International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 8 (2019) pp. 2029-2037
© Research India Publications. https://dx.doi.org/10.37622/IJAER/14.8.2019.2029-2037

Reversible Data Hiding Scheme using Prediction Error Expansion in Pixel Value Blocking and Ordering

Satyajit De
Department of Computer Science,
Maheshtala College, Budge Budge Trunck Road,
Kolkata-700141, West Bengal, India.Email:

Alok Haldar
Department of Computer Science,
Kharagpur College,West Midnapur,
West Bengal, India.

Biswapati Jana
Department of Computer Science,
Vidyasagar University, West Midnapur,
West Bengal, India.

Abstract

This paper presents a better reversible data hiding method depending on pixel value ordering and prediction-error expansion technique. A host image is first segmented into non-overlapping sub-blocks of adjacent three pixels and ordered them as ascending order. For each sub-block, the maximum pixel value is predicted by the second maximum pixel value. Then the second maximum pixel value is predicted by the minimum pixel value or minimum pixel value is predicted by the second maximum pixel value. Then by using prediction-error expansions, we can insert one or two secret bits into every sub-block pixels and also we can recover the hidden secret bits and restore the cover image fully from watermarked image. Experimental results of this method demonstrate that the embedding capacity and PSNR value is larger than another pixel value ordering and prediction error expansion based approach. Also, the visual quality of the obtained marked image is better than other PVO and PEE based method.

Keywords: Pixel-value ordering, Reversible data hiding, Prediction-error expansion, Adjacent pixel grouping.

INTRODUCTION

In the spatial domain different reversible data hiding methods have been divided as lossless compression [3-4], difference expansion (DE) [5-8], histogram shifting (HS) [11-12], prediction- error expansion (PEE) [15], etc. in the spatial domain. DE technique and LSB embedding scheme are used to get a minimum image distortion with high embedding capacity. In adjacent pixels blocks, Alattar applied difference expansion technique to embedded secret bits [7]. To increase the embedding capacity, Al-Qershi et al. used a two-dimensional difference expansion technique (2D-DE) with a threshold value depending on the image behavior [8]. The multilevel histogram technique is used by Zhao et al. to embed more secret bits [11]. In his method to enhance embedding capacity secret bits are modulated by using more peak points. Luo et al. generate a strong connection among different pixel blocks to produce a difference histogram and multi-level histogram shifting to embed the secret data [10].

A lossless data hiding method based on a histogram modification is proposed by Ni et al., where the zero or minimum points of the image histogram are utilized [17].

A reversible data hiding method with prediction-error expansion is introduced by Thodi and Rodriguez.

Hu et al. introduced a modify reversible data hiding scheme by reducing the overflow location map. Li et al. Proposed an improvement by using adaptive embedding and pixel selection. Lee et al. proposed a reversible data hiding scheme that is free of location map and a corresponding predictive value is derived from the average of its adjacency pixels to make little bit predictive errors.

Li et al. proposed a reversible data hiding scheme [1] using pixel-value-ordering and prediction-error expansion. After sorting in ascending order of every non-overlapped sub-block of equal sizes, the second maximum or minimum pixel value was used to predict the maximum pixel or minimum pixel value respectively. Then by applying prediction-error expansion technique secret data was embedded. Best result was achieved in this technique by using 2x2 sub-blocks i.e. 4 pixels' sub-block. But for this method, maximum pixels are not used to improve the embedding capacity as well as the image quality for every sub-block and this improvement is done by Jung's method.

Jung's proposed a reversible data hiding scheme [2] using pixel-value-ordering and prediction-error expansion. To improve Li et al's scheme Jung's divide the cover image into three pixels non-overlapped sub-blocks. For each sub-block, sorted the pixels in ascending order and then the second largest pixel value was used to predict the maximum pixel value. Then to embed secret data prediction error expansion was applied into it. As a result, high embedding capacity and good image quality occurs than Li et al's method. But still there is a space available to improve the embedding capacity as well as the image quality for every sub-block and this improvement is done by the proposed method.

In this paper to improve Jung's method a new reversible data hiding scheme is proposed for three pixels' sub-blocks based on pixel value ordering with prediction error expansion. Three pixels of each sub-blocks are ordered in ascending order and two prediction errors is calculated. One is between maximum and second maximum pixel value and the other between second maximum and minimum pixel value. Depending on these prediction errors it is tested that the sub-blocks can be embed two/one secret bits. Then by applying pixel-error expansion two/one secret data is hided into the sub-blocks and then all secret bits can be fetched and restore the cover image from watermarked image completely. As a result, embedding capacity and image quality becomes high.

RELATED WORK

Review of Li et al.'s Scheme

Li et al.'s used in [1] Pixel value ordered based predictor with Pixel error expansion based reversible data embedding scheme. For secret bits embedding according to the Li et al.'s method [1], the cover image is divided into different non-overlapped equal-sized sub-blocks. For a chosen block, sort the pixel values in ascending order and then it use the second largest pixel value to predict the maximum pixel or second smallest pixel value to predict minimum pixel. Best performance (For embedding capacity and PSNR value) is achieved in his paper by taking 2 × 2 size blocks. Suppose a 2 × 2 block of sorted order pixels with pixel values (si, si+1, si+2, si+3). It used the second largest value si+2 to predict the maximum value si+3. For this the prediction error is calculated as e2=si+3 - si+2. Prediction error e2 is modified to e2´ as follows.

$$e_2' = \begin{cases} e_2 & \text{if } e_2 = 0 \\ e_2 + p_2 & \text{if } e_2 = 1 \\ e_2 + 1 & \text{if } e_2 > 1 \end{cases} \qquad (1)$$

Where p_2 is a secret bit either 0 or 1 to be embedded.

Now the maximum pixel s_{i+3} is modified to s'_{i+3} to store secret bit p_2 as follows

Where p_2 is a secret bit either 0 or 1 to be embedded.

Now the maximum pixel s_{i+3} is modified to s'_{i+3} to store secret bit p_2 as follows

$$s'_{i+3} = \begin{cases} s_{i+3} & \text{if } e_2 = 0 \\ s_{i+3} + p_2 & \text{if } e_2 = 1 \\ s_{i+3} + 1 & \text{if } e_2 > 1 \end{cases} \qquad (2)$$

It also used the second smallest value s_{i+1} to predict the minimum value s_i. For this the prediction error is calculated as $e_1 = s_i - s_{i+1}$. Prediction error e_1 is modified to e_1' as follows.

$$e_1' = \begin{cases} e_1 & \text{if } e_1 = 0 \\ e_1 - p_1 & \text{if } e_1 = -1 \\ e_1 - 1 & \text{if } e_1 < -1 \end{cases} \qquad (3)$$

Where p_1 is a secret bit either 0 or 1 to be embedded.

