output: secret data $N$=$(N_1 N_2 \cdots N_{L'})$, cover image.
Step 1.   Read and divide stego-image into white and black segments as Figure 1.
Step 2.   Compute all $D_{i,j}$ in black segment by Formula (1).
Step 3.   Scan predictive errors in black segment and extract secret data by Formula (11).
Step 4.   Shift predictive errors by Formula (12).
Step 5.   Scan black segment again and return predictive errors to pixels by Formula (4).
Step 6.   Compute all $D_{i,j}$ in white segment by Formula (1).
Step 7.   Scan predictive errors in white segment and extract secret data by Formula (13).
Step 8.   Shift predictive errors by Formula (14).
Step 9.   Scan white segment again and return predictive errors to pixels by Formula (4).
Step 10.  Combine all extracted data and delete redundant digits $N_k\{N_k|k > L'\}$.
Step 11.  Output secret data $N$ and the cover image.
```

Figure 14. The extracting algorithm of TDH

embedded data, the receiver shifts the predictive errors by Formula (12) as Figure 19 shows. The receiver scans black segment again and reverses the predictive errors into pixel values by Formula (4). The results are shown in Figure 18. To extract the secret data in white segment, the receiver computes the predictive errors of white segment by Formula (1). The results are shown in Figure 17. Then the receiver scans predictive errors and extracts embedded data (0100$\bar{1}$000) by Formula (13). After extracting the embedded data, the receiver shifts the predictive errors by Formula (14) as Figure 15 shows. The receiver scans white segment again and reverses the predictive errors into pixel values by Formula (4) and gets the cover image of Figure 3. Since $L' = |H_{11}| + |H_{12}| = 13$, the receiver combines the extracted secret data of the two segments and gets the real secret data $N = (0100\bar{1}000\bar{1}0101)$.

