Arabian Journal for Science and Engineering (2021) 46:8441–8456
https://doi.org/10.1007/s13369-021-05529-3

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Efficient Image Reversible Data Hiding Technique Based on Interpolation Optimization

Fatuma Saeid Hassan[1] · **Adnan Gutub**[1]

Received: 11 December 2019 / Accepted: 4 March 2021 / Published online: 20 March 2021

Abstract
Reversible data hiding (RDH) within images is the process of hiding secret data into a cover image in a way that the image can be completely recovered after extracting the embedded data. The performance of the RDH methods depends on making a trade of between the embedding capacity and the image quality. The aim of this paper is to improve RDH method within images using interpolation-based scheme. The proposed interpolation-based RDH (IRDH) scheme improves the embedding capacity of the state-of-the-art Lee and Huang's IRDH scheme with better image quality. The proposed scheme is choosing the existing enhanced neighbor mean interpolation (ENMI) and modified neighbor mean interpolation (MNMI) techniques to scale-up the original image before embedding the secret data. It is then hiding the sensitive bits via smartly benefitting from the principle idea of Lee and Huang's embedding technique. The experiments were conducted on eight standard images to evaluate the performance of the proposed scheme. The investigational results demonstrate that the proposed technique achieves the highest embedding capacity among the most striking similar works of Jung and Yoo's scheme, Lee and Huang's method, and Chang et al.'s approach, winning with attractive image quality.

Keywords Image steganography · Interpolation · Lossless data embedding · Neighbor mean interpolation · Secret data

1 Introduction

Data hiding is an important information security technology in which the secret data is to be hidden into a cover digital media such as image [1], text [2], audio [3], or video [4] in a way that will not raise any suspicion of the existence of secret data [5]. It can be used for copyright protection [6], authentication [7], content-ownership verification [8], and sending patient information [9]. Data hiding techniques can be divided into irreversible and reversible. When data are hidden irreversibly, the receiver can recover only the exact secret message but not the exact original cover media. On the other hands, reversible data hiding (RDH) offers the ability of the exact recovery of both without any distortion. This feature of RDH makes it suitable for medical, military, and political fields where the original media are required to be losslessly restored after extracting the embedded data. Image-based RDH schemes aimed at increasing the amount of the embedded data, decreasing the distortion of the stego-image, and sometimes reducing the computational complexity of the proposed scheme. The inverse relationship between the embedding capacity and the quality of the stego-image produces the challenge to compromise between both in one RDH scheme.

RDH schemes can be classified into three categories: spatial domain, frequency domain, and compression domain. In compression domain schemes, the data are hidden into a compressed version of the image. Some of the used image compression techniques in RDH are vector quantization [10–12], block truncation coding [13–15], and JPEG [16–18]. Since these compression techniques are of lossy type, the reversibility is to use the stego-image to restore the compressed image not the original image. RDH schemes based on frequency domain transform the image into its frequency distribution then modify the frequency coefficients to hide the data. After that, the inverse transformation is performed to convert the output coefficients into a stego-image. Two commonly used transforms are discrete cosine transform [19] and discrete wavelet transform [20]. In spatial domain schemes, the embedding process is applied directly to the image pixels. Spatial domain-based RDH schemes

✉ Adnan Gutub
aagutub@uqu.edu.sa

[1] Computer Engineering Department, Umm Al-Qura University, Makkah, Saudi Arabia

can be of different types: histogram shifting (HS) schemes [21–26], difference expansion (DE) schemes [27–29], pixel-value-ordering (PVO) schemes [30–34], and interpolation-based schemes.

The first HS-based scheme was proposed by Ni et al. [21] in which the range of histogram bins between the peak bin and the zero value bin was shifted by one to vacate a room to embed data. The proposed technique achieved high quality of the stego-image but the embedding capacity depends mainly on the number of pixels having the peak value; so, more improvements were introduced in [22, 23]. Tsai et al. [24] proposed the usage of the prediction-error histogram instead of the direct intensity histogram to take advantage of the similarity between neighborhood pixels within image blocks. The residual image that has the prediction-error values was generated by dividing the image into equal-sized blocks then from each block the difference value between each pixel and the center pixel was calculated. The histogram of this residual image was generated, and multiple pairs of peak and zero value bins were extracted to embed data. This technique improved both the hiding capacity and visual quality. Pan et al. [25] generated a more concentrated prediction-error histogram by designing four kinds of predictors depending on the gradient feature of an image pixel: horizontal gradient, vertical gradient, rhombus gradient, and plane gradient. A pixel was employed for embedding, by shifting the prediction-error histogram, if it belongs to flat, semi-flat, or normal image block depending on a defined complexity measurement. The sharper distribution of the generated histogram and the careful selection of the embedding parameters improved the performance of the proposed technique. Wang et al. [26] imitated the HS technique by dividing the image intensity values into equal segments then extracting the peak value from the histogram of each segment. For data embedding, each pixel having intensity value of a segment-peak was changed into another intensity value within the segment depending on the value of the secret bits. The utilization of multiple peak points and the centralized change within the segment improved the embedding capacity and reduced the image distortion.

In 2003, Tian [27] proposed a DE-based scheme that exploited the correlation between adjacent pixels. For each pair of adjacent pixels, its difference was multiplied by 2, and then, this expanded difference was used to embed one secret bit. The stego-value of the two pixels was computed from their average value and the new difference value. The proposed technique achieved high stego-image quality, but the hiding was limited to only one bit per pixel-pair. Liu et al. [28] proposed an improved DE-based technique that embedded data adaptively by expanding the prediction-error instead of expanding the difference. They divided the image into equal-sized blocks then for each block they applied a bilinear interpolation kernel, using four reference pixels, to predict the values of the pixels. The difference between the original value of each pixel and its predicted value generated the prediction-error that was expanded to embed data. The number of embedded bits in each pixel in a block can be 0, 1, or 2 depending on the group that the block belongs to. Caciula et al. [29] proposed a multi-layer prediction-error expansion (PEE) technique that increased the embedding rate to more than 3 bits per pixel.

The first PVO-based RDH scheme was proposed by Li et al. [30] in which the technique depended on sorting the pixel values in each block to embed data into the maximum and the minimum values. For each block, the difference between the value of the maximum pixel and the second largest pixel was used as the prediction-error P_{max} of that block. Then, the histogram of the P_{max} of all the blocks was generated to embed data into bin 1 by shifting the histogram to the positive side. To increase the embedding capacity, the difference between the minimum and the second smallest pixels was used as the prediction-error P_{min} of each image block, and then, the histogram of P_{min} was generated to embed data into bin -1 by shifting the histogram to the negative side. This technique ensures that only the maximum and the minimum value of an image block is changed by one but the other pixels still the same, which reduced the distortion of the stego-image. Li et al. [30] did not utilize bin 0 in the prediction-error histogram which means that the blocks having the same value of the maximum/minimum pixel and the second largest/smallest pixel were not used for data embedding. Peng et al. [31] proposed a technique that addressed this problem by taking into consideration the locations of the maximum/minimum pixel and the second largest/smallest pixel. Many more improved PVO-based RDH schemes were proposed in the literature [32–34].

Interpolation-based RDH schemes increase the chance for embedding high payload by generating new pixels in the image. At first, an image interpolation method is used to scale-up the image to a larger size, and then, the newly generated pixels are used for data embedding while maintaining the values of the original pixels. Image interpolation techniques use the original pixels as reference values to estimate the values of the interpolated pixels. The first challenge in IRDH schemes is to select the interpolation technique that produces an interpolated image of high visual quality. Some of the used interpolation techniques in IRDH schemes are neighbor mean interpolation (NMI), interpolation by neighboring pixels (INP), enhanced neighbor mean interpolation (ENMI), modified neighbor mean interpolation (MNMI), and parabolic interpolation (PI). The second challenge is to develop an embedding technique that compromises between the hiding capacity and the stego-image quality. A detailed review of some of the existing interpolation techniques and IRDH schemes is provided in Sect. 2.

The aim of this paper is to propose an IRDH scheme that improves the embedding capacity of Lee and Huang's IRDH scheme with better image quality by combining the best interpolation technique in the literature along with the best embedding technique in the literature. The rest of this paper is organized as follows. Related works are discussed in detail in Sect. 2. Section 3 describes the proposed algorithm. Then, the experimental results are discussed in details in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 Related Works

This section reviews some of the available interpolation techniques as well as some of the proposed interpolation-based RDH schemes in the literature. Figure 1 represents the general steps of the IRDH schemes which usually start by scaling-down the input image of high resolution to the quarter of its size to produce a low-resolution image [35–37]. The resultant image is called original image which is then scaled-up to four-times of its size using an interpolation method. The resultant image is called either interpolated image or cover image. This interpolated image contains two types of pixels: the original pixels, which are kept unchanged, and the interpolated pixels, which are employed for data embedding. The final image is the stego-image that contains the secret data. Using the stego-image, we can separately recover the embedded data and the original image by performing data extraction and scaling-down, respectively. It is of important to mention that the original image is used for measuring the performance of the interpolation technique and the data embedding technique in terms of the quality of the produced image. Section 2.1 explained some of the interpolation methods followed by exploring some of the existing IRDH schemes.