Now the minimum pixel s_i is changed to s_i' to store secret bit p_1 as follows

$$s_i' = \begin{cases} s_i & \text{if } e_1 = 0 \\ s_i - p_1 & \text{if } e_1 = -1 \\ s_i - 1 & \text{if } e_1 < -1 \end{cases} \qquad (4)$$

Now, the marked block is obtained as $(s'_i, s_{i+1}, s_{i+2}, s'_{i+3})$.

Again for data extraction from maximum pixel the prediction error is calculated as $e_2' = s'_{i+3} - s_{i+2}$.

If $e_2'=0$ then this sub-block does not contain secret bit for maximum pixel and the original pixel is unchanged. If $1 \le e_2' \le 2$ then this sub-block contains a secret bit for maximum pixel and the secret bit is $p_2 = e_2' - 1$ and the original maximum pixel is restored as $s_{i+3} = s'_{i+3} - p_2$. If $e_2' > 2$ then this sub-block does not contain any secret bit for maximum pixel and the original maximum pixel is restored as $s_{i+3} = s'_{i+3} - 1$.

Again for data extraction from minimum pixel the prediction error is calculated as $e_1' = s'_i - s_{i+1}$. If $e_1'=0$ then this sub-block does not contain secret bit for minimum pixel and the original pixel is unchanged. If $-2 \le e_1' \le -1$ then this sub-block contains a secret bit for minimum pixel and the secret bit is $p_1 = |e_1'| - 1$ and the original minimum pixel is restored as $s_i = s'_i + p_1$. If $e_1' < -2$ then this sub-block does not contain any secret bit for minimum pixel and the original minimum pixel is restored as $s_i = s'_i + 1$.

But for this method, still there is a space available to improve the embedding capacity as well as the image quality for every sub-block and this improvement is done by Jung's method [2].

Review of Jung's Scheme

Jung used [2] PVO based predictor and PEE based reversible data embedding scheme to embed secret bits. In this method cover image is divided into three pixels non-overlapped sub-blocks. For a chosen sub-block, sort the pixels in ascending order and then using second largest pixel value predict the maximum pixel to embed data like Li et. al.'s scheme. Suppose a three-pixel sub-block of sorted order with pixel values is (s_i, s_{i+1}, s_{i+2}).

It used the second largest value s_{i+1} to predict the maximum value s_{i+2}. For this the prediction error is calculated as $e = s_{i+2} - s_{i+1}$. Prediction error e is modified to e´ by Jung as follows.

$$e' = \begin{cases} e & \text{if } e = 0 \\ e + p & \text{if } e = 1 \\ e + 1 & \text{if } e > 1 \end{cases} \qquad (5)$$

Where p is a secret bit either 0 or 1 to be embedded.

Now the maximum pixel s_{i+2} is changed to s'_{i+2} to store secret bit p as follows

$$s'_{i+2} = \begin{cases} s_{i+2} & \text{if } e = 0 \\ s_{i+2} + p & \text{if } e = 1 \\ s_{i+2} + 1 & \text{if } e > 1 \end{cases} \qquad (6)$$

Now, the marked sub-block is obtained as ($s_{i,}$ s_{i+1}, $s´_{i+2}$).

Again for data extraction from maximum pixel the prediction error is calculated as $e´=s´_{i+2} - s_{i+1}$.

If $e´=0$ then this sub-block does not contain secret bit for maximum pixel and the original pixel is unchanged.

If $1 \leq e´\leq 2$ then this sub-block contains a secret bit for maximum pixel and the secret bit is $p = e´ - 1$ and the original maximum pixel is restored as $s_{i+2}= s´_{i+2} - p$.

If $e´ > 2$ then this sub-block does not contain any secret bit for maximum pixel and original maximum pixel is restored as $s_{i+2}= s´_{i+2} - 1$. But in Jung's method it is not used that the minimum pixel value to predict the second maximum pixel again to embed secret bits or does not used the second largest pixel value to predict the minimum pixel to embed secret bits, therefore still there is a space available to improve the embedding capacity and image quality of every sub-block. In the previous method besides the overflow or underflow checking no other checking are used to skip the blocks which are not capable to store secret bits, therefore PSNR values becomes low. Now a better reversible data hiding scheme is proposed to improve EC, PSNR and image quality than the Li et al.'s and Jung's scheme in the following section.

Motivation and Objective

Motivation and objective of the proposed scheme is to increase embedding capacity using PVO based PEE technique within the cover image without compromising visual quality. Here, the capacity is measured by bpp and image quality is measured by PSNR (dB). The proposed scheme achieve good visual quality maintaining high embedding capacity and compred better result than the existing schemes.

Proposed Scheme

In the proposed method based on pixel-value ordering and prediction error expansion we establish a better reversible watermarking scheme for the three-pixel non-overlapped sub blocks. The embedding and extracting procedures are as follows.

Embedding Procedure

Algorithm for data embedding

The detailed algorithm for embedding secret data is described as follows:

Input: A cover image of order r × c and the sequence of secret bits which have to be embedded to this image.

Output: A watermarked image of order r × c

Step 1: Read a cover image of size r × c from user.

Step 2: This cover image is divided into three pixels non-overlapping sub-blocks.

Step 3: Consider a generalise i^{th} sub-block containing three pixels as (c_i, c_{i+1}, c_{i+2}).

Step 4: Pixels values (c_i, c_{i+1}, c_{i+2}) are sorted in ascending order to obtain (s_i, s_{i+1}, s_{i+2}) (i.e. $s_i \leq s_{i+1} \leq s_{i+2}$).

Step 5: Prediction errors are calculated as follows

$$e_2=s_{i+2} – s_{i+1}$$

$$e_1=s_{i+1} - s_i$$

Step 6: (Construction of Location map)

[Here we use an array lmap() as a location map. lmap(i)=0, indicates that i^{th} sub-block does not contain any secret bits. lmap(i)=1 indicates that secret bits exists either at the maximum pixel (s_{i+2}) or at second maximum pixel (s_{i+1}) or both of them in the i^{th} sub-block and also lmap(i)=2, indicates that secret bits exists either at the maximum pixel (s_{i+2}) or at minimum pixel (s_i) or both of them in that sub-block.]

[Skip the sub-block by setting the location map value to 0 i.e. lmap(i)=0, if it is not capable to embed data.]

