International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

Reversible data hiding scheme based on image interpolation using decimal signed digit stream

[1]Reza Ghorbandost Soveiri
[1]Science and Research Branch, Islamic Azad University, Tehran, Iran
reza.ghorbandost@srbiau.ac.ir
[2]Maryam Rajabzadeh Asaar
[2] Science and Research Branch, Islamic Azad University, Tehran, Iran
asaar@srbiau.ac.ir

Abstract
Easy and fast access to data and their exchange in various fields through social networks accessible to the public is increasing daily during the new era. Data hiding technique plays a critical role in telecommunication by providing security in data exchange and privacy protection of individuals or organizations utilizing social network channels. Reversible data hiding (RDH) technique based on image interpolation is among the methods welcomed by researchers in the field of data hiding in digital images in recent years. Here, high embedding capacity (EC) was obtained, as well as maintaining the visual quality of the stego image (SI) by applying some of the most critical interpolation techniques based on the proposed method employing n-bit division of the binary stream of secret data (SD) and their conversion into positive and negative decimal ones. Based on the results, high EC and more appropriate peak signal-to-noise ratio (PSNR) compared to the competitor method were achieved uniformly using some interpolation techniques on all of the standard tested images. On average, the EC and PSNR obtained on all tested images increased by 22.36 and 15.9% compared to the competitor method, respectively.

Keywords Reversible data hiding, Image interpolation, signed digit stream, Embedding capacity, Visual quality

Introduction

Data hiding technique embeds the information in the form of binary data in digital multimedia files such as images, audio, and video. Such embedding does not create any significant change in the multimedia, and all of the data embedded in the multimedia are well received by the receiver. Secret data (SD) cannot be accessed by persons other than the trusted sender and receiver. The above-mentioned technology has attracted much attention to transfer SD in the military, medical, authentication, or other fields on the channels of social networks accessible to the general public.

There are various methods for data hiding in digital images in the spatial domain such as least significant bit (LSB) [2-4], difference expansion (DE) [5-11], and histogram shifting (HS) [12-16], in which the data are embedded based on the location of the pixels. The LSB substitution is among the spatial domain data hiding techniques with a simple algorithm. The LSBs related to the image pixels can be a place to embed SD since the change in the LSB does not alter the pixel value significantly. However, the aforementioned technique has relatively low security and little resistance against various attacks and is considered fragile [1]. Later, two famous techniques including DE [5] and HS [12] were presented, which were successful in terms of embedding capacity (EC) and security in the process of data hiding in images. Tian [5] proposed a reversible DE data hiding method, which can embed a secret bit S into a pair of pixels (P1, P2). The DE technique expands the difference between neighboring pixels (P1, P2) to embed SD. In addition, Ni et al. [12] suggested a reversible HS data hiding method, in which the histogram of the image is created based on the frequency of pixels in the cover image (CI). Accordingly, the pixels with the highest and lowest frequency are called the peak point P and zero point Z, respectively. Pixels in the range [p+1, z] were shifted by one unit to create a hiding space.

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

Then, the pixels equal to the peak point P were utilized to embed SD.

Reversible and irreversible data hidings are among the techniques utilized in the spatial domain of data hiding in images. Reversible data hiding occurs when the embedded data can be extracted on the receiver side appropriately and the stego image (SI) can be restored to its original form. but in irreversible data hiding the stego image cannot be restored to its original form , despite the data extraction.

The interpolation is considered a familiar word in mathematics. This technique is applied in digital signal processing whenever there is a need to change from one sampling rate to another. Interpolation can be employed in the speech processing system [17]. The above-mentioned technique can be used in image processing (image resolution conversion and scale change), as well [18]. The prediction method is utilized to stretch the image based on its reference pixels. The image interpolation technique is widely applied in various fields including satellite maps, as well as medical and military imaging [33]. The data hiding employing the interpolation of digital images has attracted much attention. It has become one of the most common and critical methods in reversible hiding due to its ability to embed a large amount of data [23-39]. This issue is assessed in detail in section 2.1.

