REVERSIBLE DATA HIDING USING IMPROVED INTERPOLATION TECHNIQUE

Devendra Kumar[1], Dr. Krishna Raj[2] (Professor in ECE Department HBTI Kanpur)

[1,2]Department of Electronics & Communication Engineering
Harcourt Butler Technological Institute[1,2]
Kanpur, Uttar Pradesh, India

---***---

Abstract- *Reversible Data hiding is a process of hiding information into a cover image. There are various techniques used for hiding secret data. Audio, video, image, text, and picture can be used as a cover medium in data hiding. In images, Reversible Data Hiding (RDH) is a method which is very helpful in the recovery of original cover after the inserted message is extracted. This vital method is broadly used as a part of medical imagery, law forensics and military imagery, where no distortion of the original cover is permitted. Due to being first introduced, RDH has pulled in impressive research interest. In the last decade, reversible data hiding (RDH) has received much attention from the information hiding community. RDH is a special type of data hiding and data extraction technique and it ensures a lossless recovery of secret and cover data. Specifically, by RDH, not only the embedded message, the cover content can also be exactly extracted from the marked data.*

Index Terms-Reversible Data hiding, Medical imagery, Law forensics, Military imagery, Data extraction, Secret data, Cover data.

1. INTRODUCTION

Data hiding is a technique that conceals secret data into a carrier to convey messages. Digital images are one of the media that is suitable to convey messages due to several reasons. Firstly, digital images are often transmitted over the Internet which would arouse little suspicion. Secondly, the high correlation between pixels provides rich space for data embedding. In images, Reversible data hiding (RDH) is a method which is very helpful in the recovery of original cover after the inserted message is extracted. This vital method is broadly used as a part of medical imagery, law forensics and military imagery, where no distortion of the original cover is permitted. Due to being first introduced, RDH has pulled in impressive research interest.

2. LITERATURE SURVEY

In the practical point of view, numerous RDH strategies have developed in past few years. Fridrich *et al.* [1] developed a general system for RDH. With the initial extraction of the compressible characteristics of original cover and after this compressing them losslessly, we can spare space for embedding auxiliary data. On the basis of difference expansion (DE) [2] a more popular process works in which the difference of all pixel groups is extended, e.g., multiplied by two, and hence the LSBs of the distinction are all-zero and can be utilized for the purpose of embedding messages. Histogram shift (HS) [3] is another procedure in which space is conserved for embedding of data by means of adjusting the histogram bins of gray values. The state-of-art strategies [2]–[7] normally consolidated DE or HS to residuals of the image, e.g., the anticipated faults, to accomplish improved performance.

A few efforts on RDH in encrypted images have been accomplished. In [4], Zhang makes the division of encrypted image into many blocks. With flipping 3 LSBs of the half of pixels in each block, room can be cleared for the inserted bit. The information extraction and image recovery continue by discovering which section has been flipped in one block. This procedure can be acknowledged by means of spatial relationship in decrypted image. Hong *et al.* [5] improved Zhang's process at the decoder side with the further exploitation of the spatial correlation utilizing a dissimilar estimation equation and side match method to accomplish quite lower fault rate. These two techniques specified above depend on spatial correlation of original picture to get data. That is, the encrypted image ought to be decoded first before information extraction.

Zhang [6] suggests a novel method for separable reversible information concealing .In this, the content owner initially encrypts the original uncompressed image by making use of an encryption key to generate an encrypted image. Now, the least significant bits (LSB) of the encrypted image is compressed by the data-hider by means of a data-hiding key to develop a sparse space to put in the extra data. At the end of the receiver, the data inserted in the generated space can be effortlessly regained from the encrypted image comprising extra data as per the data-hiding key.

In [7], Lixin Luo, Zhenyong Chen, Ming Chen, Xiao Zeng, and Zhang Xiong 2010 presented the paper "Reversible image watermarking using interpolation technique." In this paper they proposed a reversible watermarking scheme based on additive interpolation-error expansion, which features very low distortion and relatively large capacity. Different from previous watermarking schemes, we utilize an interpolation technique to generate residual values named interpolation-errors and expand them by addition to embed bits. The strategy is efficient since interpolation-errors are good at de-correlating pixels and additive expansion is free of expensive overhead information. Essentially, the data embedding approach of the proposed reversible watermarking scheme, namely additive interpolation-error expansion, is a kind of Difference Expansion.

Reversible data hiding techniques can be generally classified into two frameworks

1) Vacate room after encryption
2) Reserve room before encryption

In the first system after encryption (VRAE), owner of the content first encrypts the genuine image by making use of a typical cipher with the help of an encryption key and then vacate room for data hiding. In the second framework, reserve room before encryption (RRBE), the content owner initially reserve sufficient space on original image and after this transforms the image into its encrypted format with the help of the encryption key. The process of data extraction and that of image recovery are similar to that of Framework VRAE. Clearly, general RDH algorithms are the ideal operator for the purpose of reserving room prior to the encryption and can be effortlessly used to Framework RRBE in order to get improved results. This process can acquire genuine reversibility, that is, data extraction and image recovery does not contain any error.

In this paper, we propose a reversible watermarking scheme based on improved additive interpolation-error expansion, which features very low distortion and relatively large capacity. Different from previous watermarking schemes, we utilize an improved interpolation technique to generate residual values named interpolation-errors and expand them by addition to embed bits. The strategy is efficient since interpolation-errors are good at de-correlating pixels and additive expansion is free of expensive overhead information.

