

A REVERSIBLE DATA HIDING TECHNIQUES FOR IMPROVED EMBEDDING CAPACITY USING IMAGE INTERPOLATION

SHAHI. D*, VINOD KUMAR. R. S** and BUSHARA A R +

* Assistant Professor, Department of Electronics and Communication Engineering,
Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India, shahijulian@gmail.com

** Professor, Department of Electronics and Communication Engineering,
Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India, rsvinodkumar69@yahoo.co.in

+ Associate Professor, Department of Electronics and Communication Engineering,
KMEA Engineering College, Kerala, India, bushara.ar@gmail.com

Received 1 January 2000; revised 1 January 2000; accepted 1 January 2000

Abstract

High-capacity steganography is still challenging today in the field of information security. The demand for the exact retrieval of the cover media from stego-image after the extraction of secret data is also increasing. Using reversible information hiding techniques, the cover image can be recovered at the time of extraction of secret messages. Two techniques are proposed in this paper. In the first technique, the image is interpolated using a new interpolation technique and the second technique uses a High-Capacity Reversible Steganography using Multi-layer Embedding (CRS) method for image interpolation. In both the techniques, the secret data are embedded in the cover image by Exclusive OR (XOR) operation. The proposed techniques give high embedding capacity and preserve image quality. The proposed method has given a better embedding capacity than the existing methods, with a value which is almost double that of the state of art methods, a comparable PSNR and a higher SSIM.

Key Words: Steganography, Image Interpolation, High Capacity, Embedding Capacity, Reversible Data Hiding.

1 Introduction

Nowadays, secure communication through the internet is challenging because of the development of multimedia communication. The main research areas which would improve security in communication are cryptography and steganography [1]. Steganography is a method of invisible communication and it is an important wing in information hiding, while cryptography makes the secret message undetectable if one does not possess the secret key [2]. The two types of steganography are reversible steganography and irreversible steganography. Good

Correspondence to: bushara.phd@gmail.com

Recommended for acceptance by AngelD.Sappa

ELCVIA ISSN:1577-5097

Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

embedding capacity and visual quality are achieved using irreversible steganography techniques. Using reversible steganography techniques, the cover media is precisely retrieved after the retrieval of hidden information [3]. In addition, embedding can be done in the spatial and frequency domain. Spatial domain steganography techniques embed secret bits directly in pixel values [4]. In transform-based data hiding, the cover image is first converted into the frequency domain using Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) or Discrete Fourier Transform (DFT), then secret data embedding is done [4]. High embedding capacity and imperceptibility are the two important requirements in steganography [5]. It is necessary to improve the embedding capacity while preserving image quality [6].

The performance of a data hiding technique can be analyzed by its transparency, robustness, undetectability and capacity [7]. To reduce the possibility of detecting the secret message, the distortions due to data embedding should be made as low as possible [8]. The simple spatial domain steganography technique is Least Significant Bit (LSB) substitution [9]. This technique offers better capacity, image quality and reduced complexity.

Image interpolation in image processing is used to scale up a small image into a larger image. The other names of image interpolation are Digital Zooming, Image Resizing, Image Re-sampling, and Image Magnification or Enhancement [10]. The size of the image is extended by computing the unknown pixel value at a point using the known pixel values. Image interpolation is an approximation technique which reduces the image quality. Image interpolation is used in medical imaging especially in the reconstruction of Computed Tomography (CT) or Magnetic Resonance Imaging (MRI). Another vital application of image interpolation is data hiding [11]. Image interpolation helps in the recovery of the cover image after secret data extraction, while maintaining good embedding capacity. In the proposed work, an XOR operation is utilized for embedding. This use of the XOR operation ensures that the PSNR remains higher compared to the existing methods while maintaining the embedding capacity.

In this paper, Section 2 explains the existing works in the area of reversible steganography. Section 3 illustrates the proposed methods in data hiding and how embedding capacity is improved, while preserving Peak Signal to Noise Ratio (PSNR). Numerical examples are also given to illustrate the proposed methods. Section 4 shows the results obtained from experimentation and Conclusion is given in Section 5.

2 Related Works

Active research has been carried in the area of steganography for the past two decades. A short research survey which is close to the methods proposed in this paper are discussed in this section. [3] proposed a method of data embedding using the interpolation technique known as Interpolation with Neighbouring Pixels (INP) and the embedding capacity is increased with the use of this technique. [5] presented a data concealing method in which Neighbour Mean Interpolation (NMI) is used for image interpolation and secret message embedding is done using LSB and Optimal Pixel pair Adjustment Process (OPAP). [6] presented a method of high capacity data hiding using High capacity reversible Steganography (CRS) technique. [7] presented a method of data hiding using interpolation in the spatial domain. The type of interpolation proposed is NMI. [10] proposed a method of lossless data hiding using message recoding mechanism capacity and image quality. [12] presented combined the two bit LSB replacement with XOR operation which enhanced the stego image. [13] proposed a reversible data hiding using adaptive embedding where the image is up-sampled using Simplified Parabolic Interpolation (SPI) and secret data are embedded using adaptive embedding process. [14] proposed a new interpolation technique in which the secret messages are concealed using a new adjustable data hiding algorithm. [15] used INP and Enhanced Neighbour Mean Interpolation (ENMI) for scaling up the image and introduced a novel adaptive embedding process to attain good image quality. [16] used Parabolic interpolation for getting the cover image and embedding is done using the connection between calculated pixel value and mean value. [17] proposed an interpolation based data hiding method which used an adaptive embedding scheme. [18] proposed a data hiding technique which used modulo operation to embed secret information in interpolated image.

In the existing methods, since conventional addition/subtraction are used during embedding process, large variation in the grayscale values occur while embedding, which decreases the PSNR of the stego image. The

embedding capacity and PSNR value are inversely proportional in all the existing works. As the embedding capacity increases, the PSNR decreases and vice versa. With the help of the above background knowledge, two increased capacity reversible data concealing schemes are proposed in Section 3.

3 Proposed Methods

In the proposed work, an XOR operation is used in the process of embedding. Since XOR operation is used in the process of embedding, the PSNR will be retained to a higher level than the existing methods. There are also the possibilities of increase in the embedding capacity than the previous methods while retaining the PSNR value. Because of XOR operation, the deviation in the pixel value of the embedded image is much lesser when compared to the conventional addition/subtraction which is used in the previous works.

The diagrammatic indication of the proposed data concealing technique is shown in Figure 1. The given image is first scaled down to get the input image. The input image is then interpolated using the proposed techniques to get the cover image. The confidential data is concealed in the cover media by two proposed techniques using XOR operation to get the stego image. This helps to increase the PSNR of the stego image. From the stego image the secret data can be extracted and the cover image can be recovered without any distortion.