Here, an efficient data embedding scheme is proposed, which can maintain the quality of the SI, despite the large amount of data. To this aim, the binary stream of SD is divided into n-bit binary numbers. Then, each divided n-bit number is converted into an equivalent positive or negative decimal number and embedded in the non-reference pixels of the CI (embeddable pixels) based on the method used. The aforementioned technique equally achieves high EC and visual quality on any type of image with any texture (complex and soft) utilizing the most critical interpolation techniques. The above-mentioned property can be applied as a key for trusted people to access data on the receiver's side, as well, considering the conversion of data on the sender's side. The rest of this study is organized as follows. Section 2 evaluates four standard interpolation techniques and some data hiding schemes based on image interpolation. Section 3 examines the proposed data hiding scheme in general and detailed manner, and describes the steps of data embedding and extraction, as well as original image (OI) recovery with examples. Section 4 presents the experimental results, including the performance of the proposed scheme, its general comparison with other methods, and its analysis employing relevant graphs and tables. Finally, conclusions and future objectives are presented.

2. Related work

2.1. Overviewing interpolation techniques for data hiding in images

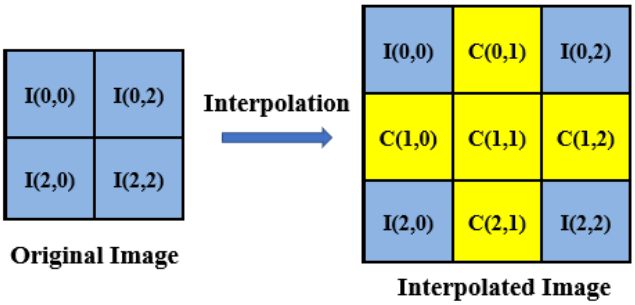


Fig.1 Expansion and interpolation process in 2 × 2 blocks

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

2.1.1. Data hiding using neighbor mean interpolation method

Jung et al. [23] proposed the neighbor mean interpolation (NMI) method for the first time. Predicting non-reference pixels in the above-mentioned method was based on Eq. (1). To this aim, the non-reference pixels in the CI were created. Then, the adjacent pixels in the interpolated image were subtracted from each other (Eq. 2) based on the created CI (Fig. 1). In the next step, the results of the subtraction were put in Eq. (3), resulting in obtaining n number of bits of SD in a string of bit stream which could be embedded for each of the non-reference pixels. In the next step, the stream of SD was divided based on n, and each string of binary data divided based on n was converted into decimal numbers. Finally, the converted data were embedded in the non-reference pixels of the CI, see Eq. (4).

$$\begin{aligned} &C(0,0) = I(0,0) \\ &C(0,1) = (I(0,0) + I(0,2))/2 \\ &C(1,0) = (I(0,0) + I(2,0))/2 \\ &C(1,1) = (I(0,0) + I(0,2) + I(2,0))/3 \end{aligned} \quad (1)$$

$$d_1 = |\,(I(0,0) - C(0,1)\,)\,|\,,\ d_2 = |\,(I(0,0) - C(1,0))\,|\,,\ d_3 = |\,(I(0,0) - C(1,1))\,| \quad (2)$$

$$n_1 = \lfloor \log_2 |d_1| \rfloor\,,\quad n_2 = \lfloor \log_2 |d_2| \rfloor\,,\quad n_3 = \lfloor \log_2 |d_3| \rfloor \quad (3)$$

$$C'(i.j) = \big(C(i.j)\big) + (b)_{10} \qquad (b)_{10} = \sum_{i=0}^{n-1} S_i . 2^i \quad (4)$$

2.1.2. Data hiding using image neighboring pixels method

Lee et al. [26] improved the method proposed by Jung et al. employing the image neighboring pixels (INP) interpolation in terms of the data EC in the image. To this aim, the non-reference pixels of the image were calculated to interpolate the image and create the CI based on Fig. 1 and prediction function in Eq. (5). Then, the largest pixel among the reference pixels of the CI was selected to hide the data in the CI, see Eq. (6). In the next step, the size of the data which can be embedded in the non-reference pixel was obtained by calculating the difference of non-reference pixels from the maximum reference one employing Eq. (7) and computing n using Eq. (3). In the next step, the bit data stream was divided based on the calculated n size. Finally, each string of the bit stream (sub-stream) was converted to decimal base data and embedded in the CI based on Eq. (4).