Additive Interpolation-Error Expansion

Essentially, the data embedding approach of the proposed reversible watermarking methodology, termed as additive interpolation-error expansion, is a type of DE. Although, it is not like the most DE approaches in two vital aspects [13]:

1) It makes use of interpolation-error, rather than using interpixel difference or prediction-error, to embed data.

2) It makes the expansion of the difference, which is interpolation-error, at this point by means of addition rather than bit-shifting.

Initially, interpolation values of pixels are estimated with the help of interpolation technique, which works by guessing a pixel value from its surrounding pixels. Then interpolation-errors are obtained through

$$e = x - x' \tag{1}$$

In the above equation, parameter x' represents the interpolation values of pixels. Suppose *LM* and *RM* represent the corresponding values of the two peak points of interpolation-errors histogram and be formulated as

$$\begin{cases} LM = \arg \max_{e \in E} hist(e) \\ RM = \arg \max_{e \in E - \{LM\}} hist(e) \end{cases} \tag{2}$$

Where, *hist*(*e*) is the number of occurrence when the interpolation-error is equal to *e* and *the parameter E* represents the interpolation-error set. Without loss of generality, assume *LM*<*RM*. Then, we classify the interpolation errors into 2 sections:

1) Right interpolation-errors (*RE*): interpolation-error *e* satisfies $e \geq RM$.

2) Left interpolation-errors (*LE*): interpolation-error *e* satisfies $e \leq LM$.

We can formulate the additive interpolation-error expansion as given and further can find the expanded interpolation error expansion using equation 3 given below.

$$e' = \begin{cases} e + sign(e)\times b, & e = LM \;\; or \;\; RM \\ e + sign(e)\times 1, & e \in (LM, LM) \cup (RM, RN) \\ e, & otherwise \end{cases} \quad (3)$$

Where e' is the expanded interpolation-error, *b* is the bit to be embedded, and *sign*(e) is a sign function defined as

$$sign(e) = \begin{cases} 1, & e \in RE \\ -1, & e \in LE \end{cases} \quad (4)$$

In (5), the parameters *LN* and *RN* are defined as

$$\begin{cases} LN = \arg \max\limits_{e \in LE} hist(e) \\ RN = \arg \max\limits_{e \in RE} hist(e) \end{cases} \quad (5)$$

Usually, *LM* is a very small integer and in most cases 0, while *LN* is a smaller integer that with no interpolation-error satisfying e= *LN.*

Similarly, in most cases, *RM* is equal to 1 and *RN* is a larger integer with no interpolation-error satisfying e= *RN* .After expansion of interpolation-errors, the watermarked pixels x'' become

$$x'' = x' + e' \quad (6)$$

During the extracting process, with the same interpolation algorithm, we can obtain the same interpolation values x' and the corresponding interpolation-errors via

$$e' = x'' - x' \quad (7)$$

Once the same *LM*, *LN*, *RM*, and *RN* are known, embedded data can be extracted through

$$b = \begin{cases} 0, & e' = LM \;\; or \;\; RM \\ 1, & e' = LM - 1 \; or \;\; RM + 1 \end{cases} \quad (8)$$

Then the inverse function of additive interpolation-error expansion is applied to recover the original interpolation-errors

$$e = \begin{cases} e' - sign(e')\times b, & e' \in [LM-1, LM] \cup [RM, RM+1] \\ e' - sign(e')\times 1, & e' \in [LN, LM-1) \cup (RM+1, RN] \\ e, & otherwise \end{cases} \quad (9)$$

Finally, we can restore the original pixels through

$$x = x' + e$$

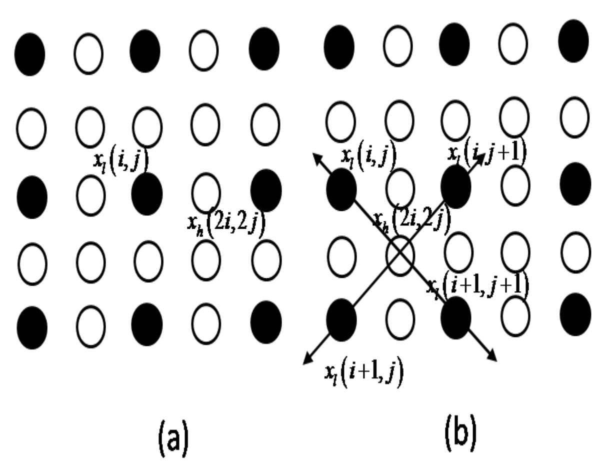
So this method offers a feasible image interpolation algorithm to obtain interpolation values and interpolation-errors and we can find the original pixels using these values [9].

Interpolation-Error

Different from the recent conventional schemes, we exploit interpolation error, the difference between pixel value and its interpolation value, to embed data. There are several reasons that lift interpolation error to be a better alternative to inter-pixel difference or prediction error. The reasons for using interpolation-error instead of inter-pixel difference in the proposed scheme can be summarized as two points.

The first one is that interpolation-error requires no blocking which can significantly reduce the amount of differences and in turn lessen the potential embedding capacity. As in this scheme, we almost get all pixels as candidates to embed messages except some negligible marginal pixels, which promise a larger embedding capacity.

The second one is that interpolation-error exploits the correlation between pixels more significantly. Fundamentally, the feasibility of reversible image watermarking is due to high inter-pixel redundancy or inter-pixel correlation existing in practical images. To maximize the capacity of image watermarking, we should exploit the correlation of pixels to the greatest extent. The proposed scheme utilizes the full-enclosing pixels to interpolate the target pixel, so the interpolation-error tends to be smaller, which means that we can obtain a higher capacity.