2.1 Review of Image Interpolation Techniques

In 2009, Jung and Yoo [35] proposed the NMI interpolation technique to be used in the first IRDH scheme. Using the scaling factor 2, each image block of size 2×2 was enlarged using NMI to be of size 3×3. As shown in the example in Fig. 2, the NMI technique kept the original pixels as they are but calculated the values of the interpolated pixels using the mean of the neighbors. The interpolated pixels in the horizontal direction, as C(0,1), and vertical direction, as C(1,0), were calculated by the mean value of two nearest original pixels. The interpolated pixels in

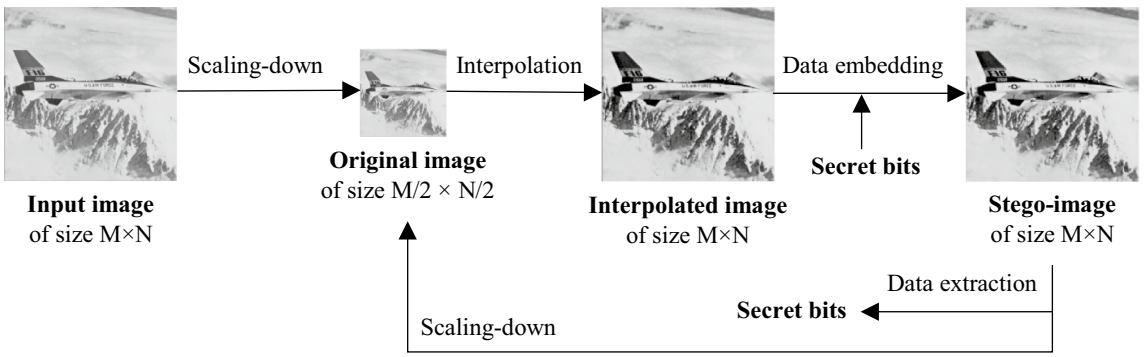


Fig. 1 The general steps of the IRDH schemes

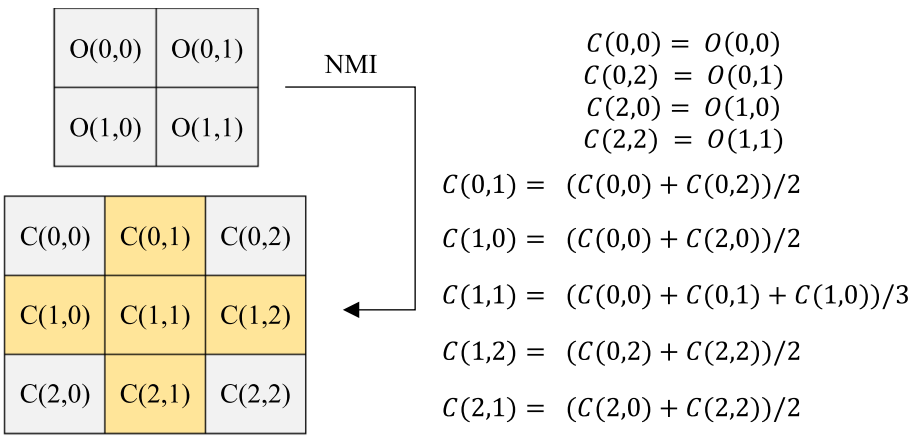


Fig. 2 Example of neighbor mean interpolation technique

the diagonal direction, as C(1,1), were assigned a value of the mean of one original pixel and two already-calculated interpolated pixels. The NMI technique is computationally simple, and it produces better image quality than the two known interpolation techniques: nearest-neighbor interpolation (NNI) and bilinear interpolation (BI).

To enhance the NMI method, Lee and Huang [38] proposed the INP interpolation method which is represented in the example in Fig. 3. The interpolated pixels in the horizontal and vertical directions were calculated from two nearest original pixels by giving more weight for one of them to generate their weighted mean value, which is the value of the interpolated pixel. The interpolated pixels in the diagonal direction were given a value of the mean of one horizontal and one vertical interpolated pixels. The INP technique outperforms the NMI technique in terms of the image quality.

Chang et al. [37] proposed the ENMI interpolation method as an improvement to NMI. ENMI achieved less image distortion than NMI by modifying the values of the diagonal interpolated pixels to be of the mean value of four nearest reference pixels. Figure 4 displays an example of ENMI.

In 2017, Malik et al. [39] proposed the modified neighbor mean interpolation technique to improve the image quality of the NMI method by giving more weight for the reference pixels while estimating the horizontal and vertical interpolated pixels. As represented in Fig. 5, the MNMI method started by calculating the interpolated pixels in the diagonal direction using the mean value of four nearest reference pixels. Then, the interpolated pixels in the horizontal and vertical directions were given a value of the weighted mean of two nearest reference pixels and one neighbor diagonal interpolated pixel.

In the same year of [39], Zhang et al. [36] proposed the parabolic interpolation technique to be used in their IRDH scheme. The PI method divided the interpolated image into 1×5 overlapping blocks of five adjacent pixels along three directions: horizontal, vertical, and diagonal. Each interpolation block consisted of three original pixels that interspersed with two interpolated pixels. The values of the three original pixels and their positions within the block were used to formulate a parabolic equation. The initial values of the two interpolated pixels were computed by substituting their positions into the produced parabolic equation. Due to the overlapped blocks, an interpolated pixel can have many

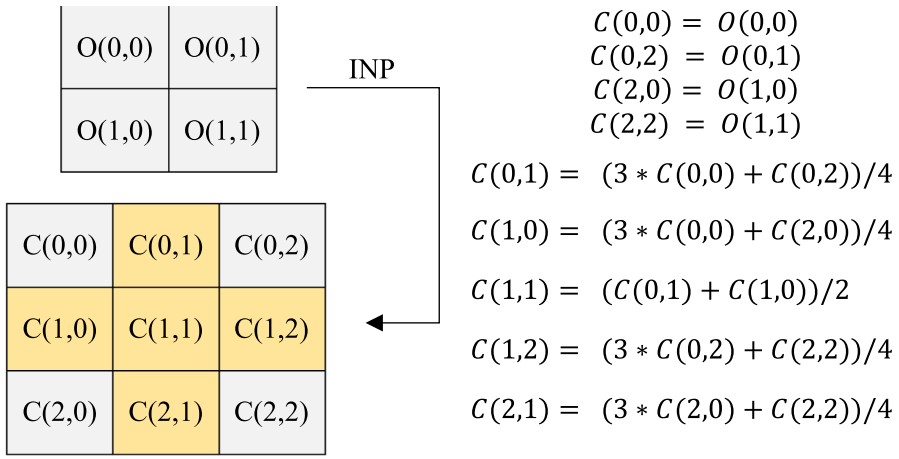


Fig. 3 Example of interpolation by neighboring pixels technique

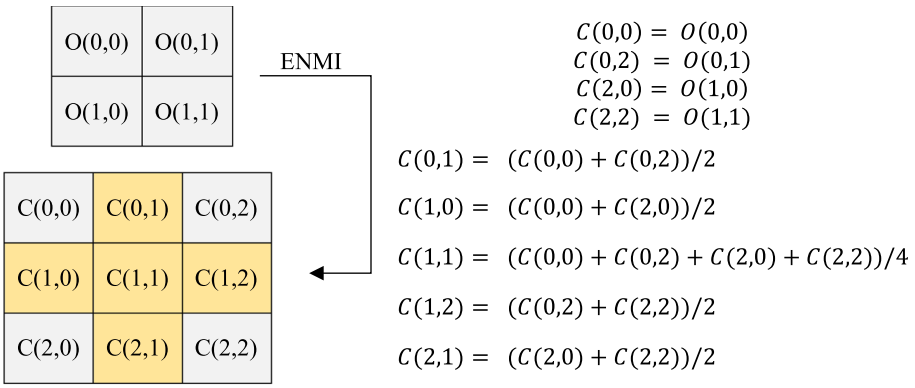


Fig. 4 Example of enhanced neighbor mean interpolation technique

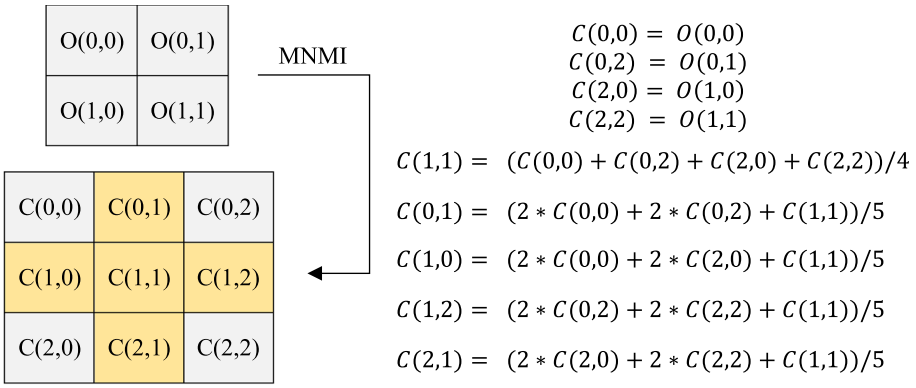


Fig. 5 Example of modified neighbor mean interpolation technique

initial values so the average of these values was taken along with the weighted average of the nearest original neighbors to calculate the final value of the interpolated pixel. The proposed PI technique produces an interpolated image of higher quality than NMI and INP. In 2019, Shaik et al. [40] improved the one-dimensional (1D) PI technique that was proposed by Zhang et al. [36] by proposing two-dimensional (2D) parabolic interpolation. The proposed 2D interpolation divided the interpolated image into 5×5 overlapping blocks and each block consisted of nine original pixels and sixteen interpolated pixels. A 2D parabolic equation was formed from the nine original pixels to be used in the estimation of the values of the new pixels. Due to the contribution of more reference pixels in calculating the interpolated pixels, the visual quality of the 2D parabolic interpolation outperformed the 1D interpolation.

In 2020, Malik et al. [41] proposed an interpolation technique that outperforms both the 1D [36] and 2D [40] parabolic interpolation techniques. They called it MNMI but we will rename it here to MNMI2 to not confuse with the MNMI [39]. MNMI2 calculated the new pixels in the horizontal and vertical directions by the weighted mean of four original pixels by giving more weight for the two nearest ones. The new pixels in the diagonal direction were computed as same as MNMI [39].