$$\begin{aligned} &C(0,0) = I(0,0) \\ &C(0,1) = (I(0,0) + (I(0,0) + I(0,2))/2)/2 \\ &C(1,0) = (I(0,0) + (I(0,0) + I(2,0))/2)/2 \\ &C(1,1) = (C(0,1) + C(1,0))/2 \end{aligned} \quad (5)$$

$$M = \max\{\, I(0,0)\,,\, I(0,2)\,,\, I(2,0)\,,\, I(2,0)\,\} \quad (6)$$

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

$$d_1 = |\ (M - C(0,1)\ )\ |\ ,\ d_2 = |\ (M - C(1,0))\ |\ ,\ d_3 = |\ (M - C(1,1))\ | \quad (7)$$

2.1.3. Data hiding using enhanced neighbor mean interpolation method

Chang et al. [27] improved the method proposed by Jung et al. [23] moderately in terms of data EC and image quality by applying the enhanced neighbor mean interpolation (ENMI) method. To this aim, the non-reference pixels of the image were calculated based on Fig. 1 and the prediction function of Eq. (8) to interpolate the image and create the CI. In the next step, the difference between the input image before rescaling and the interpolated image was calculated based on Eq. (9). The resulting matrix was based on the arrangement of non-reference pixels in the CI. In the next step, the number of sub-stream bits in the overall binary SD stream was calculated based on Eq. (10), employing the results related to the aforementioned difference. Then, the binary data stream was divided, and the achieved sub-streams were converted into decimal numbers. Finally, the converted data with the same arrangement were added to the values of the non-reference pixels in the CI, resulting in embedding the data in that image, which is shown in Eq. (11).

$$\begin{aligned} C(0,0) &= I(0,0) \\ C(0,1) &= (I(0,0) + I(0,2))\ /2 \\ C(1,0) &= (I(0,0) + I(2,0))\ /2 \\ C(1,1) &= (I(0,0) + I(0,2) + I(2,0) + I(2,2))\ /4 \end{aligned} \quad (8)$$

$$\begin{aligned} &D(0,1) = |(C(0,1) - I(0,1)\ )|\ ,\ D(1,0) = |(C(1,0) - I(1,0))|\ ,\ D(1,1) = |(C(1,1) - I(1,1))| \\ &D(1,2) = |(C(1,2) - I(1,2))|\ ,\ D(2,1) = |(C(2,1) - I(2,1))| \end{aligned} \quad (9)$$

$$\begin{aligned} &n_1 = \lfloor \log_2 |D(0,1)| \rfloor\ ,\ n_2 = \lfloor \log_2 |D(1,0)| \rfloor\ , \\ &n_3 = \lfloor \log_2 |D(1,1)| \rfloor\ , \\ &n_4 = \lfloor \log_2 |D(1,2)| \rfloor\ ,\ n_5 = \lfloor \log_2 |D(2,1)| \rfloor \end{aligned} \quad (10)$$

$$C'(i.j) = \big(C(i.j)\big) + (b)_{10} \qquad (b)_{10} = \sum_{i=0}^{n-1} S_i . 2^i \quad (11)$$

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

2.1.4. Data hiding using a new interpolation extension method

Mohammad et al. [30] proposed a new interpolation extension (NIE) method to improve the above-mentioned data EC and image quality. To this aim, the non-reference pixels in the image were calculated to interpolate the image and create the CI based on Fig. 1 and prediction function of Eq. (12). Then, the smallest pixel among the reference pixels in the CI was selected, see Eq. (13) . In the next step, the size of data which can be embedded in the non-reference pixel was obtained by calculating the difference of non-reference pixels from the minimum reference pixel (Eq. 14) and computing n utilizing Eq. (3). In the next step, the bit data stream was divided based on the calculated n size. Finally, each string of the bit stream (sub-stream) was converted to decimal base data and embedded in the CI based on Eq. (4).

$$\begin{aligned} C(0,0) &= I(0,0) \\ C(0,1) &= (I(0,0) + I(0,2))/3 + (I(2,0) + (I(2,2))/6 \\ C(1,0) &= (I(0,0) + I(2,0))/3 + (I(0,2) + (I(2,2))/6 \\ C(1,1) &= (I(0,0) + I(0,2) + I(2,0) + I(2,2))/4 \end{aligned} \quad (12)$$

$$m = \min \{ I(0,0) , I(0,2) , I(2,0) , I(2,0) \} \quad (13)$$

$$d_1 = | (m - C(0,1)) | , d_2 = | (m - C(1,0)) | , d_3 = | (m - C(1,1)) | \quad (14)$$