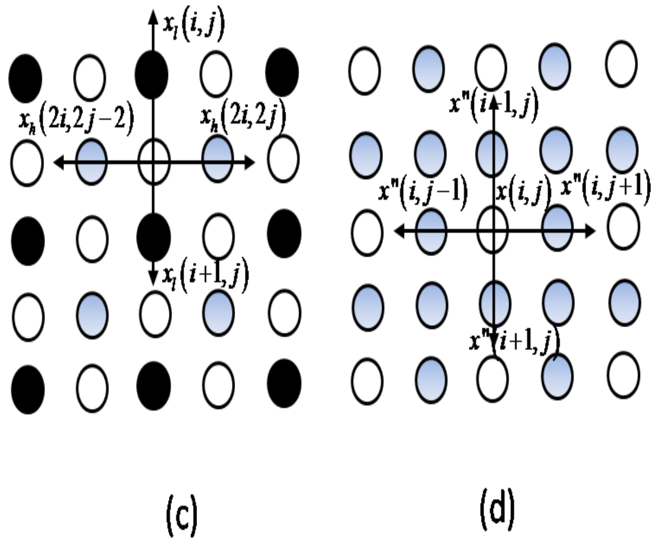


Figure 1: (a) Formation of a low-resolution image x_l from the high-resolution image x_h ; (b), (c) interpolation of residual samples of high-resolution; (d) the interpolation of the sample pixels [9].

Let us take a low-resolution image x_l is down-sampled straight-way from an associated high-resolution x_h by $x_l = x_h(2i-1, 2j-1)$, $1 \le i \le N$, $1 \le j \le M$. With the reference to Fig. 3.2(a), the black dots stands for the pixels of x_l while the missing pixels of x_h is represented by the white dots. The goal of interpolation is to make the estimation of the missing pixels in high-resolution x_h, whose size measure is 2*N*×2*M*, from the pixels in low-resolution x_l, whose size is *N*×*M*,.

The major problem of interpolation is how to infer and make use of the correlation between the missing pixels and the neighboring pixels. With the interpolation algorithm under discussion, we partition the neighboring pixels of all missing pixels into two directional subsets that are orthogonal to each other. For each subset, a directional interpolation is made, and after this, fusion of the two interpolated values is done with an optimal pair of weights to calculate x_h. We make the reconstruction of the high-resolution x_h in two step. In the first step, the missing pixels $x_h(2i, 2j)$ at the center locations clouded by four low-resolution pixels are interpolated. In the second step, the other missing pixels $x_h(2i-1, 2j)$ and $x_h(2i, 2j-1)$ are interpolated with the help of the already recovered pixels $x_h(2i, 2j)$.

Let

$$\hat{x}_h = w_\theta \hat{x}_\theta + w_{\theta+90^0} \hat{x}_{\theta+90^0} \tag{10}$$

Where, $w_\theta + w_{\theta+90^0} = 1$

The weights w_θ and $w_{\theta+90^0}$ are determined to minimize the mean square-error (MSE) of :

$$\hat{x}_h : \{w_\theta, w_{\theta+90^0}\} = \arg\min E[(\hat{x}_h - x_h)^2]_{w_\theta + w_{\theta+90^0}=1} \tag{11}$$

Then, we discuss the details of the first step using above analogy. Referring to Fig. 1(b), we can interpolate the missing high-resolution pixel $x_h(2i, 2j)$ along two orthogonal directions: 45^0 diagonal and 135^0 diagonal. Denoted by $x'_{45}(2i, 2j)$ and $x'_{135}(2i, 2j)$, the two directional interpolation results are computed as

$$\begin{cases} \{x'_{45} = ((x_l(i, j+1) + (x_l(i+1, j))/2 \\ \{x'_{135} = ((x_l(i, j) + (x_l(i+1, j+1))/2 \end{cases} \tag{12}$$

Here, we take e_{45} and e_{135} to represent the interpolation errors in the corresponding direction. Let

$$\begin{cases} \{e_{45}(2i, 2j) = x'_{45}(2i, 2j) - x_h(2i, 2j) \\ \{e_{135}(2i, 2j) = x'_{135}(2i, 2j) - x_h(2i, 2j) \end{cases} \tag{13}$$

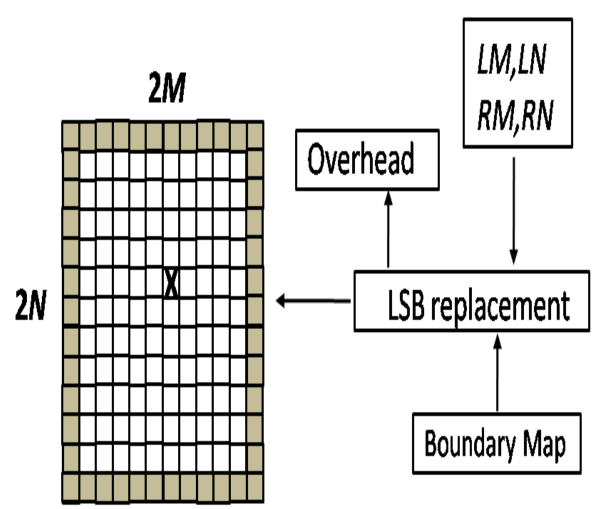


Figure 2: LSB replacement of the overhead information [9].