2.2 Review of IRDH Schemes

The first IRDH scheme was proposed by Jung and Yoo [35] in 2009. They proposed the NMI interpolation method to scale-up each 2×2 block in the original image to 3×3 block in the interpolated image. This image of a bigger size was used for hiding data in 2×2 non-overlapping blocks in which each block has one reference pixel and three interpolated pixels. Three steps were followed to hide data into each interpolated pixel: determining the number of secret bits n to be hidden by using the difference D between the interpolated pixel and the reference pixel, taking the decimal value d of the n bits, and finally forming the stego-value of the pixel by adding d to its value. Since the amount of data to be hidden in the pixel depended on the difference D, maximizing this difference will increase the embedding capacity, as done by Lee and Huang's scheme [38].

Lee and Huang [38] employed the INP technique for image interpolation. Then, they embedded the data into the interpolated pixels by maximizing the difference value. For each interpolated pixel, the difference between its value and the maximum value of the four surrounding original pixels determined the number of bits to be embedded in the pixel. Then, the pixel's new value in the stego-image was calculated by adding the decimal value of the embeddable bits to the pixel. The reported results of the scheme were better than Jung and Yoo's scheme [35] in terms of the embedding capacity and image quality.

To further improve Jung and Yoo's scheme, Chang et al. [37] proposed an IRDH scheme that embedded data in the cover image in two stages. The cover image was generated from the original image using the ENMI interpolation technique. For embedding data into the interpolated pixels, the first stage started by computing the difference D between each interpolated pixel and its corresponding pixel in the input image. This D determined the number of bits n. The initial stego-image S_1 was produced by adding or subtracting the decimal value of the n bits to/from the pixel to embed the bits in a way that reduced the distortion between the stego-image and the input image, i.e., increasing the image quality. The second embedding stage employed the histogram shifting technique on a new difference image that was produced using the image S_1. This second stage improved both the hiding capacity and the visual quality of the scheme.

Malik et al. [39] built the cover image by applying the MNMI interpolation technique to the original image. In the data embedding phase, the cover image was divided into 2×2 non-overlapping blocks. In each block, the horizontal and vertical interpolated pixels were used for embedding, while the diagonal interpolated pixel was changed accordingly. This change in the diagonal interpolated pixel eliminated the need for recalculating the interpolated pixels in

the data extraction phase, unlike the above three mentioned schemes [35, 37, 38]. The embedding in the two pixels in a block started by calculating the difference between the pixels and the diagonal interpolated pixel to be used for calculating the number of embeddable bits in each pixel. Then, the decimal value of the embeddable bits, a table called 'range table' and new difference values were used in somehow to calculate the stego-values of the three interpolated pixels. The reported experiments of the scheme showed higher hiding capacity and less image distortion than the two schemes in [35, 37].

In 2020, the same authors of the above-mentioned work used the MNMI2 interpolation technique in their proposed IRDH scheme [41]. The scheme encrypted the secret message before the embedding to increase the security. The embedding technique depended on the intensity value of the pixels. For each interpolated pixel, if its intensity value was below 16 or above 191, then four secret bits will replace its four least significant bits (LSBs). If its intensity value was in the range of 16 to 31, three secret bits will replace its three LSBs. Otherwise, two secret bits will replace its two LSBs. The scheme revealed better results of the embedding capacity and image quality than the techniques in [35, 36, 40].

From the above literature review of the IRDH schemes, we can observe the influence of the used interpolation technique on the performance of the IRDH scheme. When the interpolation technique produces a cover image of high quality, this can help more in reducing the distortion of the final stego-image. In this paper, we propose an IRDH scheme that improves the image quality and the hiding capacity of Lee and Huang's scheme [38] using the ENMI [37] and MNMI [39] interpolation techniques to produce the interpolated images. We chose these two interpolation techniques because they produced cover images of better image quality, i.e., higher value of peak signal to noise ratio (PSNR), as principally tested within multi-image secret hiding [1], as we evidenced in Table 1. We embed the data in the cover images using the proposed data embedding technique by Lee and Huang [38] as it achieved the best embedding capacity as shown in Table 2. The proposed IRDH scheme achieved the highest embedding capacity among Jung and Yoo's scheme [35], Lee and Huang's scheme [38], and Chang et al.'s scheme [37] with acceptable visual quality. The following section describes the proposed IRDH scheme.

3 The Proposed IRDH Scheme

In this section, we present our proposed high capacity IRDH scheme that takes advantage of some of the state-of-the-art techniques mentioned in Sect. 2. The scheme followed three stages: image interpolation, data embedding, and data extraction.

Figure 6 represents the general framework of the proposed IRDH scheme, which is completely different than stego hiding within texts [2] or hiding within audio files [3]. This IRDH work is a new direction of research outside the sophisticated hiding within videos [4] or watermarking [6] as well as counting-based stego schemes [5]. Although the IRDH work application can be considered common to the text sensitive authentication [8] and IoT security [9], its process is found to be different. The proposed IRDH first selects the reference pixels, i.e., from the input image of size 512×512 as by selecting one column/row from every two consecutive columns/rows. The result is the original image of size 256×256, which is then enlarged by interpolation to produce the interpolated image of size 512×512. For our interpolation, we utilize the ENMI [37], and MNMI [39] intermission techniques as they were tested to produce a better quality of interpolated images than NMI [35], INP [38] and MNMI2 [41], see Table 1. Figures 4 and 5 present examples of the ENMI and MNMI, respectively. A numerical example of the interpolation using the ENMI and MNMI techniques is provided in Sect. 3.1. The resultant interpolated image is employed in the embedding phase to

Table 1 Comparison of PSNR values in (dB) of the five interpolation techniques

Test Image	NMI	INP	ENMI	MNMI	MNMI2
Baboon	22.23	21.72	22.69	22.70	22.44
Airplane	28.93	28.04	30.12	29.98	29.03
Boat	29.22	28.30	30.53	30.39	29.32
House	27.25	26.50	28.20	28.08	27.19
Peppers	30.07	28.98	31.70	31.60	30.61
Goldhill	29.84	29.11	30.69	30.64	29.94
Man	27.46	26.67	28.33	28.27	27.54
Bridge	25.65	24.71	26.84	26.71	25.77
Average	27.58	26.75	28.64	28.55	27.73

Table 2 Comparison of embedding capacity values in (bits) of the three IRDH schemes

Test Image	Jung and Yoo's scheme [35]	Lee and Huang's scheme [38]	Chang et al.'s scheme [37]
Baboon	428,240	640,938	382,841
Airplane	177,830	342,520	184,280
Boat	216,258	384,669	219,442
House	244,607	411,072	225,602
Peppers	197,605	388,981	213,837
Goldhill	251,003	443,245	240,232
Man	285,449	488,169	261,444
Bridge	332,834	516,058	267,138
Average	266,728	451,956	249,352

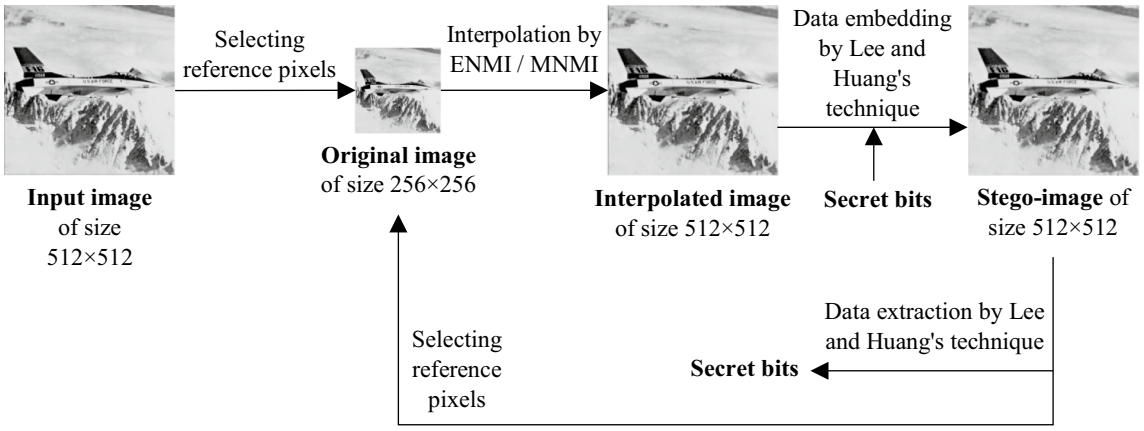


Fig. 6 The general framework of the proposed IRDH scheme

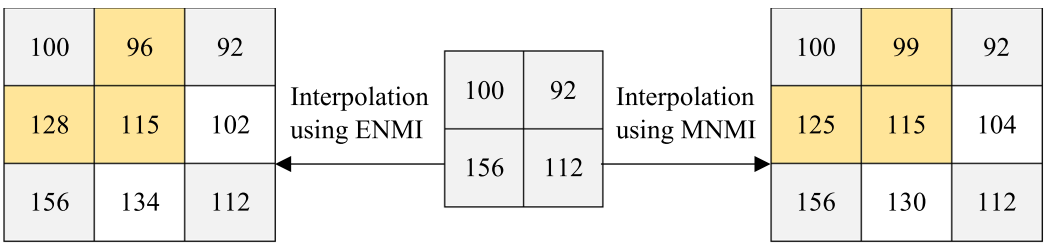
embed the secret bits and generate the stego-image. For data embedding, we use the embedding method that was proposed by Lee and Huang [38] as it achieved the best embedding capacity, see Table 2. The detailed description of the embedding phase is given in Sect. 3.2. A practical example of the embedding method is given in Sect. 3.3. Finally, the extraction phase extracts the secret bits from the stego-image using the extraction method of Lee and Huang's scheme [38]. The extraction phase is explained in Sect. 3.4 and illustrated by an example in Sect. 3.5. For image recovery, the reference pixels are selected from the stego-image in the same way as done in producing the original image.