We check the embedding capability of the i^{th} sub-block using the following conditions

i) $s_i = 0$ or $s_{i+2}=255$ [condition for underflow or overflow]

ii) $e_2 \neq 1$ and $e_1 \neq 1$

If any one of the above conditions hold then this sub-block have no capability to embed data and we skip the sub-block by setting the location map value to 0 i.e. lmap(i)=0. Store the sorted pixels (s_i, s_{i+1}, s_{i+2}) into the corresponding positions and ultimately we get ($c´_i$, $c´_{i+1}$, $c´_{i+2}$). Then go to the next sub-block by setting i=i+1 and continue from step 3 otherwise go to step 7.

Step7: Prediction error e_2 is modified as follows

$$e_2´= \begin{cases} e_2 & \text{if } e_2 = 0 \\ e_2 + p_2 & \text{if } e_2 = 1 \\ e_2 + 1 & \text{if } e_2 > 1 \end{cases} \quad (7)$$

Where p_2 is a secret bit data either 0 or 1 has to be embedded.

Step 8: Now the maximum pixel s_{i+2} is changed to $s´_{i+2}$ to store a secret bit p_2 as follows

$$s´_{i+2} = \begin{cases} s_{i+2} & \text{if } e_2 = 0 \\ s_{i+2} + p_2 & \text{if } e_2 = 1 \\ s_{i+2} + 1 & \text{if } e_2 > 1 \end{cases} \quad (8)$$

Step 9: Now the prediction error e_1 is modified as follows

$$e_1´= \begin{cases} e_1 & \text{if } e_1 = 0 \\ e_1 + p_1 & \text{if } e_1 = 1 \\ e_1 + 1 & \text{if } e_1 > 1 \end{cases} \quad (9)$$

Where p_1 is a secret bit data either 0 or 1 has to be embedded.

Step 10: If e_1=0 then skip the pixels s_{i+1} and s_i from data embedding and these pixels remains unchanged. Now the resultant watermarked ith sub-block is (s_i, s_{i+1}, $s´_{i+2}$). Set the location map value to 1 i.e. lmap(i)=1 to indicate that secret bits may exists either in maximum pixel or second maximum pixel or both. Store the pixels (s_i, s_{i+1}, $s´_{i+2}$) into the corresponding positions and ultimately we get ($c´_i$, $c´_{i+1}$, $c´_{i+2}$). Then go to next sub-block by setting i=i+1 and continue from step 3 otherwise go to step 11.

Step 11: Now the second maximum pixel s_{i+1} is changed to $s´_{i+1}$ to store a secret bit p_1 provided $s´_{i+1} < s´_{i+2}$. For that we calculate a value v as follows:

$$v = \begin{cases} s_{i+1} + p_1 & \text{if } e_1 = 1 \\ s_{i+1} + 1 & \text{if } e_1 > 1 \end{cases} \tag{10}$$

Step 12: If v < $s´_{i+2}$ then the value v is effected on the second maximum pixel s_{i+1} and it is changed to $s´_{i+1}$=v. Therefore, we get the watermarked ith sub-block as (s_i, $s´_{i+1}$, $s´_{i+2}$). Set the location map to 1 for this sub-block i.e. lmap(i)=1 to indicate that secret bits may exists either in maximum pixel or second maximum pixel or both.

Otherwise

The minimum pixel s_i is changed to $s´_i$ to store secret bit p_1 as follows

$$s´_i = \begin{cases} s_i - p_1 & \text{if } e_1 = 1 \\ s_i - 1 & \text{if } e_1 > 1 \end{cases} \tag{11}$$

Therefore, we get the watermarked ith sub-block as ($s_i´$, s_{i+1}, $s´_{i+2}$). Set the location map to 2 for this sub-block i.e. lmap(i)=2 to indicate that secret bits may exists either in maximum pixel or minimum pixel or both.

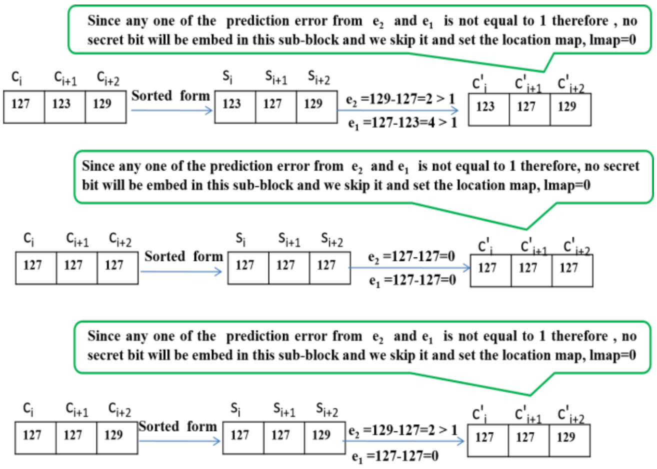
Step 13: Store the watermarked ith sub-block pixels into the corresponding positions and we get ($c´_i$, $c´_{i+1}$, $c´_{i+2}$).

Step 14: Go to the next sub-block by setting i=i+1 and continue the above procedure from step 3 until all data bits are embedded.

Step 15: Consider the last data embedding sub-block number is bl. Now the array of location map values "lmap()" is losslessly compressed by arithmetic coding and store the compressed location map with auxiliary information (i.e. last sub-block number bl and length of the compressed location map bits len_{lm}) by replacing X = $2*\lceil(\log_2 M)\rceil$ + len_{lm} [where M=total number of pixels in cover image] least significant bits of image pixels from beginning. Before replacing the least significant bits of X pixels from beginning we store the least significant bits of X pixels into remaining blocks starting from the block no (bl + 1) using the same procedure as above. Finally, we get a watermarked image.

Numerical examples for data embedding

$$a´_{i+1} = \begin{cases} a_{i+1} + p_0 & \text{if } a_{i+1} + p_0 < a´_{i+2} \\ a_{i+1} - p_0 & \text{if } a_{i+1} - p_0 > a´_i \end{cases} \tag{6}$$

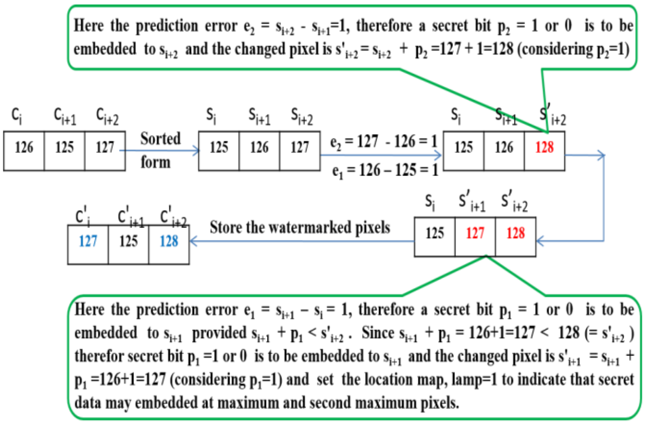

Since any one of the prediction error from e_2 and e_1 is not equal to 1 therefore , no secret bit will be embed in this sub-block and we skip it and set the location map, lmap=0

Since any one of the prediction error from e_2 and e_1 is not equal to 1 therefore, no secret bit will be embed in this sub-block and we skip it and set the location map, lmap=0

Since any one of the prediction error from e_2 and e_1 is not equal to 1 therefore , no secret bit will be embed in this sub-block and we skip it and set the location map, lmap=0

Fig 1: Example of sub-blocks which are not capable to embed secret bits.