Instead of computing the linear minimum mean square-error estimation (LMMSE) estimate of x_h, we select an optimal pair of weights to make x_h' a good estimate of x_h. The strategy of weighted average leads to significant reduction in complexity. Let

$$\begin{cases} x_h' = w_{45} \bullet x_{45}' + w_{135} \bullet x_{135}' \\ w_{45} + w_{135} = 1 \end{cases} \tag{14}$$

The weights w_{45} and w_{135} are determined to minimize the mean square-error of x_h

$$\{w_{45}, w_{135}\} = \arg \min_{w_{45}+w_{135}=1} E\left[\left(x_h' - x_h\right)^2\right] \tag{15}$$

According to abundant experiments, the correlation coefficient between e_{45} and e_{135}, which influences the correlation between x_{45}' and x_{135}' from (12), hardly changes the PSNR value and visual quality of the interpolation image. So we can show the weights are

$$w_{45} = \frac{\sigma(e_{135})}{\sigma(e_{45}) + \sigma(e_{135})}, \quad w_{135} = 1 - w_{45} \tag{16}$$

where $\sigma(e_{45})$, $\sigma(e_{135})$ are the variance estimations of e_{45} and e_{135}, respectively. They are related to the mean value of $x_h(2i,2j)$ that we will discuss. From (13), we can see intuitively how the weighting method works. For instance, for an edge in or near the 45^0 diagonal direction, $\sigma(e_{135})$ is higher than $\sigma(e_{45})$ so that w_{135} will be less than w_{45}; consequently, x_{135}' has less influence on x_h' than x_{45}', and vice versa.

Referring to Figure 1(b), the mean value of $x_h(2i,2j)$, denoted by *u*, is estimated by the available low-resolution pixels around $x_h(2i,2j)$ To balance the complexity of computation and the consistency of pixels, we compute as

$$u = \frac{\left(x_1(i, j+1) + x_1(i+1, j)\right)}{4} + \frac{\left(x_1(i, j) + x_1(i+1, j+1)\right)}{4}, \tag{17}$$

Then we return to the computation of $\sigma(e_{45})$ and $\sigma(e_{135})$, which are the variance estimations of interpolation errors in the corresponding directions. They are computed as

$$\begin{cases} \sigma(e_{45}) = \frac{1}{3}\sum_{k=1}^{3}\left(S_{45}(k) - u\right)^2 \\ \sigma(e_{135}) = \frac{1}{3}\sum_{k=1}^{3}\left(S_{135}(k) - u\right)^2 \end{cases} \tag{18}$$

where the two sets are

$$\begin{cases} S_{45} = \left\{x_l(i, j+1), x_{45}', x_l(i+1, j)\right\} \\ S_{135} = \left\{x_l(i, j), x_{135}', x_l(i+1, j+1)\right\} \end{cases} \tag{19}$$

We can use (14)–(19) to obtain the estimations of the missing high resolution pixels $x_h(2i,2j)$ and finish the first step.

After the missing high-resolution pixels $x_h(2i,2j)$ are estimated, the residual pixels $x_h(2i-1,2j)$ and $x_h(2i,2j-1)$ can be estimated similarly. Referring to Figure 1(c), the black dots represent the low-resolution pixels, the gray dots represent the estimated pixels in the first step, and the white dots represent the pixels that are to be interpolated.

3. PROPOSED METHOD OF IMPROVED INTERPOLATION-ERROR ESTIMATION

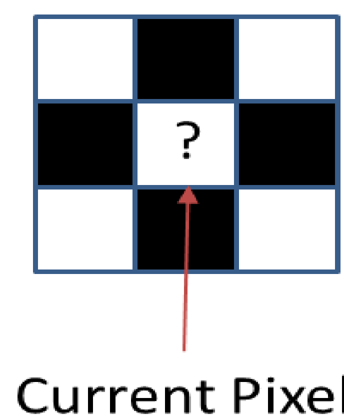


Figure 3: Estimation of center pixel through surrounding pixels using proposed method

Considering a 3×3 part of an image, whose pixel value varies from C (1, 1) to C (3, 3), then the mean value will be obtained using the pixel values of the pixels at 0^0 and 90^0 respectively. Let $x = C(2,2)$ is to be estimated, with estimated value $\hat{x}$.

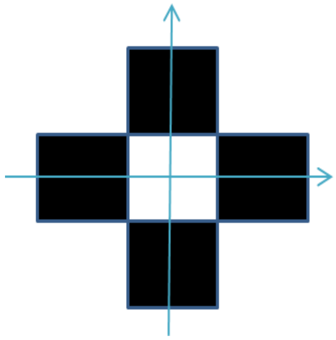


Figure 4: Estimation of center pixel through surrounding pixels at angles 0^0 and 90^0

The mean value of surrounding pixel of *x*, at 0 and 90 degree is calculated as

$$u = \frac{1}{6}\left[C(1,2)+C(2,1)+C(2,3)+C(3,2)+2C(2,2)\right] \quad (20)$$

In the mean central pixel is also included once in 0^0 and other in 90^0 directions.

