

# A Framework to Reversible Data Hiding Using Histogram-Modification

R. Neeraja<sup>1</sup>

PG Student, ECE Department  
Gokula Krishna College of Engineering  
Sullurpet, India  
e-mail: r.p.neeru@gmail.com

M. Gnana Priya<sup>2</sup>

Assoc. Prof. & Head of ECE Department  
Gokula Krishna College of Engineering  
Sullurpet, India  
e-mail: gnanasagnans@gmail.com

**Abstract**— A Novel method of Steganography to achieve Reversible Data Hiding (RDH) is proposed using Histogram Modification (HM). In this paper the HM technique is revisited and a general framework to construct HM-based RDH is presented by simply designing the shifting and embedding functions on the cover image. The Secret Image is embedded inside the cover image using several steps of specific shifting of pixels with an order. The secret image or logo is retrieved without any loss in data on the cover and as well as in the secreta image. The Experimental results show the better Peak Signal to Noise Ratio (PSNR) with the existing methods.

**Keywords**- Histogram Modification(HM); Reversible Data Hiding(RDH); Peak Signal to Noise Ratio(PSNR).

\*\*\*\*\*

## I. INTRODUCTION

In Ancient days the secreta data communication techniques were performed using chemicals, mediators etc. The Reversible Data Hiding was even done in olden days where the Host data or Cover is the mediator and the secreta data is hidden in the mediator's body or on the mediator's body. Now in the modern world the hidden data transfer from one place to other is called Steganography or Cover writing. In steganography, the embedding data is irrelevant to the cover media data and is used in the covert communication applications. Where as in water marking embedding data is relevant to the cover media and is used authentication applications. In most image data hiding methods, the host image is permanently distorted due to embedded data and it cannot be recovered back to the original image after extraction of hidden data. But in some applications such as medical image sharing, multimedia archive management, and remote sensing any distortion due to data embedding is intolerable and the availability of the original image is in high demand. To this end, a solution called "reversible data hiding" (RDH) is proposed, in which the host image can be fully restored without any distortion. Many RDH methods have been proposed in recent years, roughly the methods are classified as lossless compression method, Difference Expansion (DE) method, Histogram Modification (HM) based method and integer transform, etc. All these methods aim at increasing the Embedding Capacity (EC) as high as possible while keeping the distortion low.

## II. LITERAURE SURVEY

In this literature of Reversible data hiding algorithm, the data of the size 5-80KB which is to be hidden in the gray scale cover image or host image of size 512 X 512 X 8 is placed in the union of the minimum points of the histogram of an image and at the point where slightly modification of the pixel gray scale values is done. Using this above method about 48dB of the Peak Signal to Noise Ratio was attained from the recovered Stego image. The execution time is very less and reduced computational complexity. It has been used in various practical applications like Medical images, Aerial images, Satellite Images, etc [1].

In the present literature the data is hidden within the encrypted images was proposed using different approaches where the Reversible Hidden Data has been embedded before encryption of the host image and it is termed as "Reserving Room before Encryption (RRBE)" against the traditional method in which the Reversible Hidden Data is embedded after encrypting the host image and it is termed Vacating Room after Encryption (VRAE)" using RRBE the data which is to be hidden gets more probability of hiding due to extra space availability in the host image so the process is more convenient compared with VARE, and the former technique possess good performance with any loss in the hidden data after recovery and its efficiency is appreciable [2].

In this literature Reversible Data Hiding concept is proposed using Histogram Shifting. Here the data which is to be hidden is embedded into the host or cover image after shifting the pixels in a predefined order with the help of Histogram of the host image. After embedding the data pixels in the host image the data is smoothly recovered at the reversing the shifting process so the data can be recovered very easily without any loss in the data [3].

Reversible Data Hiding methods are increasing in number as per the requirements to attain an Optimal state, In this survey it is found that according to some predefined rules the data is embedded in the Original Image or host image by choosing an optimal value. This method is an iterative method based on the size of Host image and data the optimal value is calculated using value modification under a payload distortion criterion method and moreover practical Reversible Data Hiding is obtained. In this procedure host image is divided into subset of small size images and the differences between the sub images are calculated wherever the value of difference is less the data is embedded and recovery is done in the reverse process [4].

Histogram Shifting is recommended as one of the most important technique in the area of Reversible Data Hiding where the best results can be obtained. In this survey the author describes the overview of recent techniques involving Data Hiding using Histogram Shifting where the concentration is done on the improvement of image quality and also to increase the payload capacity in the host image. Moreover the PSNR is also has been considered to improve over the existing techniques [5].

