IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 11, Issue 5, Ver. I (Sep.-Oct .2016), PP 40-53
www.iosrjournals.org

Reversible Data Hiding Based On Multiple Histograms Modification

G. Padmavathi[1], T. Geetamma[2]
[#1] M.Tech Scholar, Department of Electronics and Communication Engineering,
[#2] Assist.Prof, Department of Electronics and Communication Engineering, GMRIT, Rajam, India.

Abstract: *Reversible data hiding (RDH) is utilized to embed secret message into a cover image by somewhat adjusting its pixel values. Embedded message and the spread image are totally recuperated from the checked substance. RDH underpins data hiding with the lossless compressibility of normal images. Lossless pressure, distinction expansion, histogram change, predection error expansion and whole number change systems are utilized for RDH process. Histogram based RDH method is isolated into two stages histogram era and histogram change. Histogram development is performed with the pixel pair's arrangements and their diverse qualities. Histogram change is done to insert data into the spread image. The unhiding procedure recuperates the message furthermore the spread image. Predection error expansion (PEE) procedure is connected for Reversible Data Hiding (RDH) process. Maybe a couple dimensional Prediction-error Histogram (PEH) are utilized as a part of the PEE systems. The two-dimensional PEH-based methods perform superior to anything one dimensional PEH. PEH alteration is settled and free of image substance. Various histograms based PEE method is received to enhance reversible data hiding (RDH) process. Numerous Histograms Modification (MHM) method utilizes an arrangement of histograms for the hiding procedure. A multifaceted nature estimation is processed for every pixel with reference to its setting. Predection Error Histogram (PEH) is produced utilizing the pixels with the multifaceted nature esteem. A grouping of histograms can be produced by differing the multifaceted nature to cover the entire image. Two expansion containers are chosen in each produced histogram. The expansion containers are chosen with reference to the image content. Data installing is done on Multiple Histograms Modification (MHM).*

Index Terms: *Reversible data hiding, prediction-error expansion, multiple histograms modification, adaptive embedding, Zhicheng's Method.*

I. Introduction

In late time the utilization of data innovation and web has been developing quickly, so the utilization of advanced data, for example, image, sound, video and content are imparted and share helpfully. Because of simple access of web the data can be effortlessly duplicated fashioned and changed. Thusly, data security is extreme inadequate and the falsifier tries to violet the substance of the mixed media. Consequently, for the insurance of the innovation and vagueness of the mixed media data, the mystery message must be covered up in the image. Data hiding is a strategy to hide mystery data into spread item, which assumes a crucial part in the assurance of the data [1]. Fundamentally in the instances of data hiding, the spread article will influenced by some bending and can't be turned around back to the first question, due to some steady contortion has been struck the spread protest even after the extraction of concealed message. Reversible data hiding encourages inconceivable plausibility of uses to connection two arrangements of data, in that way that the concealed message have been separated out without affecting the spread item. Subsequently, give an extra plausibility to taking care of two distinct data sets. Numerous reversible data hiding plans have been proposed. Tian proposed a distinction expansion data hiding plan, where the distinction and normal estimations of two neighboring pixels are figured and the mystery data to be inserted are annexed to a distinction esteem spoke to as a double number. Alattar, Kim et al, and Weng further expanded Tian's work. Ni et al. proposed a plan of utilizing top/zero focuses in the histogram of spatial area images. Fridrich et al proposed a few methods to implant data. The fundamental thought of their work is to pack the chose image highlights for gaining save space. Hong et al. displayed a plan which plays out a movement of the histogram of expectation errors. It utilize the middle edge locator (MED) to anticipate pixel values. Barton built up a reversible data implanting calculation that depends on data pressure. In this method compacted data is to be installed in an image. This paper display a distinction histogram change reversible data hiding calculation. In this proposed method the pixel pair connection is helpful in anticipating a neighborhood image locale on two dimensional space for fulfilling a succession that comprise of distinction sets. Presently by tallying the distinction matches a two dimensional contrast histogram is created. As the DPM is an injective mapping strategy which is characterized on distinction sets and it is utilized as a part of late histogram based methods by characteristic augmentation of expansion implanting and moving strategies. So at last by distinction pair mapping (DPM) procedure reversible data inserting is actualized. For edifying the installing execution the proposed method utilizes two dimensional distinction histogram and its particular DPM, as contrasted and the routine one dimensional histogram based methods which incorporate more pixels for conveying the data furthermore we can diminish the quantity of moved pixels. In the past writing investigations of implanting position and determination procedures another pixel pair choice system is proposed for finding the pixel sets in smooth image districts to install the data. Additionally, it is further utilized for upgrading the inserted execution.

II. Related work

Ni et al. proposed a reversible data hiding method in light of histogram change. In the plan, part of the spread image histogram is moved rightward or leftward to create repetition for data installing. To start with, the top and zero point containers of the first histogram are discovered signified as b(P) and b(Z), separately. At that point every one of the receptacles having a place with b(P) and b(Z) are moved rightward one level. Along these lines, the container of b(P) is exhausted and b(P + 1) turns into the new top point. Next, the classified data can be inserted by adjusting the pixel values leveling with P + 1. That is, if experience a pixel with quality breaking even with P + 1, then one piece private data can be covered up. For instance, if the present handling classified piece is "0", we alter the pixel esteem as P; while if the present preparing private piece is "1", the pixel with worth P + 1 is kept no changed. In decoder, the data extraction and image recuperation is the converse preparing of data implanting. In [3], Li et al. proposed a reversible data hiding method named nearby pixel contrast (APD) in light of the neighbor pixel contrasts adjustment. In this method, a converse "S" request is received to filter the image pixels. A 3 × 3 image piece is utilized to delineate this rule. The output heading is set apart as the blue line and the square can be modified into a pixel grouping as p1, p2, . . ., p9. Assume the host image I is a 8-bit dark level image measured as M × N. At that point a pixel succession p1, p2, . . ., pM×N are acquired through the reverse "S" request check. The distinctions of nearby pixels are processed as: (1) Considering the pixel values comparability amongst pi−1 and pi , a vast amount of di (2 ≤ i≤ M × N) is equivalent or near 0. The distinction histogram is built in light of these M × N − 1 contrast measurements. Assume the histogram receptacles from left to right are meant by b(−255), b(−254), . . ., b(−1), b(0), b(1), . . ., b(254), b(255). The 512 × 512 Lena image's distinction histogram. Clearly most contrasts are concentrated around b(0). At the point when the bend spreads away t both sides, it drops drastically and no distinctions fall into those containers a long way from b(0). Essentially, APD chooses one sets of receptacles b(p1) and b(z1) (assume p1 < z1) where b(p1) and b(z1) signify the crest point and zero point, individually. At that point the containers between [b(p1 + 1), b(z1 − 1)] are moved rightward one level. In this manner b(p1 1) are purged for data installing. That is, if a mystery bit "1" is installed, the distinctions meeting p1 are included by 1. On the off chance that "0" is embedded, they are not changed. To improve the limit, APD can likewise choose two sets of peakzero focuses, e.g. [b(p1), b(z1)] and [b(z2), b(p2)] (assume p1 < z1 and z2 < p2). At that point the receptacles between [b(p1 + 1), b(z1 − 1)] are moved rightward one level and those between [b(z2 + 1), b(p2 − 1)] are moved leftward one level. Subsequently b(p1 + 1) and b(p2 − 1) are purged for data inserting [4]. The mystery bits tweak is comparable as that in one sets of top zero focuses installing. Note the scopes of [b(p1), b(z1)] and [b(z2), b(p2)] must not be covered.

