International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(1): 52-60
The Society of Digital Information and Wireless Communications, 2011 (ISSN 2220-9085)

Reversible Data Hiding Based on Statistical Correlation of Blocked Sub-Sampled Image

Mona Nafari
Department of Electrical and Electronic Engineering of Razi University of Kermanshah, Kermanshah, Iran (e-mail: m.nafari@razi.ac.ir)

Mansour Nejati Jahromi
Department of Electrical and Electronic Engineering of Azad University South Branch, and Aeronautical College, Tehran, Iran(e-mail: Nejati@aut.ac.ir)

GholamHoseinSheisi
Department of Electrical and Electronic Engineering of Razi University of Kermanshah, Kermanshah, Iran (e-mail: sheisi@razi.ac.ir)

Abstract—**In this paper, a reversible data hiding scheme has been proposed which is based on correlation of subsample images. The proposed method modifies the blocks of sub-sampled image to prepare vacant positions for data embedding. The PSNR of the stego image produced by the proposed method is guaranteed to be above 47.5 dB, while the embedding capacity is at least, almost 6.5 times higher than that of the Kim et al. techniques with the same PSNR. This technique has the capability to control the capacity-PSNR. Experimental results support that the proposed method exploits the correlation of blocked sub-sampled image, outperforms the prior works in terms of larger capacity and stego image quality both. On various test images, we demonstrate the validity of our proposed method by comparing to other existing reversible data hiding algorithms.**

Keywords-component; Reversible data hiding, Subsampling, Correlation, Blocking, threshold

I. Introduction

During transmission, if the digital host media shows any distortion, the international unauthorized user may doubt the media which contains secret data. Therefore, the imperceptibility of embedded data is one of the most important concerns of data hiding techniques, along with its capacity [1]. A data hiding technique must extract secret data without losing or changing any bit. Therefore, the required robustness against intentional or unintentional processing or the ability to prevent bit errors during transmission or restoration, are emphasized in data hiding [2].

In image data hiding, the most important problems are the degradation of the image and permanent distortion, which sometimes leads to incorrect restoration. Therefore, for some applications, it causes critical problems such as medical or legal images [3].

Reversible data hiding is a technique that can restore the original image after data extraction. Therefore reversible data hiding leads to lower payload and higher computation. There are some reversible data hiding techniques which have been proposed and developed since 2002 [4] [5] [6]. [7] [8], [9], [10] [11].

One of the categories of data hiding is based on histogram shifting techniques. In these techniques, data can be embedded by shifting histogram bins. This technique, first proposed by Ni et al[12]. Generally, in this technique, the higher the peak of image histogram, the more the embed ding capacity is. However, payload capacity is low and it is limited by the distribution of image histogram bins.

In this paper a new reversible data hiding technique has been proposed that is based on individually the correlation of sub sample images. Detailed advantageous of sub-sampling and correlation computation will be discussed in section *C*.

The remainder of this paper is organized as follows:. Data hiding using sub-sampled images will be reviewed in Section II. Then, in Section III, our new method of data hiding and data extraction based on neighboring blocks correlation will be introduced in detail, followed by our experimental results which presented in Section IV. Finally, we conclude in Section V.

II. Data Hiding Using Subsampled Images.

Kyungsu kima et al in 2009 [13] proposed a reversible data hiding method which modifies the difference histogram between sub-sampled images. In their scheme, embedding procedure is executed after sub-sampling. Sub-sampling is the process of selecting units from an image. Suppose an image of size $M \times N$ pixels denoted by $I(x,y)$ where $x = 0,1,\ldots,M-1$ and $y = 0,1,\ldots,N-1$.

Two sampling factors Δu and Δv, set the desired sub-sampling intervals in row and column direction respectively. As illustrated in Fig.1, The 2-D image is sampled in uniform intervals. Each sub-sampled image S_k of size $N/\Delta u \times M/\Delta v$ is obtained by equation (1) as follows:

$$S_k(i,j) = I\left(i.\Delta u + floor\left(\frac{k-1}{\Delta u}\right), j\Delta v + ((k-1)\bmod \Delta v)\right) \quad (1)$$

Where $i = 0,1,\ldots, M/\Delta u - 1$, $j = 0,1,\ldots, N/\Delta v - 1$ and $k = 1,\ldots,\Delta u \times \Delta v$. If $M/\Delta u$ or $N/\Delta v$ is not an integer values, the size of all sub-sampled images is modified by flooring. The message is embedded by modifying the difference histogram between sub-sampled versions of an image. In this scheme, after generating sub-sampled versions of an original image *I* by two sampling factors Δu and Δv a reference sub-sampled image S_{ref} is determined to maximize the spatial correlation between the sub-sampled images which is defined by equation (2):

$$S_{ref} = \left(Round\left(\frac{\Delta u}{2} - 1 \right) \right) \times \Delta v + Round\left(\frac{\Delta v}{2} \right) \quad (2)$$

Then the difference images between the reference S_{ref} and other destination sub-sampled images denoted S_{Des} are created. By preparing empty bins in each histogram H of the difference images, messages are embedded. Thus in Kyungsu scheme secret bits are embedded in difference histogram of sub-sampled images [13][14].

The data hiding described above, is different in data hiding from the scheme which is explained in the next section. In fact the concept of correlation based data hiding [15] is a statistical approach which is combined with sub-sampling procedure for higher capacity and security. Furthermore, in contrast to Kim et al method, the proposed scheme doesn't select any reference for subsample images and equally consider all of them.

III. THE PROPOSED METHOD.

Correlation quantifies the strength of a linear relationship between two variables [16]. When there is no correlation between two quantities, then there is no tendency for the values of one quantity to increase or decrease with the values of the second one. Correlation based data hiding is a technique for embedding data into the blocked image based on the correlation value of each block with neighboring blocks. The distortion resulting from correlation based embedding depends on the average correlation of all blocks with correspondent neighboring blocks. The higher the correlation, the less the distortion is. In this section, a reversible data hiding technique based on correlation is proposed which is performed on sub sampled images.

One of the advantages of this approach is that the sub-sampling procedure increases the number of embeddable blocks in comparison to the previous paper [15]. Subsequently increases the payload capacity. Due to the computation of neighboring blocks correlation, the distortion tends to become less. Therefore capacity – PSNR control can be achieved.

A. subsampling

As it is noted in section II , sampling is the process of selecting pixels of an image that leads to a determined number of smaller images depends on sampling factors Δu and Δv . By using equation (1) and defining Δu and Δv , all sub-sampled images is defined. As illustrated in Fig.2, the 2-D image is sampled at different sizes of intervals.

The basic idea of sub-sampling is to increase the security of embedded data by utilizing a permutation key on sub-sampled images after embedding in addition to increase the payload capacity.