In the above examples (i.e. in Fig 2) of sub-blocks since no prediction errors e_1 or e_2 have value 1, therefore these sub-blocks have no capability to embed secret bits, hence skip these blocks from embedding. Similarly, the sub-blocks in Fig 3, have no capability to embed data and skip them from embedding.


Here the prediction error e_2 = s_{i+2} - s_{i+1}=1, therefore a secret bit p_2 = 1 or 0 is to be embedded to s_{i+2} and the changed pixel is s'_{i+2}= s_{i+2} + p_2 =127 + 1=128 (considering p_2=1)

Here the prediction error e_1 = s_{i+1} – s_i = 1, therefore a secret bit p_1 = 1 or 0 is to be embedded to s_{i+1} provided s_{i+1} + p_1 < s'_{i+2} . Since s_{i+1} + p_1 = 126+1=127 < 128 (= s'_{i+2}) therefor secret bit p_1 =1 or 0 is to be embedded to s_{i+1} and the changed pixel is s'_{i+1} = s_{i+1} + p_1 =126+1=127 (considering p_1=1) and set the location map, lamp=1 to indicate that secret data may embedded at maximum and second maximum pixels.
c_i c_{i+1} c_{i+2}
126 125 127
Sorted form
s_i s_{i+1} s_{i+2}
125 126 127
e_2 = 127 - 126 = 1
e_1 = 126 – 125 = 1
s_i s_{i+1} s'_{i+2}
125 126 128
s_i s'_{i+1} s'_{i+2}
125 127 128
Store the watermarked pixels
c'_i c'_{i+1} c'_{i+2}
127 125 128

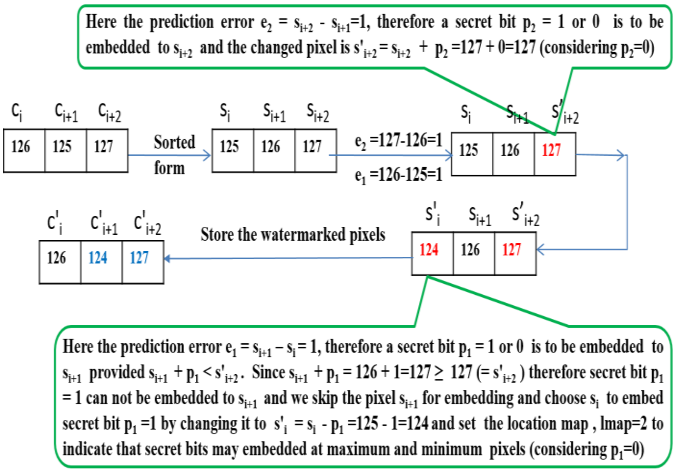
Fig 2: Example of a sub-block which is capable to embed two secret bits into maximum and second maximum pixels with location map value 1.

In Fig 2, consider a sub-block of pixel values c_i=126, c_{i+1}= 125, c_{i+2}=127. After sorting them in ascending order we get s_i=125, s_{i+1}=126 and s_{i+2}=127. Calculate the prediction errors e_2=s_{i+2} – s_{i+1}=127 - 126=1 and e_1=s_{i+1} - s_i=126 - 125=1. Suppose we want to store secret bits p_1=1 and p_2=1. Since e_2 =1, therefore it is changed to $e_2´$=e_2 + p_2 =1 + 1 =2 and the maximum pixel embed the bit p_2 and it is changed to $s´_{i+2}$=s_{i+2}

$+ p_2=127 + 1=128$. Similarly, since $e_1=1$ therefore s_{i+1} embed the bit $p_1=1$ and s_{i+1} is changed to $s'_{i+1}=s_{i+1} + p_1 =126 + 1 =127 < s'_{i+2} = 128$ and set the location map to 1. So, we get the watermarked sub-block as $(s_i, s_{i+1}', s_{i+2}') = (125,127,128)$ and they are stored as $(c'_i, c'_{i+1}, c'_{i+2}) = (127,125,128)$. We accept s'_{i+1} because $s'_{i+1} < s'_{i+2}$, otherwise we can embed data into s_i and set location map to 2 shown in Fig 3. The data extraction for this watermarked sub-blocks are also shown in Fig 4 and

Fig 5 respectively.

Others embedding and extracting situations are as obvious as previous.



Here the prediction error $e_2 = s_{i+2} - s_{i+1}=1$, therefore a secret bit $p_2 = 1$ or 0 is to be embedded to s_{i+2} and the changed pixel is $s'_{i+2} = s_{i+2} + p_2 =127 + 0=127$ (considering $p_2=0$)

Here the prediction error $e_1 = s_{i+1} - s_i = 1$, therefore a secret bit $p_1 = 1$ or 0 is to be embedded to s_{i+1} provided $s_{i+1} + p_1 < s'_{i+2}$. Since $s_{i+1} + p_1 = 126 + 1=127 \geq 127 (= s'_{i+2})$ therefore secret bit $p_1 = 1$ can not be embedded to s_{i+1} and we skip the pixel s_{i+1} for embedding and choose s_i to embed secret bit $p_1 =1$ by changing it to $s'_i = s_i - p_1 =125 - 1=124$ and set the location map , lmap=2 to indicate that secret bits may embedded at maximum and minimum pixels (considering $p_1=0$)

Fig 3: Example of a sub-block which is capable to embed two secret bits into maximum and minimum pixels with location map value is 2

Data Extraction Procedure

A watermarked image is divided into three pixels non-overlapping sub-blocks. In Extracting method, we extract the secret bits and restore the pixel values in its original form as follows.

Algorithm for data extraction and image recovery

Input: A watermarked image of order r × c.

Output: Original image i.e. cover image and hided secret bits.

Step 1: To get the auxiliary information [i.e. last embedding block number bl and length of the compressed location map len_{lm}] from the watermarked image we read first $L=2*\lceil(\log_2 M)\rceil$ [where M=total number of pixels in watermarked image] least significant bits of pixels. Then read next len_{lm} lease significant bits of pixels to get the compressed location map and then decompressing it to get the array of location map values "lamp()".