Defining average value in 0 and 90 degree respectively as

$$x_0 = \frac{1}{3}\left[C(2,1)+C(2,2)+C(2,3)\right] \quad \text{and}$$

$$x_{90} = \frac{1}{3}\left[C(1,2)+C(2,2)+C(3,2)\right]. \quad (21)$$

Considering,

$$S_0 = [C(2,1),(2,3),x_0] \text{ And } S_{90} = [C(1,2),(3,2),x_{90}] \quad (22)$$

The variance in 0^0 and 90^0 direction is

$$\begin{cases} \sigma(e_0) = \frac{1}{3}\sum_{k=1}^{3}\left(S_0(k)-u\right)^2 \\ \sigma(e_{90}) = \frac{1}{3}\sum_{k=1}^{3}\left(S_{90}(k)-u\right)^2 \end{cases} \quad (23)$$

The weight in 0^0 and 90^0 direction is given by

$$w_0 = \frac{\sigma_{90}}{\sigma_0 + \sigma_{90}} \text{ and } w_{90} = \frac{\sigma_0}{\sigma_0 + \sigma_{90}} \quad (24)$$

The estimated value of the pixel is obtained by

$$\hat{x} = w_0 x_0 + w_{90} x_{90} \quad (25)$$

So now the improved estimated interpolation error will be

$$e = x - \hat{x} \quad (26)$$

Generally, for any pixel value $B_{i,j}$ assessing error is estimated by means of $e_{i,j} = B_{i,j} - B'_{i,j}$ and afterward certain data can be inserted into the estimating error arrangement with histogram shift, Then, further calculation of the estimating errors of black pixels is performed by surrounding white pixels in which modification may have been made. After this, other estimating error sequence is developed which can provide accommodation to the messages too. Moreover, we can likewise execute multilayer embedding plan by considering the changed B as "original" one when required. Finally, the output is that in order to exploit all pixels of B, 2 estimating error sequences are developed for embedding messages in all single-layer embedding methods.

Error sequence – data can be added using Histogram shift

Divide the Histogram in two parts L and R and search for highest point in each part LM and RM. In general LM = -1 and RM = 0.

Search for zero in each part and define as LN, RN respectively.

In order to insert messages into positions with an estimating error which is equivalent to RM, move each error value between *RM*+1 and *RN*-1 with one step to right, then, the bit 0 with RM and the bit 1 with *RM*+1 could be represented by us. The process of embedding in the left part is same with the exception of that the direction of shifting is left, and the shift is acknowledged by removing 1 from the relating pixel values.

Generating the Marked Decrypted Image:

In order to develop the marked decrypted image X'' which is buildup of A'' and B'', the content owner of the content should follow the under given steps.

Step 1. With the help of the encryption key, the owner makes the decryption of the image putting aside the LSB-planes of A_E. We can calculate the decrypted version of E' having the embedded data by

$$X''_{i,j}(k) = E'_{i,j}(k) \oplus r_{i,j}(k) \quad (27)$$

And

$$X''_{i,j} = \sum_{k=0}^{7} X''_{i,j}(k) \times 2^k, \quad (28)$$

Above the parameters $E^{'}_{i,j}(k)$ and $X^{''}_{i,j}(k)$ represents the binary bits of $E^{'}_{i,j}(k)$ and $X^{''}_{i,j}(k)$, acquired through (27) respectively.

Step 2. Extract *SR* and *ER* in marginal zone of $B^{''}$. With the rearrangement of $A^{''}$ and $B^{''}$ to its original form, the plain image comprising embedded data is acquired. As we can notice that the marked decrypted image $X^{''}$ is indistinguishable to modified X with the exception of LSB-planes of A. Meanwhile, it maintains perceptual transparency compared with original image C.

More particularly, the distortion is proposed by means of two different manners: the embedding procedure by changing the LSB-planes of A and self-reversible embedding procedure by inserting LSB planes of A into B. The initial step distortion is quite managed through exploiting the LSB-planes of just A and the next part can give advantage from phenomenal execution of current RDH procedures.

Data Extraction and Image Restoration

After the generation of the marked decrypted image, the owner of the content can moreover extract the data and makes the recovery of original image. The procedure is basically like that of typical RDH techniques.

The below describes the outlines the particular steps:

Step 1. Make the Recording and decryption of the LSB-planes of $A^{''}$ as per the data hiding key; extract the data till reach the end label.

Step 2. Extract *LN*, *RN*, *LM*, *RM*, *LP*, *RP*, R_b, *x* and boundary map from the LSB of marginal area of $B^{''}$. After this, scan $B^{''}$ to undertake the below mentioned steps.

Step 3. If R_b is equal to 0, which means no black pixels participate in embedding process, go to Step 5.

Step 4. Calculate estimating errors $e'_{i,j}$ of the black pixels $B''_{i,j}$. If $B''_{i,j}$ belongs to [1, 254], recover the estimating error and original pixel value in a reverse order and extract embedded bits when $e'_{i,j}$ is equal to *LN*, *LM* (or *LP*), *RM* (*RP*) and *RN*. Else, if $B''_{i,j} \in \{0,255\}$, refer to the corresponding bit *b* in boundary map. If *b*=0, skip this one, else operate like $B''_{i,j} \in [1,254]$. Repeat this step until the part of payload R_b is extracted. If extracted bits are LSBs of pixels in marginal area, restore them immediately.

Step 5. Make the calculation of estimating errors $e'_{i,j}$ of the white pixels $B''_{i,j}$, and extract embedded bits and carry out the recovery of white pixels in the same way with Step 4. In the event of the extracted bits are LSBs of pixels in marginal zone, restore them at once.

Step 6. Carry on with Step 2 to Step 5 x-1 rounds on $B^{''}$ and merge each extracted bit to develop LSB-planes of A. This should be continued till we recover B perfectly.

Step 7. Marked LSB-planes of $A^{''}$ should be replaced with its original bits extracted from $B^{''}$ in order to get original cover image C.