B. Correlation calculation based on proposed method

In the proposed scheme each sub-sampled image $S_k(i, j)$, $k = 1,...,\Delta u \times \Delta v$ is divided into non-overlapping 1×2 blocks. In dividing process, block size plays an important role in the number of bits to be embedded. Since one bit is embedded in each block which can be 2×2 , 3×3 , 4×4 , …. Thus it is more advantageous to apply small sized blocks.

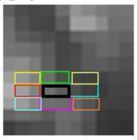

Fig **1**:1×2image blocking

Each block has 8 adjacent blocks (Fig.2) which each one is used in calculating the mean correlation of central blocks. If the size of a sub-sampled image is $M/_{\Delta u} \times N/_{\Delta v}$ pixels with $M/_{\Delta u}$ rows and $N/_{\Delta v}$ columns, the number of blocks is calculated by equation (3):

$$n = \left(\frac{M \times N/_{\Delta u \times \Delta v} - \left(M/_{\Delta u} \times 2 + N/_{\Delta v} \times 2 - 4 \right)}{2} \right) \quad (3)$$

Where $M/_{\Delta u} \times 2 + N/_{\Delta v} \times 2 - 4$ is the number of pixels at the border of a sub-sampled image. These pixels are used only in calculating the correlation matrix not in the procedure of data embedding; because these pixels decrease the quality of stego image at borders.

If A and B are two matrixes, the correlation of these two matrices is calculated by equation (4):

$$c = \frac{\sum_i \sum_j \left(A_{ij} - \overline{A} \right)\left(B_{ij} - \overline{B} \right)}{\sqrt{\left(\sum_i \sum_j \left(A_{ij} - A \right)^2 \right)\left(\sum_i \sum_j \left(B_{ij} - \overline{B} \right)^2 \right)}} \quad (4)$$

where A is the central block where secret bit have to be embedded within. There are n central blocks in the sub-sampled image. B is the adjacent block of each 1×2 block A . $\overline{A}$ and $\overline{B}$ are the mean values of A and B respectively. In other words for each correlation calculation between block A and one of neighboring block B , equation (5) is calculated. i and j illustrate the pixel located at i th row and j th column of A and B . For a sub-sampled image which has n blocks, there is $8\times n$ correlation calculation. Then only the mean values are saved for each block. The mean value of these 8 values constitute a matrix C which its size is the same as sub-sampled images and each element is the correlation value of each 1×2 block with the 8 neighboring block.

In this study, correlation is a statistical concept which is based on image blocks. The correlation is used for determining if the block is embeddable or not embeddable.

In the next section, the overall procedure of data hiding will be discussed which is based on sub-sampling.

C. data embedding based on subsampling

Data hiding process involves sampling the image before data embedding. Fig.4 (a, b) shows an overall illustration of data embedding and data extraction procedure. For each subsample image, the embedding algorithm is executed which is explained in section 1.

1) embedding algorithm for each subsample image.

If the procedure of correlation computation is performed for each block, a correlation matrix will be generated as C . By

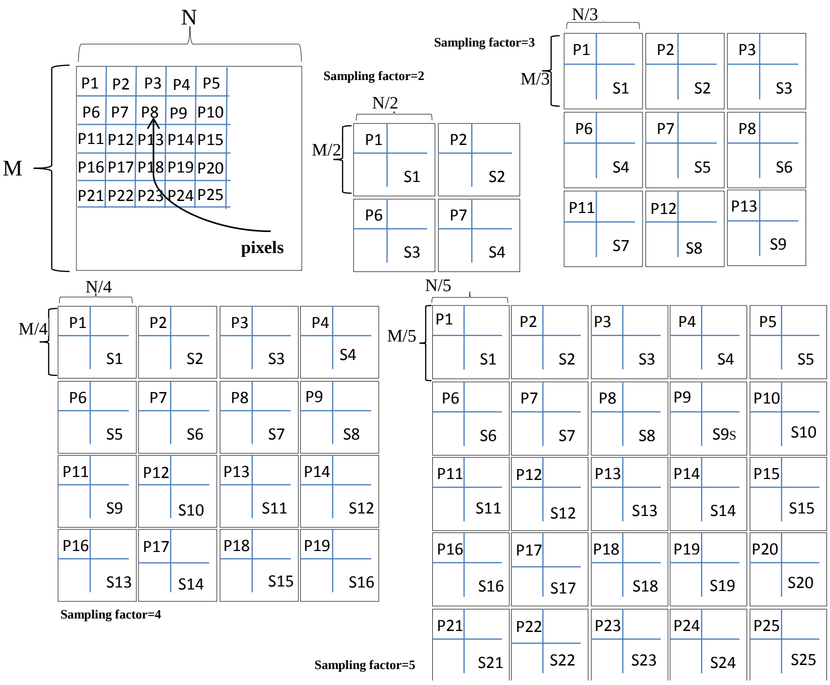

Fig 2: Sub-sampling at different sampling factors :(a) original (b): $\Delta u=2,\Delta v=2$, (c) $\Delta u=3,\Delta v=3$,(d) $\Delta u=4,\Delta v=4$, (e) $\Delta u=5,\Delta v=5$

normalizing the correlation matrix elements, correlation values are in the range of 0 to 1. In this section data hiding method is proposed that is based on threshold selection and correlation of adjacent blocks.

The following steps and their explanations illustrate the procedure of data embedding algorithm in each subsample.

Step1: Divide the image in to 1×2 blocks.

Step 2: Calculate correlation between each block and its 8 adjacent blocks, then calculate the mean value of these 8 values for each block and put these values in a matrix which its size is $M/\Delta u \times N/\Delta v$ to have a matrix with the same size as the sub-sampled original image. The correlation matrix is concatenated to zero row and zero column before and after the first and the last row and column respectively. So the correlation matrix size is $M/\Delta u \times N/\Delta v$.

Step 3: Specify a threshold T for data embedding. This threshold determines the embeddable blocks of original image which its correlation value in the correlation matrix is greater than threshold or equal. It is clear that the less the threshold, the more the blocks have the condition of data embedding.

Step 4: Sum the values of 2 pixels in each block as equation (5). The correlation value of each 1×2 block have to be greater than or equal to the defined threshold. Then calculate the residue of this value in dividing by 2.