3.1 Illustrative Example of the Interpolation Phase

A minimal numerical example of the interpolation phase is presented in Fig. 7. For 2×2 original image as in Fig. 7a, the interpolated image is built by firstly adding a new empty row and column between two adjacent original rows and columns, respectively, so the result is 3×3 cover image. The values of the interpolated pixels in the new rows and columns are calculated using ENMI, Fig. 7b, and MNMI, Fig. 7c. In the given example, the four gray pixels in the cover images are original pixels which are C(0,0) = 100, C(0,2) = 92, C(2,0) = 156 and C(2,2) = 112.

Using ENMI, for each 2×2 inner block in the cover image, the value of the horizontal interpolated pixel, C(0,1), and the vertical interpolated pixel, C(1,0), is computed by the mean of the two nearest original pixels. So, C(0,1) = (100 + 92)/2 = 96 and C(1,0) = (100 + 156)/2 = 128. However, the value of the diagonal interpolated pixel, C(1,1), is computed by the mean of the four nearest original pixels. So, C(1,1) = (100 + 92 + 156 + 112)/4 = 115.

Using MNMI, for each 2×2 inner block in the cover image, the value of the diagonal interpolated pixel, C(1,1), is computed by the mean of the four nearest original pixels. So, C(1,1) = (100 + 92 + 156 + 112)/4 = 115. Then, the value of the horizontal and vertical interpolated pixels is computed



$$C(0,1) = (100 + 92)/2 = 96$$
$$C(1,0) = (100 + 156)/2 = 128$$
$$C(1,1) = 100 + 92 + 156 + 112)/4 = 115$$

(b)

$$C(1,1) = (100 + 92 + 156 + 112)/4 = 115$$
$$C(0,1) = (2 * 100 + 2 * 92 + 115)/5 = 99$$
$$C(1,0) = (2 * 100 + 2 * 156 + 115)/5 = 125$$

(a) **(c)**

Fig. 7 A numerical example of the interpolation **a** The input image, **b** ENMI-cover image, and **c** MNMI-cover image

by the weighted mean of two nearest original pixels and one nearest diagonal interpolated pixel. So, C(0,1) = (2 × 100 + 2 × 92 + 115)/5 = 99 and C(1,0) = (2 × 100 + 2 × 156 + 115) /5 = 125.

3.2 Data Embedding Phase

The result of the previous interpolation phase is an interpolated image that is used for embedding the secret message to generate the final stego-image. The components of the embedding phase are shown in Fig. 8 in which the interpolated pixels in the cover image are used for embedding, whereas the original pixels are kept untouched.

In our IRDH scheme, we use the embedding method that was proposed by Lee and Huang [38] because it attained the best embedding capacity among Jung and Yoo's scheme [35] and Chang et al.'s scheme [37] as shown in Table 2. The reason of the high capacity of Lee and Huang's scheme is that their embedding technique maximizes the difference by using the maximum of four original pixels. The detailed explanation of the embedding technique is given below.

The technique used 3 × 3 overlapping blocks for data embedding. When cover pixels and stego-pixels are given as Fig. 8, the embedding process is started by extracting the maximum value M of the four original pixels in the corners as in Eq. 1.

$$M = \max(C(0,0), C(0,2), C(2,0), C(2,2)) \tag{1}$$

The three interpolated pixels within the inner 2 × 2 block are used for embedding which are C(0,1), C(1,0), and C(1,1). For each of these three pixels, a value D_t, (t = 1, 2, 3), is calculated which is the difference between the pixel value and the maximum value M. So, D_1, D_2, and D_3 are calculated as in Eq. 2.

$$\begin{aligned} D_1 &= M - C(0,1) \\ D_2 &= M - C(1,0) \\ D_3 &= M - C(1,1) \end{aligned} \tag{2}$$

After that, for each of the three interpolated pixels, the $\log_2$ of the difference D_t is used to calculate the number of embeddable bits n_t in the pixel. So, n_1, n_2, and n_3 are calculated as in Eq. 3.

$$n_t = floor(\log_2 |D_t|), t = 1, 2, 3 \tag{3}$$

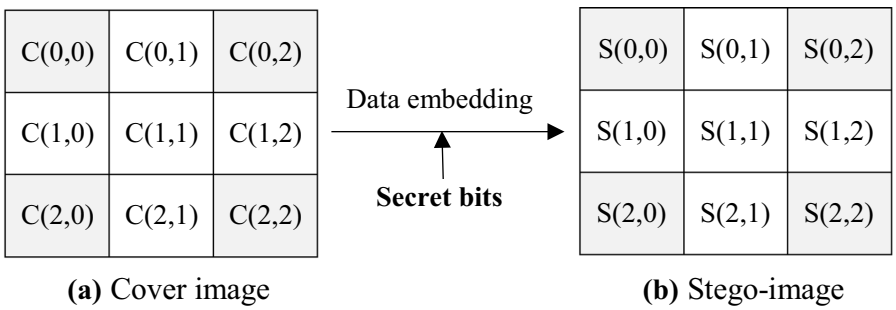
Then, for each of the three pixels, the decimal value d_t of the n_t embeddable bits is added to the pixel to form its new value in the final stego-image. So, the values of S(0,1), S(1,0), and S(1,1) are calculated using Eq. 4.

$$\begin{aligned} S(0,1) &= C(0,1) + d_1 \\ S(1,0) &= C(1,0) + d_2 \\ S(1,1) &= C(1,1) + d_3 \end{aligned} \tag{4}$$

3.3 Practical Example of the Embedding Method

Using the cover image in Fig. 9 (a, left), the three pixels in the inner 2 × 2 block are used for embedding which are: C(0,1) = 96, C(1,0) = 128 and C(1,1) = 115. Assuming that the secret bit stream is $(00,111,001,101,000)_2$, the process of the embedding is as follows.

At first, the M value is calculated by the maximum of the four surrounding original pixels, C(0,0) = 100, C(0,2) = 92, C(2,0) = 156 and C(2,2) = 112, so M = 156. Then, the difference values D_1, D_2, and D_3 are computed using Eq. 2 as D_1 = 156–96 = 60, D_2 = 156–128 = 28, and D_3 = 156–115 = 41. The number of embeddable bits in each of the three pixels are $n_1 = \text{floor}(\log_2(|60|)) = 5$, $n_2 = \text{floor}(\log_2(|28|)) = 4$ and $n_3 = \text{floor}(\log_2(|41|)) = 5$. So, $d_1 = (7)_{10}$ which is the decimal value of the first five secret bits $(00,111)_2$, $d_2 = (3)_{10}$ which is the decimal value of the next four secret bits $(0011)_2$ and $d_3 = (8)_{10}$ which is the decimal value of the next five secret bits $(01,000)_2$. At last, the new values of the three bits are calculated using Eq. 4 to form the values of the corresponding stego-pixels, S(0,1) = 96 + 7 = 103, S(1,0) = 128 + 3 = 131 and S(1,1) = 115 + 8 = 123 as shown in Fig. 9 (a, right). Figure 9a represents the cover image produced by the ENMI interpolation technique and its corresponding stego-image. Figure 9b represents the cover image produced by the MNMI and its corresponding stego-image.



(a) Cover image **(b)** Stego-image

Fig. 8 Components of the embedding phase

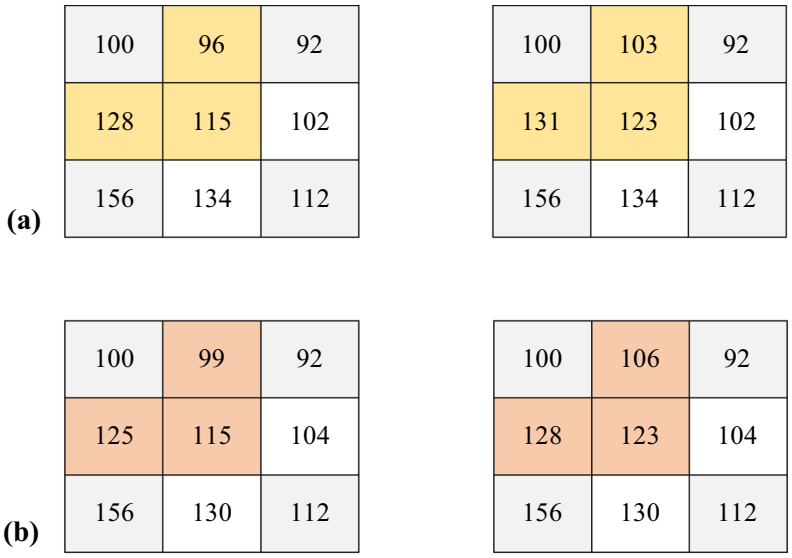


Fig. 9 Example of the data embedding phase **a** ENMI-cover image (left) and its corresponding stego-image (right) and **b** MNMI-cover image (left) and its corresponding stego-image (left)

3.4 Data Extraction Phase

The result of the previous embedding phase is a stego-image that is used for extracting the secret message and recovering the original image. When stego-pixels are given as Fig. 8b, the gray pixels are original pixels that do not contain embedded bits as the rest of the pixels. In our IRDH scheme, we use the extraction method of Lee and Huang's scheme [38]. The extraction method is started by using the original pixels to recalculate the cover image, Fig. 8a, using either the ENMI interpolation technique or the MNMI technique depending on the used interpolation technique in the interpolation phase. For each 3×3 overlapping block in the cover image, the maximum value M of the four original pixels in the corners is calculated as in Eq. 1. To calculate the number of the embedded bits in the three pixels in the inner 2×2 block of the corresponding stego-block, we need to calculate D_1, D_2 and D_3 using Eq. 2 then calculate n_1, n_2 and n_3 using Eq. 3. At last, the decimal value of the embedded bits in each of the three pixels in the stego-block is calculated using Eq. 5. Then, each decimal value d_t, ($t=1, 2, 3$), is converted to binary value of n_t digits. Then, the bits are concatenated together to form the secret bit stream. To recover the original image, the stego-image is down-sampled by removing the new rows and columns that were inserted in the interpolation phase.