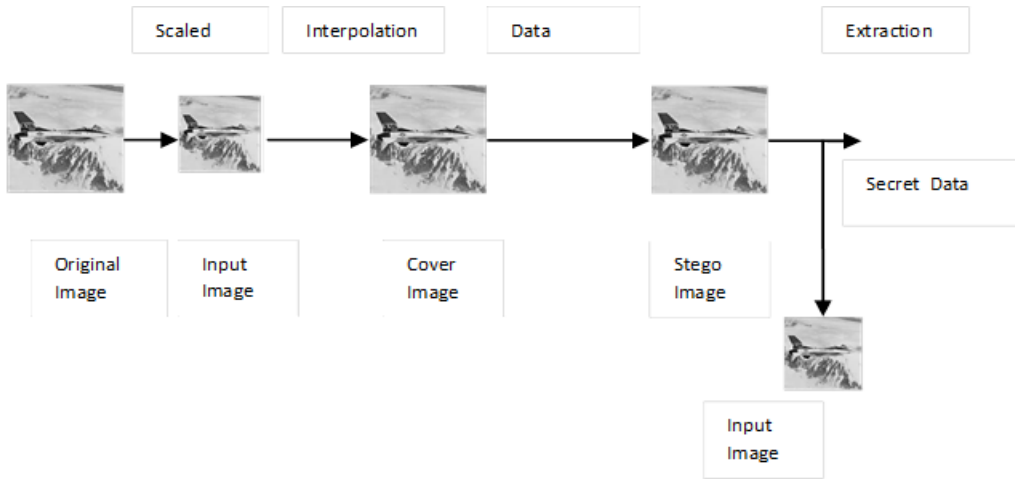


Figure 1: Reversible Image Steganography using Image Interpolation and XOR operation (RSIX)

In this section, two novel methods in reversible image steganography are proposed. They are reversible image steganography with high capacity using image interpolation and XOR operation (RSIX) and reversible image steganography using improved CRS technique (RSIC).

3.1 Method of Interpolation

In the proposed data hiding technique, the interpolation of input image is done at first. The input image (I_m) of size $\frac{X}{2} \times \frac{Y}{2}$ is interpolated to get the cover image (C_{er}) of size and its schematic representation is given in Figure 2. Here $C_{er}(c, d)$, $C_{er}(c, d + 2)$, $C_{er}(c + 2, d)$ and $C_{er}(c + 2, d + 2)$ represent the pivot pixels and $C_{er}(c, d + 1)$, $C_{er}(c + 1, d)$, $C_{er}(c + 1, d + 1)$, $C_{er}(c + 1, d + 2)$ and $C_{er}(c + 2, d + 1)$ represent the interpolated pixels.

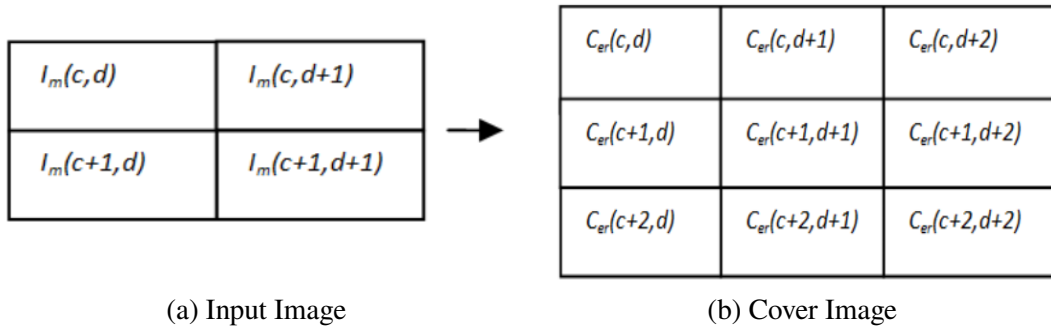


Figure 2: Diagrammatic Representation of the Interpolation Method

The steps involved in this interpolation technique are listed below. They are

1. Center pixel in the cover image is the mean of the 4 diagonal pixels of the input image.
2. Since the center pixel is used for the calculation of other pixels, it is calculated first.
3. The other pixels are calculated as the mean of the neighbouring pixels and center pixel (just calculated).

This procedure is represented mathematically using Eq. (1).

$$\left. \begin{aligned} C_{er}(c, d) &= I_m(s, t) \\ C_{er}(c+1, d+1) &= \frac{C_{er}(c, d) + C_{er}(c, d+2) + C_{er}(c+2, d) + C_{er}(c+2, d+2)}{4} \\ C_{er}(c, d+1) &= \frac{C_{er}(c, d) + C_{er}(c, d+2) + C_{er}(c+1, d+1)}{3} \\ C_{er}(c+1, d) &= \frac{C_{er}(c, d) + C_{er}(c+2, d) + C_{er}(c+1, d+1)}{3} \end{aligned} \right\} \quad (1)$$

The pixel location $C_{er}(c, d) = I_m(c, d)$, $c = 2x$, $d = 2y$ is not changed during data embedding.

3.2 Embedding Procedure

The interpolation technique is followed by the data embedding process. The formulae involved in data embedding are discussed in this section. Let the difference values $db1$, $db2$ and $db3$ between pixels $C_{er}(c, d+1)$, $C_{er}(c+1, d)$ and $C_{er}(c+1, d+1)$ are calculated as in Eq. (2)

$$\left. \begin{aligned} C_{er_{max}} &= \max(C_{er}(c, d), C_{er}(c, d+2), C_{er}(c+2, d), C_{er}(c+2, d+2)) \\ db1 &= C_{er_{max}} - C_{er}(c, d+1) \\ db2 &= C_{er_{max}} - C_{er}(c+1, d) \\ db3 &= C_{er_{max}} - C_{er}(c+1, d+1) \end{aligned} \right\} \quad (2)$$

The number of bits of secret data to be concealed is calculated using Eq. (3)

$$n_{bi} = \log_2(db_i), \text{ where } i = 1, 2 \text{ and } 3 \quad (3)$$

If the neighboring pixels have the same value, the number of bits embedded in that locations are chosen as $n = 1, 2, 3$ and 4 . The secret bits S_i are embedded in pixel locations $C_{er}(c, d+1)$, $C_{er}(c+1, d)$ and $C_{er}(c+1, d+1)$ by XOR operation. The stego pixels values S_t are calculated as in Eq. (4).

$$\left. \begin{aligned} S_t(c, d+1) &= XOR(C_{er}(c, d+1), S_1) \\ S_t(c+1, d) &= XOR(C_{er}(c+1, d), S_2) \\ S_t(c+1, d+1) &= XOR(C_{er}(c+1, d+1), S_3) \end{aligned} \right\} \quad (4)$$

where S_1 , S_2 and S_3 are the secret bits.

The various steps involved in the implementation of RSIX are concised in Section Algorithm 1.