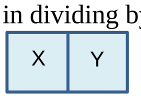

Fig 1:block pixels

$$sumblock = X + Y \tag{5}$$

Step 5: Determine if the sum-block has an even or odd value. If the value is odd, the bit to be embedded is 0, otherwise the bit to be embedded is 1. Then according to equation (6) secret bit has to be embedded to the sum-block. Finally add this value to the pixel. In this equation $datastream$ is the secret bit to be embedded according to equation (7). 1 is used for the data extraction process.

$$X(i)=X(i)+1+(sumblock+datastreem(i))\bmod 2 \tag{6}$$

$$datastreem=\{b_1,b_2,...,b_n\} \tag{7}$$

where X illustrates the left pixel value of each 1×2 block. (Fig.3)

It is clear that there is four conditions in embedding process:

a) If the sum-block is odd, its residue in dividing by 2 is 1. Then if the bit to be embedded is 0, 2 is added to the pixel value X.

b) If the sum-block is even, its residue in dividing by 2 is 0. Then if the bit to be embedded is 1, 2 is added to the pixel value X.

c) If the sum-block is odd, its residue in dividing by 2 is 1. Then if the bit to be embedded is 1, 1 is added to the pixel value X and no bit is embedded. So the block is skipped.

d) If the sum-block is even, its residue in dividing by 2 is 0. Then if the bit to be embedded is 0, 1 is added to the pixel value X and no bit is embedded. So the block is skipped.

In other words, in the case of embedding secret bit, 2 is added to the value of X. But if no bit is embedded, then l is added to the value X. Fig 5(a) illustrates the embedding algorithm.

After embedding algorithm, correlation matrix is calculated like the process mentioned in step 2.An additional bit has to be concatenated to each elements of correlation matrix at the left pixel of each block. This bit shows whether the sum-block is odd or even, if the sum-block is odd, the bit is 1, else if the sum-block is even, the bit is 0. This bit is a virtual one and only works to recover the original image from stego image. i indicates the i th block that satisfies the embedding condition, thus it is index of X and $datastreem$ both.

If the mean square error (MSE) of correlation matrix elements before and after embedding process is determined, the error of correlation matrix for each sub-sampled image is calculated by using equation (8) where m and n are the m throw and n th column of correlation matrix.

$$error = \frac{\sqrt{\left(\sum_{m}\sum_{n}\left(C_{befor-embedding}(m,n) - C_{after-embedding}(m,n)\right)\right)^2}}{M/_{\Delta u} \times N/_{\Delta v}} \quad (8)$$

Where $C_{befor-embedding}$ and $C_{after-embedding}$ in equation (8) are the correlation values at mth row and nth column of correlation matrix, before and after embedding respectively. $C_{befor-embedding}$ is the same as C but from now on it is called $C_{befor-embedding}$ to be comparable to $C_{after-embedding}$. Where $M/_{\Delta u} \times N/_{\Delta v}$ illustrates the number of all pixels of each sub-sampled image or correlation matrix.

Clearly in each iteration of this algorithm, the correlation matrix before embedding is different from the correlation after embedding. This difference or error which here after is called (MSE) is important in selection of the best threshold. By decreasing this error, minimum degradation is achieved. Because error decreasing leads to enhancement of the quality of the image. But the capacity has to be high enough. So this target will be met by a trade off between error and payload capacity. By executing all these process in each threshold from 0 to 1 with the step of 0.l, the best threshold can be determined. Because as it is mentioned, when the threshold is decreased, the payload capacity increases, on the other hand, the quality of the stego image decreases.

By Calculating error values from lower threshold to higher ones, the decreasing pattern of this error can be recognized. This pattern is depending on, image type and payload capacity, but undoubtedly it has a descending gradient. Then sub-sampled images are combined to constitute the stego image.

2) Extraction process:

For the extraction process, correlation matrix C , threshold and sub-sampling factor $(\Delta u, \Delta v)$ is required as the overhead information. Sub-sampling factor is applied to generate the sub-sampled stego images from stego image in addition to generating sub-sampled correlation matrix $C_k(i,j)$ for each sub-sampled original images. This is done by performing sub-sampling procedure like the process described in section 4.1. Secret bit is extracted by the following steps from each sub-sampled stego image:

Step1: Compute the sum of two pixels of each block in each sub-sampled stego image according to the same block in $C_k(i,j)$ which its correlation is higher or equal to the threshold defined at the beginning of embedding.

Step2: extract secret bits of each sub-sampled stego image as $extracted(i)$ which is derived by equation (9) to be:

$$extracted(i) = 1 - sumblock(i) \bmod 2 \quad (9)$$

Step3: Recover sub-sampled original image blocks as follows:

-If $extracted(i) = 0$ and $C_k(i,j)$ of $block(i)$ is more or equal to "1" $(C(i) \geq 1)$, subtract 2 from the upper left pixel of each block: $X = X - 2$

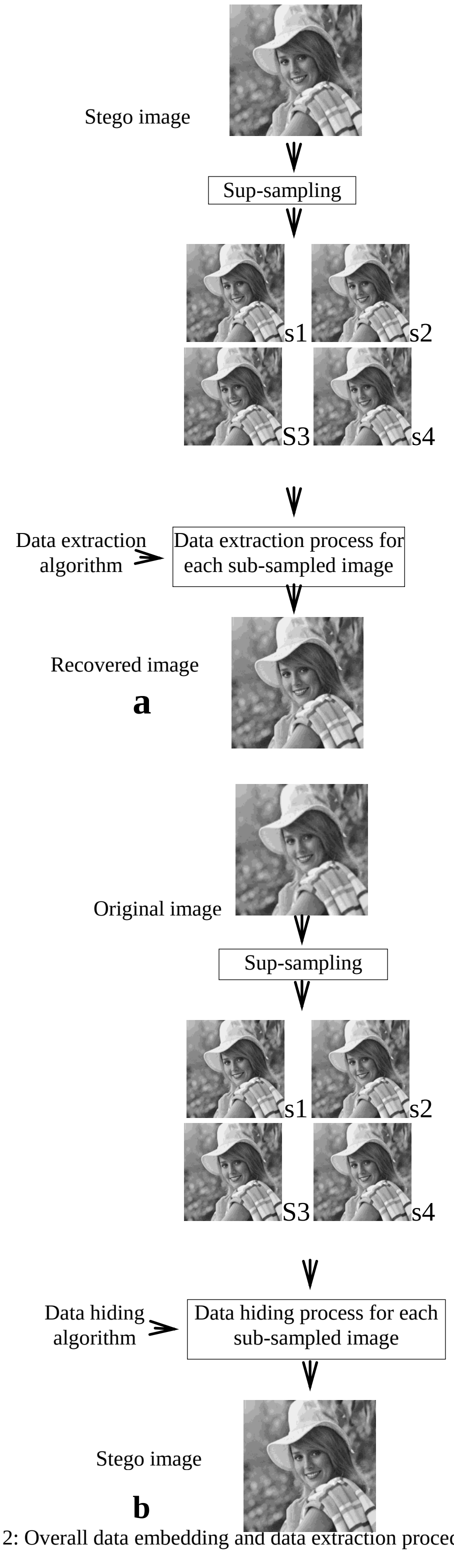

Fig 2: Overall data embedding and data extraction procedure. (a) data embedding and (b) data extraction procedure

- If $extracted(i) = 1$ and $C_k(i,j)$ of $block(i)$ is located between "1" and "T" $(T \leq C(i) \leq 1)$, subtract 2 from the upper left pixel of each block: $X = X - 2$

-If $extracted(i) = 1$ and $C_k(i,j)$ of $block(i)$ is more or equal to 1" $(C(i) \geq 1)$, subtract 1 from the upper left pixel of each block: $X = X - 1$

- If $extracted(i) = 0$ and $C_k(i,j)$ of $block(i)$ is located between "1" and "T", $(T \leq C(i) \leq 1)$, subtract 1 from the upper left pixel of each block: $X = X - 1$

Step4: Rearrange each restored sub-sampled image. The block diagram of extraction algorithm is illustrated in Fig.4 (b).