III. Problem Statement

Prediction-error expansion (PEE) technique is applied for Reversible Data Hiding (RDH) process. One- or two-dimensional Prediction-error Histogram (PEH) are used in the PEE techniques. The two-dimensional PEH-based methods perform better than one dimensional PEH. PEH modification is fixed and independent of image content. Multiple histograms based PEE method is adopted to improve reversible data hiding (RDH) process. Multiple Histograms Modification (MHM) method uses a sequence of histograms for the hiding process. A complexity measurement is computed for each pixel with reference to its context. Prediction-Error Histogram (PEH) is generated using the pixels with the complexity value. A sequence of histograms can be generated by varying the complexity to cover the whole image. Two expansion bins are selected in each generated histogram. The expansion bins are selected with reference to the image content. Data embedding is carried out on Multiple Histograms Modification (MHM). The following problems are identified from the existing system. • Embedding capacity (EC) is low • Predictor selection is not optimized • Limited embedding performance • Complexity measure selection is not optimized • Limited secret data security.[10]

IV. Methodology

In this work, a novel reversible data embedding algorithm using contrast stretching is proposed. The contrast stretching technique is used instead of lossless compression to explore the redundancy in the image histogram and to create greater embedding space. In addition, such a technique is designed to reversibly embed the payload into the image histograms without embedding the original values concerning the host image. In addition, the proposed technique achieved satisfactory and stable performance both on embedding capacity and visual quality. Figure 1 and 2 respectively summarizes the proposed algorithm. Figure.1 shows a schematic diagram of the proposed reversible data embedding system that will be applied to the output image obtained using Zhicheng's method. First, the output image from zhicheng's method, Io, is divided into equally, non-overlapping blocks, which are the inputs to the system. And then, the system computes the histogram statistics, Ho, of each image block, which includes the gray level with the most counts and the number of pixels used to calculate the histogram. The contrast of the histogram of each image block is then stretched to create an extra embedding space, with the reverse-aid information A produced as well. The watermark W thus consists of the data payload and the reverse-aid. Finally, the watermark is embedded into the generated space in the stretched image, Is, producing the watermarked image, Iw. [3]

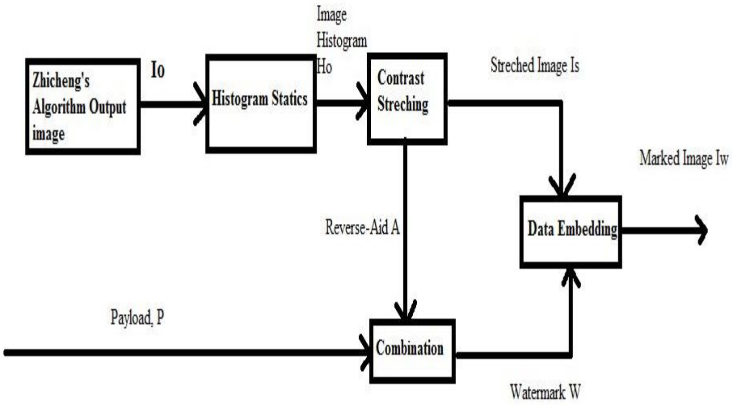


Figure 1: The Embedding Process of the Removable Data Using Proposed Method.

The data extraction and restoration process of the removable data embedding using proposed method is illustrated in Figure.2. The restoration process starts with the division of the received image I'w into equally sized NxN blocks. Next, the system computes the histogram statistics, H'w, of each image block and extracts the watermark, W'. During restoration, the embedded watermark is removed from the received image, thus restoring it to its original state. [1][2]

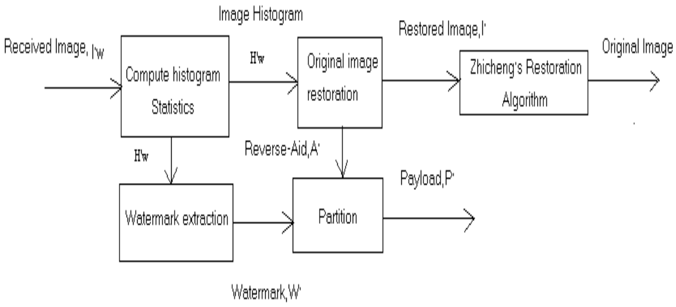


Figure 2: The Data Extraction and Restoration Process of the Removable Data Embedding using Proposed Method.

Zhicheng's Method

Zhicheng et. el. utilized the redundancy of the image histogram and slightly modified the pixel values to embed the data. The algorithm is simple and efficient. Therefore, it can keep high visual quality for all images with a PSNR value guaranteed to be higher than 48 dB. However, the capacity only equals to the peak of the image histogram and is about 0.05 bits/pixel. This capacity belongs to the range of (5K-80K) bits for a 512x512x8 grayscale image. Furthermore, their approach cannot be generalized since the capacity variation is very high and may change dramatically from image to image.

An Embedding Algorithm Procedure

The "lena" image is used as an example to illustrate this algorithm. For a given grayscale image, say lena image (512x512x8), its histogram is first generated as shown in figure 3.