$$\begin{aligned} d_1 &= S(0,1) - \mathrm{C}(0,1) \\ d_2 &= S(1,0) - \mathrm{C}(1,0) \\ d_3 &= S(1,1) - \mathrm{C}(1,1) \end{aligned} \tag{5}$$

3.5 Practical Example of the Extraction Method

Using the stego-image in Fig. 9 (a, right), the three pixels in the inner 2×2 block contain embedded data. The pixels are S(0,1)=103, S(1,0)=131 and S(1,1)=123. The four gray pixels, S(0,0)=100, S(0,2)=92, S(2,0)=156 and S(2,2)=112, are original pixels and they are used to recalculate the cover image, Fig. 9 (a, left), using the ENMI interpolation technique. The four gray pixels in the stego-image are as same as the four gray pixels in the cover image. From the cover image, the maximum value of the original pixels is extracted as $M=156$. Then, the three pixels C(0,1)=96, C(1,0)=128 and C(1,1)=115 are used to calculate the difference values D_1, D_2, and D_3 using Eq. 2 as $D_1=156–96=60$, $D_2=156–128=28$, and $D_3=156–115=41$. The number of embedded bits in each of the corresponding three pixels in the stego-image is $n_1=\mathrm{floor}(\log_2(|60|))=5$, $n_2=\mathrm{floor}(\log_2(|28|))=4$ and $n_3=\mathrm{floor}(\log_2(|41|))=5$. Using Eq. 5, the decimal values d_1, d_2 and d_3 are computed as $d_1=103–96=(7)_{10}$, $d_2=131–128=(3)_{10}$ and $d_3=123–115=(8)_{10}$. Every decimal value d_t, (t=1, 2, 3), is converted to binary value of n_t digits. Therefore, from d_1, the secret bits are $(00,111)_2$, from d_2, the secret bits are $(0011)_2$ and from d_3, the secret bits are $(01,000)_2$. By concatenation, the secret bit stream is formed as $(00,111,001,101,000)_2$. To recover the original image, the stego-image is down-sampled by removing the new rows and columns that were inserted in the interpolation phase so the resultant image is the original image in Fig. 7a.

4 Comparisons and Analysis

This section discusses the experimental results and comparisons of our proposed scheme and some of the reviewed IRDH schemes in Sect. 2 [35, 37–39]. The performance of the schemes is evaluated based on the quality of the produced images, the maximum number of bits that can be embedded by each scheme, and the running time. In Sect. 4.1, we describe the experimental study including the used software and hardware platform and the used evaluation parameters. Section 4.2 and Sect. 4.3 represent comparative analysis of the results of the image interpolation phase and the data embedding phase, respectively.

4.1 Experimental Study

The work in this study was implemented using MATLAB R2018a running on a computer having a 2.9 GHz Intel core i7 CPU and 8 GB memory. We applied our proposed scheme and the existing schemes [35, 37–39] on eight grayscale test images of size 512×512, see Fig. 10. These images are from CVG-UGR image database to be similar in testing benchmark images to the work in [42], using these CVG-UGR images that are openly provided for academic testing and experimentation purposes. These images has been also studied in [43] showing attractive remarks worth making them our bench mark here too.

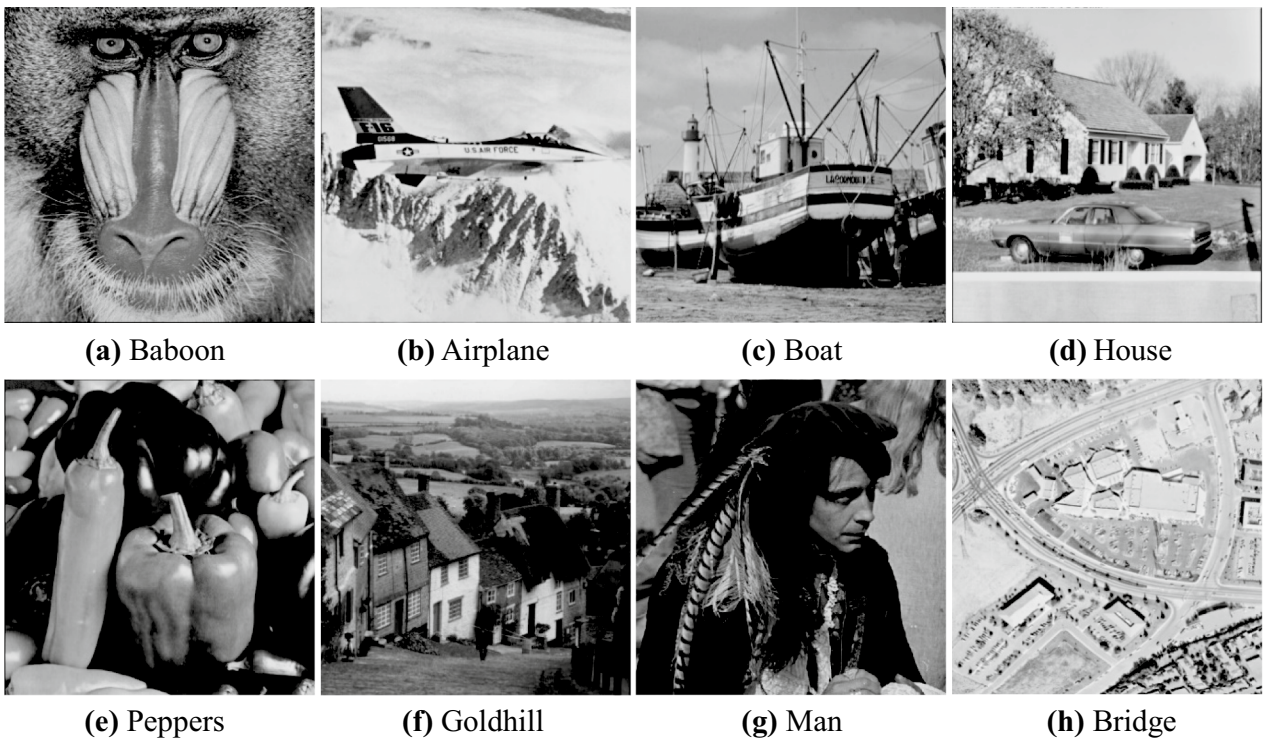
We use the peak signal to noise ratio PSNR to evaluate the quality of the interpolated image or the stego-image with respect to the input image, all of size 512×512, as the following equation.

$$PSNR = 10 \times \log_{10} \frac{255^2}{\frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i,j)-C(i,j))^2}{M \times N}} \quad (6)$$

where the I(i,j) and C(i,j) are the intensity value of the pixels exist in position (i,j) in the input image I and cover image/stego-image C, respectively, and both images are of size M×N. The larger PSNR value, the higher visual quality and less distortion of the interpolated image/stego-image. For measuring the performance of the embedding, we calculate the number of bits that can be hidden by the scheme; this is known as embedding capacity. Finally, we measure the running time in seconds to evaluate the computational complexity. The less running time, the less computational complexity.

4.2 Analysis of the Interpolation Results

In this section, we discuss the experimental results of the five existing interpolation techniques: NMI [35], INP [38],



(a) Baboon **(b)** Airplane **(c)** Boat **(d)** House

(e) Peppers **(f)** Goldhill **(g)** Man **(h)** Bridge

Fig. 10 The eight standard test images [42, 43]

ENMI [37], MNMI [39], and MNMI2 [41]. Our proposed scheme started by generating the original images of size 256×256 from the provided test images in Fig. 10 by scaling-down the images. After that, the five interpolation techniques are applied to the original images to produce the cover images. We measure the performance of the interpolation techniques in terms of the visual quality of the produced cover images using PSNR (dB) as illustrated in Table 1. It is evident from Table 1 that the sequence of the techniques from the highest average PSNR to the lowest is: ENMI, MNMI, MNMI2, NMI, and finally INP.

The best three techniques, i.e., ENMI, MNMI, and MNMI2, calculated the diagonal interpolated pixels by the average of the four nearest original pixels, while the other two techniques, i.e., NMI and INP, involved some of the already-computed interpolated pixels in computing the diagonal pixels. This involvement of the already-computed interpolated pixels in the calculation causes the reduction of the image quality of the two methods. The ENMI is the best among the MNMI and MNMI2 methods because it considered two points while calculating the horizontal and vertical interpolated pixels. The first point was to avoid the involvement of any of the already-computed interpolated pixels in the calculation, unlike the MNMI method. The second point was to involve only the very close original pixels in the calculation, unlike the MNMI2 method. In our proposed IRDH scheme, we use the ENMI and MNMI methods in the interpolation phase because they scored the best image quality.

4.3 Analysis of the Data Embedding Results

This section discusses the experimental results of the proposed scheme and the existing schemes [35, 37–39] after embedding the secret data into the cover images. The experiments were conducted on a secret bit stream that was generated by the MATLAB software using a pseudorandom number generator different than the cybercrime vulnerabilities presented in [44] and securing data via cryptography and steganography of [45] as well as the cyber-attacks prediction [46], LSB image security [47], and detailed medical security [47].

In this proposed scheme, the cover images that were produced by the ENMI and MNMI interpolation techniques are used here to embed the bits using the embedding method of Lee and Huang's scheme [38]. For comparison, we also embed the secret bits using the other schemes that were proposed by Jung and Yoo [35], Lee and Huang [38], Chang et al. [37], and Malik et al. [39]. When we implemented the Malik et al.'s embedding scheme [39], we noticed that some of the pixels in the stego-image may have intensity values out of the range, i.e., below 0 or above 255, so we do not evaluate here the embedding results of this scheme.