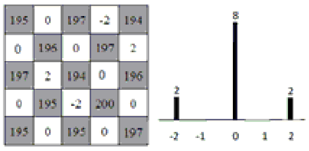
3.2. **Coded-Ternary Data Hiding(C-TDH).** Since the capacity of TDH depends on the number of the highest point of the histogram, how to improve the capacity of TDH becomes an important issue. According to the characteristics of NAF, 1 and $\bar{1}$ are never adjacent, we propose Coded-Ternary Data Hiding (C_TDH) by using the concept of coding. We first find that secret data can be represented by three symbols, 10, $\bar{1}0$, and 0. We then try to embed at most two secret digits in one pixel.

The embedding and extracting algorithm of C-TDH is almost the same with that of TDH. But when the sender wants to embed ternary secret data into an image, the sender uses Formula (15) and Formula (16) to embed secret data. In addition, when the sender wants to extract secret data from the image, the sender uses Formula (17) and Formula (18) to extract secret data.



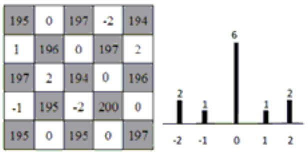
195	0	197	-2	194
0	196	0	197	1
197	1	194	0	196
0	195	-2	200	0
195	0	195	0	197

Figure 15. The predictive errors of white segment



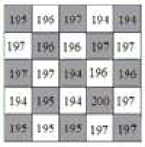
195	0	197	-2	194
0	196	0	197	2
197	2	194	0	196
0	195	-2	200	0
195	0	195	0	197

Figure 16. The shifted predictive errors of white segment



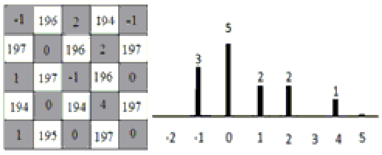
195	0	197	-2	194
1	196	0	197	2
197	2	194	0	196
-1	195	-2	200	0
195	0	195	0	197

Figure 17. Embedded histogram of white segment



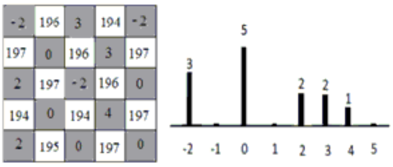
195	196	197	194	194
197	196	196	197	197
197	197	194	196	196
194	195	194	200	197
195	195	195	197	197

Figure 18. The stego-image after first algorithm



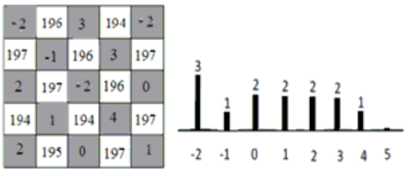
-1	196	2	194	-1
197	0	196	2	197
1	197	-1	196	0
194	0	194	4	197