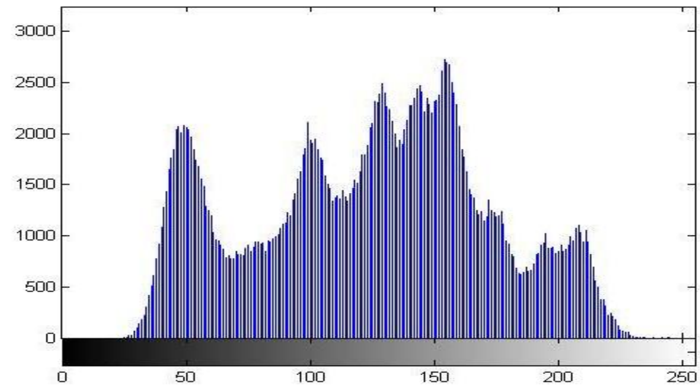


Figure 3: Lena Original Histogram.

An illustration of the Embedding Algorithm Using an Example With One Zero Point and One Peak Point is summarized as follows: In the histogram, we first find a zero point, and then a peak point. A zero point corresponds to the grayscale value which has zero frequency and belongs to the image domain, e.g., h(255) as shown in Figure 3 . A peak point corresponds to the grayscale value which has the maximum number of repeated pixels within the given image, e.g., h(154) as shown in Figure 3. To illustrate the principle of the algorithm,and for the sake of notational simplicity, only one zero point and one peak point are used in this example. The objective of finding the peak point is to increase the embedding capacity to be as large as possible since in this algorithm, as shown below, the number of bits that can be embedded into an image equals to the number of pixels which are associated with the peak point. [10]

The whole image is scanned in a sequential order, say, row-by-row, from top to bottom, or, column-by-column, from left to right. The grayscale value of pixels between 155 (including 155) and 254 (including 254) is incremented by "1." This step is equivalent to shifting the range of the histogram, [155 254], to the right-hand side by 1unit, leaving the grayscale value 155 empty as shown in Figure 4.

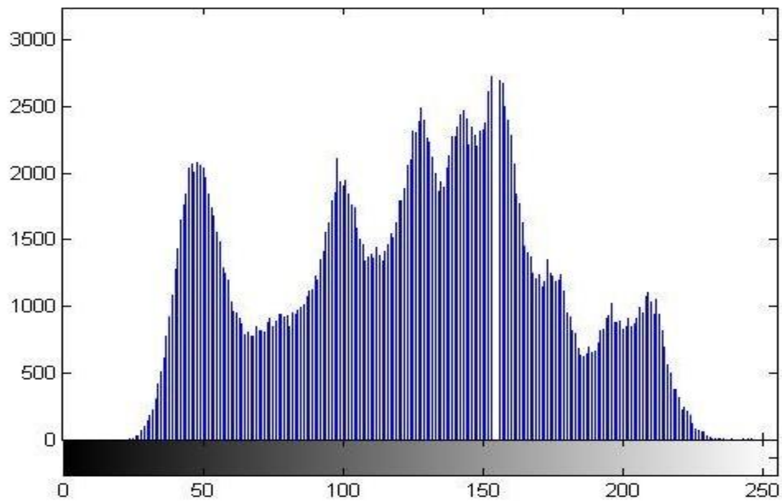


Figure 4: Lena Histogram after Shifting.

Now, the whole image is scanned once again in the same sequential order. Once a pixel with grayscale value of 154 is encountered, we check the to-be-embedded data sequence. If the corresponding to-be-embedded bit in the sequence is binary "1", the pixel value is incremented by 1. Otherwise, the pixel value remains intact.(For illustration purposes, it is preferred to present the embedding algorithm in these three steps only.)

The above three steps complete the data embedding process. Now we can observe that the data embedding capacity of this algorithm, when only one pair of zero and peak points is used, equals to the number of pixels that assume the grayscale value of the peak point as mentioned above. The histogram of the marked Lena image is plotted in Figure 5. It is noteworthy that the peak at 154, which was shown in Figure 3 is no longer the peak value in Figure 5.

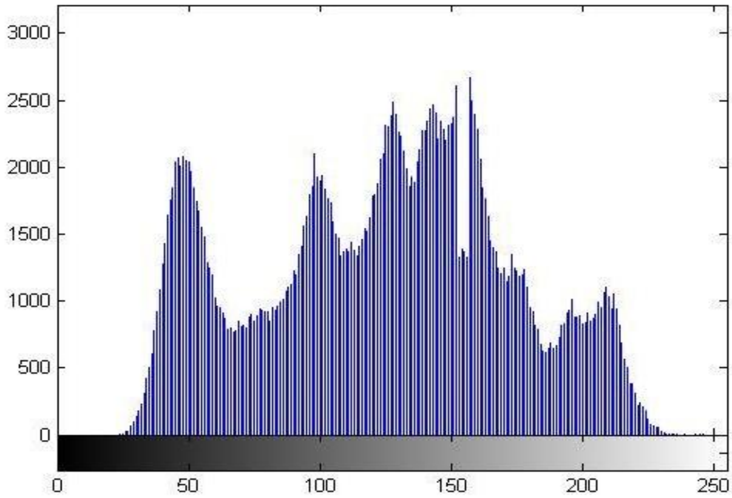


Figure 5: Lena Histogram after Zhicheng's et. el. Method.

In very rare cases, the zero point is not able to be found in a histogram. In such situations, the minimum point is used instead of the zero point. Then, the gray value and the coordinates of the minimum point are recorded as an overhead part of the embedded data. This book-keeping information will be used later to recover the minimum point data retrieval. If there are multiple pairs of zero points and peak points, it is possible to further increase the payload by adding complexity to this algorithm.[8]

Data Extraction Procedure

Only the simple case of one pair of minimum point and maximum point is described here because, as shown above, the general cases of multiple pairs of maximum and minimum points can be decomposed as a few one pair cases. Assuming the grayscale value of the maximum point and the minimum points are a and b, respectively. Without loss of generality, assume a<b. The marked image is of size (MxN), and each pixel grayscale value x belongs to the interval between [0,255] (assuming 8 bit pixel representation). The next step is to scan the marked image in the same sequential order as that used in the embedding procedure. If a pixel with its grayscale value (a+1) is encountered, a bit "1" is extracted. If a pixel with its value (a) is encountered, a bit "0" is extracted. Scan the image again, for any pixel whose grayscale value between [a,b], the pixel value is subtracted by 1.

If there is overhead bookkeeping information found in the extracted data, set the pixel grayscale value (whose coordinate (i,j) is saved in the overhead) as b.