For Natural images Reversible Data Hiding concept is applied, here depending upon the relationship between the neighboring pixels the data hiding is done. The histogram is obtained on the differences between the differences of neighboring pixels wherever in the histogram the peak point is obtained the data is hidden in that place and even wherever the histogram value is zero even in that point the data is embedded, so multilevel embedding is performed for better performance. The recovery is done to obtain lossless information [6].

### III. PROPOSED FRAMEWORK

In this method, the host image is first divided into non-overlapping blocks such that each block contains  $n$  pixels. Then, an  $n$ -dimensional histogram is generated by counting the frequency of the pixel-value-array sized  $n$  of each divided block. Finally, by modifying the resulting  $n$ -dimensional histogram data embedding is implemented. Each the pixel-value-array is an element of  $Z^n$  and is divided into two disjoint sets, one set is used to carry hidden data, and the other set is simply shifted to create vacant spaces to ensure the reversibility.

Let  $S$  and  $T$  be a partition of  $Z^n$ :  $S \cup T = Z^n$  and  $S \cap T = \emptyset$ . Define three functions  $g: T \rightarrow Z^n$ ,  $f_0: S \rightarrow Z^n$  and  $f_1: S \rightarrow Z^n$  satisfy the following conditions: C1: The functions  $g$ ,  $f_0$  and  $f_1$  are injective. C2: The sets  $g(T)$ ,  $f_0(S)$  and  $f_1(S)$  are disjointed with each other. Here,  $g$  is called “shifting functions” and will be used to shift pixel values;  $f_0$  and  $f_1$  are called “embedding functions” and will be used to embed data. Each block with value  $x \in T$  will be shifted to  $g(x)$ , and the block with value  $x \in S$  will be expanded to either  $f_0(x)$  or  $f_1(x)$  to carry one data bit. The shifting and embedding functions will give a HM-based RDH algorithm where the reversibility can be guaranteed by the conditions C1 and C2.

The underflow/overflow is an inevitable problem of RDH, i.e., the shifted and expanded values should be restricted in the range of  $[0, 255]$  for gray-scale image and it is expressed as in (1).

$$A_n = \{X = (x_1, \dots, x_n) \in Z^n : 0 \leq x_i \leq 255\} \quad (1)$$

Let  $A_n$  is the set of all pixel-value-arrays of length  $n$  of gray-scale image. The time shifted and expanded values of underflow/overflow are given in (2).

$$\left. \begin{aligned} T_s &= A_n \cap g^{-1}(A_n) \\ S_e &= A_n \cap f_0^{-1}(A_n) \cap f_1^{-1}(A_n) \\ T_{u,o} &= A_n \cap T - T_s \\ S_{u,o} &= A_n \cap S - S_e \end{aligned} \right\} \quad (2)$$

The suffix “s”, “e” and “u, o” mean “shift”, “embed” and “underflow/overflow”, respectively. Obviously, the four sets  $T_s$ ,  $S_e$ ,  $T_{u,o}$  and  $S_{u,o}$  are disjointed with each other and constitute a partition of  $A_n$ , i.e.,

$$A_n = T_s \cup S_e \cup T_{u,o} \cup S_{u,o} \quad (3)$$

The sets  $g(T_s)$ ,  $f_0(S_e)$  and  $f_1(S_e)$  are contained in  $A_n$  and the condition C2 ensures that they are also disjointed. Each block

with value  $x \in T_s$  will be shifted, each block with value  $x \in S_e$  will be expanded to carry one data bit, and the block with value  $x \in T_{u,o} \cup S_{u,o}$  will remain unchanged since it cannot be shifted or expanded due to underflow/overflow. Take  $T_{u,o} \cup S_{u,o} = T \cup S$  for simplicity in the following context.

### IV. DATA EMBEDDING

The embedding procedure contains many steps

Step 1: Divide the host image into  $k$  non-overlapping blocks  $\{X_1, \dots, X_k\}$  such that each  $X_i$  contains  $n$  pixels. Assume the value of  $X_i$  is  $x_i \in A_n$ .