2.2. Proposing some data hiding schemes to improve the efficiency of interpolation techniques

Malik et al. [28] proposed the modified neighbor mean interpolation (MNMI) technology, which considered all neighboring pixels and their influence on the reference pixels to present better interpolated image after embedding the data. To this aim, the secret data were embedded in the interpolated pixels, considering the human vision system to maintain the quality of the resulting image. Then, the smooth and complex regions in the interpolated or cover image were identified by dividing the image into blocks to embed more bits in these regions, increase the data EC, and improve the image quality. In addition, Malik et al. [33] proposed an interpolation and a new reversible data hiding scheme to upscale the OI and hide SD in the image (interpolated upscale). The proposed scheme considered the characteristics of the human visual system while embedding SD. To this aim, the image was divided into groups in terms of pixel intensity ranges. Then, the SD bits were adaptively embedded in the pixels based on the pixel intensity values. The proposed scheme could maintain the visual quality of the stego image (SI). Based on the results, the proposed scheme performed better, especially in terms of visual quality, compared to some interpolation-based data hiding schemes [23,29,30]. Further, Chen et al. [34] proposed a data hiding scheme based on image interpolation, which could be applied in different image interpolation techniques. To this aim, the interpolated CI was divided into 3×3 blocks. Then, the non-reference pixels in the upper horizontal row, as well as the left and center vertical rows, were separately differentiated with their neighboring reference pixels. In the next step, the primary binary stream data was divided

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

into binary sub-streams and converted into decimal data based on the maximum calculated difference size. Comparing the size of the embeddable pixels in the CI with the main one was regarded as the embedding criterion during the data embedding stage, resulting in adding or subtracting the data in the embeddable pixels. The proposed method established a relative balance between data EC and image quality. Furthermore, Bai et al. [38] proposed a new scheme for efficient data hiding employing image interpolation. Previously, the division of the SD stream was calculated based on the difference between reference and non-reference pixels. Then, the data size was determined based on decimals for embedding in each of the non-reference pixels. Finally, the non-reference pixels were embedded in the order of priority in a zigzag manner from top to bottom and left to right. However, four sizes were calculated for the primary non-reference pixels using the four reference pixels and predicting their method. To this aim, the calculated non-reference pixels were sorted based on the order of binary numbers and in ascending order in the case of weighted similarity of two pixels among the four primary non-reference pixels. Then, one of the two similar pixels was removed, and the larger one was considered by adding a unit as the fourth non-reference pixel. Finally, the original and last non-reference pixels were determined, except for the smaller pixel among the last three ones. The process was the same in the case of non-similarity of the primary non-reference pixels. However, the smaller pixel was removed and the last three ones were considered as the original and final non-reference pixels in the CI, respectively. The arrangement of the last three non-reference pixels in the CI was considered after dividing the SD into one, two, or three-bit based on their correspondence with the numbers assigned to the non-reference pixels. The arrangement of non-reference pixels in order of priority (center, top, and left) was interpolated for each 3×3 block in the CI, resulting in achieving a relatively high and identical data EC and maintaining the appropriate visual quality for all of the tested images.

In another study, Zhang et al. [39] proposed an data hiding scheme based on circular predictor interpolation (CPI). The algorithm of the proposed scheme initially divided an image into regions based on texture degree and selected different prediction radii for different regions. Then, a parameter named distance coefficient was suggested for prediction accuracy. In the next step, the complexity of the non-reference pixels was calculated to ensure better visual quality under low load conditions. In the next step, the non-reference pixels were sorted in ascending order to hide SD with priority in pixels with low complexity. The overflow and underflow problems were defined by a pixel value correction function which utilized reference pixels to calculate two non-reference pixels and guaranteed that each non-reference pixel can embed SD. Finally, synchronization was designed to create the interpolation and embedding regions and reduce the distortion under low load conditions due to its ability to be set simultaneously based on the number of embedded bits. The method proposed by Zhang et al. presented appropriate results regarding EC and visual quality compared to previous ones [23,26,34,35,37,38]. In our proposed method, The most critical interpolation techniques were applied like previous ones. To this aim, the binary data stream was divided into equal parts and converted the divided data into decimal basis. Then, the segmented data were converted into positive and negative decimal data intelligently, reducing the distortion in embeddable pixels after embedding the data. Converting the stream of SD into that of decimal signed digit creates an appropriate embedding space for the non-reference pixels in the interpolated images equally unlike the previous methods which assigned a different EC to each non-reference pixel, resulting in generating the same and high EC with proper visual quality for any type of image. Section 3 investigates the issue in detail.