In this way, the original image can be recovered without any distortion, but the total capacity is limited to the sum of peaks count that is used in embedding process which will not be able to give a capacity more than 0.05 bpp. To improve the capacity of this algorithm, the next section proposes an enhancement method that will provide better results compared to this method. [7]

V. Properties and Comparative Analysis of Contemporary Researches

We present in this section some RDH works related to PEE including C-PEE, the PEE with adaptive embedding (A-PEE), the PEE with optimal expansion bins selection (O-PEE) and the PEE which combines both adaptive embedding and optimal expansion bins selection (AO-PEE). The first step of these PEE-based methods is the generation of PEH. First of all, in a specific scanning order, the cover pixels are collected into a one-dimensional sequence as $(x_1,..., x_N)$ where N is the total number of collected pixels. Then, a predictor is employed to predict each x_i, and the prediction value denoted by x_i should be rounded off if it is not an integer. Next, the prediction-error is computed by

$e_i = x_i - x_i$. (1)

Finally, the prediction-error sequence $(e_1,...,e_N)$ is derived and the corresponding PEH denoted by h can be established as

$h(e) = \#\{1 \le i \le N : ei = e\}, \quad \forall e \in Z$ (2)

where # means the cardinal number of a set. The second step of PEE-based methods is the modification of PEH. We now introduce the PEH modification mechanisms for the above four types of PEE-based methods, respectively.

A. C-PEE

After PEH generation, the C-PEE embedding procedure contains following steps. First, for a prediction-error e_i, it is

$$\widetilde{e}_i = \begin{cases} e_i + m, & \text{if } e_i = 0 \\ e_i - m, & \text{if } e_i = -1 \\ e_i + 1, & \text{if } e_i > 0 \\ e_i - 1, & \text{if } e_i < -1 \end{cases} \quad (3$$

where $m \in \{0,1\}$ is a to-be-embedded data bit. With (3), the bins −1 and 0 are expanded to embed data, while other bins are shifted to create vacancies to ensure the reversibility.

Then, the cover pixel x_i is modified to $x_i = \widehat{x}_i + e_i$ to generate the marked pixel. Notice that the above procedure will stop once all data bits are embedded, i.e., only the cover pixels $(x_1,\ldots, x_{N\text{end}})$ need to be processed where $N_{\text{end}} \leq N$ is the smallest index such that the payload can be embedded into the first N_{end} cover pixels.

An illustration of histogram modification mechanism for C-PEE is shown in Fig. 1. Particularly, the mapping of bins for C-PEE is shown in Fig. 1(b), in which the red points (bins −1 and 0) are expanded while black points (other bins) are shifted.

The C-PEE extraction and image restoration procedure can be summarized as follows. First, determine the predictionx_i of marked pixel x_i for each $i \in \{1,\ldots, N_{\text{end}}\}$. The marked

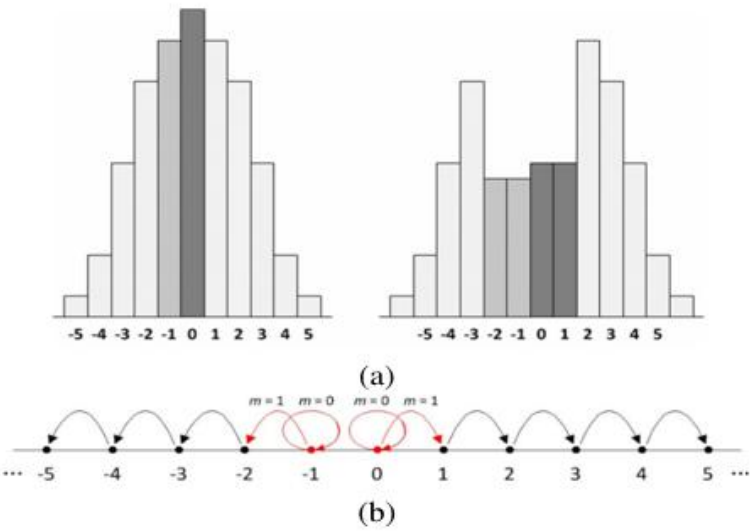


Fig. 6. Histogram modification mechanism for C-PEE. (a) PEH before (left)and after (right) C-PEE embedding. (b) Mapping of bins for C-PEE.

prediction-error is thus $e_i = \widetilde{x}_i - x_i$. Then, for each e_i, the original prediction-error can be recovered as

$$e_i = \begin{cases} \widetilde{e}_i, & \text{if } \widetilde{e}_i \in \{-1, 0\} \\ \widetilde{e}_i - 1, & \text{if } \widetilde{e}_i > 0 \\ \widetilde{e}_i + 1, & \text{if } \widetilde{e}_i < -1. \end{cases} \quad (4)$$

Meanwhile, the embedded databit can be extracted as $m = 0$ if $e_i \in \{-1,0\}$, or $m = 1$ if $e_i \in \{-2,1\}$. Finally, restore the cover pixel as $x_i = \widehat{x}_i + e_i$.

A key issue for the reversibility of C-PEE is that the prediction values obtained by decoder should be the same as those of encoder. For example, by using median-edge-detector (MED [28]) or gradient-adjusted-predictor (GAP [42]) which is based on half-enclosing casual pixels for prediction, the decoder can inversely scan and process pixels to get the same prediction values.

B. A-PEE

As an extension to C-PEE, the adaptive embedding strategy has been proposed to better exploit the image redundancy [40]–[43]. Specifically, for A-PEE, a complexity measurement denoted by n_i is computed for each x_i according to its context. Then, only the pixels satisfying $n_i < T$ will be embedded, where $T > 0$ is a pre-selected threshold. This means, for each x_i with $n_i < T$, it will be processed according to the C-PEE embedding procedure. Otherwise, i.e., $n_i \geq T$, x_i is ignored and its value keeps unchanged. Here, the threshold T is an

important factor for the embedding performance of A-PEE. To better utilize smooth pixels, T is taken as the smallest positive integer such that the payload can be successfully embedded.
The extraction procedure of A-PEE is almost the same as that of C-PEE. The difference is that, only the pixels with $n_i < T$ need to be processed, while other pixels can be directly restored as themselves since they are unmodified during data embedding.