Algorithm 1 Embedding Algorithm

Input:

Original Image, I_m is of size $\frac{X}{2} \times \frac{Y}{2}$

Output:

Stego Image, S_t is of size $X \times Y$

begin

step 1. In the beginning c is set to zero and d is set to zero

step 2. A cover image C_{er} of size $X \times Y$ is obtained using Eq. (1)

step 3. Select a block in cover image (C_{er}) and compute d_{bi} using Eq. (2)

step 4. Calculate n_{bi} using Eq. (3)

step 5. The secret message is concealed in cover image (C_{er}) with the use of Eq. (4)

step 6. If all the information to be hided are embedded, then stego image

S_t of size $X \times Y$ is obtained. Proceed to step 7.

Else

Proceed to step 3

End If

step 7. End (Algorithm)

3.3 Numerical Example

This proposed steganography method can be precised with the help of sub-image given in Figure 3.

160	140	180
136	100	102
76	88	90

(a) Original Image

160	144	140
143	134	123
136	112	100

(b) Cover Image

160	153	140
133	138	108
136	125	100

(c) Stego Image

Figure 3: Numerical Example for RSIX

Let the secret bits are 1001101011000101101110101011011011. $C_{er}(0, 1)$, $C_{er}(1, 0)$, $C_{er}(1, 1)$, $C_{er}(1, 2)$ and $C_{er}(2, 1)$ are calculated using Eq. (1). C_{ermax} is calculated using Eq. (2). The embedding procedure is briefed as follows. Let C_{ermax} represents the maximum value of corner pixels of 3×3 matrix of the cover image.

$$C_{ermax} = \max\{160, 140, 136, 100\} = 100$$

Using Eq. (3).

$$n_{b1} = \log_2 16 = 4$$

$$n_{b2} = \log_2 17 = 4$$

$$n_{b3} = \log_2 26 = 4$$

The Stego image is obtained using the following procedure. Let 1001 represents the secret data 1 (S_1). Padding four zeros to the left yields an 8-bit pattern 00001001. The 8-bit binary pattern of 144 ($C_{er}(0, 1)$) is

10010000. Performing XOR between secret data 1 (S_1) and the pixel value 144, 153 is obtained. The 8-bit binary pattern of 143 ($C_{er}(1, 0)$) is 10001111. Select 1010 as secret data 2 (S_2) and pad four zeros to the left to get 8-bit pattern 00001010. Performing XOR between secret data 2 (S_2) and the pixel value 143, 133 is obtained. The 8-bit binary pattern of 134 ($C_{er}(1, 1)$) is 10000110. Select 1100 as secret data 3 (S_3) and pad four zeros to the left to get 8-bit pattern 00001100. Similarly, performing XOR between secret data 3 (S_3) and the pixel value 134, 138 is obtained. Repeating the aforesaid procedure, Figure 3(c) is obtained.

3.4 Extraction

The cover media can be easily retrieved, since the pivot pixels $C_{er}(0, 0)$, $C_{er}(0, 2)$, $C_{er}(2, 0)$ and $C_{er}(2, 2)$ remain unchanged when Eq. (1) is used. The number of bits to be concealed is evaluated using Eqs. (2) and (3). The secret data is retrieved by doing XOR operation among the stego image and cover image.

In Figure 3 (c), the stego pixels $S_t(0, 1)$, $S_t(1, 0)$ and $S_t(1, 1)$ are 153, 133 and 138 respectively. Since $S_t(0, 0) = C_{er}(0, 0)$, $S_t(0, 2) = C_{er}(0, 2)$, $S_t(2, 0) = C_{er}(2, 0)$ and $S_t(2, 2) = C_{er}(2, 2)$, cover image in Figure 3(b) can be recovered from stego image in Figure 3 (c) using Eq. (1). The number of bits to be concealed is evaluated using Eqs. (2) and (3) as

$$\begin{aligned} n_{b1} &= \log_2 16 = 4 \\ n_{b2} &= \log_2 17 = 4 \\ n_{b3} &= \log_2 26 = 4 \end{aligned}$$

The embedded secret information can be retrieved from the 8 bit binary pattern (S_{8i}) of secret data, which is obtained from the XOR operation of Stego image and Cover image. The 8 bit pattern of secret data (S_{8i}), where $i=1, 2$ and 3 are calculated as follows.

$$S_{81} = \text{XOR}(S_t(0,1), C_{er}(0,1)) = 00001001$$

$$S_{82} = \text{XOR}(S_t(1,0), C_{er}(1,0)) = 00001010$$

$$S_{83} = \text{XOR}(S_t(1,1), C_{er}(1,1)) = 00001100$$

Since $n_{b1}=4$, extracting the four LSB of S_{81} , secret data 1 (S_1) becomes 1001

Since $n_{b2}=4$, extracting the four LSB of S_{82} , secret data 2 (S_2) becomes 1010

Since $n_{b3}=4$, extracting the four LSB of S_{83} , secret data 3 (S_3) becomes 1100

4 Reversible Image Steganography using Improved CRS (RSIC)

4.1 Method of Interpolation

The input is first scaled down and then interpolated using CRS [6] to get the cover image. The secret data is embedded by XOR operation. The cover image in Figure 2(b) is obtained using Eq. (5).

$$\left. \begin{aligned} C_{er_{max}} &= \max(C_{er}(c, d), C_{er}(c, d+2), C_{er}(c+2, d), C_{er}(c+2, d+2)) \\ C_{er_{min}} &= \min(C_{er}(c, d), C_{er}(c, d+2), C_{er}(c+2, d), C_{er}(c+2, d+2)) \\ A &= \frac{3 \times C_{er_{min}} + C_{er_{max}}}{4} \\ C_{er}(c, d+1) &= \frac{A + \frac{C_{er}(c, d) + C_{er}(c, d+2)}{2}}{2} \\ C_{er}(c+1, d+1) &= \frac{(C_{er}(c, d) + C_{er}(c+1, d) + C_{er}(c, d+1))}{3} \end{aligned} \right\} \quad (5)$$

The pixel location $C_{er}(c, d) = I_m(c, d)$, $c=2x$, $d=2y$ is not changed during data embedding.

4.2 Embedding Procedure

The embedding method in this proposed technique is explained in this section.