As an example, suppose an image with 12×12 pixels $I_{original-image}$ and secret data as follows:

And the secret data are:

$$data = \begin{Bmatrix} \langle 1,1,0,1,1,0,0,1 \rangle, \langle 1,1,0,1,1,0,1,0 \rangle \\ \langle 0,0,1,1,1,1,0,1 \rangle, \langle 1,1,1,1,0,1,0,1 \rangle \end{Bmatrix}$$

If the image is sub-sampled with the sampling factor of $\Delta u = 2, \Delta v = 2$, four sub-sampled images is generated which are $I_1(i,j)$, $I_2(i,j)$, $I_3(i,j)$ and $I_4(i,j)$ and divided into 1×2 blocks:

$$I1 = \begin{bmatrix} 114 & 128 & 134 & 142 & 151 & 156 \\ 116 & 126 & 140 & 142 & 148 & 157 \\ 117 & 126 & 133 & 143 & 150 & 151 \\ 114 & 127 & 139 & 141 & 147 & 151 \\ 117 & 125 & 136 & 145 & 148 & 155 \\ 120 & 127 & 135 & 143 & 148 & 152 \end{bmatrix}, I2 = \begin{bmatrix} 121 & 134 & 136 & 146 & 157 & 154 \\ 120 & 133 & 139 & 148 & 152 & 153 \\ 121 & 130 & 138 & 146 & 152 & 153 \\ 121 & 133 & 141 & 144 & 148 & 156 \\ 121 & 132 & 141 & 147 & 151 & 152 \\ 122 & 131 & 141 & 143 & 153 & 151 \end{bmatrix}$$

$$I3 = \begin{bmatrix} 111 & 127 & 138 & 147 & 152 & 151 \\ 114 & 130 & 135 & 143 & 148 & 152 \\ 116 & 125 & 136 & 142 & 149 & 149 \\ 116 & 127 & 138 & 142 & 147 & 155 \\ 117 & 126 & 136 & 145 & 148 & 152 \\ 116 & 128 & 132 & 140 & 151 & 153 \end{bmatrix}, I4 = \begin{bmatrix} 118 & 133 & 140 & 152 & 153 & 150 \\ 121 & 137 & 138 & 145 & 152 & 154 \\ 121 & 129 & 140 & 145 & 150 & 152 \\ 121 & 133 & 140 & 146 & 150 & 157 \\ 121 & 132 & 142 & 146 & 151 & 148 \\ 120 & 129 & 138 & 145 & 151 & 154 \end{bmatrix}$$

The blocked correlation matrix of each sub-sampled image are:

$$C1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.875 & 0 & 0.87 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, C2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 0.87 & 0 & 0.875 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, C4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 0.875 & 0 & 0.875 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The sum-block of each block of I_1 are: 266,259,266,261,290,293,288,293.

By considering *T*=0.1 for the correlation matrix of I_1 which is C_1, all of the blocks can be used for embedding. According to being even or odd, the secret data are embedded in the left pixel of each block. Therefore I_{stego1} is obtained by the equation (6) as follows:

$$X(1) = X(1) + 1 + (sumblock(1) + datastream(1)) \bmod 2$$
$$= 126 + 1 + (266 + 1) \bmod 2 = 128$$

$$X(2) = X(2) + 1 + (sumblock(2) + datastream(2)) \bmod 2$$
$$= 126 + 1 + (259 + 1) \bmod 2 = 127$$

$$I_{original-image} = \begin{bmatrix} 114 & 121 & 128 & 134 & 134 & 136 & 142 & 146 & 151 & 157 & 156 & 154 \\ 111 & 118 & 127 & 133 & 138 & 140 & 147 & 152 & 152 & 153 & 151 & 150 \\ 116 & 120 & 126 & 133 & 140 & 139 & 142 & 148 & 148 & 152 & 157 & 153 \\ 114 & 121 & 130 & 137 & 135 & 138 & 143 & 145 & 148 & 152 & 152 & 154 \\ 117 & 121 & 126 & 130 & 133 & 138 & 143 & 146 & 150 & 152 & 151 & 153 \\ 116 & 121 & 125 & 129 & 136 & 140 & 142 & 145 & 149 & 150 & 149 & 152 \\ 114 & 121 & 127 & 133 & 139 & 141 & 141 & 144 & 147 & 148 & 151 & 156 \\ 116 & 121 & 127 & 133 & 138 & 140 & 142 & 146 & 147 & 150 & 155 & 157 \\ 117 & 121 & 125 & 132 & 136 & 141 & 145 & 147 & 148 & 151 & 155 & 152 \\ 117 & 121 & 126 & 132 & 136 & 142 & 145 & 146 & 148 & 151 & 152 & 148 \\ 120 & 122 & 127 & 131 & 135 & 141 & 143 & 143 & 148 & 153 & 152 & 151 \\ 116 & 120 & 128 & 129 & 132 & 138 & 140 & 145 & 151 & 151 & 153 & 154 \end{bmatrix}$$

$$X(3) = X(3) + 1 + (sumblock(3) + datastream(3)) \bmod 2$$
$$= 127 + 1 + (266 + 0) \bmod 2 = 128$$

$$X(4) = X(4) + 1 + (sumblock(4) + datastream(4)) \bmod 2$$
$$= 125 + 1 + (261 + 1) \bmod 2 = 126$$

$$X(5) = X(5) + 1 + (sumblock(5) + datastream(5)) \bmod 2$$
$$= 142 + 1 + (290 + 1) \bmod 2 = 144$$

$$X(6) = X(6) + 1 + (sumblock(6) + datastream(6)) \bmod 2$$
$$= 143 + 1 + (293 + 0) \bmod 2 = 145$$

$$X(7) = X(7) + 1 + (sumblock(7) + datastream(7)) \bmod 2$$
$$= 141 + 1 + (288 + 0) \bmod 2 = 142$$

$$X(8) = X(8) + 1 + (sumblock(8) + datastream(8)) \bmod 2$$
$$= 145 + 1 + (293 + 1) \bmod 2 = 146$$