C. O-PEE

We use the example introduced in to illustrate O-PEE. Consider here a specific PEH satisfying

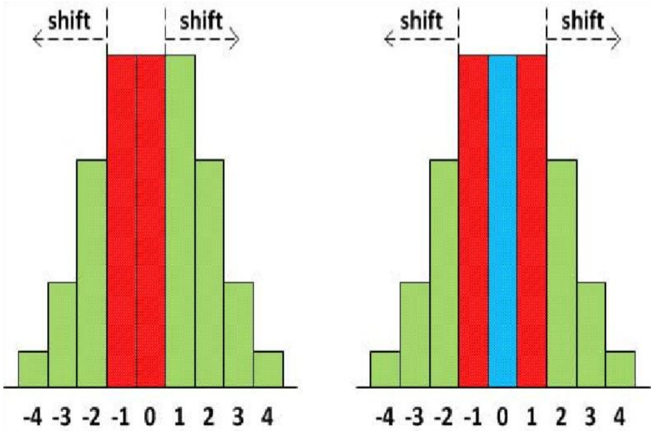


Fig. 7. C-PEE (left) v.s. O-PEE (right).

$h(-1) = h(0) = h(1) = H$ and suppose that EC is $2H$.
In this case, for C-PEE (see the left figure of Fig. 2 for an illustration), the expected value of embedding distortion in l^2-norm can be formulated as

$$\mathbf{E}(\|\widetilde{I} - I\|_2^2) = \sum_{i=1}^{N} \mathbf{E}((\widetilde{x}_i - x_i)^2) = \frac{1}{2}\Big(h(-1) + h(0)\Big) + \sum_{e \notin \{-1,0\}} h(e) = N - H \qquad (5)$$

where I and I are cover and marked images, respectively. However, as pointed out in [44], the choice of expansion bins as −1 and 0 is not mandatory. For this specific PEH, as shown in the right figure of Fig. 2, one can select the bins −1 and 1 for expansion. In this situation, only the bins larger than 1 or smaller than −1 need to be shifted, while the bin 0 may remain unchanged. As a result, compared with C-PEE, the expected value of embedding distortion is reduced from $N-H$ to $N - 2H$.

Through this example, we see that it is possible to improve C-PEE by suitably selecting expansion bins. For a given EC, one can select two expansion bins $a < b$ to minimize the embedding distortion, and the optimal expansion bins can be determined by repeated embedding for a collection of (a,b). An example for determining (a,b) will be given later at the end of this section. We now describe the O-PEE embedding procedure when a and b are already determined. This is a natural extension of C-PEE embedding. First, for a predictionerror e_i, it is modified as

$$\widetilde{e}_i = \begin{cases} e_i, & \text{if } a < e_i < b \\ e_i + m, & \text{if } e_i = b \\ e_i - m, & \text{if } e_i = a \\ e_i + 1, & \text{if } e_i > b \\ e_i - 1, & \text{if } e_i < a \end{cases} \qquad (6)$$

where $m \in \{0,1\}$ is a to-be-embeddeddata bit. Then, x_i is modified to $\widetilde{x}_i = \widehat{x}_i + e_i$ to generate the marked pixel. Clearly, O-PEE turns out to be C-PEE when $(a,b) = (-1,0)$. An illustration of O-PEE with $(a,b) = (-3,2)$ is shown in Fig. 3. One can see the difference between C-PEE and O-PEE by comparing this figure with Fig. 1.

The O-PEE extraction is just the inverse of data embedding. The obvious data extraction and image restoration procedure of O-PEE is omitted.

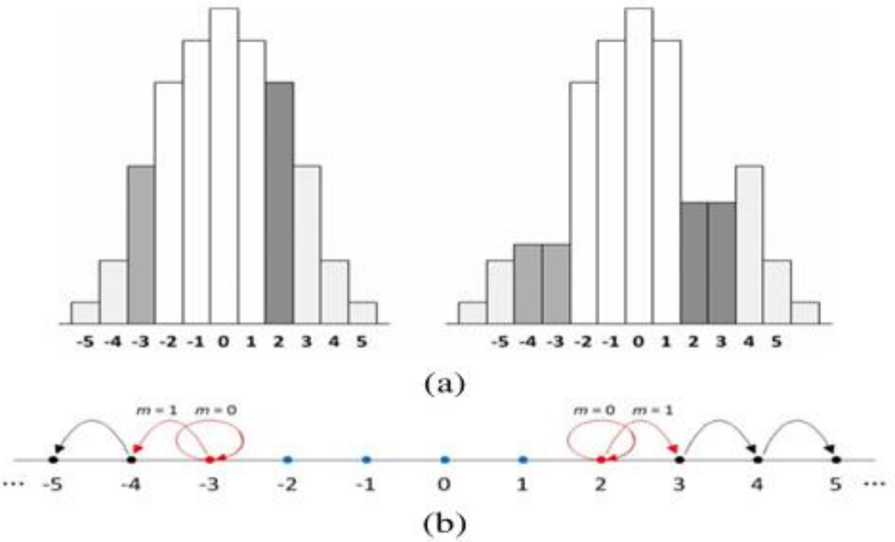


Fig. 8. Histogram modification mechanism for O-PEE with *(a,b)* = *(*−3,2*)*. Here, the bins {−2,−1,0, 1} are unmodified. (a) PEH before (left) and after (right) O-PEE embedding. (b) Mapping of bins for O-PEE.

x_i	u
v	w

Fig. 4. Context of a pixel *xi*.

D. AO-PEE

If the aforementioned two improvements for C-PEE, adaptive embedding and optimal expansion bins selection, are combined together, better performance can be expected [55], [56]. This combined embedding is called AO-PEE in our work. We now briefly describe the AO-PEE embedding procedure as follows. First, for a cover pixel x_i, if its complexity measurement $n_i < T$, the prediction-error e_i is modified according to (6) to get e_i. Then, x_i is modified to $x_i = x_i + e_i$ to get the marked pixel. Here, the same as A-PEE, the pixels with $n_i \geq T$ are ignored and unmodified. There are three parameters to be determined in AO-PEE: the complexity threshold T and expansion bins *(a,b)*. In [55], a recursive search method is proposed for determining these parameters. Specifically, for given *(a,b)*, T is taken as the smallest positive integer such that the payload can be successfully embedded and the corresponding embedding distortion is recorded as $D_{a,b}$. Then, repeat the above step for a collection of *(a,b)*, and finally obtain the optimal *(a,b, T)* as the one providing the smallest $D_{a,b}$.

We now give some experimental results of C-PEE, A-PEE, O-PEE and AO-PEE for a better explanation of these methods. The experiment is conducted as follows.