1	195	0	197	0

Figure 19. The predictive errors of black segment



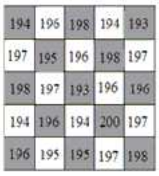
-2	196	3	194	-2
197	0	196	3	197
2	197	-2	196	0
194	0	194	4	197
2	195	0	197	0

Figure 20. The shifted predictive errors of black segment



-2	196	3	194	-2
197	-1	196	3	197
2	197	-2	196	0
194	1	194	4	197
2	195	0	197	1

Figure 21. Embedded histogram of black segment



194	196	198	194	193
197	195	196	198	197
198	197	193	196	196
194	196	194	200	197
196	195	195	197	198

Figure 22. The stego-image

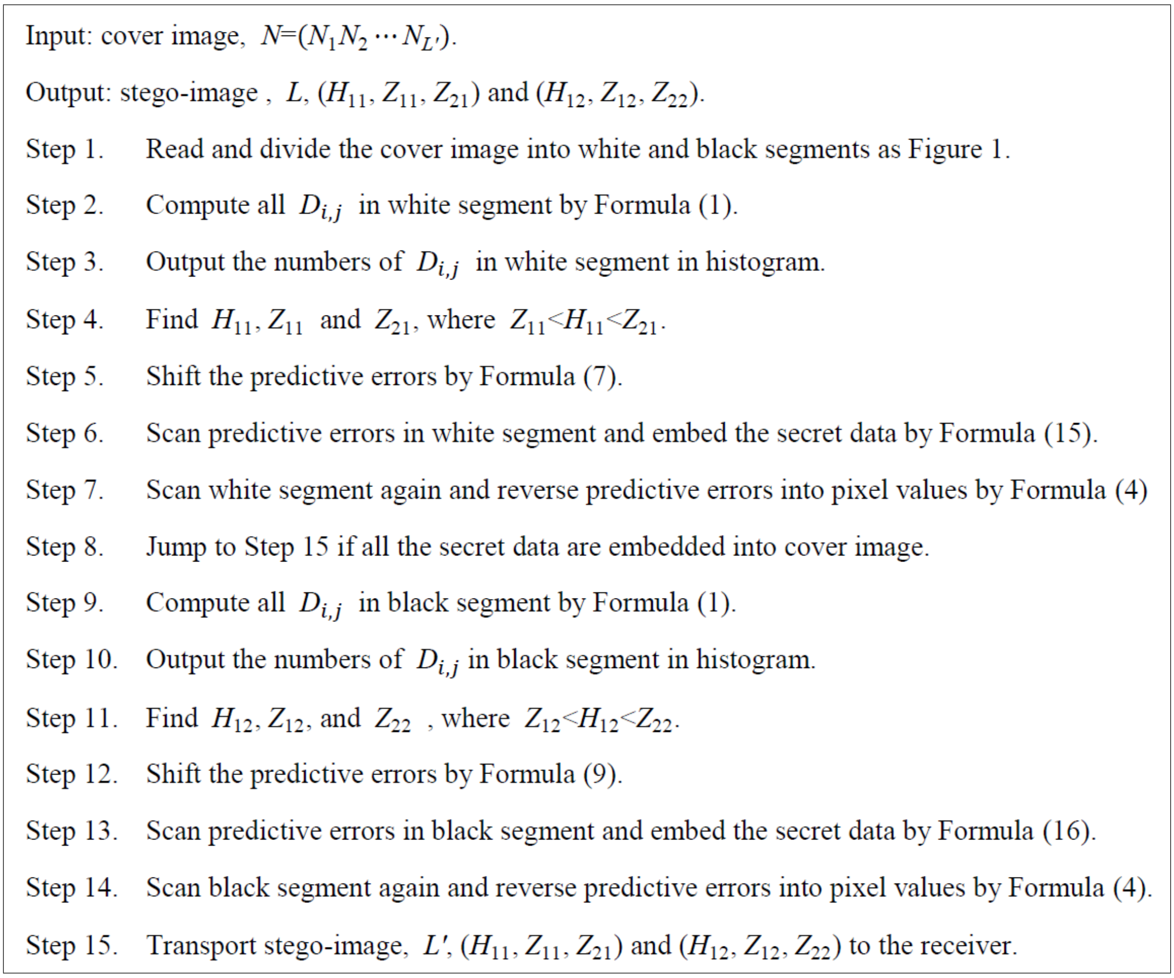
$$D_{i,j} = \begin{cases} D_{i,j}+1, k=k+2, & \text{if } D_{i,j}=H_{11} \text{ and } N_k=10 \\ D_{i,j}-1, k=k+2, & \text{if } D_{i,j}=H_{11} \text{ and } N_k=\bar{1}0 \\ D_{i,j}, k=k+1, & \text{if } D_{i,j}=H_{11} \text{ and } N_k=0 \end{cases} \tag{15}$$

$$D_{i,j} = \begin{cases} D_{i,j}+1, k=k+2, & \text{if } D_{i,j}=H_{12} \text{ and } N_k=10 \\ D_{i,j}-1, k=k+2, & \text{if } D_{i,j}=H_{12} \text{ and } N_k=\bar{1}0 \\ D_{i,j}, k=k+1, & \text{if } D_{i,j}=H_{12} \text{ and } N_k=0 \end{cases} \tag{16}$$

$$\begin{cases} N_kN_{k+1} = 10, k = k+2, & \text{if } D_{i,j} = H_{12}+1 \\ N_kN_{k+1} = \bar{1}0, k = k+2, & \text{if } D_{i,j} = H_{12}-1 \\ N_k = 0, k = k+1, & \text{if } D_{i,j} = H_{12}+1 \end{cases} \tag{17}$$

$$\begin{cases} N_kN_{k+1} = 10, k = k+2, & \text{if } D_{i,j} = H_{11}+1 \\ N_kN_{k+1} = \bar{1}0, k = k+2, & \text{if } D_{i,j} = H_{11}-1 \\ N_k = 0, k = k+1, & \text{if } D_{i,j} = H_{11}+1 \end{cases} \tag{18}$$

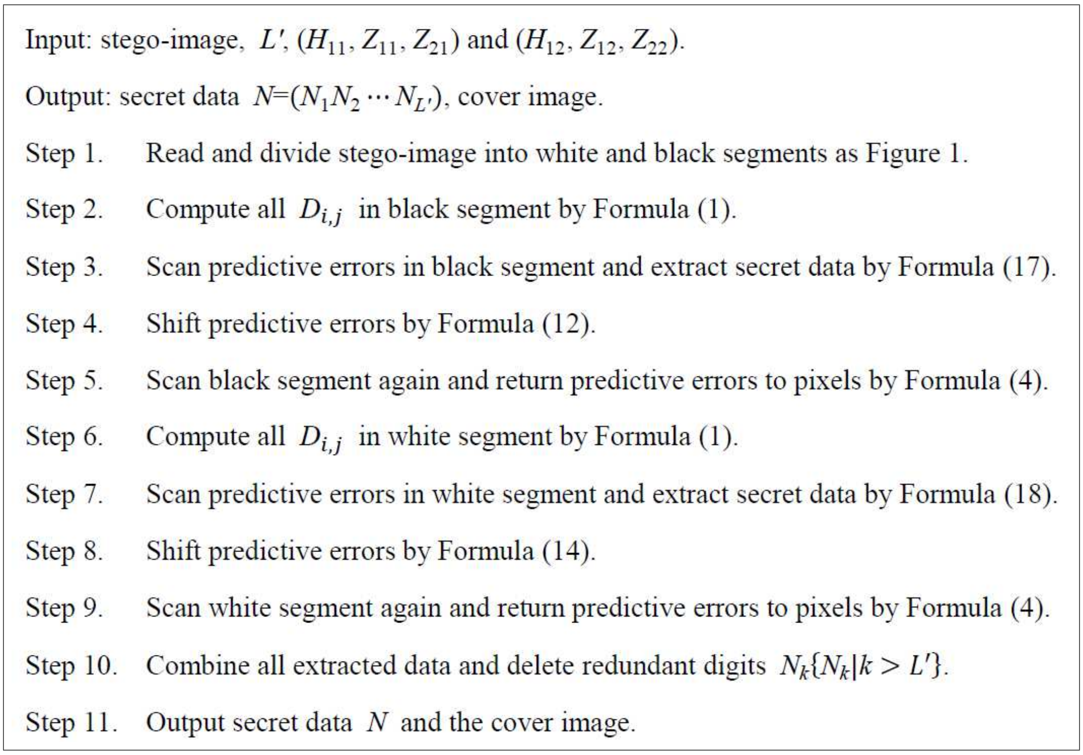
The embedding algorithm and the extracting algorithm of C-TDH are shown in Figure 23 and Figure 24, respectively.



```
Input: cover image, $N$=($N_1N_2 \cdots N_{L'}$).
Output: stego-image , $L$, ($H_{11}$, $Z_{11}$, $Z_{21}$) and ($H_{12}$, $Z_{12}$, $Z_{22}$).