For simplicity the permutation process is not mentioned in this example. If this procedure is done on all other sub-sampled images, the following sub-sampled stego images are obtained:

$$Istego1 = \begin{bmatrix} 114 & 128 & 134 & 142 & 151 & 156 \\ 116 & 128 & 140 & 144 & 148 & 157 \\ 117 & 127 & 133 & 145 & 150 & 151 \\ 114 & 128 & 139 & 142 & 147 & 151 \\ 117 & 126 & 136 & 145 & 148 & 155 \\ 120 & 127 & 135 & 143 & 148 & 152 \end{bmatrix}, Istego2 = \begin{bmatrix} 121 & 134 & 136 & 146 & 157 & 154 \\ 120 & 134 & 139 & 149 & 152 & 153 \\ 121 & 129 & 138 & 145 & 152 & 153 \\ 121 & 132 & 141 & 143 & 148 & 156 \\ 121 & 132 & 141 & 148 & 151 & 152 \\ 122 & 131 & 141 & 143 & 153 & 151 \end{bmatrix}$$

$$Istego3 = \begin{bmatrix} 111 & 127 & 138 & 147 & 152 & 151 \\ 114 & 130 & 135 & 143 & 148 & 152 \\ 116 & 125 & 136 & 142 & 149 & 149 \\ 116 & 127 & 138 & 142 & 147 & 155 \\ 117 & 125 & 136 & 145 & 148 & 152 \\ 116 & 128 & 132 & 140 & 151 & 153 \end{bmatrix}, Istego4 = \begin{bmatrix} 111 & 127 & 138 & 147 & 152 & 151 \\ 114 & 130 & 135 & 143 & 148 & 152 \\ 116 & 125 & 136 & 142 & 149 & 149 \\ 116 & 127 & 138 & 142 & 147 & 155 \\ 117 & 125 & 136 & 145 & 148 & 152 \\ 116 & 128 & 132 & 140 & 151 & 153 \end{bmatrix}$$

Thus the stego image is obtained:

$$Istego = \begin{bmatrix} 114 & 121 & 128 & 134 & 134 & 136 & 142 & 146 & 151 & 157 & 156 & 154 \\ 111 & 118 & 127 & 133 & 138 & 140 & 147 & 152 & 152 & 153 & 151 & 150 \\ 116 & 120 & 128 & 135 & 140 & 139 & 144 & 150 & 148 & 152 & 157 & 153 \\ 114 & 121 & 131 & 139 & 135 & 138 & 145 & 147 & 148 & 152 & 152 & 154 \\ 117 & 121 & 127 & 131 & 133 & 138 & 145 & 147 & 150 & 152 & 151 & 153 \\ 116 & 121 & 126 & 131 & 136 & 140 & 144 & 146 & 149 & 150 & 149 & 152 \\ 114 & 121 & 128 & 134 & 139 & 141 & 142 & 145 & 147 & 148 & 151 & 156 \\ 116 & 121 & 128 & 134 & 138 & 140 & 143 & 148 & 147 & 150 & 155 & 157 \\ 117 & 121 & 126 & 134 & 136 & 141 & 146 & 149 & 148 & 151 & 155 & 152 \\ 117 & 121 & 127 & 133 & 136 & 142 & 147 & 147 & 148 & 151 & 152 & 148 \\ 120 & 122 & 127 & 131 & 135 & 141 & 143 & 143 & 148 & 153 & 152 & 151 \\ 116 & 120 & 128 & 129 & 132 & 138 & 140 & 145 & 151 & 151 & 153 & 154 \end{bmatrix}$$

For the procedure of data extraction, by considering correlation matrix of each subsample image and the defined threshold, in addition to using equation.8, the extracted secret data is obtained:

$$restored-data1 = \{1,1,0,1,1,0,0,1\}$$
$$restored-data2 = \{1,1,0,1,1,0,1,0\}$$
$$restored-data3 = \{0,0,1,1,1,1,0,1\}$$
$$restored-data4 = \{0,0,1,1,1,1,0,1\}$$

For the restoration of original image, by considering condition of step3 and 4, after computing sum-blocks of each sub-sampled stego images, recovered image is obtained:

IV. EXPERIMENTAL RESULTS

.In all experiments, six images of 512×512 pixels with 256 gray scales from USC-SIPI [17] is tested as depicted in Fig.5: a) Man b) Elaine c) Pentagon d) Lena e) F-16 f) Zelda, of the same size 512×512.

All the concerned data hiding algorithms are run by the operating system windows XP and the program developing environment is MATLAB 7.8.The proposed method has been assessed by five aspects: sampling factor, threshold, correlation error, PSNR and pay load capacity.

The PSNR value can be computed by the following equation:

Fig.5. Test images: a) Man b) Elaine c) Pentagon d) Lena e) F-16 f) Zelda

$$PSNR = 10\log_{10}\left(\frac{R^2}{MSE}\right) \quad (10)$$

Where MSE (Mean Square Error) can be computed by Equation (11) as follows:

$$MSE = \frac{\sum_{M,N}\left[I_1(m,n) - I_2(m,n)\right]^2}{M \times N} \quad (11)$$

Where M & N illustrate row and column of each image respectively and R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255.If the distortion between the cover image and the stego-image is small, the PSNR value is large. Thus, a larger PSNR value means that the quality of the stego-image is better than the smaller one. Secret data is generated by using pseudo random number generator.

In terms of embedding capacity and quality, the performance of our algorithm is measured by comparing with other reversible data hiding schemes. The message bit is generated by using pseudo random number generator using the MATLAB function. Embedding variables such as sampling factor (Δu and Δv) and threshold is adjusted for comparing the payload capacity and image quality results.