- The MED predictor is employed for prediction, i.e., referring to Fig. 4, the prediction value of x_i is

$$\hat{x}_i = \begin{cases} \min(u,v), & \text{if } w \geq \max(u,v) \\ \max(u,v), & \text{if } w \leq \min(u,v) \\ u+v-w, & \text{otherwise.} \end{cases} \qquad (7$$

=

x_i ⎩max*(u,v)*, if $w \leq \min(u,v)$ (7) $u + v - w$, otherwise.

- For A-PEE and AO-PEE, the complexity measurement n_i is computed as the mean of pixel differences in the context of x_i, i.e.,

$$n_i = \left\lfloor \frac{|u-v| + |v-w| + |w-u|}{3} \right\rfloor \qquad (\; 3$$

where · is the floor function.

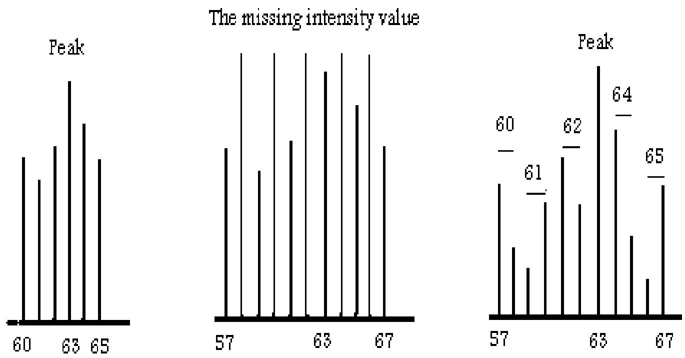
- For O-PEE and AO-PEE, the bins *(a,b)* satisfying $-7 \leq a < b \leq 7$ are tested to determine the optimal expansion bins.

The embedding results for the standard 512 × 512 sized gray-scale image Lena are listed in Table I. The corresponding parameters including the complexity threshold T (for A-PEE and AO-PEE) and optimal

expansion bins *(a,b)* (for O-PEE and AO-PEE) are also listed in this table, e.g., for AO-PEE with EC of 20,000 bits, $(a,b,T) = (-2,2,3)$.
According to this table, compared with C-PEE, better performance is achieved using A-PEE, O-PEE and AO-PEE. Moreover, with the combined improvements, AO-PEE performs the best among these PEE-based methods.

VI. Contrast Stretching

Contrast stretching is one of the histogram manipulation techniques. It tries to extend the narrow range of image pixel values over a wider range. It is often used to improve the image quality by stretching the range of intensity values such that the image contrast could be enhanced. Contrast enhancement will produce a histogram which contains many gaps. These gaps represent a missing set of pixels of particular intensity values. This additional created empty space would be used for data embedding. Figure 6 illustrates the proposed method of contrast enhancement for reversible data embedding. In this method, every intensity value in the original histogram is stretched to a position either to the left or to the right of the peak value, hence creating the missing intensity values. As large gaps of missing intensity values together could produce a downgraded look; the width of a gap is defined as one grayscale value in this work. The embedding process is implemented as fowllows: the image block is scanned pixel by pixel. The pixel values associated with the histogram peak are kept intact for reversibility. The rest may be selected to embed the watermark. If the current watermark bit is “1,” the pixel value is added or subtracted by 1, otherwise, they are kept intact. For example, Figures 6.a and 6.b show the original and stretched histograms of an image block, respectively. Pixels with gray level 57 are selected for embedding and gray level 58 is a missing intensity value. While scanning the image block, if the watermark bit is “1,” the value 57 of the current pixel is incremented by 1, whereas if the watermark bit is “0,” nothing is done. Figure 6.c shows the histogram after embedding the watermark. The total number of pixels of gray levels 57 and 58 is equal to the number of pixels of gray level 60 in the original histogram. The embedding capacity in this image block is the number of selected pixels, i.e., the number of pixels of gray levels 60, 61, 62, 64 and 65. [4]



(a) (b) (c)

Figure 9: Removable data embedding by contrast stretching: (a) original histogram of the image block, (b) stretched histogram, (c) histogram after embedding.

Let m be the image block of size N×N and m(i, j) denotes the original pixel value at position (i, j), where $0 \leq i, j \leq N-1$. The contrast stretching is defined as
$m'(i, j) = p + 2*[m(i, j) - p] = 2m(i, j) - p$, (1)
where m'(i, j) is the pixel value at i,j position after stretching and p is the gray level value with the highest count of pixels in the block's histogram. A threshold value τ is used to balance between the shift widths and the number of selected gray levels for embedding. In Figure 6.a, the pixels with gray values of 61, 62, 64 and 65 are selected to embed the watermark, therefore τ is selected to be 2. The original pixel values that satisfy $p - \tau \leq b(i, j) \leq p + \tau$ are mapped to the output value using the contrast stretching function in equation 1. Thus the range of $[p-2\times\tau, p+2\times\tau]$ is called the embedding area. Other pixel values will be shifted to the right or left by τ levels. This means
$b'(i, j) = 2*b(i, j) - p$, if $p - \tau \leq b(i, j) \leq p + \tau$ (2)
$b'(i, j) = b(i, j) - \tau$, if $b(i, j) < p - \tau$ (3)
$b'(i, j) = b(i, j) + \tau$, if $b(i, j) > p + \tau$ (4)
As grayscale values are bounded to the interval [0, 255], then $0 \leq b'(i, j) \leq 255$, which is equivalent to $0 \leq 2\times b(i, j) - p \leq 255$. To overcome the over/underflow problems (the range of shifted pixel values to be less than 0 or greater than 255) and the extreme value problem (if intensity value 0 or 255 exists, there is no room for

stretching). Accordingly, extra condtions should be applied to avoid such problems. The pixel values in an image block must satisfy the following condition:

$p/2 < b(i, j) < (p + 255)/2$ (5)

Considering the threshold, the pixel values violating condition (3) might not cause any problem if they satisfy another constraint:

$\tau < b(i, j) < 255 - \tau$ (6)

In the cases where the value in an image block satisfies equation (5) & (6) , then it could be stretched via equations (2) ,(3) & (4); otherwise, the following equations are used instead without causing the over flow or underflow problem:

$b'(i, j) = 2*b(i, j)-p$, if $p-\varepsilon_1 \le b(i, j) \le p + \varepsilon_2$ (7)

$b'(i, j) = b(i, j)- \tau$, if $b(i, j) < p - \varepsilon_1$ (8)

$b'(i, j) = b(i, j)+ \tau$, if $b(i, j) > p + \varepsilon_2$ (9)

where $\varepsilon 1$ and $\varepsilon 2$ are the numbers of empty bins in the histogram from 0 to the minimum gray level and from the maximum gray level to 255, respectively. The range of $[p-2\times\varepsilon 1, p + 2\times\varepsilon 2]$ is also called the embedding area. In addition, the number of the gray levels to be stretched will be the number of the missing intensity values clustered at one end of the histogram. If the range of grayscale values in the histogram covers the full possible set of values, straightforward contrast stretching will achieve nothing; in other words, the histogram should have at least one gap on the left or right side of the peak in order to provide room for stretching. Usually, the pixel values in an image block spread only within a small range so that they could be stretched.