Step 1.   Read and divide the cover image into white and black segments as Figure 1.
Step 2.   Compute all $D_{i,j}$ in white segment by Formula (1).
Step 3.   Output the numbers of $D_{i,j}$ in white segment in histogram.
Step 4.   Find $H_{11}$, $Z_{11}$ and $Z_{21}$, where $Z_{11}$<$H_{11}$<$Z_{21}$.
Step 5.   Shift the predictive errors by Formula (7).
Step 6.   Scan predictive errors in white segment and embed the secret data by Formula (15).
Step 7.   Scan white segment again and reverse predictive errors into pixel values by Formula (4)
Step 8.   Jump to Step 15 if all the secret data are embedded into cover image.
Step 9.   Compute all $D_{i,j}$ in black segment by Formula (1).
Step 10.  Output the numbers of $D_{i,j}$ in black segment in histogram.
Step 11.  Find $H_{12}$, $Z_{12}$, and $Z_{22}$ , where $Z_{12}$<$H_{12}$<$Z_{22}$.
Step 12.  Shift the predictive errors by Formula (9).
Step 13.  Scan predictive errors in black segment and embed the secret data by Formula (16).
Step 14.  Scan black segment again and reverse predictive errors into pixel values by Formula (4).
Step 15.  Transport stego-image, $L'$, ($H_{11}$, $Z_{11}$, $Z_{21}$) and ($H_{12}$, $Z_{12}$, $Z_{22}$) to the receiver.
```

Figure 23. The embeding algorithm of C-TDH

We give an example of embedding secret data $N = (0100\bar{1}000\bar{1}0101)$ in Figure 3. According to three symbols, 10, $\bar{1}0$, and 0, secret data N can be viewed as $N = (\underline{0}\ \underline{10}\ \underline{0}\ \underline{\bar{1}0}\ \underline{0}\ \underline{0}\ \underline{\bar{1}0}\ \underline{10}\ \underline{1})$. The sender divides the cover image into white and black segments and computes all the predictive errors of white segment by Formula (1). Then the sender outputs the numbers of predictive errors in white segment and gets the histogram of predictive errors in Figure 25. The sender gets $H_{11} = 0$, $Z_{11} = -1$ and $Z_{21} = 2$. Then the sender shifts the predictive errors in the range of (-1, 0) and (0, 2) to generate the spaces nearby peak point H_{11} by Formula (7). The results are shown in Figure 26. The sender

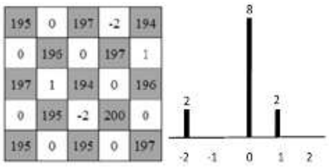


```
Input: stego-image, $L'$, $(H_{11}, Z_{11}, Z_{21})$ and $(H_{12}, Z_{12}, Z_{22})$.
Output: secret data $N$=$(N_1 N_2 \cdots N_{L'})$, cover image.
Step 1.  Read and divide stego-image into white and black segments as Figure 1.
Step 2.  Compute all $D_{i,j}$ in black segment by Formula (1).
Step 3.  Scan predictive errors in black segment and extract secret data by Formula (17).
Step 4.  Shift predictive errors by Formula (12).
Step 5.  Scan black segment again and return predictive errors to pixels by Formula (4).
Step 6.  Compute all $D_{i,j}$ in white segment by Formula (1).
Step 7.  Scan predictive errors in white segment and extract secret data by Formula (18).
Step 8.  Shift predictive errors by Formula (14).
Step 9.  Scan white segment again and return predictive errors to pixels by Formula (4).
Step 10. Combine all extracted data and delete redundant digits $N_k\{N_k | k > L'\}$.
Step 11. Output secret data $N$ and the cover image.
```

Figure 24. The extracting algorithm of C-TDH

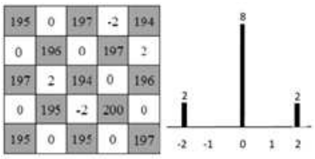
scans the predictive errors and embeds secret data $(\underline{0}\ \underline{10}\ \underline{0}\ \underline{\bar{1}0}\ \underline{0}\ \underline{0}\ \underline{\bar{1}0}\ \underline{10})$ by Formula (15). After embedding the secret data, the sender gets the embedded histogram as Figure 27 shows. The sender reverses the predictive errors into pixel values by Formula (4). The results are shown in Figure 28.