We evaluate the performance of the schemes in terms of the visual quality of the final stego-images, using PSNR (dB), and the embedding capacity, in (bits). Table 3 provides a comparison of the PSNR values and the embedding capacity values of each of the following schemes: Jung and Yoo's scheme [35], Lee and Huang's scheme [38], and Chang

Table 3 Comparison of PSNR values in (dB) and embedding capacity values in (bits) of the five IRDH schemes

Test Image	Metric	Jung and Yoo's scheme [35]	Lee and Huang's scheme [38]	Chang et al.'s scheme [37]	Proposed with ENMI	Proposed with MNMI
Baboon	PSNR	21.93	21.20	29.32	21.53	21.51
	Capacity	428,240	640,938	382,841	665,849	683,032
Airplane	PSNR	28.60	27.38	36.19	28.11	27.98
	Capacity	177,830	342,520	184,280	347,062	363,959
Boat	PSNR	28.73	27.41	36.88	28.13	27.97
	Capacity	216,258	384,669	219,442	398,200	418,003
House	PSNR	27.06	26.09	34.68	26.67	26.55
	Capacity	244,607	411,072	225,602	425,779	445,098
Peppers	PSNR	29.91	28.66	38.05	29.56	29.43
	Capacity	197,605	388,981	213,837	398,262	415,125
Goldhill	PSNR	29.44	28.35	37.13	28.90	28.80
	Capacity	251,003	443,245	240,232	462,914	482,572
Man	PSNR	26.94	25.66	34.91	26.12	26.02
	Capacity	285,449	488,169	261,444	506,671	523,938
Bridge	PSNR	25.13	23.75	33.41	24.46	24.29
	Capacity	332,834	516,058	267,138	537,345	559,105
Average	PSNR	27.22	26.06	35.07	26.69	26.57
	Capacity	266,728	451,956	249,352	467,760	486,354

et al.'s scheme [37], as well as our proposed scheme with ENMI, and our proposed scheme with MNMI.

From Table 3, it is obvious that our proposed scheme by either using ENMI or MNMI achieved the highest embedding capacity among the other three schemes [35, 37, 38] in all of the eight images. In addition, our scheme achieved higher PSNR values than Lee and Huang's scheme in all of the eight images. However, the PSNR values of the other two schemes [35, 37] are better than our scheme. Let us discuss this observation as followed.

Chang et al.'s scheme [37] embedded the data into the cover image by the help of the difference value between the interpolated pixels and their corresponding pixels in the input image. This usage of the input image in data embedding made the stego-image closer to the input image, i.e., higher image quality. However, our scheme used the input image only for evaluating the image quality not for hiding data because practically the input image does not always exists. Jung and Yoo's scheme [35] achieved higher image quality than ours because our scheme embeds at minimum 169,232 more bits and maximum 254,792 more bits than Jung and Yoo's scheme so this huge increasing in the embedding capacity reduces a bit the image quality.

On the other hands, our scheme with MNMI has higher embedding capacity than our scheme with ENMI but the later achieved a little bit higher image quality than the

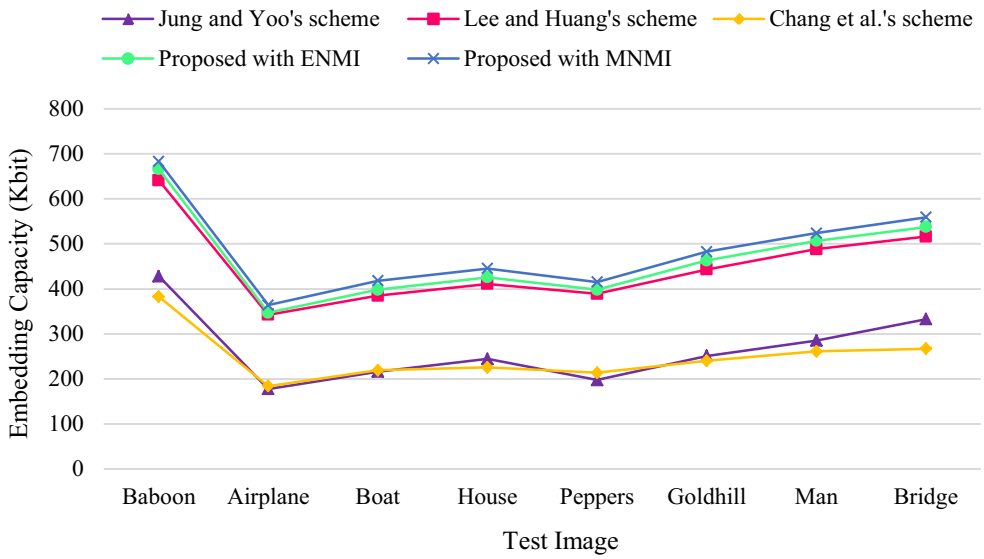


Fig. 11 Comparison of embedding capacity values of the five IRDH schemes

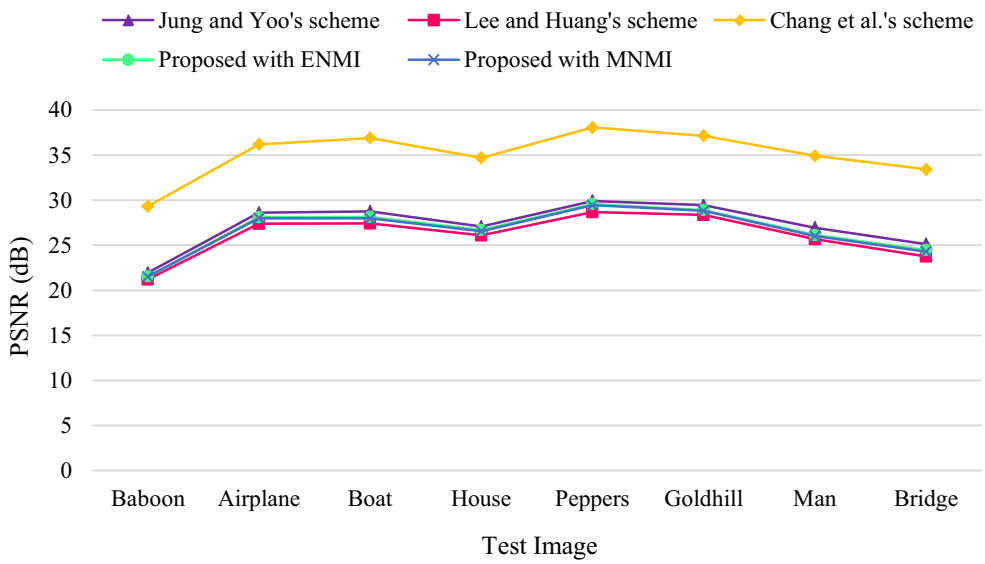
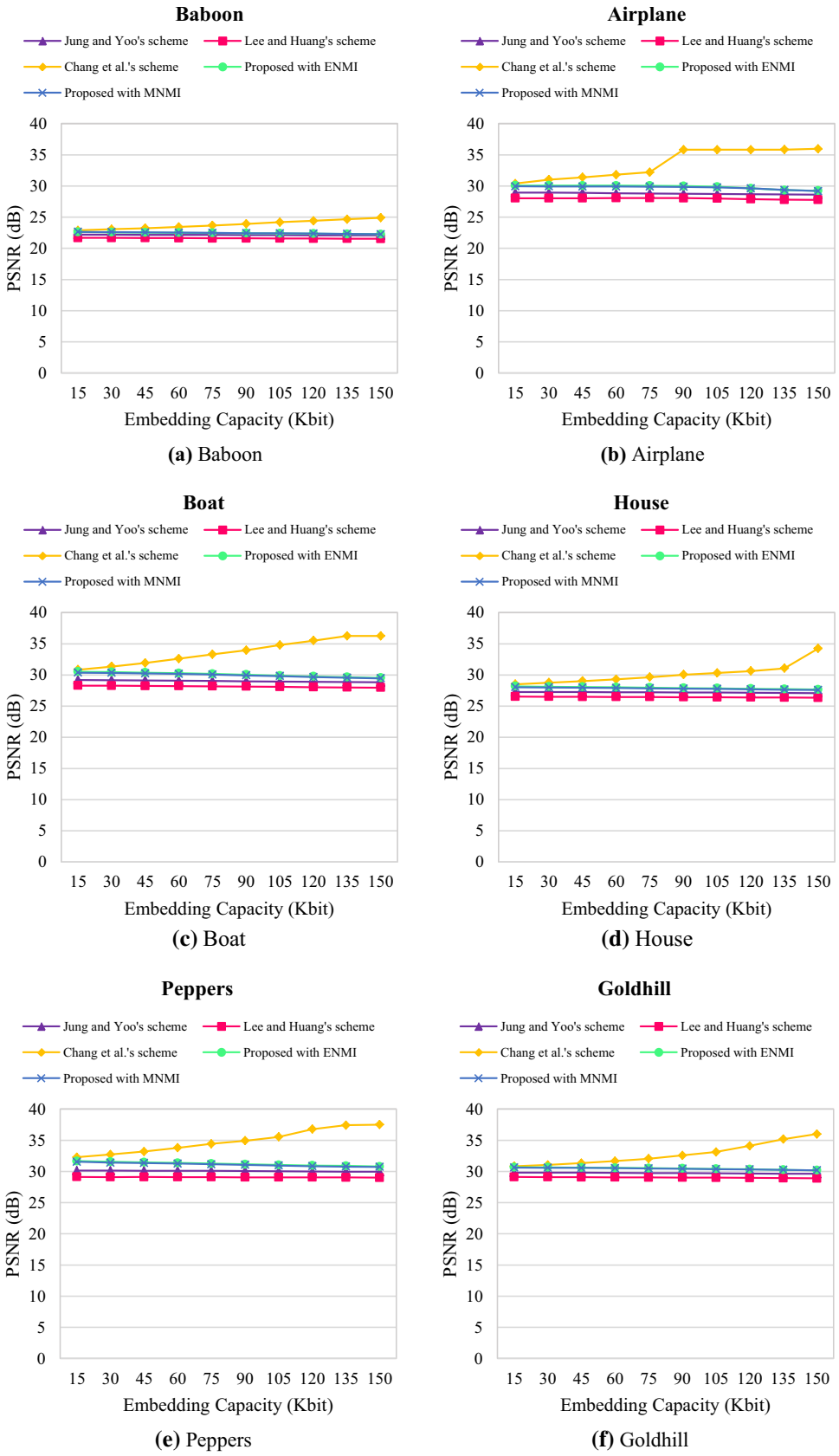