Step 2: Divide the watermarked image into 3 pixels non-overlapped sub-blocks starting from pixel number L+ len_{lm}+1. Firstly, by using the following data extraction and image restoration procedure extract the sequence of X = L+ len_{lm} hided bits from the sub-block number (bl + 1) to the last block. Now replace the least significant bits of first X pixels by the sequence of X extracted bits. Further we apply the following procedure to retrieve the watermark bit sequence and restore the pixels in its original form.

Step 3: i = 1 //This variable i indicates the block number.

Step 4: Consider the ith sub-block containing pixels as $(c'_i, c'_{i+1}, c'_{i+2})$. Now pixels values are sorted in ascending order and we obtain $(s'_{i,}, s'_{i+1}, s'_{i+2})$. [i.e. $s'_{i,} \leq s'_{i+1} \leq s'_{i+2}$].

[Here the location map lmap(i)=0, indicates that ith sub-block does not contain any secret bits. lmap(i)=1 indicates that secret bits may exists either at the maximum pixel (s'_{i+2}) or at second maximum pixel (s'_{i+1}) or both of them in the ith sub-block and also lmap(i)=2, indicates that secret bits may exists either at the maximum pixel (s'_{i+2}) or at minimum pixel (s'_i) or both of them in that sub-block.]

Step 5: if lmap(i)=0 then no secret bits exists and pixel values are unchanged. So, store the pixel values of ith sub-block i.e. $(s'_{i,}, s'_{i+1}, s'_{i+2})$ to their original positions to get the cover image sub-block as (c_i, c_{i+1}, c_{i+2}). Then go to the next sub-block by setting i=i+1 and continue from step 4 otherwise go to next step 6. [end of if]

Step 6: If lmap(i)=1 then

a) [extract the 2nd secret bit p_1 for the ith sub-block as follows]

Calculate the prediction error $e'_1 = s'_{i+1} - s'_i$, then the secret bit extraction and image restoration for the pixel s'_{i+1} is as follows.

- If $e'_1 > 2$ then no secret bit exists in s'_{i+1} and the recovered cover pixel value is $s_{i+1} = s'_{i+1} - 1$
- If $1 \leq e'_1 \leq 2$ then the recovered secret bit is $p_1= e'_1 - 1$ and the recovered cover pixel value is $s_{i+1} = s'_{i+1} - p_1$.
- If $e'_1 =0$ no secret bits are there and s'_{i+1} becomes unchanged and the cover pixel value is $s_{i+1} = s'_{i+1}$

b) [extract the 1st secret bit p_2 for the ith sub-block as follows]

Calculate the prediction error $e_2'=s'_{i+2} - s'_{i+1}$, then the secret bit extraction and image restoration for the pixel s'_{i+2} as follows.

- If $e_2'=0$, no secret bits exist in s'_{i+2} and s'_{i+2} becomes unchanged and the cover pixel value is $s_{i+2}=s'_{i+2}$
- If $1 \leq e_2' \leq 2$, secret bit is there and the secret bit is $p_2=e_2'-1$ and the recovered cover pixel value is $s_{i+2}=s'_{i+2} - p_2$
- If $e_2' > 2$, no secret bit is there and sub-block is shifted in data embedding and its recovered cover pixel value is $s_{i+2}=s'_{i+2}-1$

So, the required fetched bit sequence is $p_2\ p_1$ (if they exists) and the recovered cover pixels of ith sub-block is $(s'_{i,}, s_{i+1}, s_{i+2})$. Now go to step 8.

[end of outer if of step 6]

Step 7: If lmap(i)=2 then

a) [extract 1st secret bit p_2 for the ith sub-block as follows]

Calculate the prediction error $e_2'=s'_{i+2} - s'_{i+1}$, then the secret bit extraction and image restoration for the pixel s'_{i+2} as follows.

- If $e_2'=0$, no secret bits exist in s'_{i+2} and s'_{i+2} becomes unchanged and the cover pixel value is $s_{i+2}=s'_{i+2}$
- If $1 \le e_2' \le 2$, secret bit is there and the secret bit is $p_2=e_2'- 1$ and the recovered cover pixel value is $s_{i+2}=s'_{i+2} - p_2$
- If $e_2' > 2$, no secret bit is there and sub-block is shifted in data embedding and its recovered cover pixel value is $s_{i+2}=s'_{i+2}-1$

c) [extract the 2nd secret bit p_1 for the ith sub-block as follows]

Calculate the prediction error $e'_1 = s'_{i+1} - s'_i$, then the secret bit extraction and image restoration for the pixel s'_i as follows.

- If $e_1'=0$, no secret bits are there and s'_i becomes unchanged and the cover pixel value is $s_i=s'_i$
- If $1 \le e_1' \le 2$, secret bit is there and the secret bit is $p_1=e_1'-1$ and the recovered cover pixel value is $s_i=s'_i + p_1$
- If $e_1' > 2$, no secret bit is there and sub-block is shifted in data embedding and its recovered cover pixel value is $s_i=s'_i + 1$

So, the required fetched bit sequence is $p_2\, p_1$ (if they exists) and the recovered cover pixels of ith sub-block is ($s_{i,}$, s'_{i+1}, s_{i+2}).

[end of outer if of step 7]

Step 8: Store the recovered cover pixels of ith sub-block into its original positions (c_i, c_{i+1}, c_{i+2}). Set i=i +1 and the above process continued until all sub-blocks are completed.

Step 9: Ultimately we fetched all hided secret bits from the watermarked image and the cover image restored completely.