To embed remaining secret data $(\underline{1})$, the sender swaps white segment for black segment and computes all the predictive errors of black segment by Formula (1). Given the numbers of predictive errors in black segment, the sender gets the histogram of predictive errors in Figure 29. Obviously, the sender gets $H_{12} = 0$, $Z_{12} = -2$ and $Z_{22} = 3$. The sender shifts the predictive errors in the range of (-2, 0) and (0, 3) to generate the spaces nearby peak point by Formula (16) as show in Figure 30. The sender scans the predictive errors of black segment and embeds secret data $(\underline{1})$ into the image by Formula (10). After embedding secret data, the sender gets the embedded histogram as Figure 31 shows. The sender reverses the predictive errors into pixel values by Formula (4). The results are shown in Figure 32. Now the secret data are completely embedded into the cover image. The sender transports stego-image, $L' = 13$, $H_{11} = 0$, $Z_{11} = -1$, $Z_{21} = 2$, $H_{12} = 0$, $Z_{12} = -2$ and $Z_{22} = 3$ to the receiver. After receiving the information, the receiver divides the stego-image into white and black segments as Figure 32 shows. Then the receiver computes all the predictive errors of black segment by Formula (1). The results are shown in Figure 31. The receiver scans the predictive errors and extracts secret data $(\underline{10}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0})$ by Formula (17). After extracting the embedded data, the receiver shifts the predictive errors by Formula (12) as Figure 29 shows. The receiver scans black segment again and reverses the predictive errors into pixel values by Formula (4). The results are shown in Figure 28. To extract the secret data in white segment, the receiver computes the predictive errors of white segment by Formula (1). The results are shown in Figure 27. Then the receiver scans the predictive errors and extracts secret data $(\underline{0}\ \underline{10}\ \underline{0}\ \underline{\bar{1}0}\ \underline{0}\ \underline{0}\ \underline{\bar{1}0}\ \underline{10})$ by Formula (18). After extracting embedded data, the receiver shifts the predictive errors by Formula (14) as Figure 25 shows. The receiver scans white segment again and reverses the predictive errors into pixel values by Formula (4) and gets the cover image as Figure 3 shows. Since the redundant secret data are extracted when $L' < |H_{11}|+|H_{12}|$, the receiver has to combine the extracted secret data $(\underline{0}\ \underline{10}\ \underline{0}\ \underline{\bar{1}0}\ \underline{0}\ \underline{0}\ \underline{\bar{1}0}\ \underline{10}\ \underline{10}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0})$ of two segments and delete redundant N_k where $k > L'$ to get the real secret data $N = (0100\bar{1}000\bar{1}0101)$ in Figure 3. According to three symbols, 10, $\bar{1}0$, and 0, secret data N can be viewed as $N = (\underline{0}\ \underline{10}\ \underline{0}\ \underline{\bar{1}0}\ \underline{0}\ \underline{0}\ \underline{\bar{1}0}\ \underline{10}\ \underline{1})$.



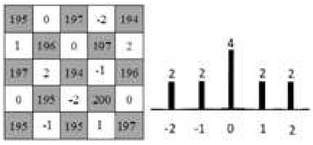
195	0	197	-2	194
0	196	0	197	1
197	1	194	0	196
0	195	-2	200	0
195	0	195	0	197

FIGURE 25. The predictive errors of white segment



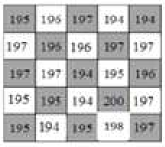
195	0	197	-2	194
0	196	0	197	2
197	2	194	0	196
0	195	-2	200	0
195	0	195	0	197

FIGURE 26. The shifted predictive errors of white segment



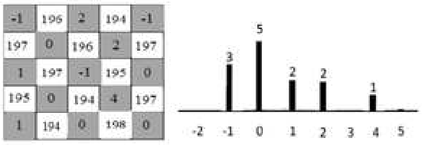
195	0	197	-2	194
1	196	0	197	2
197	2	194	-1	196
0	195	-2	200	0
195	-1	195	1	197

FIGURE 27. Embedded histogram of white segment



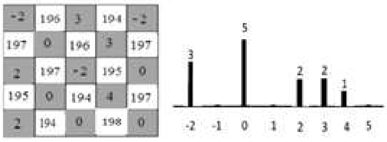
195	196	197	194	194
197	196	196	197	197
197	197	194	195	196
195	195	194	200	197
195	194	195	198	197

FIGURE 28. The stego-image after first algorithm



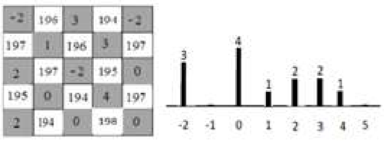
-1	196	2	194	-1
197	0	196	2	197
1	197	-1	195	0
195	0	194	4	197
1	194	0	198	0

FIGURE 29. The predictive errors of black segment



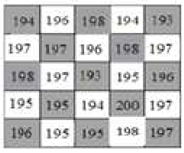
-2	196	3	194	-2
197	0	196	3	197
2	197	-2	195	0
195	0	194	4	197
2	194	0	198	0

FIGURE 30. The shifted predictive errors of black segment



-2	196	3	194	-2
197	1	196	3	197
2	197	-2	195	0
195	0	194	4	197
2	194	0	198	0

FIGURE 31. Embedded histogram of black segment



194	196	198	194	193
197	197	196	198	197
198	197	193	195	196
195	195	194	200	197
196	195	195	198	197

FIGURE 32. The stego-image

4. **Experimental result.** In this section, we show the experimental results of Yang and Tsai,s and our methods. Let M_1 be the number of highest peak point and M_2 be the number of second highest peak point. From the example in Section 2, the number of the highest peak point H_{11} in white segment is 8 and that of the highest peak point H_{21} in black segment is 4. We get $M_1 = 8 + 4 = 12$. On the other hand, the number of the

second highest peak point H_{21} in white segment is 2 and that of the second highest peak point H_{22} in black segment is 3. We have $M_2 = H_{21} + H_{22} = 5$.