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

3. Proposed scheme

The strategy of converting binary SD into signed digit is studied here. Then, the data embedding process is explained by citing an example. Finally, an example is presented for extracting the data and recovering the OI.

3.1. Strategy of SD conversion and created EC

The binary SD stream is converted into the signed digit in the proposed method based on the influence of the embedded data on the distortion of CI. To this aim, the binary data stream is divided into four-bit sub-streams, considering n=4. In other words, the main SD stream is divided into four-bit numbers, as well as converting each four-bit sub-stream into an equivalent decimal number. Employing the decimal equivalent from the four-bit number $(1000)_2$-$(1111)_2$ creates a significant distortion in the pixels, which can be embedded in the CI. Thus, the decimal number -1 was put based on mode 3 in Table 1 instead of the four-bit binary number $(1000)_2$, except the four-bit numbers $(0000)_2$-$(0111)_2$. the same way, decimal numbers -2, -3, -4, -5, -6, -7, and -8 were replaced instead of four-bit binary ones $(1001)_2$, $(1010)_2$, $(1011)_2$, $(1100)_2$, $(1101)_2$, $(1110)_2$, and $(1111)_2$. As indicated in Table 1, modes 1 and 2 are similar, creating a maximum of ±15 unit difference in embeddable pixels, which is not suggested. In addition, modes 3 and 4 are similar, generating a maximum of ±8 unit difference in embeddable pixels, which is about half of modes 1 and 2. Modes 3 and 4 are used in the conversion, which exhibit similar behaviors. Similarly, the binary data stream is divided into five-bit sub-streams, considering n=5. To this aim, the original SD stream is divided into five-bit numbers, and each five-bit sub-stream is converted into an equivalent decimal number. Utilizing the decimal equivalent from the five-bit numbers $(00000)_2$-$(01111)_2$ creates a significant distortion in the pixels, which can be embedded in the CI. Therefore, the decimal number -1 was applied based on the mode 3 in Table 3 instead of the five-bit binary one $(10000)_2$, except the five-bit numbers $(00000)_2$-$(01111)_2$. Similarly, decimal numbers -2, -3, -4, -5, -6, -7, -8, -9, -10, -11, -12, -13, -14, -15, and -16 were replaced instead of five-bit binary ones $(10001)_2$, $(10010)_2$, $(10011)_2$, $(10100)_2$, $(10101)_2$, $(10110)_2$, $(10111)_2$, $(11000)_2$, $(11001)_2$, $(11010)_2$, $(11011)_2$, $(11100)_2$, $(11101)_2$, $(11110)_2$, and $(11111)_2$. As shown in Table 2, modes 1 and 2 are similar, creating a maximum of ±31 unit difference in embeddable pixels, which is not recommended. Modes 3 and 4 are considered similar, resulting in creating a maximum of ±16 unit difference in embeddable pixels. Modes 3 and 4 are employed about half of modes 1 and 2, which exhibit similar behaviors in the conversion. Fig. 2 shows the flow chart related to converting the binary data stream into that of marked numbers for n=4 and n=5 in order to understand the issue better.