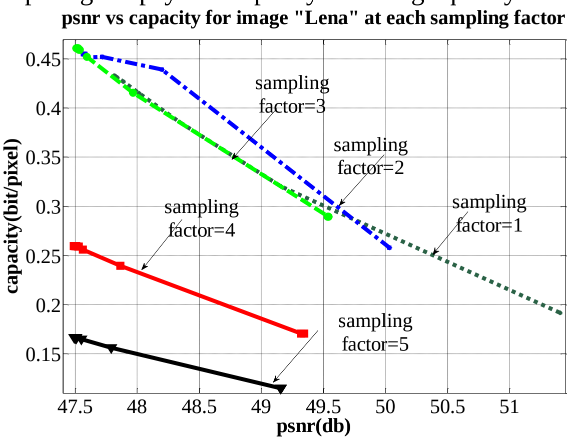

Fig 3(a):Capacity and quality illustration for the selected sampling factor: $\Delta u = 2, \Delta v = 2$, $\Delta u = 3, \Delta v = 3$, $\Delta u = 4, \Delta v = 4$, $\Delta u = 5, \Delta v = 5$ quality vs capacity

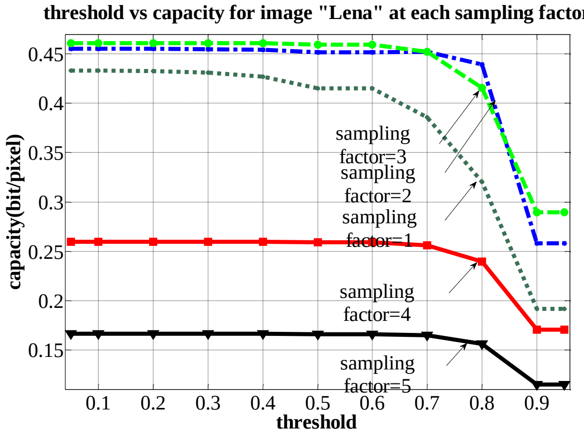

Fig 4(b):Capacity and quality illustration for the selected sampling factor: $\Delta u = 2, \Delta v = 2$, $\Delta u = 3, \Delta v = 3$, $\Delta u = 4, \Delta v = 4$, $\Delta u = 5, \Delta v = 5$

capacity vs threshold

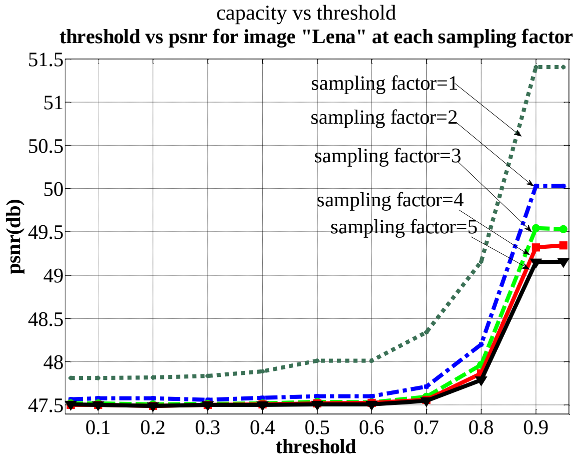

Fig 5(c):Capacity and quality illustration for the selected sampling factor: $\Delta u = 2, \Delta v = 2$, $\Delta u = 3, \Delta v = 3$, $\Delta u = 4, \Delta v = 4$, $\Delta u = 5, \Delta v = 5$

quality vs threshold.

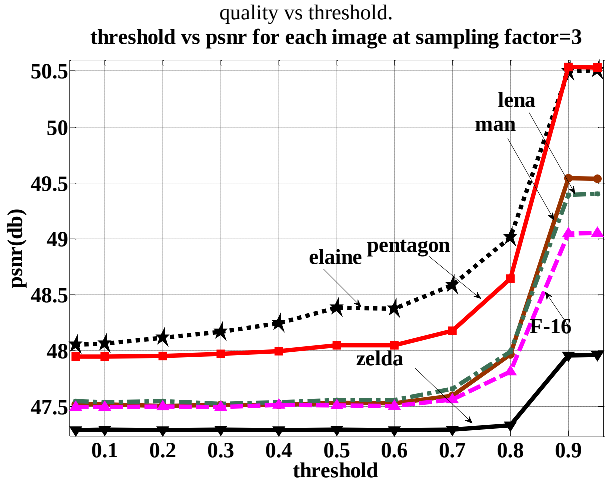

Fig 6(a) Comparison of quality (dB) versus threshold.

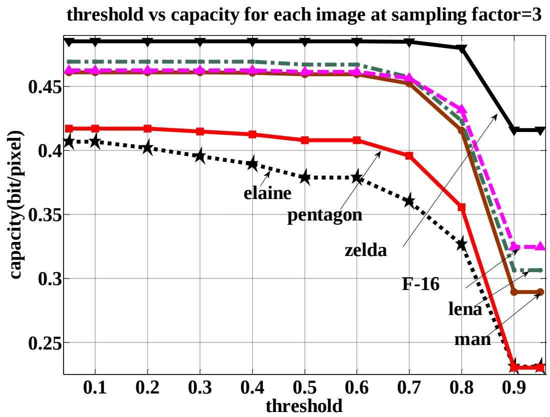

Fig 7: b) Comparison of embedding capacity (bpp) versus threshold

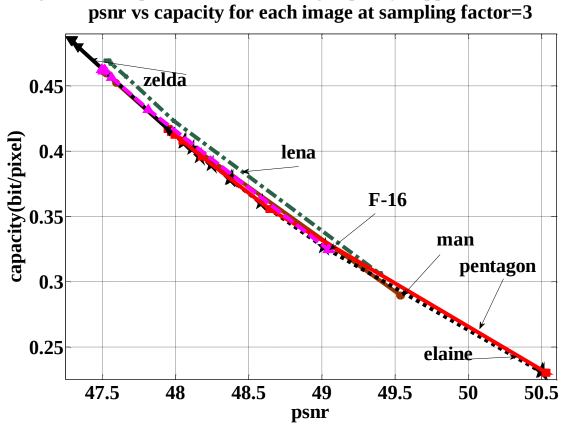

Fig 8 (c) Comparison of embedding capacity (bpp) versus quality(dB)

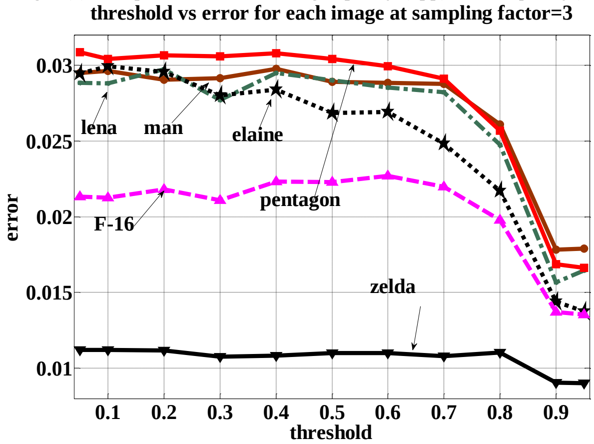

Fig 9 (d) Comparison of error versus threshold for test images.

Table 1 Comparison results of proposed scheme in terms of the payload (bits) and the PSNR (dB) for Lena, Girls, Liberty statue, Baby, Sidewalk and Washington. DC at sampling factor $\Delta u=3, \Delta v=3$ and two threshold of higher case and lower one .