Step 2: Define the location map  $LM$ :  $LM(i) = 0$  if  $x_i \in T_s \cup S_e$ , and  $LM(i) = 1$  if  $x_i \in T \cup S$ .  $LM$  is a binary sequence of length  $k$ . Denote  $k^l = \lceil \log_2(k+1) \rceil$ , where  $\lceil \cdot \rceil$  is the ceiling function. Take  $l = \{i : LM(i) = 1\}$  which is the amount of underflow/overflow blocks. Then define a binary sequence  $LMc$  of length  $lc = (l+1)k$  to record all underflow/overflow locations.

a)  $(LMc(1), \dots, LMc(k^l))$  is the binary representation of  $l$ .

b) For each  $j \in \{1, \dots, l\}$ ,  $(LMc(jk^l+1), \dots, LMc(jk^l+k^l))$  is the binary representation of  $i$ , where  $i$  is the  $j$ -th index such that  $LM(i) = 1$ .

Step 3: Divide the  $k$  blocks into three parts to get  $I_1$ ,  $I_2$  and  $I_3$ .

a)  $I_1 = \{X_1, \dots, X_{k_1}\}$  with  $k_1 = \left\lfloor \frac{lc}{n} \right\rfloor$ .

b)  $I_2 = \{X_{k_1+1}, \dots, X_{k_1+k_2}\}$  such that it contains exactly  $lc$  embeddable blocks

c)  $I_3 = \{X_{k_1+k_2+1}, \dots, X_k\}$  is the set of rest blocks.

Step 4: Embed the hidden data into  $I_1$  and  $I_3$ , i.e., for any  $I \in \{1, \dots, k_1, k_1 + k_2 + 1, \dots, k\}$ .

a) If  $x_i \in T_s$ , replace the value of  $X_i$  by  $g(x_i)$ .

b) If  $x_i \in S_e$ , replace the value of  $X_i$  by  $f_m(x_i)$ , where  $m \in \{0, 1\}$  is the data bit to be embedded.

c) If  $x_i \in T \cup S$ , do nothing with  $X_i$ .

Step 5: Record LSBs of the first  $lc$  pixels of  $I_1$  to get a binary sequence  $S_{LSB}$ . Then replace these LSBs by the sequence  $LMc$  defined in Step 2.

Step 6: Embed the sequence  $S_{LSB}$  into  $I_2$  in the same way as Step 4. Since the length of  $S_{LSB}$  is  $lc$ ,  $S_{LSB}$  can be embedded exactly into the embeddable blocks of  $I_2$ . Finally, the marked image is obtained.

## V. DATA EXTRACTION

The data extraction procedure also contains many steps.

Step 1: Divide the marked image into  $k$  non-overlapping blocks  $\{Y_1, \dots, Y_k\}$ . Assume the value of  $Y_i$  is  $y_i \in \mathbb{A}_n$ .

Step 2: Firstly, determine the amount of problematic locations  $l$ , by reading LSBs of the first  $k^1 = \lfloor \log_2(k+1) \rfloor$  pixels. Secondly, read LSBs of the first  $lc = (l+1)k$  pixels to get the sequence LMc. Then we can get the location map LM. Finally, with  $k_1 = \lfloor \frac{lc}{n} \rfloor$ , LM, and by identifying embeddable blocks, we can obtain the same partition as defined in Step 3 of data embedding.

Step 3: Extract data from  $I_2$  and recover original pixel values of  $I_2$ , i.e., for any  $i \in \{k_1+1, \dots, k_1+k_2\}$ .

a) If  $LM(i) = 0$  and  $y_i \in g(Ts)$ , the original pixel value is  $g^{-1}(y_i)$  and there is no embedded data.

b) If  $LM(i) = 0$  and  $y_i \in f_m(Se)$  holds for a certain  $m \in \{0, 1\}$ , the original pixel value is  $f_m^{-1}(y_i)$  and the embedded data bit is  $m$ .

c) If  $LM(i) = 1$ , the original pixel value is  $y_i$  itself and there is no embedded data.

The sequence  $S_{LSB}$  defined in Step 5 of data embedding is extracted in this step.

Step 4: Replace LSBs of the first  $lc$  pixels of  $I_1$  by  $S_{LSB}$ .

Step 5: Extract the embedded hidden data and recover original pixel values in  $I_1 \cup I_3$  in the same way as Step 3.

Finally, the hidden data is extracted and the original image is restored.