The difference values d_{b1} , d_{b2} and d_{b3} between pixels $C_{er}(c, d + 1)$, $C_{er}(c + 1, d)$ and $C_{er}(c + 1, d + 1)$ respectively are calculated using Eq. (6)

$$\begin{aligned} d_{b1} &= \begin{cases} C_{ermax} - C_{er}(c, d + 1) & C_{er}(c, d + 1) < C_{ermax} + C_{ermin}/2 \\ C_{er}(c, d + 1) - C_{ermin}, & otherwise \end{cases} \\ d_{b2} &= \begin{cases} C_{ermax} - C_{er}(c + 1, d) & C_{er}(c + 1, d) < \frac{C_{ermax} + C_{ermin}}{2} \\ C_{er}(c + 1, d) - C_{ermin}, & otherwise \end{cases} \\ d_{b3} &= \begin{cases} C_{ermax} - C_{er}(c + 1, d + 1), & C_{er}(c + 1, d + 1) < \frac{C_{ermax} + C_{ermin}}{2} \\ C_{er}(c + 1, d + 1) - C_{ermin}, & otherwise \end{cases} \end{aligned} \quad (6)$$

The number of secret message bits to be concealed is evaluated using Eq. (3). If the neighboring pixels have the same value, the bits embedded in that locations are chosen as $n = 4$. The secret bits S_i are embedded in pixel locations of $C_{er}(c, d + 1)$, $C_{er}(c + 1, d)$ and $C_{er}(c + 1, d + 1)$ by XOR operation using Eq. (4). The various steps involved in the implementation of RSIC is briefed in Section Algorithm 2.

4.3 Embedding Algorithm

Algorithm 2 Embedding Algorithm

Input:

Original Image, I_m is of size $\frac{X}{2} \times \frac{Y}{2}$

Output:

Stego Image, S_t is of size $X \times Y$

begin

step 1. In the beginning c is set to zero and d is set to zero

step 2. A cover image C_{er} of size $X \times Y$ is obtained using Eq. (5)

step 3. Select a 3×3 block in cover image (C_{er}) and compute d_{bi} using Eq. (6)

step 4. Calculate n_{bi} using Eq. (3)

step 5. The secret message bits is concealed in cover image (C_{er}) using Eq. (4)

step 6. If the entire secret information bits are embedded, then stego image S_t of size $X \times Y$ is attained. follow step 7

Else

Proceed to step 3

End If

step 7. End (Algorithm)

4.4 Illustration using a Sub-Image

This proposed steganography technique can be explained using the example shown in Figure (4). Let the secret bits are 1001101011000101101110101011011. $C_{er}(0, 1)$, $C_{er}(1, 0)$, $C_{er}(1, 1)$, $C_{er}(1, 2)$ and $C_{er}(2, 1)$ are calculated based on Eq. (5). The difference in values between the reference pixel and $C_{er}(c, d + 1)$, $C_{er}(c + 1, d)$ and $C_{er}(c + 1, d + 1)$ calculated using Eq. (6).

$$C_{errmax} = \max\{160, 140, 136, 100\} = 160$$

$$C_{ermin} = \min\{160, 140, 136, 100\} = 100$$

$$n_{b1} = \log_2 32 = 5$$

$$n_{b2} = \log_2 31 = 4$$

$$n_{b3} = \log_2 41 = 5$$

The stego image is obtained using the following procedure. Consider 10011 as secret data1 (S_1) and pad three zeros to the left to get 8-bit pattern 00010011. The 8-bit binary pattern of $132(C_{er}(0, 1))$ is 10000100. Performing XOR between secret data1 (S_1) and the pixel $C_{er}(0, 1)$, 151 is obtained. Select 0101 as secret data 2 (S_2) and pad four zeros to the left to get 8-bit pattern 00000101. The 8-bit binary pattern of $131(C_{er}(1, 0))$ is 10000011. Performing XOR operation between secret data 2 (S_2) and the pixel $C_{er}(1, 0)$, 134 is obtained. Consider 10001 as secret data 3(S_3) and pad three zeros to the left to get 8-bit pattern 00010001. The 8-bit binary pattern of $141(C_{er}(1, 1))$ is 10001101. After doing XOR between secret data 3 (S_3) and the pixel $C_{er}(1, 1)$, 156 is obtained. Repeating the above operation for the pixels in Fig 4 (b), the stego image shown in Figure 4(c) is obtained.

160	140	180
136	100	102
76	88	90

(a) Original Image

160	132	140
131	141	120
136	104	100

(b) Cover Image

160	151	140
134	156	98
136	115	100

(c) Stego Image

Figure 4: Numerical Illustration for RSIC

4.5 Extraction

The cover image can be recovered since the pivot pixels $C_{er}(0, 0)$, $C_{er}(0, 2)$, $C_{er}(2, 0)$ and $C_{er}(2, 2)$ remain unchanged when Eq. (5) is used. The number of bits to be hidden is estimated using Eqs. (6) and (3). The secret data is extracted by doing XOR operation among the stego image and cover image.

From Figure 4(c), the stego pixels $S_t(0, 1)$, $S_t(1, 0)$ and $S_t(1, 1)$ are 151, 134 and 156 respectively. Since $S_t(0, 0) = C_{er}(0, 0)$, $S_t(0, 2) = C_{er}(0, 2)$, $S_t(2, 0) = C_{er}(2, 0)$ and $S_t(2, 2) = C_{er}(2, 2)$, cover image in Figure 4(b) can be recovered from stego image in Figure 4(c) using Eq. (5). Using Eqs. (6) and (3).

$$n_{b1} = \log_2 32 = 5$$

$$n_{b2} = \log_2 31 = 4$$

$$n_{b3} = \log_2 41 = 5$$

The hidden secret information can be retrieved from the 8 bit binary pattern (S_{8i}) of secret data, where $i=1, 2, 3$ and 4 are obtained from the XOR operation of Stego image and Cover image. The 8 bit pattern of secret data (S_{8i}) is calculated as follows.

$$S_{81} = \text{XOR}(S_t(0,1), C_{er}(0,1)) = 00010011$$

$$S_{82} = \text{XOR}(S_t(1,0), C_{er}(1,0)) = 00000101$$

$$S_{83} = \text{XOR}(S_t(1,1), C_{er}(1,1)) = 00010001$$

Since $n_{b1}=5$, extracting the five LSB of S_{81} , secret data 1 (S_1) becomes 10011

Since $n_{b2}=4$, extracting the four LSB of S_{82} , secret data 2 (S_2) becomes 0101

Since $n_{b3}=5$, extracting the five LSB of S_{83} , secret data 3 (S_3) becomes 10001

5 Experimental Results and Discussion

The proposed methods were tested using gray scale images and the performance of the methods proposed was assessed using peak signal to noise ratio (PSNR) and embedding capacity (EC). The implementation of the algorithm was done using MATLAB (R2016b) and execution is done in i5 Intel (R) Core (TM) processor @ 1.7 GHz which has 4GB RAM, and they were compared with INP, CRS and image interpolation based data hiding (IIDH) techniques. In the encoder the original image is of size 256×256 pixels, the down sampled input

image I_m is of size 128×128 pixels. The stego image S_t and cover image C_v are of size 256×256 pixels. The proposed methods gave better performance when compared with INP, CRS and IIDH techniques. Table 2 shows the results of proposed methods and their comparison with the INP, CRS and (IIDH) techniques. The test images are shown in Figure 5.