VII. Reversible Data Embedding

The reversible data embedding algorithm can be summarized in the following steps:

The original image is divided into equally-sized (N×N), non-overlapping image blocks, and N is set to be less than 16 such that the full-coverage problem is avoided.

The histogram statistics is computed for each image block, including the gray level with the most counts, p, and the number of pixels. If there is more than one peak in a block, the peak is defined as the one with greater gray level. Figure 10 shows one sample for a block histogram before stretching. [5][6]

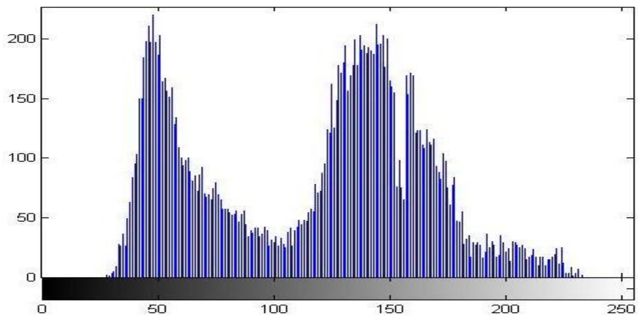


Figure 10: A Sample of Block Histogram before Stretching.

Contrast stretching is performed on each image block to create extra embedding space. Figure 11 shows the histogram of a block after contrast stretching.

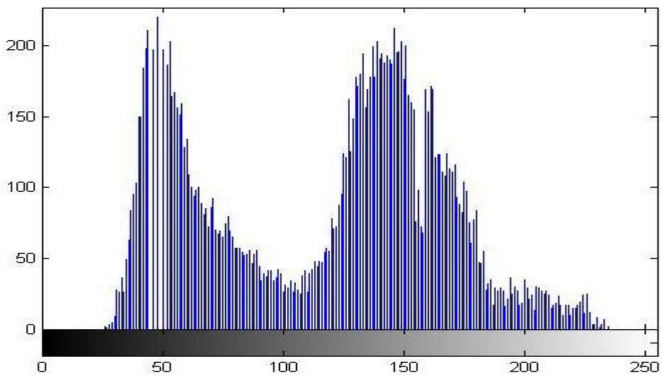


Figure 11: Block Histogram after Stretching.

Each block is divided into two sets of pixels, L (left of the histogram) and R (right of the histogram), according to the pixel values being less or greater than p. In some exceptional cases, if the peak is the minimum or maximum gray level, this block is only divided into R or L accordingly. If all of the pixels in the block have the same value, this block is discarded.

The watermark is embedded into the created space after contrast stretching and the watermarked image is obtained. The data embedding starts with scanning the pixel values, and pixel values in the embedding area are modified by either adding or subtracting one bit:

$b''(i, j) = b'(i, j)-1, \text{ if } b'(i, j) > p$ (10)

$b''(i, j) = b'(i, j)+1, \text{ if } b'(i, j) < p$ (11)

Where b"(i, j) denotes the pixel value after data embedding. Figure 9 shows the block histogram after data embedding.

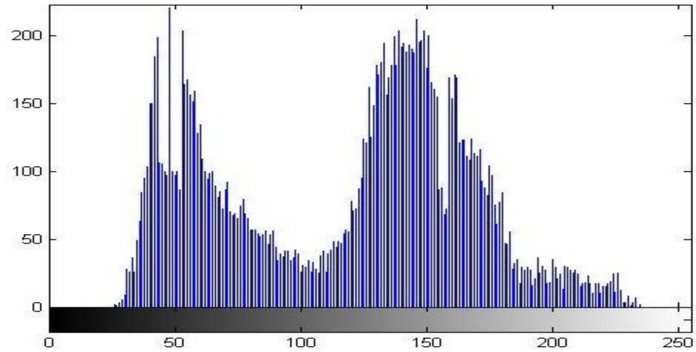


Figure 12:Block Histogram after Data Embedding.

VIII. Experimental Results

To verify the algorithm, we make use of standard color grayscale images in the experiment, color and grayscale watermark were visibly embedded to verify whether the lossless representation of the original image. We are comparing our watermarking and watermark removal method using four techniques,

1) Linear Histogram modification using Encryption
2) Linear Histogram modification without using Encryption
3) Equalized Histogram modification using Encryption
4) Equalized Histogram modification without using Encryption

Fig. 13 Original image

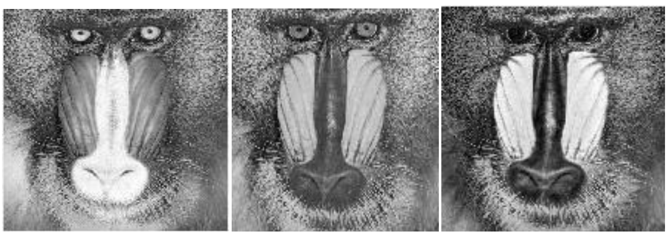
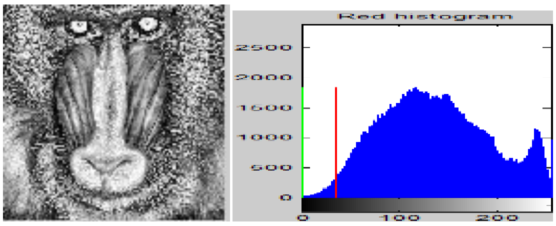
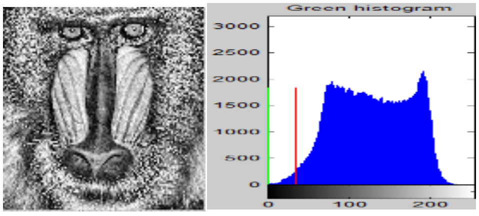