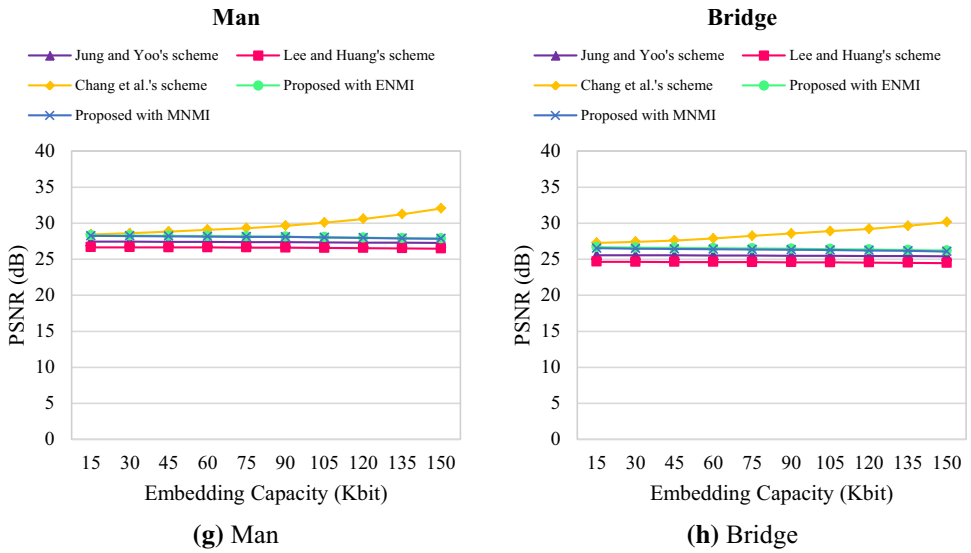
Fig. 12 Comparison of PSNR values of the five IRDH schemes



(a) Baboon **(b)** Airplane **(c)** Boat **(d)** House **(e)** Peppers **(f)** Goldhill

Fig. 13 Comparison of PSNR values of the five IRDH schemes for varying embedding capacity in each test image

Fig. 13 (continued)



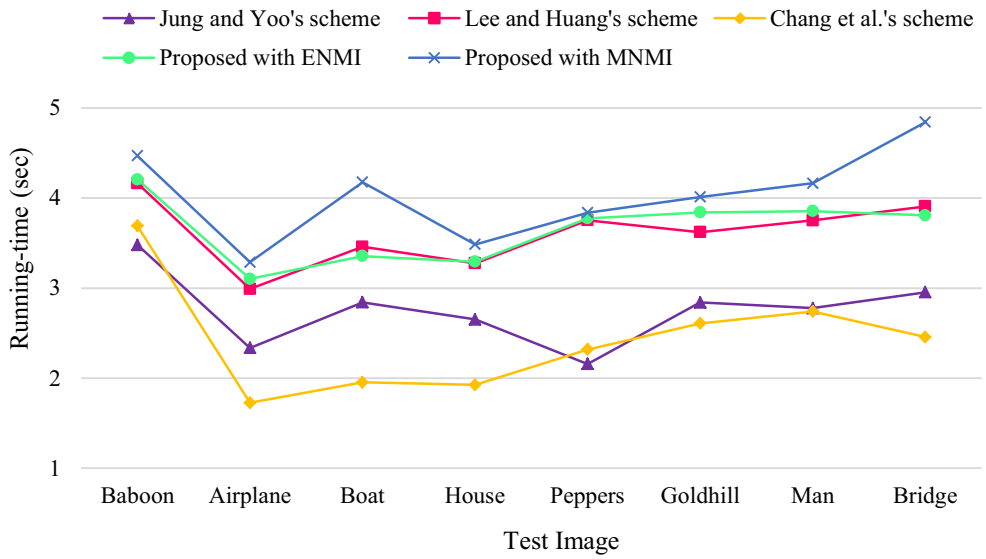


Fig. 14 Comparison of the run-time values in (sec) of the five IRDH schemes

former. Figures 11 and 12 visualize the comparisons of the embedding capacity and the image quality, respectively, of the schemes. The image quality of the schemes is also analyzed for varying embedding capacity in each of the test images as illustrated in Fig. 13.

For analyzing the running time of our proposed scheme, we calculate the running time of the embedding phase of our proposed scheme as well as the other three closely related existing schemes [35, 37, 38] by taking the average of many consecutive runs of each scheme in each image as represented in Fig. 14. It is clearly shown in Fig. 14 that the general sequence of the techniques from the highest running time to the lowest is: our scheme with MNMI, our scheme with ENMI, Lee and Huang's scheme [38], Jung and Yoo's scheme [35] and finally Chang et al.'s scheme [37].

Although our proposed scheme used the same embedding technique of Lee and Huang's scheme [38], ours takes higher time. This is because our scheme embeds more bits into the images so the scheme requires reading more data from the secret bit stream which takes more time.

5 Conclusion

In this paper, we have proposed a high capacity interpolation-based reversible data hiding (IRDH) scheme that outperformed the embedding capacity of state-of-art Lee and Huang's scheme with better image quality. The work considered several techniques in the literature for selection of the best two interpolation techniques, in terms of the image quality, and the best embedding technique, in terms of the embedding capacity, to form this new interpolation-based reversible data hiding scheme. Interestingly, our scheme used the enhanced neighbor mean interpolation (ENMI) and the modified neighbor mean interpolation (MNMI) techniques to produce the cover images that were used later to embed the secret bits. For secret data embedding, we principally adopted the known Lee and Huang's technique adjusted to produce the best results.

The experimental remarks showed that our proposed improved method achieved the highest embedding practical capacity among all the reported schemes that have been involved in similar experimentations, such as the proposals of Jung and Yoo's scheme [35], as well as Lee and Huang's approach [38] and Chang et al.'s scheme [37]. The testing results of our embedding capacity are found scoring the highest PSNR values in addition to embedding 34,398 more bits, on average, than the reported scheme with the best embedding capacity, making us very attractive. Furthermore, among our both approaches using the two interpolation techniques, our scheme with ENMI interpolation achieved better image quality than with MNMI interpolation, but the latter achieved better embedding capacity, as an innovative dimension of trade-off research making the option within the application and user preferences or requirements as complete variation related to secrecy investigation in [45, 47, 48] and cybercrime weaknesses [44] as well as attacks prediction [46].

On the other hands, the embedding phase of our presented scheme suffers from the high running time, which can be a tolerable delay complexity considering todays enhanced computing high speed flexibility. In fact, as future work, this running time of our scheme can be further investigated and studied in depth, either to be reduced or accommodated by proposing parallelization and multi-threading technologies utilizing the multicore processors or via using a more simplified embedding technique or smart machine learning adaptability.

Acknowledgment Thanks to Umm Al-Qura University, Saudi Arabia, for the educational support given toward this research.

Funding There is no funding source.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed Consent Informed consent was obtained from all individual participants included in the study.

References

1. Gutub, A.; Al-Shaarani, F.: Efficient implementation of multi-image secret hiding based on LSB and DWT steganography comparisons. Arab. J. Sci. Eng. **45**, 2631–2644 (2020). https://doi.org/10.1007/s13369-020-04413-w
2. Alanazi, N.; Khan, E.; Gutub, A.: Inclusion of Unicode Standard Seamless Characters to Expand Arabic Text Steganography for Secure Individual Uses. Journal of King Saud University-Computer and Information Sciences, In press (2020) https://doi.org/10.1016/j.jksuci.2020.04.011
3. Al-Juaid, N.; Gutub, A.: Combining RSA and audio steganography on personal computers for enhancing security. SN Appl. Sci. **1**, 830 (2019). https://doi.org/10.1007/s42452-019-0875-8
4. Al-Juaid, N., Gutub, A., Khan, E.: Enhancing PC data security via combining RSA cryptography and video based steganography. J. Inf. Secur. Cybercrimes Res. (JISCR) **1**(1), 8–18 (2018).https://doi.org/10.26735/16587790.2018.006
5. AlKhodaidi, T.; Gutub, A.: Trustworthy target key alteration helping counting-based secret sharing applicability. Arab. J. Sci. Eng. **45**, 3403–3423 (2020). https://doi.org/10.1007/s13369-020-04422-9
6. Gutub, A.; Al-Haidari, F.; Al-Kahsah, K.; Hamodi, J.: e-Text watermarking: Utilizing "Kashida" extensions in Arabic language electronic writing. J. Emerg. Technol. Web Intell. (JETWI) **2**(1), 48–55 (2010). https://doi.org/10.4304/jetwi.2.1.48-55
7. Alotaibi, M., Al-hendi, D., Alroithy, B., AlGhamdi, M., Gutub, A.: Secure mobile computing authentication utilizing hash, cryptography and steganography combination. J. Inf. Secur. Cybercrimes Res. (JISCR) **2**(1), 9–20 (2019).https://doi.org/10.26735/16587790.2019.001
8. Almazrooie, M.; Samsudin, A.; Gutub, A.; Salleh, M.S.; Omar, M.A.; Hassan, S.A.: Integrity verification for digital Holy Quran verses using cryptographic hash function and compression. J. King Saud University-Comput. Inf. Sci. **32**(1), 24–34 (2020). https://doi.org/10.1016/j.jksuci.2018.02.006
9. Alassaf, N.; Gutub, A.: Simulating light-weight-cryptography implementation for IoT Healthcare data security applications. Int. J. E-Health Med. Commun. (IJEHMC) **10**(4), 1–15 (2019). https://doi.org/10.4018/IJEHMC.2019100101
10. Gutub, A.: Pixel indicator technique for RGB image steganography. J. Emerg. Technol. Web Intell. (JETWI) **2**(1), 56–64 (2010). https://doi.org/10.4304/jetwi.2.1.56-64
11. Li, Y.; Chang, C.C.; Mingxing, H.: High capacity reversible data hiding for VQ-compressed images based on difference transformation and mapping technique. IEEE Access. **8**, 32226–32245 (2020). https://doi.org/10.1109/ACCESS.2020.2973179