We construct the secret data randomly and embed the secret data into ten images in Figure 33. In Table 1, we compute the maximum of capacity and the PSNR of Yang and Tsai's method and TDH in average case. Then, we show the capacity of TDH and C-TDH in Table 2. From the experimental results, THD gets higher PSNR's than Yang and Tsai's method. Furthermore C-TDH produces higher capacity than THD.

Figure 33. Ten images in our experiment

Table 1. The average PSNR of Yang and Tsai's method and TDH

Image Name	M_1	M_2	M_1/M_2	Yang and Tsai's method	TDH
				PSNR	PSNR
Airplane	11976	7099	1.687	62.52	66.30
Peppers	45572	23765	1.918	56.92	60.50
Boat	57120	28196	2.026	56.02	59.52
Lena	61481	29854	2.059	55.72	59.20
Holidayhouse	114699	38156	3.006	53.48	56.49
Swimmingpool	100613	19511	5.157	54.53	57.06
Snowtree	90318	15678	5.761	55.07	57.53
Dubai	112162	17352	6.464	54.20	56.59
Room	176220	21630	8.147	52.36	54.63
Cloudsea	195970	21070	9.301	51.96	54.17

5. **Discussion.** In this section, we compare Yang and Tsai's method with ours using the performance measures of capacity and time complexity. First, we compute the capacity of Yang and Tsai's method and ours as follows. Given an example of histogram in Figure 34, we can find the first two highest peak points, H_1 and H_2, and their corresponding zero points, Z_1 and Z_2. We let M_1 be the number of H_1, M_2 be the number of H_2, O_1 be the sum of the numbers in the range of (H_1, Z_1), O_2 be the sum of the numbers in the range of (H_1, H_2), and O_3 be the sum of the numbers in the range of (H_2, Z_2).

Assume that the secret data is random. In Yang and Tsai's method, the secret data B is represented by 0, 1. The number of pixel values which should be shifted is $O_1 + O_3 + \frac{1}{2}(M_1 + M_2)$, where $\frac{1}{2}$ denotes the probability of appearing 1 in B. In TDH, the secret

Table 2. The capacity of TDH and C-TDH

Image Name	TDH	C-TDH
	Max capacity	Max capacity
Airplane	11976	17964
Peppers	45572	68358
Boat	57120	85680
Lena	61481	92221
Holidayhouse	114699	172048
Swimmingpool	100613	150919
Snowtree	90318	135477
Dubai	112162	168243
Room	176220	264330
Cloudsea	195970	293955

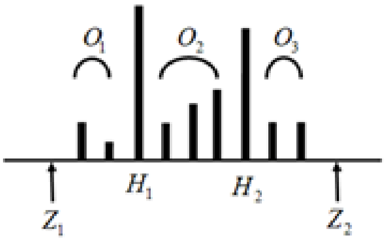
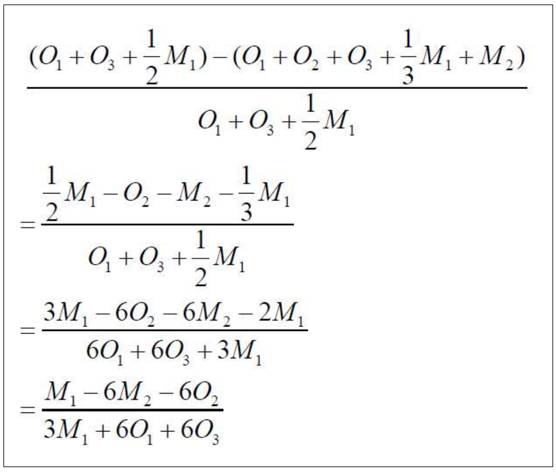


Figure 34. The distribution of histogram

data N is represented by 0, 1, $\bar{1}$. Therefore, the number of pixel values which should be shifted is $O_1 + O_2 + O_3 + M_2 + \frac{1}{3}M_1$, where $\frac{1}{3}$ denotes the probability of appearing 1 and $\bar{1}$ in N. Because the length of N is larger than that of B by one, we assume that N and B have the same length. In general, we have that O_2 is zero. Thus, if $M_1 > 6M_2$, the number of shifted pixel values in TDH is less than that in Yang and Tsai's method. In smooth images where the differences among the intensity levels of neighboring pixels may be small, it is expected that M_1 is much larger than M_2. According to Figure 35, TDH reduces 33% of shifted values of Yang and Tsai's method in best case if $M_1 > 6M_2$.