Test image	**Proposed scheme (at T=0.1)** $\Delta u=3, \Delta v=3$		**Proposed scheme (at T=0.9)** $\Delta u=3, \Delta v=3$	
	Capacity	***Psnr***	Capacity	***Psnr***
Pentagon	0.4171	47.95	0.0357	48.9
Elaine	0.4067	48.05	0.0250	48.7
Lena	0.4693	47.55	0.0762	49.0
Zelda	0.4853	47.30	0.1208	48.9
Man	0.4609	47.51	0.0829	48.1
F-16	0.4625	47.50	0.0445	49.6

In this paper a new reversible data hiding technique has been proposed that is based on individually the correlation of sub sample images.

Fig.6(a) analyzes capacity versus quality performance for each sampling factor for image *Lena*. When $\Delta u=5, \Delta v=5$ (sampling factor) is chosen, the performance is more satisfactory among subsampled stego images or resultant stego image in terms of capacity and quality than other sampling factor. For *Lena* image, the sampling factor (3,3) achieved 0.48 bpp at *T*=0.1,whereas the sampling factor (5,5)achieved 0.17 bpp at the same threshold although it utilized more subsampled images than those of the sampling factor (3,3). Because pixel redundancy and correlation between subsampled image's blocks with neighboring ones are high at the selected sampling factors. Fig.6(b) plots the embedding capacity for *Lena* image, with the different sampling factors $(\Delta u=3, \Delta v=3)$ of: (2,2),(3,3), (4,4) and (5,5).The performance of algorithm is degraded when the factor values is large (5,5).This is due to the fact that the larger the sampling factors are, the weaker the correlation between subsampled image blocks is. Fig.6(c).shows the quality of the image *Lena* at various sampling factor and various threshold up to 0.95. From this result, the performance of quality at any threshold for $\Delta u=3, \Delta v=3$ is the lowest one, and quality at any threshold for $\Delta u=1, \Delta v=1$(no sampling factor) is the highest one.

Table 2: Comparison results of proposed schemeat and threshold 0.1and Kim et al scheme in terms of the payload (bits) and the PSNR (dB) at sampling factor $\Delta u=3, \Delta v=3$.

Test image	**Proposed scheme (at T=0.1)** $\Delta u=3, \Delta v=3$ **Capacity(bpp)**	**Psnr(db)**	**Kim et al scheme** $\Delta u=3, \Delta v=3$ **Capacity(bpp)**	**Psnr(db)**
Pentagon	0.4171	47.95	0.0357	48.9
Elaine	0.4067	48.05	0.0250	48.7
Lena	0.4693	47.55	0.0762	49.0
Zelda	0.4853	47.30	0.1208	48.9
Man	0.4609	47.51	0.0829	48.1
F-16	0.4625	47.50	0.0445	49.6

Table 3:Comparison results of proposed schemeat and threshold 0.1andNi et aland MPE schemes respectively in terms of the payload (bits) and the PSNR (dB) at sampling factor $\Delta u = 3, \Delta v = 3$.

Test image	**Proposed scheme (at T=0.1) $\Delta u = 3, \Delta v = 3$ Capacity(bpp)**	**Psnr(db)**	**Ni et al.'sscheme Capacity(bpp)**	**Psnr(db)**	**MPE scheme Capacity(bpp)**	**Psnr(db)**
Pentagon	0.4171	47.95	0.0823	47.6993	0.2726	33.5809
Elaine	0.4067	48.05	0.0088	49.2065	0.1562	34.1688
Lena	0.4693	47.55	0.0119	48.4521	0.4121	29.7486
Zelda	0.4853	47.30	0.0093	55.3048	0.1742	32.7845
Man	0.4609	47.51	0.0163	52.3028	0.4927	28.4108
F-16	0.4625	47.50	0.0105	51.0766	0.2963	30.3416

Table 4: Results of proposed scheme at best threshold in terms of the payload (bits) and the PSNR (dB) and Error /pixel

Test image	**Threshold(T)** $\Delta u = 3, \Delta v = 3$	**Embedded capacity (bbp)**	**PSNR (db)**	**Error /pixel**
Pentagon	0.7	0.4	48.17	0.0291
Elaine	0.7	0.36	48.58	0.0248
Lena	0.7	0.45	47.66	0.0282
Zelda	0.8	0.48	47.33	0.0110
Man	0.7	0.45	47.6	0.0288
F-16	0.7	0.45	47.56	0.0220

Because clearly, by increasing the sampling factor image quality will decline.

Fig. 7 shows the performance of the marked images at various embedding capacities up to 0.5bpp when the sampling factor is $\Delta u = 3, \Delta v = 3$. Fig.7 (a) shows the payload capacity in each threshold from 0.05 to 1. As it is shown at threshold 0.1 and 0.9, the algorithm can embed 0.4503bpp and 0.3478 bpp respectively. At threshold 0.1 almost all blocks can be used for data embedding. PSNR as a measurement of image quality is used for all threshold. Fig.7 (b) shows PSNR for six test images. As it is shown, at $T = 0.9$ the mean PSNR for the images is more than 49db and by decreasing the threshold, PSNR decreases. This effect leads to keep the tradeoff between image quality (PSNR) and capacity, although the PSNR at worst case is not lower than 47.5db .If it decrease to lower than this value, It doesn’t affect the image quality so much.

From this result, as it is shown in Fig. 7(c), the performance of capacity versus quality depends on the characteristics of the images. Some of them, especially *Lena* and *Zelda* images, which containing more low frequency components than middle and high-frequency components, achieved high embedding capacity while keeping the PSNR value low at any threshold. For instance, the capacity is 0.49bpp with PSNR of 47.8 dB at threshold (*T*) equal to 0.1 for *Zelda* image, whereas the capacity is0.41 bpp with PSNR of 48.1dB for *Pentagon* image.

The presented method achieved the capacity from 107480 bits to 128450 bits and the PSNR value from 47.8 to 48.1dB for all test images. Based on test images, the pay load capacity at threshold $T = 0.1$ (most. capacity) and PSNR at worst case (T=0.9) are shown in Table 1.