12. Huang, C.T.; Wang, W.J.; Yang, C.H.; Wang, S.J.: A scheme of reversible information hiding based on SMVQ. Imaging Sci. J. **61**, 195–203 (2013). https://doi.org/10.1179/1743131X11Y.0000000031
13. Kim, C.; Shin, D.; Leng, L.; Yang, C.N.: Lossless data hiding for absolute moment block truncation coding using histogram modification. J. Real-Time Image Process. **14**, 101–114 (2018). https://doi.org/10.1007/s11554-016-0641-8
14. Lin, C.C.; Chang, C.C.; Wang, Z.M.: Reversible data hiding scheme using adaptive block truncation coding based on an edge-based quantization approach. Symmetry (Basel) (2019). https://doi.org/10.3390/sym11060765
15. Malik, A.; Sikka, G.; Verma, H.K.: An AMBTC compression based data hiding scheme using pixel value adjusting strategy. Multidimens. Syst. Signal Process. **29**, 1801–1818 (2018). https://doi.org/10.1007/s11045-017-0530-8
16. Hou, D.; Wang, H.; Zhang, W.; Yu, N.: Reversible data hiding in JPEG image based on DCT frequency and block selection. Signal Process. **148**, 41–47 (2018). https://doi.org/10.1016/j.sigpro.2018.02.002
17. Kim, S.; Huang, F.; Kim, H.J.: Reversible data hiding in JPEG images using quantized DC. Entropy **21**, 1610–1621 (2019). https://doi.org/10.3390/e21090835
18. Chang, J.C.; Lu, Y.Z.; Wu, H.L.: A separable reversible data hiding scheme for encrypted JPEG bitstreams. Signal Process. **133**, 135–143 (2017). https://doi.org/10.1016/j.sigpro.2016.11.003
19. Lin, Y.K.: High capacity reversible data hiding scheme based upon discrete cosine transformation. J. Syst. Softw. **85**, 2395–2404 (2012). https://doi.org/10.1016/j.jss.2012.05.032
20. Agrawal, S.; Kumar, M.: Reversible data hiding for medical images using integer-to-integer wavelet transform. IEEE Students' Conf. Electr. Electron. Comput. Sci. (SCEECS) 18–22 (2016) https://doi.org/10.1109/SCEECS.2016.7509266
21. Ni, Z.; Shi, Y.Q.; Ansari, N.; Su, W.: Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. **16**, 354–361 (2006). https://doi.org/10.1109/TCSVT.2006.869964
22. Hwang, J.H.; Kim, J.W.; Choi, J.U.: A reversible watermarking based on histogram shifting. Lect. Notes Comput. Sci. (LNCS) **4283**, 348–361 (2006). https://doi.org/10.1007/11922841_28
23. Kuo, W.C.; Jiang, D.J.; Huang, Y.C.: Reversible data hiding based on histogram. Lect. Notes Comput. Sci. (LNCS) **4682**, 1152–1161 (2007). https://doi.org/10.1007/978-3-540-74205-0_119
24. Tsai, P.; Hu, Y.C.; Yeh, H.L.: Reversible image hiding scheme using predictive coding and histogram shifting. Signal Processing. **89**, 1129–1143 (2009). https://doi.org/10.1016/j.sigpro.2008.12.017
25. Pan, Z.; Gao, X.; Wang, L.; Gao, E.: Effective reversible data hiding using dynamic neighboring pixels prediction based on prediction-error histogram. Multimed. Tools Appl. **79**, 12569–12595 (2020). https://doi.org/10.1007/s11042-019-08335-0
26. Wang, Z.H.; Lee, C.F.; Chang, C.Y.: Histogram-shifting-imitated reversible data hiding. J. Syst. Softw. **86**, 315–323 (2013). https://doi.org/10.1016/j.jss.2012.08.029
27. Tian, J.: Reversible Data Embedding Using a Difference Expansion. IEEE Trans. Circuits Syst. Video Technol. **13**, 890–896 (2003). https://doi.org/10.1109/TCSVT.2003.815962
28. Liu, Y.C.; Wu, H.C.; Yu, S.S.: Adaptive DE-based reversible steganographic technique using bilinear interpolation and simplified location map. Multimed. Tools Appl. **52**, 263–276 (2011). https://doi.org/10.1007/s11042-010-0496-0
29. Caciula, I.; Coanda, H.G.; Coltuc, D.: Multiple moduli prediction error expansion reversible data hiding. Signal Process. Image Commun. **71**, 120–127 (2019). https://doi.org/10.1016/j.image.2018.11.005
30. Li, X.; Li, J.; Li, B.; Yang, B.: High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. Signal Processing. **93**, 198–205 (2013). https://doi.org/10.1016/j.sigpro.2012.07.025
31. Peng, F.; Li, X.; Yang, B.: Improved PVO-based reversible data hiding. Digit. Signal Process. A Rev. J. **25**, 255–265 (2014). https://doi.org/10.1016/j.dsp.2013.11.002
32. Qu, X.; Kim, H.J.: Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. Signal Process. **111**, 249–260 (2015). https://doi.org/10.1016/j.sigpro.2015.01.002
33. Kim, S.; Qu, X.; Sachnev, V.; Kim, H.J.: Skewed Histogram Shifting for Reversible Data Hiding Using a Pair of Extreme Predictions. IEEE Trans. Circuits Syst. Video Technol. **29**, 3236–3246 (2019). https://doi.org/10.1109/TCSVT.2018.2878932
34. Wu, H.; Li, X.; Zhao, Y.; Ni, R.: Improved reversible data hiding based on PVO and adaptive pairwise embedding. J. Real-Time Image Process. **16**, 685–695 (2019). https://doi.org/10.1007/s11554-019-00867-w
35. Jung, K.H.; Yoo, K.Y.: Data hiding method using image interpolation. Comput. Stand. Interfaces. **31**, 465–470 (2009). https://doi.org/10.1016/j.csi.2008.06.001
36. Zhang, X.; Sun, Z.; Tang, Z.; Yu, C.; Wang, X.: High capacity data hiding based on interpolated image. Multimed. Tools Appl. **76**, 9195–9218 (2017). https://doi.org/10.1007/s11042-016-3521-0
37. Chang, Y.T.; Huang, C.T.; Lee, C.F.; Wang, S.J.: Image interpolating based data hiding in conjunction with pixel-shifting of histogram. J. Supercomput. **66**, 1093–1110 (2013). https://doi.org/10.1007/s11227-013-1016-6
38. Lee, C.F.; Huang, Y.L.: An efficient image interpolation increasing payload in reversible data hiding. Expert Syst. Appl. **39**, 6712–6719 (2012). https://doi.org/10.1016/j.eswa.2011.12.019
39. Malik, A.; Sikka, G.; Verma, H.K.: Image interpolation based high capacity reversible data hiding scheme. Multimed. Tools Appl. **76**, 24107–24123 (2017). https://doi.org/10.1007/s11042-016-4186-4
40. Shaik, A.; Thanikaiselvan, V.: High capacity reversible data hiding using 2D parabolic interpolation. Multimed. Tools Appl. **78**, 9717–9735 (2019). https://doi.org/10.1007/s11042-018-6544-x
41. Malik, A.; Sikka, G.; Verma, H.K.: A Reversible Data Hiding Scheme for Interpolated Images Based on Pixel Intensity Range. Multimed. Tools Appl., In press (2020) https://doi.org/10.1007/s11042-020-08691-2
42. Hassan, F., Gutub, A.: Efficient reversible data hiding multimedia technique based on smart image interpolation. Multimed. Tools Appl. **79**(39), 30087–30109 (2020). https://doi.org/10.1007/s11042-020-09513-1
43. Hassan, F., Gutub, A.: Novel embedding secrecy within images utilizing an improved interpolation-based reversible data hiding scheme. J. King Saud Univ. Comput. Inf. Sci. (2020). https://doi.org/10.1016/j.jksuci.2020.07.008
44. Al-Shaarani, F.; Basakran, N.; Gutub, A.: Sensing e-Banking Cybercrimes Vulnerabilities via Smart Information Sciences Strategies. RAS Eng. Technol. **1**(1), 1–9 (2020)
45. Alkhudaydi, M.; Gutub, A.: Securing Data via Cryptography and Arabic Text Steganography. SN Comput. Sci. **2**, 46 (2021). https://doi.org/10.1007/s42979-020-00438-y
46. Altalhi, S.; Gutub, A.: A survey on predictions of cyber-attacks utilizing real-time twitter tracing recognition. Journal of Ambient Intelligence and Humanized Computing (2021). https://doi.org/10.1007/s12652-020-02789-z
47. Al-Roithy, B., Gutub, A.: Trustworthy image security via involving binary and chaotic gravitational searching within PRNG selections. Int. J. Comput. Sci. Network Secur. (IJCSNS) **20**(12), 167–176 (2020). https://doi.org/10.22937/IJCSNS.2020.20.12.18
48. Bin-Hureib, E., Gutub, A.: Enhancing medical data security via combining elliptic curve cryptography with 1-LSB and 2-LSB image steganography. Int. J. Comput. Sci. Network Secur. (IJCSNS) **20**(12), 232–241 (2020). https://doi.org/10.22937/IJCSNS.2020.20.12.26