We further compare our methods: TDH and C_TDH. If C-TDH and TDH embed the same secret data of length less than M_1, the average number of shifted pixel values is $O_1 + O_2 + O_3 + M_2 + \frac{1}{3}M_1$ in our methods. Next, we discuss the capacity of C-TDH. Assume that the length of secret data is L' in C-TDH. On the average, there are $\frac{1}{3}L'$ symbols of 10 or $\bar{1}0$, and $\frac{1}{3}L'$ symbols of 0. Thus, we require $\frac{1}{3}L' + \frac{1}{3}L'$ highest peak points. This implies that $\frac{1}{3}L' + \frac{1}{3}L' = M_1$. We have $L' = \frac{3}{2}M_1$. So, as compared with TDH, C-TDH increases the capacity by 50% on average.

At last, we compare the capacity of C-TDH and Yang and Tsai's method. From the previous discussion, C-TDH can embed $\frac{3}{2}M_1$ secret data. In Yang and Tsai's method, the length of embedded secret data is $M_1 + M_2$. Obviously, if $M_1 > 2M_2$, the capacity of C-TDH is larger than that of Yang and Tsai's method.



$$\frac{(O_1+O_3+\frac{1}{2}M_1)-(O_1+O_2+O_3+\frac{1}{3}M_1+M_2)}{O_1+O_3+\frac{1}{2}M_1}$$

$$=\frac{\frac{1}{2}M_1-O_2-M_2-\frac{1}{3}M_1}{O_1+O_3+\frac{1}{2}M_1}$$

$$=\frac{3M_1-6O_2-6M_2-2M_1}{6O_1+6O_3+3M_1}$$

$$=\frac{M_1-6M_2-6O_2}{3M_1+6O_1+6O_3}$$

FIGURE 35. The ratio of reduced number in TDH

Next, we discuss the difference on the time complexity of Yang and Tsai's method, as well as ours. Like Ni *et al.*'s algorithm [15], we assume that the image height is H and the width is W. According to [15], the computational complexity of Ni *et al.*'s algorithm is $\mathcal{O}(3HW)$. If the ternary secret data can be embedded in one image, we need to scan the whole image eleven times during the process of embedding as discussed in the secret data embedding. Hence, the computational complexity is $\mathcal{O}(11HW)$. On the other hand, we need to scan the whole image nine times in the extracting algorithm. Therefore, the computational complexity of the extracting algorithm is $\mathcal{O}(9HW)$. According to Figure 2, Yang and Tsai's method has the same number in scanning the whole image. Therefore, we conclude that the time complexity of Yang and Tsai's method and ours is the same.

6. **Conclusion.** In this paper, we propose two reversible data hiding method, TDH and C-TDH, which can be used in ternary computers. We embed ternary secret data into cover images. Since the probability of appearing 0 is about 66% in ternary secret data, TDH reduces 33% of shifted values of Yang and Tsai's method in best case if $M_1 > 6M_2$ in the histogram. Obviously, TDH provides higher PSNR in smooth images. Furthermore, we propose C-TDH method to increase the amount of the embedded secret data in TDH method. According to our analysis, as compared with TDH, C-TDH increases the capacity by 50% on average.

REFERENCES

[1] S. Arno, and F. S. Wheeler, Signed Digit Representations of Minimal Hamming Weight, *Circuits, Systems, and Signal Processing*, vol. 42, no. 5, pp. 1007–1010, 1993.