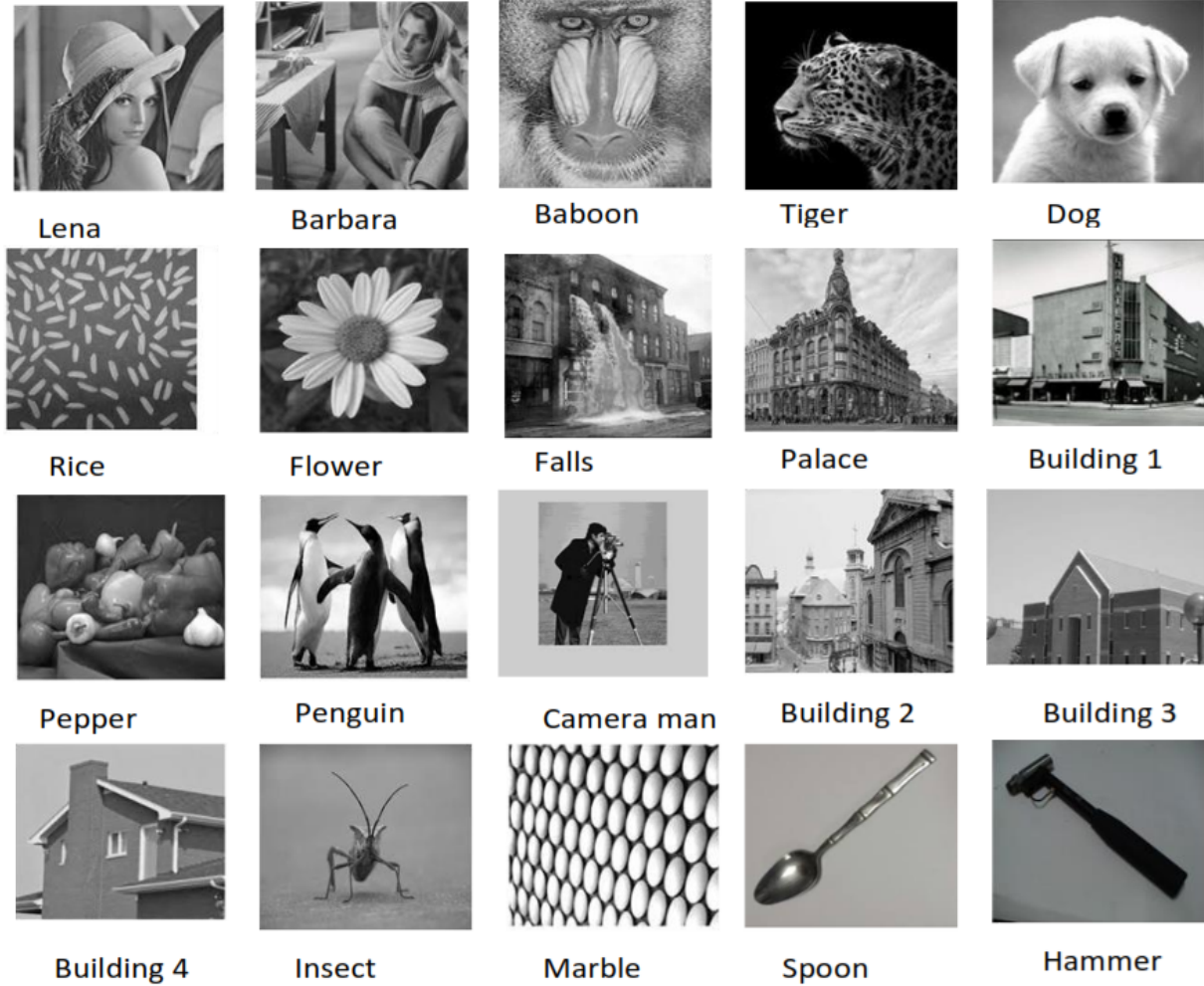


Figure 5: Sample Gray Scale Images used in the Proposed Work

Table 1: Performance Comparison of the RSIX using PSNR and Embedding Capacity (EC)

Test Images	Bits Embedded during identical neighboring values							
	n = 1		n = 2		n = 3		n = 4	
	PSNR (dB)	EC (bpp)	PSNR (dB)	EC (bpp)	PSNR (dB)	EC (bpp)	PSNR (dB)	EC (bpp)
Lena	34.21	2.24	34.39	2.26	34.32	2.28	34.22	2.30
Barbara	33.98	2.49	34.05	2.50	33.98	2.51	33.82	2.52
Baboon	34.29	2.98	34.24	2.98	34.28	2.98	34.30	2.98
Tiger	31.93	2.50	31.71	2.89	31.38	3.28	29.71	3.67
Dog	39.19	1.58	39.07	1.64	38.91	1.70	37.89	1.76

Rice	32.35	2.47	32.32	2.54	32.31	2.62	31.84	2.70
Flower	34.61	2.24	34.78	2.25	34.67	2.25	34.61	2.26
Falls	34.53	2.58	34.56	2.60	34.66	2.62	34.35	2.64
Palace	34.04	2.47	34.02	2.47	33.92	2.48	33.93	2.49
Building 1	33.36	2.04	33.38	2.08	33.40	2.12	33.15	2.17
Pepper	33.18	1.82	32.76	2.16	32.46	2.51	30.69	2.86
Penguin	32.44	1.91	32.55	2.16	32.09	2.42	30.89	2.67
Camera man	35.61	1.54	35.55	2.06	34.81	2.59	32.20	3.12
Building 2	33.85	2.48	33.78	2.65	33.62	2.82	33.38	3.00
Building 3	37.91	1.31	38.01	1.61	36.92	1.92	35.01	2.20
Building 4	36.26	1.60	35.93	1.78	36.22	1.96	34.87	2.14
Insect	40.66	1.04	40.45	1.27	39.26	1.49	36.85	1.71
Marble	27.49	2.56	27.56	2.93	27.39	3.29	26.59	3.66
Spoon	39.09	1.09	38.72	1.66	37.23	2.22	34.15	2.79
Hammer	43.83	0.59	43.91	0.62	43.43	0.66	41.91	0.70

Table 1, shows the Embedding Capacity (EC) and PSNR values for the RSIX technique for 20 gray scale images. The Embedding Capacity (EC) and the PSNR values were computed for various values of $n = 1, 2, 3$ and 4, where n is the number of bits embedded in pixel location if the neighbouring pixels have the same value. The value of embedding capacity shows increase in value with the value of n and the PSNR shows decrease in value with the value of n , which is evident from Table 1.

Table 2, shows the Embedding Capacity (EC) and PSNR values for the RSIX using 10 gray scale image with homogenous background. This proposed method gives improved capacity and PSNR values for gray scale images with homogenous background.