Numerical examples of data extraction

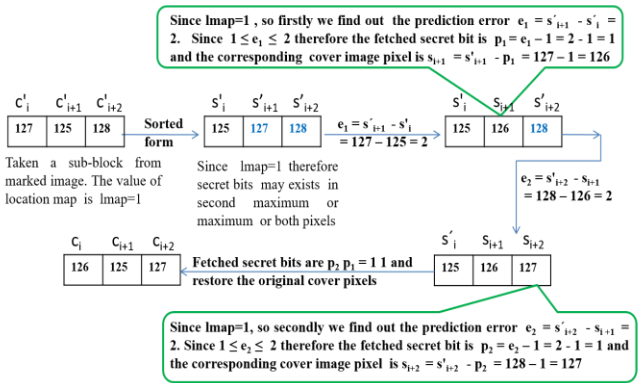

Since lmap=1 , so firstly we find out the prediction error $e_1 = s'_{i+1} - s'_i = 2$. Since $1 \le e_1 \le 2$ therefore the fetched secret bit is $p_1 = e_1 - 1 = 2 - 1 = 1$ and the corresponding cover image pixel is $s_{i+1} = s'_{i+1} - p_1 = 127 - 1 = 126$
c'_i c'_{i+1} c'_{i+2}
127 125 128
Sorted form
s'_i s'_{i+1} s'_{i+2}
125 127 128
$e_1 = s'_{i+1} - s'_i = 127 - 125 = 2$
s'_i s_{i+1} s'_{i+2}
125 126 128
Taken a sub-block from marked image. The value of location map is lmap=1
Since lmap=1 therefore secret bits may exists in second maximum or maximum or both pixels
$e_2 = s'_{i+2} - s_{i+1} = 128 - 126 = 2$
c_i c_{i+1} c_{i+2}
126 125 127
Fetched secret bits are $p_2\, p_1 = 1\,1$ and restore the original cover pixels
s'_i s_{i+1} s_{i+2}
125 126 127
Since lmap=1, so secondly we find out the prediction error $e_2 = s'_{i+2} - s_{i+1} = 2$. Since $1 \le e_2 \le 2$ therefore the fetched secret bit is $p_2 = e_2 - 1 = 2 - 1 = 1$ and the corresponding cover image pixel is $s_{i+2} = s'_{i+2} - p_2 = 128 - 1 = 127$

Fig 4: Example of watermarked sub-blocks whose location map value is 1 and two secret bits are fetched from second maximum and maximum pixels depending on prediction error and restore the cover pixels for that sub-block.

In the above fig 4 a watermarked sub-block is considering for data extraction and image recovery. The pixel values of this sub-block are $c'_i=127$, $c'_{i+1}= 125$, $c'_{i+2}=128$ and location map value is 1 i.e. lmap=1. After sorting these pixels in ascending order we get $s'_i=125$, $s'_{i+1}=127$ and $s'_{i+2}=128$. Since location map value is 1, therefore minimum pixel s'_i does not contain any secret bit but secret bits may exist at maximum (s'_{i+2}) and second maximum (s'_{i+1}) pixels. First we have to fetched secret bit (if exists) from second maximum pixel and recovered the pixel value in its original form. Then by using this recovered value of second maximum pixel we have to fetched the secret bit (if exists) from maximum pixel and recovered the pixel it in its original form. The required bit sequence is fetched bit from maximum pixel followed by fetched bit from second maximum pixel. Now calculate the prediction error $e_1=s'_{i+1} - s'_i =127 - 125 = 2$. Since $e_1=2$ ($1 \le e_1 \le 2$) therefore the fetched secret bit is $p_1=e_1 -1=2 - 1=1$ and the recovered cover pixel is $s_{i+1}=s'_{i+1} - p_1=127 - 1=126$. Again calculate the prediction error $e_2=s'_{i+2} - s_{i+1}=128 - 126=2$. Since $e_2=2$ therefore the fetched secret bit $p_2=e_2 -1=2 - 1=1$ and the recovered cover pixel is $s_{i+2}=s'_{i+2} - p_2=128 - 1=127$. So, the required fetched bit sequence is $p_2p_1=11$ and the cover pixel sub-block as (s'_i, s_{i+1}, s_{i+2}) = (125,126,127) and they are stored into their original positon as (c_i, c_{i+1}, c_{i+2}) = (126,125,127).

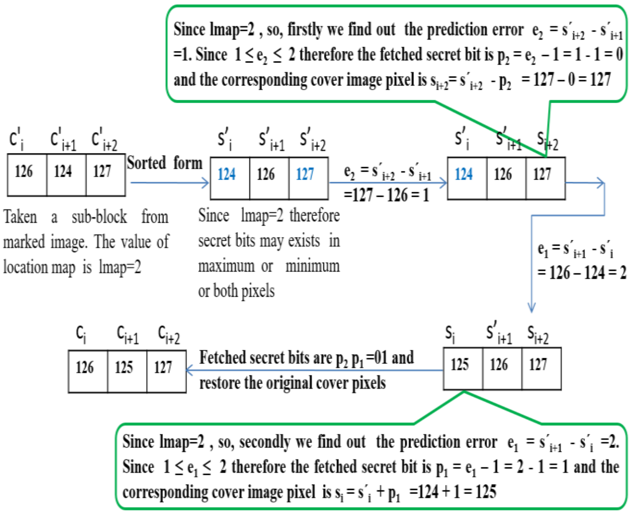

Since lmap=2 , so, firstly we find out the prediction error $e_2 = s'_{i+2} - s'_{i+1} =1$. Since $1 \le e_2 \le 2$ therefore the fetched secret bit is $p_2 = e_2 - 1 = 1 - 1 = 0$ and the corresponding cover image pixel is $s_{i+2}= s'_{i+2} - p_2 = 127 - 0 = 127$
c'_i c'_{i+1} c'_{i+2}
126 124 127
Sorted form
s'_i s'_{i+1} s'_{i+2}
124 126 127
$e_2 = s'_{i+2} - s'_{i+1} =127 - 126 = 1$
s'_i s'_{i+1} s_{i+2}
124 126 127
Taken a sub-block from marked image. The value of location map is lmap=2
Since lmap=2 therefore secret bits may exists in maximum or minimum or both pixels
$e_1 = s'_{i+1} - s'_i = 126 - 124 = 2$
c_i c_{i+1} c_{i+2}
126 125 127
Fetched secret bits are $p_2\, p_1 =01$ and restore the original cover pixels
s_i s'_{i+1} s_{i+2}
125 126 127
Since lmap=2 , so, secondly we find out the prediction error $e_1 = s'_{i+1} - s'_i =2$. Since $1 \le e_1 \le 2$ therefore the fetched secret bit is $p_1 = e_1 - 1 = 2 - 1 = 1$ and the corresponding cover image pixel is $s_i = s'_i + p_1 =124 + 1 = 125$

Fig 5: Example of watermarked sub-blocks whose location map value is 2 and two secret bits are fetched from minimum and maximum pixels depending on prediction error and restore the cover pixels for that sub-block.

Experimental Results and Comparisons

The proposed method is implemented by using MATLAB. The experiment is done by using 512 × 512 grey scale cover images taking from USC-SIPT image database and researchget.net. The secret bits are randomly generated in MATLAB using the function randi([0,1],1,1).

Images are used in the proposed scheme as follows.