In this paper, a novel method is proposed for RDH in encrypted images, for which we don't "vacate room after encryption" as carried out in [16]–[18], but "reserve room before encryption". In the proposed technique, as a first step, we empty out room by means of embedding LSBs of some pixels into other pixels with a traditional RDH method and then encrypt the image, so the positions of these LSBs in the encrypted image can be used to embed data. Not only does the proposed method separate data extraction from image decryption but also achieves excellent performance in two different prospects: Real reversibility is realized, that is, data extraction and image recovery are free of any error.

For given embedding rates, the PSNRs of decrypted image containing the embedded data are significantly improved; and for the acceptable PSNR, the range of embedding rates is greatly enlarged as it has been discussed that the smaller interpolation error will be advantageous as the embedding area increases on the cover image the results in better PSNR and increased embedding capacity.

4. EXPERIMENTAL RESULTS AND COMPARISON

In the MATLAB simulation six public available standard images 'Airplane', 'Baboon', 'Barbara', 'Boat', 'Lenna' and 'Peppers' in pgm format is considered. Each image is of size 512×512×8 and of 256KB. The objective criteria PSNR and SSIM are employed to evaluate the quality of marked decrypted image. the results are drawn for proposed method at 0.05bpp and the original image, watermarked image and SSIM trends are shown. Results show as the embedding rate (bits per pixel) is increased, the PSNR and SSIM values reduce. In the table 1, and 2 results for RDH process is shown in which PSNR(dB), SSIM values are detailed. Again here as the BPP increases, the SSIM values are detailed. Again here as the BPP increases, the SSIM and PSNR value decreases.

Figure 5: Airplane (a) Original Image (b) Watermarked Image (c) SSIM

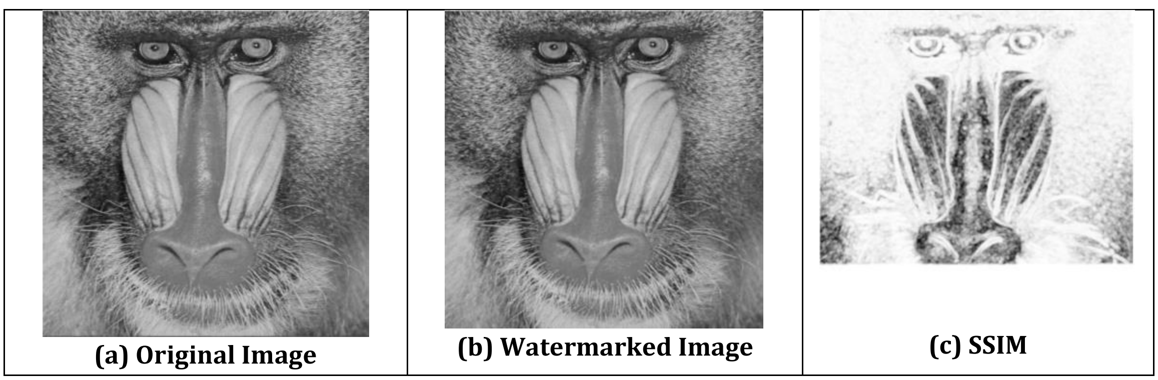


Figure 6: Baboon (a) Original Image (b) Watermarked Image (c) SSIM

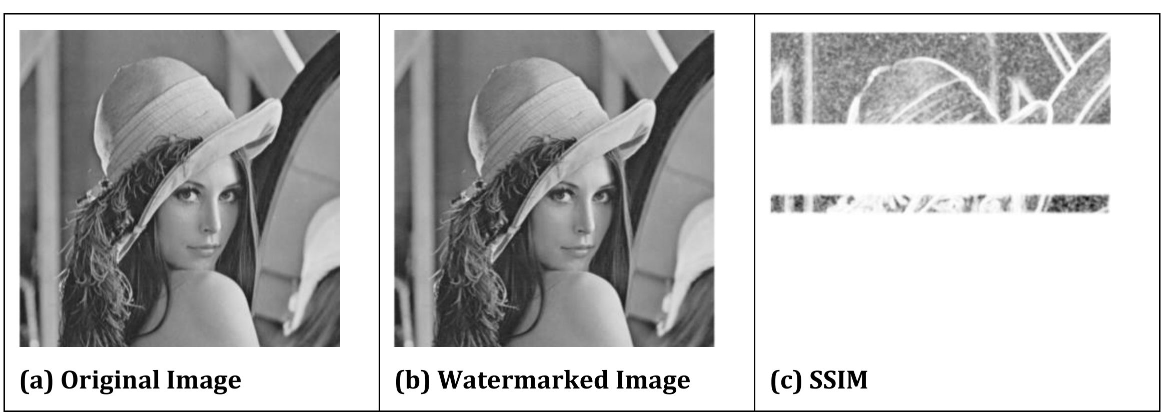


Figure 8: Lenna (a) Original Image (b) Watermarked Image (c) SSIM

Table 1: PSNR for RDH process for different images (Proposed method)

	Bit Per Pixels					
	0.05	0.1	0.2	0.3	0.4	0.5
Image	**PSNR (dB)**					
Airplane	**58.56**	**55.86**	52.57	49.32	48.06	46.18
Baboon	**53.14**	**50.00**	45.06	41.40	38.50	35.71
Barbara	**57.52**	**53.93**	50.81	47.32	44.48	**42.30**
Boat	58.77	**55.35**	51.56	48.45	46.01	**44.06**
Lenna	57.82	54.42	51.41	48.32	45.70	**44.85**
Peppers	56.18	53.35	50.20	46.62	43.91	41.39