Fig. 14 Separation of Red, Green and Blue component

Red Component



Green Component



Blue Component

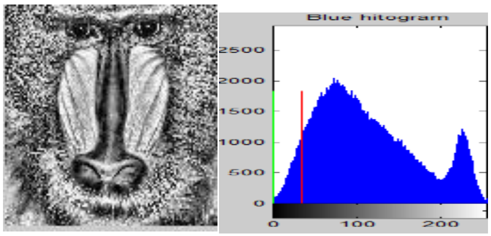
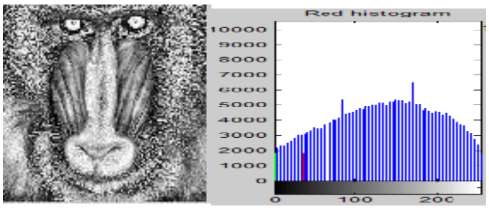
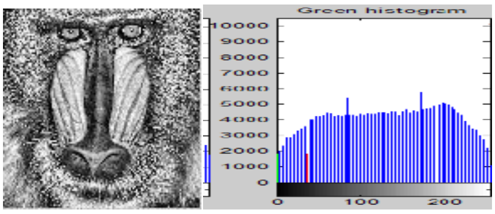


Fig. 15 Histogram of Red, Green and Blue component

Red Component



Green Component



Blue Component

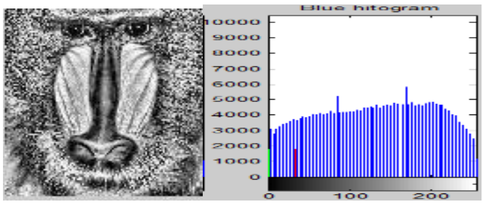


Fig. 16 Equalized Histogram of Red, Green and Blue component

a2. Original Image **b2.** Watermarking

c2. Watermarked image **d2.** Lossless restore image

Fig. 17the visible lossless Digital watermarking experiments

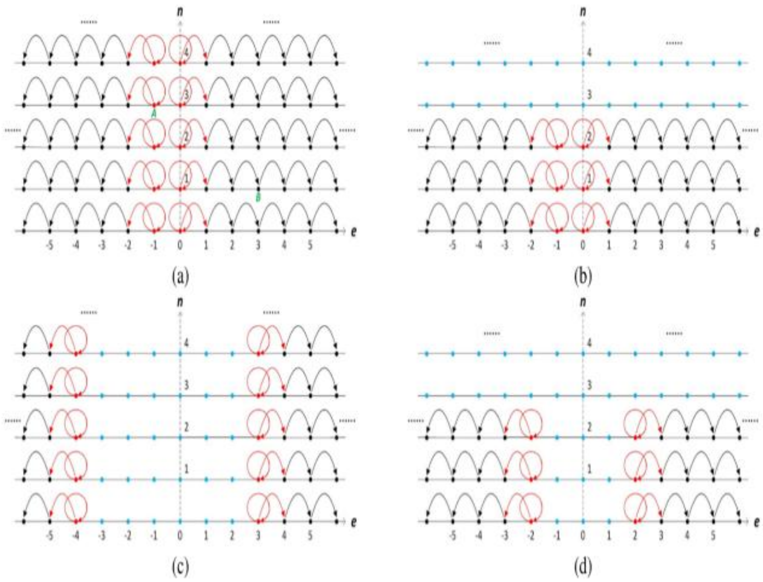


Fig. 18 Four Methods Graphical Representation

Table 1. Comparisons of four methods

methods	Embedded capacity(bits)	PSNR VALUE(dB)
C-PEE	16,384	Y=59.8245
A-PEE (T=3)	16,384	Y=67.1486
O-PEE(a= 4,b=3)	16,384	Y=60.8521
AO-PEE (a=-2,b=2,T=3)	16,384	Y=63.7073

IX. Conclusion

This proposed work, focus on a new reversible data compression hiding scheme for hiding message in JPEG image. In this to location map will be compressed to increase in order to increase the embedding capacity. The location map depend on the payload or hidden message. Comparing with other RDH techniques and literature, this present work often has enhanced flexibility to different images and larger embedding capacity for the same image quality.So the proposed framework hasa potential to provide excellent RDH algorithms and Zhicheng's Method. Finally in this paper, we compared with C-PEE, better performance is achieved using A-

PEE, O-PEE and AO-PEE. Moreover, with the combined improvements, AO-PEE performs the best among these PEE-based methods

References

[1]. Zhenxing Qian, Xinpeng Zhang and ShuozhongWang, "Reversible Data Hiding in Encrypted JPEG Bitstream." IEEE Trans. on Multimedia, VOL. 16, NO.5, Apr. 2014.

[2]. W. Hong, T. S. Chen, and C. W. Shiu, " Reversible Data Hiding Based on Histogram Shifting of Prediction Errors," in Intelligent Onformation Technology Application Workshops, pp. 292-295, 2008.

[3]. J.M. Barton, "Method and Apparatus for Embedding Authentication Information Within Digital Data," U. S. Patent 5646997, 1997.

[4]. A. M. Alattar, " Reversible Watermark Using the Difference Expansion of a generalized Integer Transform, IEEE Transaction Image processing, VOL. 13, no. 8 pp. 1147-1156, 2004.

[5]. H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. G. Choo, A Novel Difference Expansion Transform for Reversible Data Embedding, IEEE Transaction Information Forensics and Security, vol. 3, no. 3, pp. 456-465, 2008.

[6]. M. Fallahpour, "Reversible image data hiding based on gradient adjusted prediction," IEICE Electron. Express, vol. 5, no. 20, pp.870–876, Oct. 2008.

[7]. S. W. Wang, Y. Zhao, and J. S. Pan, "A Novel Reversible Data Hiding Scheme," International Journal of Innovative Computing, Information and control, vol. 4, no. 2, pp. 351-358, 2008.

[8]. Xinpeng Zhang, "Separable Reversible Data Hiding in Encrypted Image." IEEE Trans. on information forensics and security, VOL. 7, NO.2, Apr. 2012.

[9]. J. Tian, "Reversible Data Embedding Using a Difference Expansion" IEEE Transaction Circuits Syst. Video Technology VOL. 13, NO.8, Aug. 2003.

[10]. J. Fridrich, M. Goljan, and R. Du, "Lossless data Embedding – New Paradigm in Digital Watermarking," EURASIP J. Application Signal Process., vol. 2002, no. 2, pp. 185-196, Feb 2002.