Table 2: Performance Evaluation of PSNR and Embedding Capacity (EC) for the RSIX with the Existing Techniques in Homogenous Background Grayscale Images

Test Images	INP [3]		CRS [6]		IIDH [5]			RSIX (n=4) (Proposed Method)	
	PSNR (dB)	EC (bpp)	PSNR (dB)	EC (bpp)	PSNR (dB) LSB	PSNR (dB) OPAP	EC (bpp)	PSNR (dB)	EC (bpp)
Tiger	27.51	1.96	27.43	2.37	36.05	41.56	1.29	29.71	3.7
Pepper	28.84	1.75	28.55	2.32	37.06	42.06	1.06	30.69	2.86
Penguin	28.08	1.50	27.97	1.96	37.22	42.14	0.88	30.89	2.67
Camera man	31.22	0.92	30.74	1.18	38.86	45.14	0.54	32.20	3.12
Building 2	30.14	2.12	28.53	2.66	35.74	43.00	1.30	33.38	3.0
Building 3	34.94	0.92	32.84	1.22	40.96	47.23	0.48	35.01	2.22
Building 4	32.81	1.30	31.20	1.69	38.83	45.76	0.67	34.87	2.14
Insect	36.80	0.80	35.17	1.04	41.83	49.77	0.34	36.85	1.71
Marble	22.76	1.91	22.79	2.51	32.77	36.93	1.47	26.59	3.66
Spoon	35.22	0.49	33.35	0.63	41.76	48.76	0.99	34.15	2.79

Table 3, indicates the Embedding Capacity (EC) and PSNR values for the RSIX using 10 gray scale images. While the proposed method gives improved PSNR values for gray scale images, the embedding capacity is

slightly lower than the CRS technique, which is evident from Table 3.

Table 3: Performance Evaluation of PSNR and Embedding Capacity (EC) for the RSIX with the Existing Techniques in Grayscale Images

Test Images	INP [3]		CRS [6]		IIDH [5]			RSIX (n=4) (Proposed Method)	
	PSNR (dB)	EC (bpp)	PSNR (dB)	EC (bpp)	PSNR (dB) LSB	PSNR (dB) OPAP	EC (bpp)	PSNR (dB)	EC (bpp)
Lena	30.58	2.08	29.35	2.60	36.98	43.54	1.13	34.22	2.29
Barbara	29.67	2.21	28.84	2.88	36.39	43.54	1.35	33.82	2.52
Baboon	30.44	2.73	29.22	3.41	35.87	43.66	1.70	34.30	3.0
Dog	34.66	1.40	33.91	1.85	40.97	48.86	0.69	36.25	1.76
Rice	27.96	2.17	27.09	2.75	35.51	41.96	1.30	31.84	2.70
Flower	30.26	2.02	29.47	2.62	37.85	44.25	1.13	34.61	2.26
Falls	30.73	2.35	29.63	2.95	37.15	44.32	1.35	34.35	2.64
Palace	30.09	2.23	28.84	2.85	36.47	43.65	1.32	34.04	2.47
Building 1	29.14	1.86	28.17	2.34	37.49	42.76	0.99	33.15	2.17
Hammer	40.53	0.55	39.53	0.81	46.69	53.88	0.13	41.91	0.70

Table 4, shows the Embedding Capacity (EC) and PSNR values for RSIC using 20 gray scale images. RSIC gives improved capacity and PSNR values for all the gray scale images. Even though the IIDH technique gives high PSNR, the embedding capacity is very low, which is lesser than one half of the proposed technique.

Table 4: Performance Evaluation of PSNR and Embedding Capacity (EC) for the RSIC with the Existing Techniques

Test Images	INP [3]		CRS [6]		IIDH [5]			RSIC (n=4) (Proposed Method)	
	PSNR (dB)	EC (bpp)	PSNR (dB)	EC (bpp)	PSNR (dB) LSB	PSNR (dB) OPAP	EC (bpp)	PSNR (dB)	EC (bpp)
Lena	30.58	2.08	29.35	2.60	36.98	43.54	1.13	32.19	2.89
Barbara	29.67	2.21	28.84	2.88	36.39	43.54	1.35	31.65	2.92
Baboon	30.44	2.73	29.22	3.41	35.87	43.66	1.70	32.22	3.41
Tiger	27.51	1.96	27.43	2.37	36.05	41.56	1.29	28.29	3.93
Dog	34.66	1.40	33.91	1.85	40.97	48.86	0.69	36.25	2.07
Rice	27.96	2.17	27.09	2.75	35.51	41.96	1.30	29.95	3.06
Flower	30.26	2.02	29.47	2.62	37.85	44.25	1.13	32.44	2.63
Falls	30.73	2.35	29.63	2.95	37.15	44.32	1.35	32.61	3.02
Palace	30.09	2.23	28.84	2.85	36.47	43.65	1.32	31.88	2.88
Building 1	29.14	1.86	28.17	2.34	37.49	42.76	0.99	31.33	2.69
Pepper	28.84	1.75	28.55	2.32	37.06	42.06	1.06	29.75	3.10
Penguin	28.08	1.50	27.97	1.96	37.22	42.14	0.88	29.32	2.96
Camera man	31.22	0.92	30.74	1.18	38.86	45.14	0.54	31.17	3.29
Building 2	30.14	2.12	28.53	2.66	35.74	43.00	1.30	31.09	3.34

Building 3	34.94	0.92	32.84	1.22	40.96	47.23	0.48	33.69	2.43
Building 4	32.81	1.30	31.20	1.69	38.83	45.76	0.67	33.43	2.41
Insect	36.80	0.80	35.17	1.04	41.83	49.77	0.34	35.61	1.93
Marble	22.76	1.91	22.79	2.51	32.77	36.93	1.47	24.87	3.98
Spoon	35.22	0.49	33.35	0.63	41.76	48.76	0.99	33.40	2.89
Hammer	40.53	0.55	39.53	0.81	46.69	53.88	0.13	40.79	0.96



Figure 6: Performance Comparison of the Proposed Method (Columns 5 & 6) with CRS method (Columns 1 & 2) and An improved Capacity data hiding Technique [14] (Columns 3 & 4); Columns 1, 3 & 5: Interpolated images; Columns 2, 4 & 6: Stego Images; Input Images (Row 1 to 10): Lena, Barbara, Baboon, Airfield, Boats, Bridge, Couple, Elaine and Goldhill



Figure 7: Performance Comparison of the Proposed Method (Columns 5 & 6) with CRS method (Columns 1 & 2) and An improved Capacity data hiding Technique [14] (Columns 3 & 4); Columns 1, 3 & 5: Interpolated images; Columns 2, 4 & 6: Stego Images; Input Images (Row 1 to 9): Tiger, Dog, Rice, Flower, Falls, Palace, Building 1, Pepper and Building 3

In addition, the comparison of the proposed work with the recent relevant works is done with standard images,

namely Lena, Barbara, Baboon, Airfield, Boats, Bridge, Couple, Elaine and Goldhill. Embedding Capacity, SSIM and PSNR are the quantifying parameters used for comparison and the obtained output images are shown in Figures. 6 and 7. Also, the obtained readings are depicted in Tables 5, 6 and 7.