It is obvious that there isn’t so much difference in payload capacity and PSNR between *side Lena*, *Man* and *F-16* for the reason of existence of middle frequency components, and moderate correlation in all part of these images. But *Zelda* have slow changes in edges and low frequency components, so the correlation of this image is much higher than the previous three images that leads to higher capacity. This high correlation is maintained before and after embedding secret data. Therefore the correlation error is low and PSNR is high. Because these errors show that the changes in correlation are so low. *Pentagon* and *Elain* have sudden changes in edges and high frequency components, so the correlation of this image is lower than the previous images and leads to lower capacity

Fig 9 shows the error of correlation matrix before and after data embedding in each threshold from 0.1 to 1 .These values are all divided by 10^3 . It is clear that the error decreases by increasing of threshold. In fact, the number of positions for embedding will decrease. Thus when the threshold is 1 the error is zero. Because almost there is no position for embedding, so the error is close to zero. There is some difference in all six test images. In *Pentagon* this error is much more than other images but in *Zelda*, This error is lower than other images because of existence of high frequency components in this image and consequently the embedding positions is less than other images. Now if a practical standard is defined for image quality it can be obtained by the correlation error, before and after data embedding. Any of the test images have a specified threshold that shows the best tradeoff between image quality, capacity and correlation error. This practical standard is based on comments of 30 people. The results illustrated in Table(4)shows which threshold is the best one for maintaining image quality and high capacity in each image; based on the correlation error between correlation matrix before and after data hiding.

Table.2 shows the comparison results of proposed scheme at threshold 0.1and Kim et al scheme in terms of the payload (bits) and the PSNR (dB) at sampling factor $\Delta u = 3, \Delta v = 3$. As it is seen, PSNR and capacity of proposed scheme in comparison with Ni et al method is much higher.

Tables(3)show The comparison of PSNR and payload capacity between proposed scheme and Ni et al method and MPE method[18] at $T = 0.1$ respectively. As it is seen, PSNR and capacity of proposed scheme in comparison with Ni et al method is much higher.

Right column of Table (3) shows the comparison of proposed scheme and MPE method in pay load capacity and PSNR. Higher thresholds act more strictly than lower ones and don't let us embed secret data in any part of the image. Even in these threshold such as $T = 0.9$, some test images show higher capacity and PSNR. As it is mentioned, the reason of lower capacity in Washington DC, is the existence of high frequency components which leads to slightly lower capacity than other images at $T = 0.9$. Thus MPE shows higher capacity than the proposed method. MPE method embeds secret data based on error prediction. So it can provide high capacity for embedding but PSNR in proposed scheme is much more than MPE method. The average PSNR in proposed method is more than 48 db, but its average is 31 db in MPE method.

Table 4 shows the results of proposed scheme at best threshold in terms of the payload (bits), the PSNR (dB) and Error /pixel.

V. CONCLUSIONS

In this paper, a simple and efficient reversible data hiding scheme have been proposed that is based on statistical correlation of subsampled images for gray level images under the given threshold. The proposed scheme embeds secret data by modifying pixel values. The overhead information of this method is the thresholds sampling factor and correlation matrix. Experimental results supported that this method achieved higher embedding capacity with a higher PSNR than other reversible scheme. This algorithm has the following advantages: 1) simplicity and effectiveness, 2) adjustability of embedding capacity and quality based on threshold and sampling factor 3) applicability to almost all kinds of grey images. Whereas, a clear disadvantage of our algorithm is the necessity of using correlation matrix in the extraction process. Therefore, it is aimed to plan a study of how to achieve lossless recovery without using correlation matrix. In future works the security of the secret data can be further safeguarded by using encryption-decryption techniques before it is embedded into the host images.

References

[1]Wang, C., Wu, N., Tsai, C., Hwang, M., 2008. A high quality steganographic method with pixel-value differencing and modulus function.Journal of Systems andSoftware 81, 150–158.

[2]Yu, Y.H., Chang, C.C., Lin, I.C., 2007. A new steganographic method for color and grayscale image hiding.Computer Vision and Image Understanding 107, 183–194.

[3]Wien Hong ,Tung-Shou Chen, Chih-Wei Shiu : Reversible data hiding for high quality images using modification of prediction errors.The Journal of Systems and Software 82 (2009) 1833–1842..

[4]Fridrich, J., Goljan, M., Du, R., 2002. Lossless data embedding new paradigm in digital watermarking.Special Issue on Emerging Applications of MultimediaData Hiding 2002 (2), 185–196.

[5]Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E., 2002. Reversible data hiding in: Proceedings of the International Conference on Image Processing, pp. 157–160.Data Hiding 2002 (2), 185–196.

[6]Awrangjeb, M., Kankanhalli, M.S., 2005. Reversible watermarking using a perceptual model.Journal of Electronic Imaging 14 (1), 013014.

[7]Tian, J., 2003. Reversible data embedding using a difference expansion.IEEETransactions on Circuits and Systems for Video Technology 13 (8), 890–896.

[8]Alattar, A.M., 2004. Reversible watermark using the difference expansion of a generalized integer transform. IEEE Transactions on Image Processing 13 (8),1147–1156.

[9]Kamstra, L., Heijmans, H.J.A.M., 2005. Reversible data embedding into images using wavelet techniques and sorting. IEEE Transactions on Image Processing 14 (12),2082–2090.

[10]Thodi, D.M., Rodriguez, J.J., 2007. Expansion embedding techniques for reversible watermarking.IEEE Transactions on Image Processing 16 (3), 721–730.

[11]Kim, H.J., Sachnev, V., Shi,Y.Q,Nam, J,Choo, H.G., 2008. A novel difference expansion transform for reversible data embedding. IEEE Transactions onInformation Forensics and Security 3 (3), 456–465.

[12]Ni, Z., Shi, Y.Q., Ansari, N., Su, W., 2006. Reversible data hiding. IEEE Transactions on Circuits and Systems for Video Technology 16 (3), 354–362.

[13]Kyung-Su Kima, Min-JeongLeea, Hae-YeounLeeb, Heung-KyuLeea:Reversible data hiding exploiting spatial correlation between sub-sampled images.Pattern Recognition 42 (2009) 3083 -- 3096

[14]Yasaman Zandi Mehran, Mona Nafari, Alireza Nafari, and Nazanin Zandi Mehran: Histogram Shifting as a Data Hiding Technique: An Overview of Recent Developments; DICTAP 2011, Part I, CCIS 166, pp. 770–786, 2011. Springer-Verlag Berlin Heidelberg 2011.

[15]Mona Nafari, Gholam Hossein Sheisi, and Mansour Nejati Jahromi: New Data Hiding Method Based on Neighboring Correlation of Blocked Image; DICTAP 2011, Part I, CCIS 166, pp. 787–801, 2011. Springer-Verlag Berlin Heidelberg 2011.

[16]Lim, Jae S., Two-Dimensional Signal and Image Processing, Englewood Cliffs, NJ, Prentice Hall, 1990, pp. 218-237

[17]The USC-SIPI Image Database: http://sipi.usc.edu/database

[18]Wien Hong, Tung-Shou Chen, Chih-Wei Shiu:*Reversible data hiding for high quality images using modification of prediction errors* The Journal of Systems and Software 82 (2009) 1833–1842