Table 2: SSIM for RDH process for different images (Proposed method)

	Bit Per Pixels					
	0.05	0.1	0.2	0.3	0.4	0.5
Image	**SSIM**					
Airplane	**0.9992**	**0.9984**	**0.9971**	0.9946	0.9927	0.9876
Baboon	**0.9995**	**0.9987**	0.9956	0.9894	0.9809	0.9654
Barbara	**0.9994**	**0.9987**	**0.9972**	0.9942	0.9827	**0.9836**
Boat	0.9992	**0.9984**	0.9967	0.9944	0.9889	**0.9826**
Lenna	0.9990	0.9981	**0.9963**	0.9931	0.9871	**0.9806**
Peppers	0.9989	0.9979	0.9955	0.9898	0.9816	0.9691

Table 3: Boundary map for RDH process for different images (Proposed)

	Bit Per Pixels					
	0.05	0.1	0.2	0.3	0.4	0.5
Image	**Length of boundary map**					
Airplane	0	0	0	0	0	0
Baboon	0	0	0	0	1	5
Barbara	0	0	0	0	0	0
Boat	0	0	0	0	0	0
Lenna	0	0	0	0	0	0
Peppers	9	68	132	192	390	831

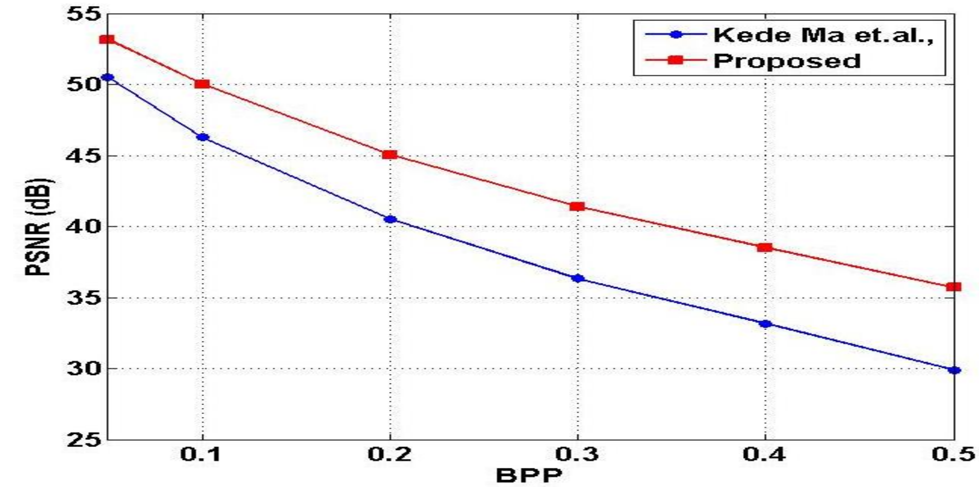


Chart 1: Comparison of PSNR vs BPP for baboon image.

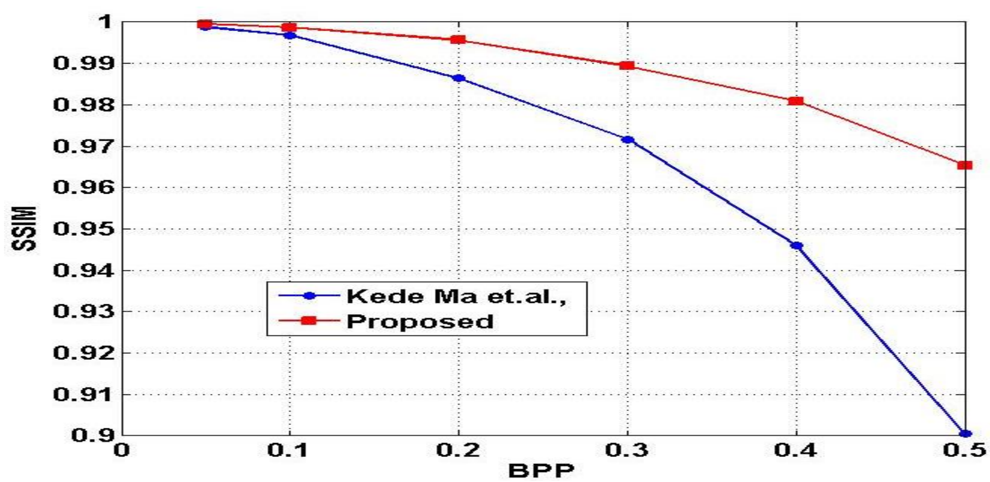


Chart 2: Comparison of SSIM vs BPP for baboon image.

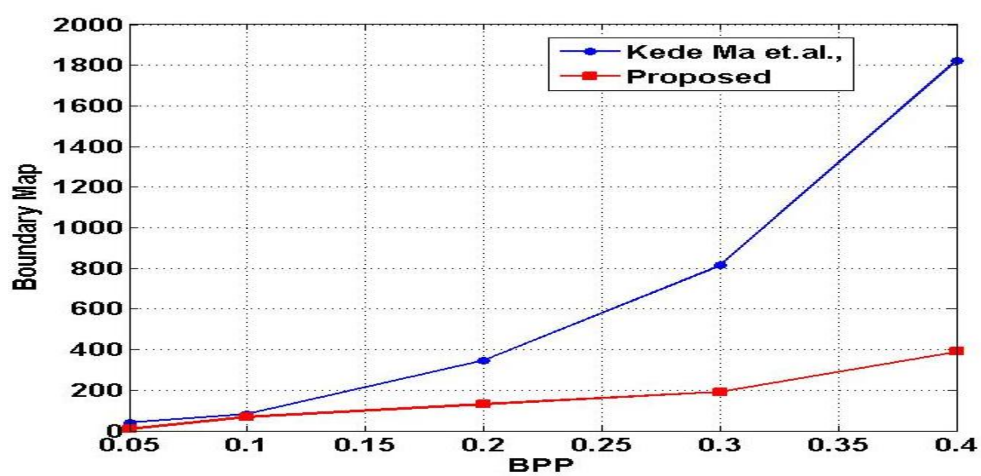


Chart 3: Comparison of Boundary Map vs BPP for baboon image.