Table 5: Performance Evaluation of Embedding Capacity (bpp) of the Proposed Method with the Existing Methods

Images	[13]		[14]	[15]		[16]	RSIC (n=4) (Proposed Method)
	T=4	T=5		With INP T=3	With ENMI T=3		
Lena	2.505	1.876	1.68	2.499	2.502	1.7954	2.89
Barbara	2.399	1.786	1.92	2.378	2.385	2.2511	2.92
Baboon	2.585	1.887	2.24	2.579	2.586	2.9923	3.41
Airfield	2.914	2.179	2.34	2.915	2.922	2.6661	3.45
Boats	2.652	2.006	2.09	2.621	2.625	1.6814	2.91
Bridge	2.381	1.765	2.32	2.360	2.367	2.7986	3.5
Couple	2.547	1.856	2.06	2.535	2.542	2.2357	3.14
Elaine	2.077	1.073	1.061	2.063	2.062	2.0832	2.30
Goldhill	2.351	1.726	1.78	2.334	2.335	2.1545	2.88
Tiger	2.99	2.02	2.12	3.010	3.012	2.9812	3.93
Dog	0.99	0.94	0.957	0.980	0.989	0.9734	2.07
Rice	2.199	2.001	2.012	2.198	2.221	2.1452	3.06
Flower	2.184	2.103	2.112	2.276	2.289	2.1003	2.63
Falls	2.497	2.428	2.489	2.513	2.617	2.5864	3.02
Palace	2.378	2.356	2.387	2.418	2.502	2.4981	2.88
Building 1	2.112	2.106	2.098	2.188	2.198	2.2011	2.69
Pepper	2.398	2.367	2.298	2.326	2.344	2.4902	3.10
Penguin	2.198	2.196	1.085	2.166	2.187	2.2073	2.96
Camera man	2.409	2.407	2.356	2.400	2.496	2.5264	3.29
Building 2	2.677	2.672	2.487	2.513	2.545	2.6843	3.34
Building 3	1.788	1.778	1.767	1.795	1.826	1.9886	2.43
Building 4	1.828	1.824	1.798	1.815	1.842	1.8956	2.41
Insect	1.094	1.086	1.099	1.159	1.164	1.1824	1.93
Marble	2.489	2.487	2.506	2.592	2.653	2.7753	3.98
Spoon	2.188	1.985	2.037	2.086	2.095	2.1896	2.89
Hammer	0.291	0.288	0.296	0.303	0.317	0.3352	0.96

Table 6: Performance Evaluation of SSIM of the Proposed Method with the Existing Methods

Images	[13]		[14]	[15]		[16]	RSIC (n=4) (Proposed Method)
	T=4	T=5		With INP T=3	With ENMI T=3		
Lena	0.8327	0.9352	0.9501	0.7084	0.7042	0.9607	0.9612
Barbara	0.9087	0.9600	0.8316	0.8036	0.7943	0.8614	0.9406

Baboon	0.9707	0.9922	0.8149	0.8353	0.8211	0.7739	0.9191
Airfield	0.9131	0.9691	0.9723	0.7419	0.7248	0.8530	0.9257
Boats	0.9009	0.9615	0.9661	0.7132	0.7042	0.9493	0.9365
Bridge	0.9787	0.9924	0.9612	0.8739	0.8616	0.8479	0.9218
Couple	0.9069	0.9690	0.9692	0.7476	0.7375	0.9230	0.9319
Elaine	0.9032	0.9679	0.9722	0.6936	0.6837	0.9255	0.9490
Goldhill	0.9269	0.9664	0.9756	0.8076	0.7968	0.9285	0.9389
Tiger	0.9228	0.9683	0.9723	0.8274	0.8163	0.9143	0.9483
Dog	0.9425	0.9629	0.9816	0.7995	0.7841	0.9471	0.9635
Rice	0.9319	0.9636	0.9725	0.8312	0.8285	0.9463	0.9541
Flower	0.9523	0.9666	0.9618	0.8398	0.8374	0.9405	0.9484
Falls	0.9427	0.9613	0.9583	0.8817	0.8739	0.9262	0.9385
Palace	0.9561	0.9663	0.9515	0.7925	0.7839	0.9466	0.9629
Building 1	0.9529	0.9552	0.9582	0.7938	0.7927	0.9238	0.9382
Pepper	0.9592	0.9583	0.9526	0.8417	0.8364	0.9381	0.9472
Penguin	0.9528	0.9665	0.9631	0.7327	0.7263	0.9473	0.9515
Camera man	0.9513	0.9683	0.9526	0.7516	0.7482	0.9283	0.9372
Building 2	0.9621	0.9674	0.9512	0.8331	0.8274	0.9372	0.9421
Building 3	0.9581	0.9673	0.9527	0.8673	0.8562	0.9261	0.9475
Building 4	0.9552	0.9627	0.9513	0.7978	0.7921	0.9287	0.9419
Insect	0.9327	0.9606	0.9528	0.8626	0.8571	0.9419	0.9492
Marble	0.9537	0.9673	0.9662	0.8573	0.8485	0.9428	0.9571
Spoon	0.9603	0.9624	0.9623	0.8226	0.8163	0.9427	0.9582
Hammer	0.9518	0.9626	0.9519	0.7942	0.7826	0.9241	0.9373

Table 7: Performance Evaluation of PSNR (dB) of the Proposed Method with the Existing Methods

Images	[13]		[14]	[15]		[16]	RSIC (n=4) (Proposed Method)
	T=4	T=5		With INP T=3	With ENMI T=3		
Lena	41.52	46.88	30.22	29.44	29.46	31.5249	32.19
Barbara	42.22	47.27	23.84	30.19	30.22	24.0956	31.65
Baboon	41.26	47.19	21.23	29.28	29.28	20.1189	32.22
Airfield	39.53	45.67	30.73	27.59	27.55	22.5284	28.57
Boats	40.52	45.96	33.85	28.67	28.71	29.1458	32.08
Bridge	42.39	47.42	33.34	30.29	30.33	22.2311	31.03
Couple	41.53	47.45	33.44	29.48	29.62	27.4339	31.52
Elaine	41.04	47.03	35.11	28.93	28.94	30.7798	38.87
Goldhill	42.80	47.87	35.76	30.66	30.65	29.2010	33.77
Tiger	32.16	33.34	29.14	24.36	24.33	23.2781	28.29
Dog	37.88	38.89	37.15	31.49	31.47	30.7424	36.25
Rice	34.38	34.45	30.12	24.70	24.66	24.5641	29.95
Flower	36.52	36.63	33.74	29.89	29.86	29.4821	32.44
Falls	35.23	35.72	33.17	29.08	29.05	28.9673	32.61