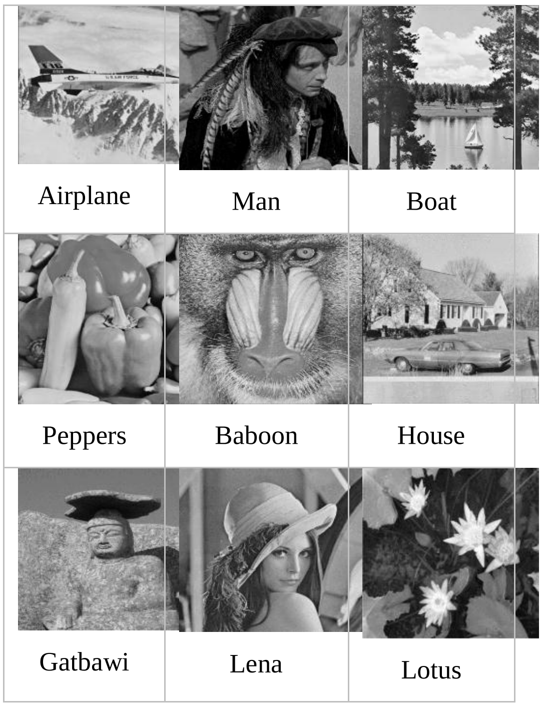


Fig 6: All are gray scale images of size 512 × 512 taken from USC-SIPI image database and researchget.net and from google for experiment.

The embedding rate (ER), Peak signal to noise ratio (PSNR) and universal image quality index Q are generated from the formulas represented in [2] to evaluate the watermarked image quality. In the proposed method each 3 pixels' sub-block could contain maximum two bits. Therefore, the total number of secret bits could be embedded in a cover image of order r × c is (r × c) × $\frac{2}{3}$. So for a 512 × 512 grey scale image, maximum secret bits could be embedded in the proposed scheme is (512 × 512) × $\frac{2}{3}$ = 174762 bits.

In the proposed scheme if we neglect a few sub-blocks that contains auxiliary information and location map, then for each other 3-pixels sub-blocks maximum two pixels will be modified to embed secret bits and the other pixels remains unchanged.

Therefore, in the proposed scheme the embedding capacity is more high than the Jung's method and the Li et al.'s method.

Also theoretically the PSNR of the marked image versus its cover image is at least 10 $\log_{10}\frac{255^2}{MSE} \geq$ 10 $\log_{10}\frac{255^2 \times 3}{2}$ = 49.89dB.

But practically in the proposed scheme since we skipped all the 3-pixels sub-blocks which have no capability to embed at least one secret bit (i.e. we avoid unnecessary expansions of pixels) therefore the average PSNR value is higher than Jung's method and Li et al.'s method. Also the average quality index of different images is higher than the Jung's method and Li et al.'s method.

Experimental results of proposed scheme with existing scheme

The results of proposed scheme compared with Li et al.'s method [1] and Jung's method [2] using nine standards 512×512 sized grey scale images are shown in Table 1 and Table 2 as follows.

Table 1: Comparisons of experimental results (Embedding capacity and embedding rate) of proposed method with other method [1,2]

	Li et al.'s method [1]		Jung's method [2]		Proposed method	
Cover Image	EC	ER	EC	ER	EC	ER
Airplane	12260	0.047	16098	0.061	45417	0.1733
Man	8275	0.032	9533	0.036	23651	0.0902
Boat	7043	0.027	8450	0.032	32415	0.1237
Lena	9799	0.037	11048	0.042	38813	0.1481
Peppers	8918	0.034	10805	0.041	31763	0.1212
Baboon	3839	0.015	4508	0.017	11839	0.0452
House	9910	0.038	13194	0.050	32824	0.1252
Lotus	11423	0.044	12685	0.048	44775	0.1708
Gatbawi	7105	0.027	9361	0.036	31948	0.1219
Average	8730	0.033	10631	0.040	32605	0.1244

Table 2: Comparisons of experimental results (PSNR and Quality Index) of proposed method with other method [1,2]

	Li et al.'s method [1]		Jung's method [2]		Proposed method	
Cover Image	PSNR	Q	PSNR	Q	PSNR	Q
Airplane	55.82	0.9889	56.85	0.9910	57.38	1.0000
Man	45.32	0.9965	45.40	0.9971	58.60	1.0000
Boat	55.91	0.9976	57.05	0.9983	58.30	1.0000
Lena	55.56	0.9955	57.09	0.9968	57.03	0.9999
Peppers	55.66	0.9965	57.30	0.9976	57.37	1.0000
Baboon	56.04	0.9995	56.97	0.9996	61.02	1.0000
House	55.95	0.9852	57.07	0.9887	58.59	1.0000
Lotus	41.98	0.9935	42.58	0.9949	58.03	1.0000
Gatbawi	34.24	0.9888	31.84	0.9909	59.23	1.0000
Average	50.72	0.9936	51.31	0.9950	58.39	0.9999

EC: Embedded capacity, ER: Embedded rate, PSNR: Peak signal to noise ratio, Q: Image quality index

On the experiment, compressed location map bits with auxiliary information are not stored into the watermarked image. On average the embedding capacity of Li et al.'s method is 8730 bits and Jung's method is 10631 bits, but in the proposed method it is 32605. This shows that proposed method hides 21974 bits more on average than Jung's method and 23875 bits more than Li et al.'s method. Also the average value of ER is 0.0841 more than Jung's method and 0.091 more than Li et al.'s method.

On the other hand, average PSNR of Li et al.'s method is 50.72 dB and that of Jung's method is 51.31 dB, but in the proposed method it is 58.39dB. So, the proposed method improves the average PSNR 7.08 dB more than Jung's method and 7.67 dB more than Li et al.'s method. PSNR of all images in the proposed scheme are better than Li et al.'s method and Jung's method except Lena image. The PSNR value of Lena image for proposed scheme is 1.47 dB more than Li et al.'s method but .06 dB less than Jung's method.

From the above table it is also shown that the average quality index Q is better than the Li et al.'s and Jung's method.

Comparison Graphs

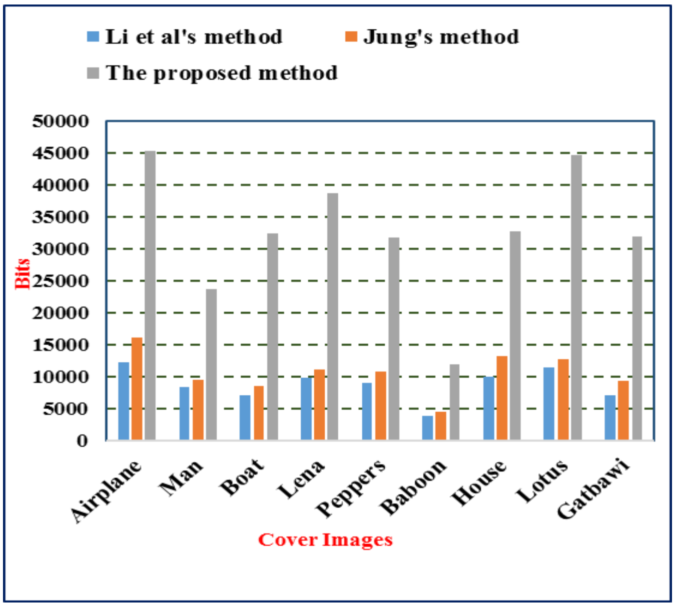


Fig 7: Graph to show a comparison of embedding bits of proposed scheme with existing scheme