## VI. EMBEDDING CAPACITY

According to the data embedding procedure, the EC of this method can be formulated as in (4)

$$\sum_{x \in S_e} h_n(x) - \lfloor \log_2(k+1) \rfloor \sum_{x \in TS_u} h_n(x) - \lfloor \log_2(k+1) \rfloor \quad (4)$$

Where  $h_n$  is the  $n$ -dimensional histogram:  $h_n(x) = \{i : x_i = x\}$ .

smallest integer such that it is capable to embed the required payload. This novel algorithm achieves a better performance compared with previous works. It can provide a larger PSNR whatever the test image.

## VII. A NOVAL HM BASED RDH ALGORITHM

A novel and efficient RDH algorithm is introduced to demonstrate the universality and applicability of this framework. It is developed by exploiting nine-dimensional histogram. By incorporating existing PEE and pixel selection techniques into the proposed framework, one can achieve an excellent performance. For a  $3 \times 3$  block  $\mathbf{x} = (x_1, \dots, x_9)$ , shown below "Fig. 1".

$x_1$	$x_2$	$x_3$
$x_4$	$x_5$	$x_6$
$x_7$	$x_8$	$x_9$

Figure 1. 3X3 Block

Take the linear predictor with non-uniform weight is given as (5)

$$\tilde{x}_5 = \frac{1}{16}(x_1 + x_3 + x_7 + x_9) + \frac{3}{16}(x_2 + x_4 + x_6 + x_8) \quad (5)$$

to predict  $x_5$ . The prediction-error is denoted as in (6)

$$e_5 = \tilde{x}_5 - x_5 \quad (6)$$

Take the following function (7)

$$C(x) = \max\{x_1, \dots, x_4, x_6, \dots, x_9\} - \min\{x_1, \dots, x_4, x_6, \dots, x_9\} \quad (7)$$

To measure the local complexity of pixel  $x_5$  and use an integer-valued parameter  $s$  to select smooth pixels.

For an integer  $t > 0$ , take  $t_l = \lfloor \frac{t}{2} \rfloor$  and  $t_r = \lceil \frac{t}{2} \rceil$  then define in (8) and (9)

$$(1) S = \{x \in Z^9 : -t_l \leq e_5 < t_r, C(x) < s\} \text{ and } T = Z^9 - S. \quad (8)$$

(2) For  $x \in T$ ,

$$g(x) =$$

$$\begin{cases} (x_1, \dots, x_4, x_5 + t_r, x_6, \dots, x_9), & \text{if } e_5 \geq t_r \text{ and } C(x) < s, \\ (x_1, \dots, x_4, x_5 - t_l, x_6, \dots, x_9), & \text{if } e_5 < -t_l \text{ and } C(x) < s, \\ x, & \text{if } C(x) \geq s. \end{cases} \quad (9)$$

(3) For  $x \in S$  and  $m \in \{0, 1\}$ ,

$$F_m(x) = (x_1, \dots, x_4, x_5 + \lfloor e_5 \rfloor + m, x_6, \dots, x_9). \quad (10)$$

To minimize the embedding distortion, the parameter  $s$  is first fixed as its maximum 256 and the parameter  $t$  is taken as the smallest integer such that it is capable to embed the required payload.

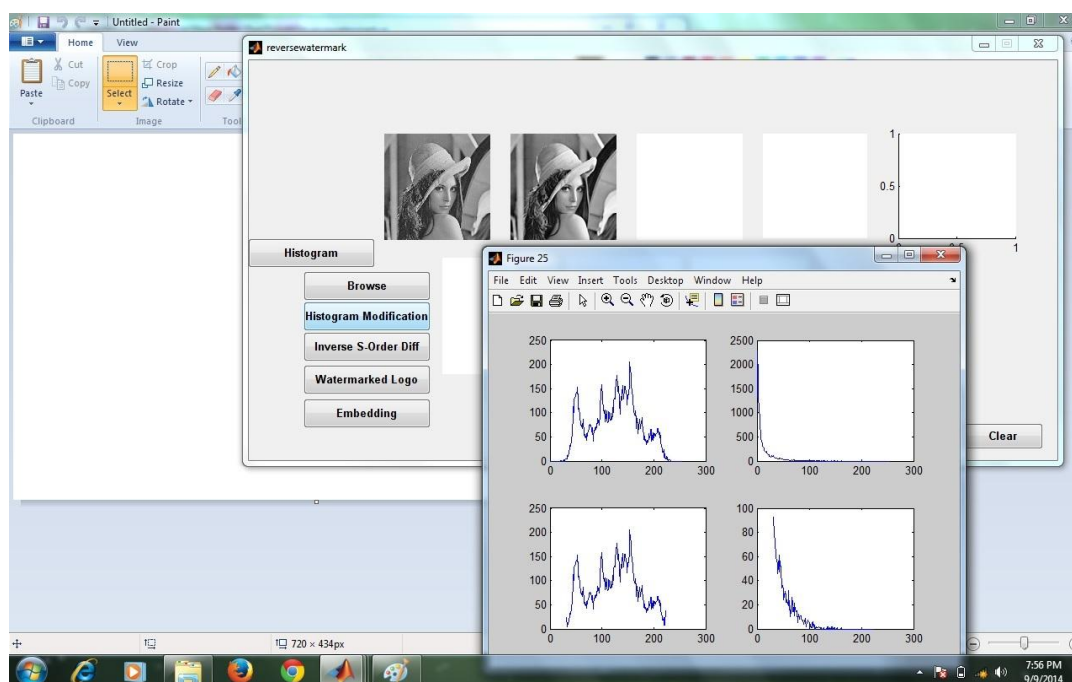
When  $t$  is determined, to take advantage of smooth pixels as much as possible, vary the parameter  $s$  and then take it as the