Palace	33.07	34.25	33.05	26.97	26.94	26.7491	31.88
Building 1	33.77	34.74	33.27	27.89	27.85	27.1637	31.33
Pepper	32.64	33.61	31.26	26.27	26.26	25.3176	29.75
Penguin	34.86	33.63	29.31	25.79	25.78	24.7836	29.32
Camera man	34.37	34.42	32.85	27.78	27.75	26.2614	31.17
Building 2	34.72	34.74	33.06	26.17	26.13	25.0372	31.09
Building 3	35.95	36.18	34.75	27.98	27.97	27.6537	33.69
Building 4	34.89	35.17	34.86	26.96	26.94	26.7436	33.43
Insect	37.61	37.84	36.72	28.98	28.94	28.5248	35.61
Marble	26.74	26.82	25.17	22.66	22.62	21.3135	24.87
Spoon	35.97	36.81	34.69	27.24	27.18	26.7437	33.40
Hammer	43.83	44.06	42.07	33.77	33.76	33.1853	40.79

5.1 Comparison with the Existing Methods

The proposed method has given better embedding capacity than the six existing methods namely [13], [14], [15], [16], [17] and [18]. This is evident from Table 8. The SSIM of the proposed method is greater than [13], [15] and [16], which is evident from Table 6. The PSNR of the proposed method is also higher than [15] and [16], which is understood from Table 7. Even though the PSNR of [13], [14], [17] and [18] are slightly higher than the proposed method, their Embedding Capacity is very much lesser, which is known from Tables 5, 6, 7 and 8. The above experimentation is done for different values of predefined number of unchanged MSBs, T for [13]. The values of PSNR, SSIM and EC represented in Table 8 are the average of 20 sample images namely, Lena, Barbara, Baboon, Airfield, Boats, Bridge, Couple, Elaine, Goldhill, Tiger, Dog, Rice, Flower, Falls, Palace, Building 1, Pepper, Building 3, Spoon and Hammer.

Table 8: Performance Comparison of the proposed work with the State-of-Art-Methods

Authors/Methods		PSNR (dB)	SSIM	EC (bpp)
[16]		26.67	0.9201	1.989
[15]	with INP, $T = 3$	30.32	0.8008	2.148
	with ENMI, $T = 3$	28.16	0.7937	2.216
[14]		32.38	0.9510	1.811
[13]	with $T = 4$	37.21	0.9377	2.197
	with $T = 5$	39.39	0.9654	1.892
[17]		38.79	-	1.455
[18]		45.67	-	1.400
RSIC (n=4) (Proposed Method)		32.22	0.9441	2.881

In the decoder, after the extraction of secret data, the input image (I_m) of size 128×128 pixels and the up sampled cover image (C_v) of size 256×256 pixels are retrieved.

6 Conclusion

Two reversible data hiding techniques for improved Embedding Capacity using XOR operation are proposed. The first method used a novel interpolation technique and the second method used CRS technique for interpolation. Both the methods have used XOR operation for embedding. While the first method is exclusively

applicable for homogenous images, the second method is applicable to all types of images which are understood from the results. Also, the proposed methods are robust to attacks due to noise. The proposed method has given a better embedding capacity than the existing methods, with a value which is almost double that of the state of art methods, moderate PSNR and a better SSIM. The proposed method gave an embedding capacity of 2.88bpp, SSIM of 0.9441 and PSNR of 32.22dB. The work can be modified to develop an improved steganography approach in which the gray scale cover image shall be replaced by an RGB image.

References

- [1] Chao, R. M., Wu, H. C., Lee, C. C., & Chu, Y. P. (2009). A novel image data hiding scheme with diamond encoding. *EURASIP Journal on Information Security*, 2009(1), 1-9.
- [2] Fridrich, J., Goljan, M., & Du, R. (2001). Detecting LSB steganography in color, and gray-scale images. *IEEE multimedia*, 8(4), 22-28.
- [3] Lee, C. F., & Huang, Y. L. (2012). An efficient image interpolation increasing payload in reversible data hiding. *Expert systems with applications*, 39(8), 6712-6719.
- [4] Tang, M., Hu, J., Song, W., & Zeng, S. (2015). Reversible and adaptive image steganographic method. *AEU-International Journal of Electronics and Communications*, 69(12), 1745-1754.
- [5] Yang, C. N., Hsu, S. C., & Kim, C. (2017). Improving stego image quality in image interpolation based data hiding. *Computer Standards & Interfaces*, 50, 209-215.
- [6] Tang, M., Hu, J., & Song, W. (2014). A high capacity image steganography using multi-layer embedding. *Optik*, 125(15), 3972-3976.
- [7] Tang, M., Zeng, S., Chen, X., Hu, J., & Du, Y. (2016). An adaptive image steganography using AMBTC compression and interpolation technique. *Optik*, 127(1), 471-477.
- [8] Sharma, K., & Joshi, K. (2017). A novel approach of data hiding technique using interpolation and Braille method. *International Journal of Engineering and Technology*, 9(3S), 320-327.
- [9] Al-Dmour, H., & Al-Ani, A. (2016). A steganography embedding method based on edge identification and XOR coding. *Expert systems with Applications*, 46, 293-306.
- [10] Lu, T. C. (2017). An interpolation-based lossless hiding scheme based on message recoding mechanism. *Optik*, 130, 1377-1396.
- [11] Jung, K. H., & Yoo, K. Y. (2009). Data hiding method using image interpolation. *Computer Standards & Interfaces*, 31(2), 465-470.
- [12] Joshi, K., Yadav, R., & Chawla, G. (2017). An enhanced method for data hiding using 2-bit xor in image steganography. *International Journal of Engineering and Technology*, 9(4), 143.
- [13] Wahed, M. A., & Nyeem, H. (2019). High capacity reversible data hiding with interpolation and adaptive embedding. *PloS one*, 14(3), 1-23.
- [14] Mohammad, A. A., Al-Haj, A., & Farfoura, M. (2019). An improved capacity data hiding technique based on image interpolation. *Multimedia Tools and Applications*, 78, 7181-7205.
- [15] Wahed, M. A., & Nyeem, H. (2019). Reversible data hiding with interpolation and adaptive embedding. *Multimedia Tools and Applications*, 78, 10795-10819.

- [16] Zhang, X., Sun, Z., Tang, Z., Yu, C., & Wang, X. (2017). High capacity data hiding based on interpolated image. *Multimedia Tools and Applications*, 76, 9195-9218.
- [17] Zhong, S., Lu, Y., & Xiong, X. (2023). Reversible data hiding algorithm in encrypted domain based on image interpolation. *IEEE Access*, 11, 108281-108294.
- [18] Fan, M., Zhong, S., & Xiong, X. (2023). Reversible data hiding method for interpolated images based on modulo operation and prediction-error expansion. *IEEE Access*, 11, 27290-27302.