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

Table 1- Conversion and assignment of signed digit code

4 -bit binary digits	states of signed-digit				
	case 1	case 2	case 3	case 4	selected case
0 0 0 0	0	0	0	0	0
0 0 0 1	1	-1	1	1	1
0 0 1 0	2	-2	2	2	2
0 0 1 1	3	-3	3	3	3
0 1 0 0	4	-4	4	4	4
0 1 0 1	5	-5	5	5	5
0 1 1 0	6	-6	6	6	6
0 1 1 1	7	-7	7	7	7
1 0 0 0	8	-8	-1	8	-1
1 0 0 1	9	-9	-2	-1	-2
1 0 1 0	10	-10	-3	-2	-3
1 0 1 1	11	-11	-4	-3	-4
1 1 0 0	12	-12	-5	-4	-5
1 1 0 1	13	-13	-6	-5	-6
1 1 1 0	14	-14	-7	-6	-7
1 1 1 1	15	-15	-8	-7	-8

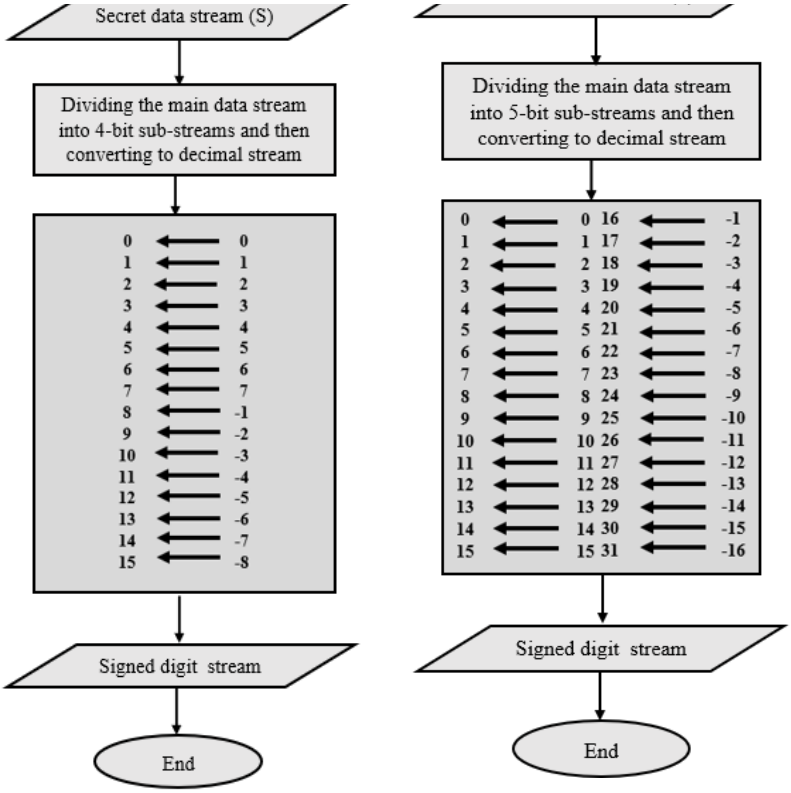


Fig. 2 Flowchart of converting the stream of binary data into the stream of signed digits for (n=4) and

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

Table 2 - Conversion and assignment of signed digit code

5 -bit binary digits	states of signed-digit				
	case 1	case 2	case 3	case 4	selected case
0 0 0 0 0	0	0	0	0	0
0 0 0 0 1	1	-1	1	1	1
0 0 0 1 0	2	-2	2	2	2
0 0 0 1 1	3	-3	3	3	3
0 0 1 0 0	4	-4	4	4	4
0 0 1 0 1	5	-5	5	5	5
0 0 1 1 0	6	-6	6	6	6
0 0 1 1 1	7	-7	7	7	7
0 1 0 0 0	8	-8	8	8	8
0 1 0 0 1	9	-9	9	9	9
0 1 0 1 0	10	-10	10	10	10
0 1 0 1 1	11	-11	11	11	11
0 1 1 0 0	12	-12	12	12	12
0 1 1 0 1	13	-13	13	13	13
0 1 1 1 0	14	-14	14	14	14
0 1 1 1 1	15	-15	15	15	15
1 0 0 0 0	16	-16	-1	16	-1
1 0 0 0 1	17	-17	-2	-1	-2
1 0 0 1 0	18	-18	-3	-2	-3
1 0 0 1 1	19	-19	-4	-3	-4
1 0 1 0 0	20	-20	-5	-4	-5
1 0 1 0 1	21	-21	-6	-5	-6
1 0 1 1 0	22	-22	-7	-6	-7
1 0 1 1 1	23	-23	-8	-7	-8
1 1 0 0 0	24	-24	-9	-8	-9
1 1 0 0 1	25	-25	-10	-9	-10
1 1 0 1 0	26	-26	-11	-10	-11
1 1 0 1 1	27	-27	-12	-11	-12
1 1 1 0 0	28	-28	-13	-12	-13
1 1 1 0 1	29	-29	-14	-13	-14
1 1 1 1 0	30	-30	-15	-14	-15
1 1 1 1 1	31	-31	-16	-15	-16