[2] X. Bo, Y. Lizhi, and H. Yongfeng, Reversible Data Hiding Using Histogram Shifting inSmall Blocks, *Proc. of the IEEE International Conference on Communications (ICC)*, Cape Town, South Africa, pp. 1–6, 2010.

[3] M. Muzzarelli, M. Carli, G. Boato, and K. Egiazarian, Reversible Data Hiding Using Histogram Shifting inSmall Blocks, *Proc. of the the 12th ACM workshop on Multimedia and security*, pp. 147–152, 2010.

[4] M. U. Celik, G. Sharma,A. M. Tekalp, and E. Saber, Reversible Data Hiding, *IEEE International Conference on Image Process*, vol.2, no. 2-3, pp. 157–160, 2002.

[5] M. Chen, Z. Chen, X. Zeng, and Z. Xiong, Reversible Data Hiding Using Additive Prediction-ErrorExpansion, *Proc. of the 11th ACM workshop on Multimedia and security*, Princeton, New Jersey, USA, pp. 19–24, 2009.

[6] D. Coltuc, Improved Embedding for Prediction Based Reversible Watermarking, *IEEE Trans. on Information Forensics and Security*, vol. 6, no. 3, pp. 873–882, 2011.

[7] M. Glusker, D. M. Hogan, and P. Vass The Ternary Calculating Machine of Thomas Fowler, *IEEE Annals of the History of Computing*, vol. 27, no. 3, pp. 4–22, 2005.

[8] B. Hayes, Third Base, *American Scientist*, vol. 89, no. 6, pp. 490-494, 2001.

[9] H. C. Huang and W. C. Fang, Authenticity Preservation with Histogram-Based Reversible Data Hiding and Quadtree Concepts, *Sensors*, vol. 11, no. 10, pp. 9717–9731, 2011.

[10] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. G. Choo, A Novel Difference Expansion Transform for Reversible Data Embedding, *IEEE Trans. on Information Forensics and Security*, vol. 3, pp. 456–465, 2008.

[11] D.E. Knuth, *The Art of Computer Programming*, Seminumerical Algorithms, vol. 2, third ed. Addison-Wesley, 1998.

[12] C. F. Lee, C. C. Chang, P. Y. Pai, and C. M. Liu, An Adjustable and Reversible Data Hiding Method Based on Multiple-base Notational System without Location Map, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 1–28, 2015.

[13] C. C. Lin, W. L. Tai, and C. C. Chang, Multilevel Reversible Data Hiding Based on Histogram Modification of Difference Images, *Pattern Recognition*, vol. 41, pp. 3582–3591, 2008.

[14] J. Marin and F. Y. Shih, Reversible Data Hiding Techniques Using Multiple Scanning Difference Value Histogram Modification, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 3, pp. 451–460, 2014.

[15] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, Reversible Data Hiding, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, pp. 354–362, 2006.

[16] P. Pear, A.Ramak, N. Zimic, M. Mraz, and I. L.Bajec, Adiabatic Pipelining : a Key to Ternary Computing with Quantum Dots, *Nanotechnology*, vol. 19, no. 49, pp. 1–12, 2008.

[17] W. L. Tai, C. M. Yeh, and C. C. Chang, Reversible Data Hiding Based on Histogram Modification of Pixel Differences, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 19, pp. 906–910, 2009.

[18] P. Tsai, Histogram-Based Reversible Data Hiding for Vector Quantization Compressed Images, *IET Image Processing*, vol. 3, pp. 100–114, 2009.

[19] P. Tsai, Y. C. Hu, and H. L. Yeh, Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting, *Signal Processing*, vol. 89, pp. 1129–1143, 2009.

[20] C. M. Wang, N. I. Wu, C. S. Tsai, and M. S. Hwang, A High Quality Steganographic Method with Pixel-Value Differencing and Modulus Function, *Journal of Systems and Software*, vol. 81, pp. 150–158, 2008.

[21] H. W. Yang, K. F. Hwang, and S. S. Chou, Interleaving Max-Min Difference Histogram Shifting Data Hiding Method, *Journal of Software*, vol. 5, pp. 615–621, 2010.

[22] C. H. Yang, and M. H. Tsai, Improving Histogram-Based Reversible Data Hiding by Interleaving Predictions, *IET Image Processing*, vol. 4, pp. 223–234, 2010.

[23] J. Yi, H. Huacan, and L. Yangtian, Ternary Optical Computer Architecture, *PhysicaScripta*, vol. T118, pp. 98–101, 2000.