On the basis of results obtained, it can be found that that the chosen interpolation methods have deep impact on the over-all performance of the RDH process. And the interpolation values should be predicted in such a way that the error value attains -1, 0 and 1 as error, which achieves higher embedding capacity. Thus inclusion of actual value of pixel in the mean value calculation which is to be estimated, improves results significantly.

Discussion on Boundary Map

Boundary map in this paper, is used for distinguishing between natural and pseudo boundary pixels and its size is critical to practical applicability of proposed approach. Table 3 shows the boundary map size of six standard images. In most cases, no boundary map is needed. Even for Peppers image, the largest size is 831 bits (with a large embedding rate 0.5 bpp by adopting embedding scheme 4 rounds) and the marginal area (512*4*4 bits) is large enough to accommodate it.

5. CONCLUSION

Reversible data hiding in encrypted images is a new topic drawing attention because of the privacy-preserving requirements from cloud data management. Previous methods implement RDH in encrypted images by vacating room after encryption, as opposed to which we proposed by reserving room before encryption. Thus the data hider can benefit from the extra space emptied out in previous stage to make data hiding process effortless. The proposed method can take advantage of all traditional RDH techniques for plain images and achieve excellent performance without loss of perfect secrecy. Furthermore, this novel method can achieve real reversibility, separate data extraction and greatly improvement on the quality of marked decrypted images.

6. FUTURE WORKS

1. In the future more better interpolation technique can be explored.

2. A better image partition method can be searched.

3. M-SSIM can be used as image quality measure index.

4. A relation between PSNR and SSIM can be modeled, as some relation exists between them.

7. REFERENCES

[1] J. Fridrich and M. Goljan, "Lossless data embedding for all image formats," in Proc. SPIE Proc. Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, San Jose, CA, USA,Jan. 2002, vol. 4675, pp. 572–583

[2] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[3] W. Zhang, B. Chen, and N. Yu, "Improving various reversible data hiding schemes via optimal codes for binary covers," IEEE Trans. Image Process., vol. 21, no. 6, pp. 2991–3003, Jun. 2012.

[4] L. Luo et al., "Reversible image watermarking using interpolation technique," IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193,Mar. 2010.

[5] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.Q. Shi, "Reversible watermarking algorithm using sorting and prediction," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 7, pp. 989– 999, Jul. 2009.

[6] Xinpeng Zhang, Member, IEEE "Reversible Data Hiding With Optimal Value Transfer" IEEE Transactions On Multimedia, VOL. 15, NO. 2, FEBRUARY 2013.

[7] X. Zhang, "Separable reversible data hiding in encrypted image" IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012.

[8] Che-Wei Lee and Wen-Hsiang Tsail "A Lossless Data Hiding Method by Histogram Shifting Based on an Adaptive Block Division Scheme" 2010 River Publishers.

[9] Xinpeng Zhang, "Reversible Data Hiding in Encrypted Image", in IEEE signal processing letters, VOL. 18, NO. 4, pp 255-258, April 2011

[10] K. Ma, Weiming Zhang, X. Zhao, "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption", Information Forensics and Security, IEEE Transactions on Vol.8, No.3, 2013.

[11] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted grayscale images, IEEE Trans. Image Process" vol.19, no.4, pp. 1097–1102, Apr. 2010.

[12] Xiang Wang , "Efficient generalized integer transform for reversible watermarking, "IEEE Signal Processing Letters, Vol. 17, No. 6, June 2010.

[13] Mehmet Utku Celik ,"Lossless generalized LSB data embedding", IEEE Transactions On Image Processing, Vol.14, No. 2, February 2005.

[14] D.M. Thodi and J.J .Rodriguez, "Expansion embedding techniques for reversible watermarking" IEEE Trans. Image Process., vol.16, no.3, pp. 721–730, Mar. 2000.

[15] Vidyadhar Gupta, and Krishna Raj. "An Efficient Modified Lifting Based 2-D Discrete Wavelet Transform Architecture." Recent Advances in Information Technology (RAIT), 2012 1st International Conference on " Recent Advances in Information Technology (RAIT-2012), Date of Conference:17-19 Dec. 2012 Page(s):1 - 6 Print ISBN:978-1-4673-1047-5 INSPEC Accession Number:13488575 Conference Location :Allahabad DOI:10.1109/ICPCES.2012.6508051. Dec. 2012.

[16] Krishna Raj, and Vedvrat. "A Study of VLSI Architectures for 2-D Discrete Wavelet Transform." IEEE International Conference on Power, Control and Embedded Systems (ICPCES), 2010 International Conference Page(s):1 - 3 Print ISBN:978-1-4244-8543-7 INSPEC Accession Number:11763651 DOI:10.1109/ICPCES.2010.5698715. Dec. 2010