## VIII. SIMULATION RESULTS

Table 1 shows the comparison of PSNR (in db) between the proposed algorithms with existed four methods, for an ER of 0.5 BPP.

**Table 1. Comparison of PSNR in dB**

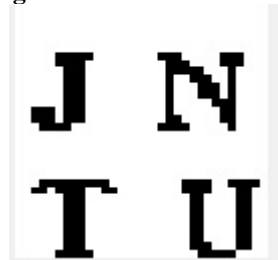
Image	Hu et al.	Luo et al.	Li et al.	Hong	Proposed algorithm I
Lena	40.73	41.33	42.37	41.25	<b>43.14</b>
Baboon	30.65	29.47	31.42	31.10	<b>32.11</b>
Sailboat on lake	35.33	35.67	36.63	35.66	<b>37.47</b>
Fishing boat	36.45	36.05	37.82	36.79	<b>38.51</b>



**Figure 2. Histogram Modification**



**Figure 3. Host/Cover Image**



**Figure 4. Secrete Data**



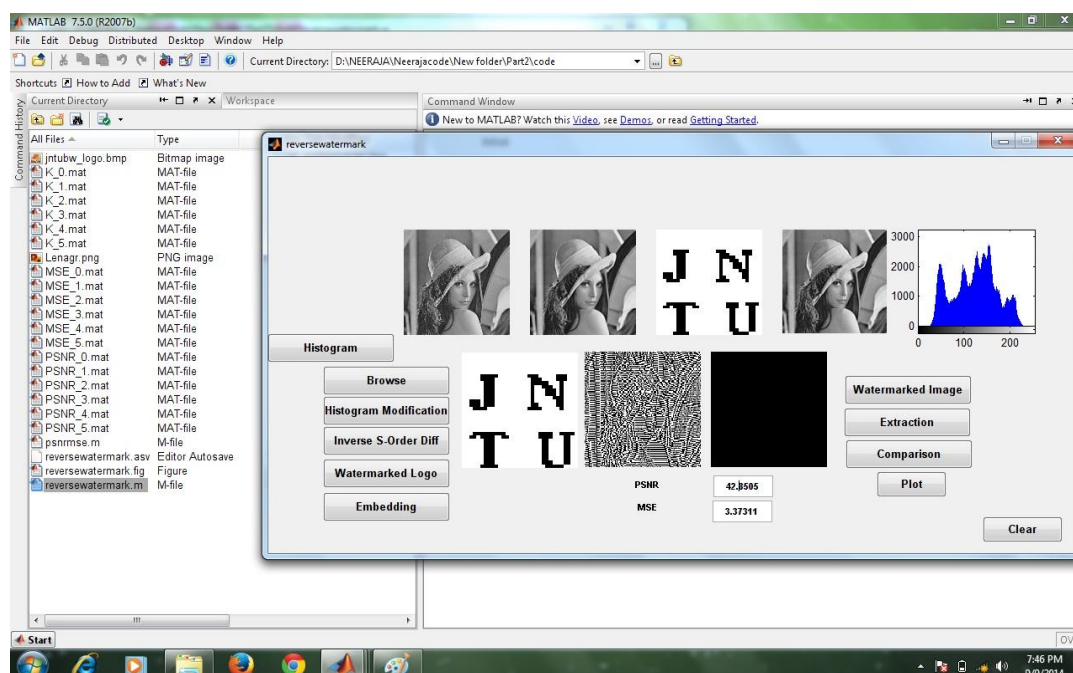


Figure 5. Final Gui Output

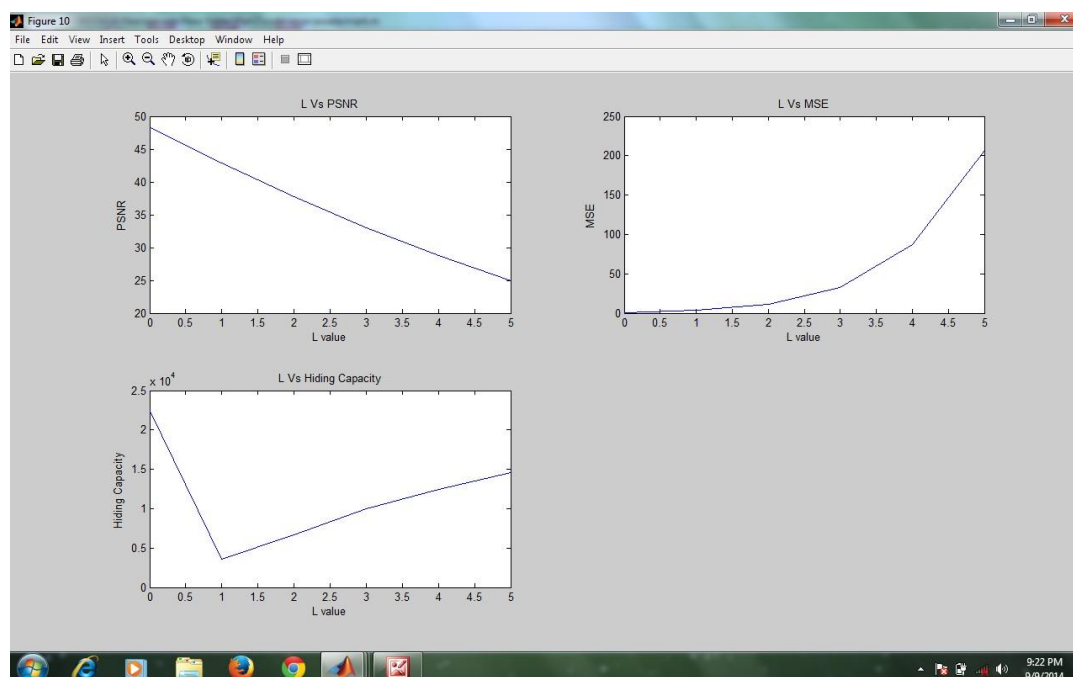


Figure 6. PSNR, Hiding Capacity and MSE Graphs

## IX. CONCLUSION

By revisiting existing algorithms, a general framework to construct HS-based RDH is proposed. According to this framework, by defining the shifting and embedding functions one can obtain a RDH algorithm. This work will facilitate the design of RDH. Furthermore, by incorporating this framework with the PEE a novel RDH algorithm is also introduced. This algorithm can achieve a better performance compared with the state-of-the-art works. So the proposed framework has a potential to provide excellent RDH algorithms.

## X. REFERENCES

- [1] Z. Ni, Y.Q. Shi, N. Ansari and W. Su, "Reversible Data Hiding," IEEE Transaction on Circuits Systems for Video Technology, Vol. 16, No. 3, pp. 354 –362, March.2006.
- [2] Keda Ma et al, "Reversible Data Hiding in Encrypted Images by Reserving Room before Encryption", IEEE Transaction on Information Forensics and security, Vol. No. 3, pp. 553-562, March 2013.
- [3] X. Li, B. Li, B. Yang and T. Zeng, "General Framework to Histogram Shifting based Reversible Data Hiding",

- 
- IEEE Transaction on Image Processing, Vol. 22, No. 6, June 2013.
- [4] X. Zhang, “Reversible Data Hiding with Optimal Value Transfer”, IEEE Transaction on Multimedia, Vol. 15, No. 2, February 2013.
- [5] Y. Z. Mehran et al, “Histogram Shifting as a Data Hiding Technique: An Overview of Recent Developments”, Springer- Verlag, pp. 770 – 786, 2011.
- [6] Z. Zhao et al, “Reversible Data Hiding based on Multilevel Histogram Modification and Sequential Recovery”, ELSVIER – International Journal of Electronics and Communication, pp.813 – 824, 2011.