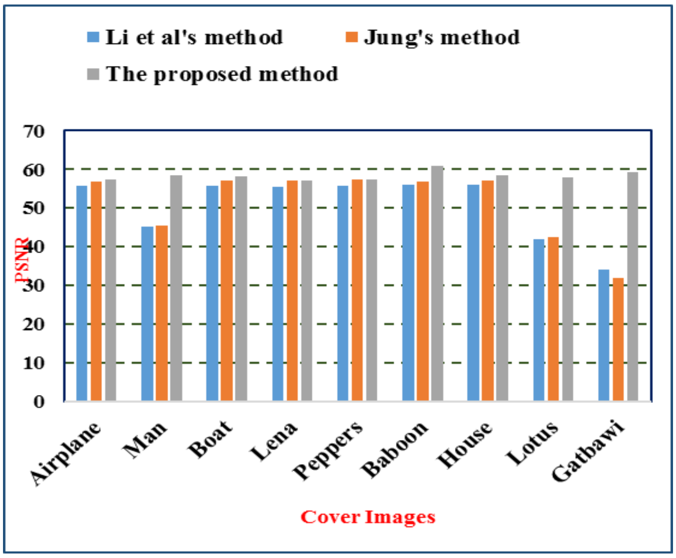


Fig 8: Graph to show a comparison of PSNR proposed scheme with existing schemes

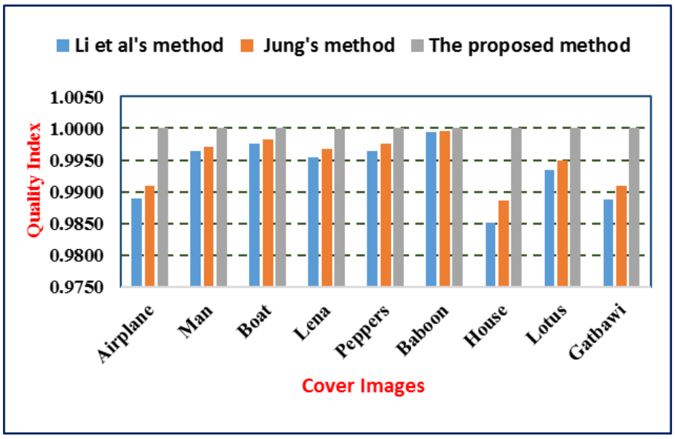


Fig 9: Graph to show a comparison of visual quality index between proposed schemes with existing schemes.

CONCLUSION

The paper presented a new Pixel-value ordering and Prediction error expansion based reversible watermarking scheme to improve Li et al.'s method and Jung's method. In this scheme, the prediction errors on ordered pixels of three pixels sub-blocks calculated and then skip those blocks which are not capable to embed data. Two secret bits are embeded each remaining sub-block depending on Prediction error expansion technique and finally from the watermarked image we can be fetched the secret bits and restore the cover image completely. Experimental results of the proposed scheme show that the embedding capacity, PSNR, and the quality index is better than Li et al.'s method and Jung's method. In our future work, we try to embed three secret bits into three values pixel blocks to improve embedding capacity as well as keeping good visual quality in different domain using PVO based PEE technique.

REFERENCES

[1] Li, X., Li, J., Li, B. and Yang, B. Hsiao, J.Y. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Processing 93*(2013) 198-205.

[2] Jung, K.H. Data Hiding Scheme with Reversibility Based on Neighboring Pixel Value Grouping and Ordering. *Jour of Adv Research in Dynamical & Control Systems*10 (2)(2018)227-233.

[3] J. Fridrich, M. Goljan, R. Du, Lossless data embedding - new paradigm in digital watermarking, EURASIP Journal on Applied Signal Processing 2002 (2) (2002) 185–196.

[4] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized- LSB data embedding, IEEE Transactions on Image Processing 14 (2) (2005) 253–266.

[5] Tai, W.L., Yeh, C.M. and Chang, C.C. Reversible data hiding based on histogram modification of pixel

differences. *IEEE Transactions on Circuits and Systems for Video Technology* **19** (2009) 906-910.

[6] Tian, J. Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology* **13** (2003) 890-896.

[7] Alattar, A.M. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Transactions on Image Processing* **13** (2004) 1147-1156.

[8] Al-Qershi, O.M. and Khoo, B.E. Two-dimensional difference expansion (2D-DE) scheme with a characteristics-based threshold. *Signal Processing* **93** (1) (2013) 154-162.

[9] F. Peng, X. Li, B. Yang, Adaptive reversible data hiding scheme based on integer transform, Signal Processing 92 (1) (2012) 54–62.

[10] Luo, H., Yu, F.X., Chen, H., Huang, Z.L., Li, H. and Wang, P.H. Reversible datahiding based on block median preservation. *Information Sciences* **181** (2011) 308-328.

[11] Zhao, Z., Luo, H., Lu, Z.M. and Pan, J.S. Reversible data hiding based on multilevel histogram modification and sequential recovery. *AEU -International Journal of Electronics and Communications* **65** (2011) 814-826.

[12] Huang, L.C., Tseng, L.Y. and Hwang, M.S. A reversible data hiding method by histogram shifting in high quality medical images. *The Journal of Systems and Software* **86** (2013) 716-727

[13] M. Liu, H.S. Seah, C. Zhu, W. Lin, F. Tian, Reducing location map in prediction-based difference expansion for reversible image data embedding, Signal Processing 92 (3) (2012) 819–828.

[14] Y. Hu, H.K. Lee, J. Li, DE-based reversible data hiding with improved overflow location map, IEEE Transactions on Circuits and Systems for Video Technology 19 (2) (2009) 250–260.

[15] W. Hong, T.S. Chen, C.W. Shiu, Reversible data hiding for high quality images using modification of prediction errors, Journal of Systems and Software 82 (11) (2009) 1833–1842.

[16] C.C. Lin, N.L. Hsueh, A lossless data hiding scheme based on three- pixel block differences, Pattern hiding based on histogram modification of pixel differences, IEEE Transactions on Circuits and Systems for Video Technology 19 (6) (2009) 906–910. Recognition 41 (4) (2008) 1415–1425.

[17] W.L. Tai, C.M. Yeh, C.C. Chang, Reversible data [18] P. Tsai, Y.C. Hu, H.L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, Signal Processing 89 (6) (2009) 1129–1143.