A direct relationship is observed between the maximum EC created in the proposed method and n. An increase in n raises the EC. The created EC is calculated by Eq. (15) since three - quarters of the block can embed the data each 2×2 block. For example, EC of 786432 bits is observed in the CI for an image with dimensions of 512×512 when (n=4). The aforementioned value increases to 983040 bits when (n=5).

$$EC = (3/4)\times(\text{Image Size})\times n \qquad (15)$$

3.2. Embedding and extracting the data and recovering the OI

Fig. 3 illustrates the block diagram related to the proposed scheme. The OI is converted to half of its original size in the encoder or sender part. Then, the interpolation operation is performed on the image using one of the interpolation methods. In the next step, the secret data are embedded in the CI. In the next step, the SI is received on the side of the decoder or receiver, followed by extracting the data and restoring the CI. On the receiving side, the SD can be extracted without errors, as well as restoring the original CI correctly.

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

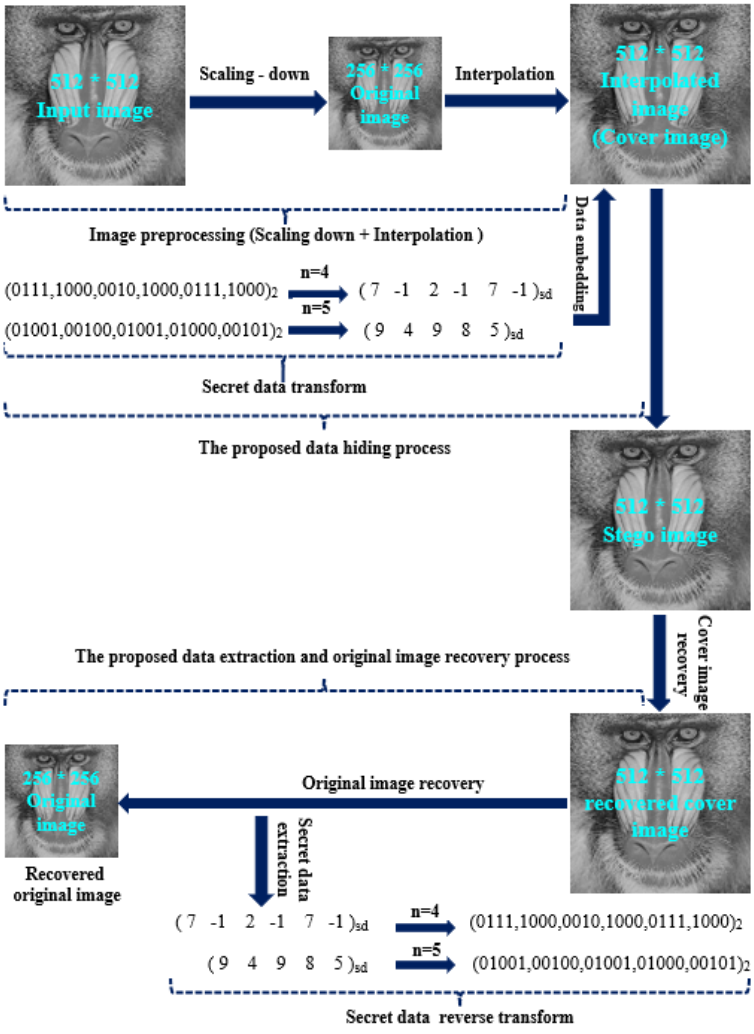


Fig. 3 Framework of our proposed scheme

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

3.2.1. The process of data embedding in the proposed scheme

The steps of the data embedding process are as follows:
Input: CI (P) and SD (S)
Output: SI (P')
Step 1: The input image is rescaled to half its original size for interpolation.

$$P'(0,1) = P(0,1) + b_1$$
$$P'(1,0) = P(1,0) + b_2 \quad (16)$$
$$P'(1,1) = P(1,1) + b_3$$

Step 2: Interpolating the reference pixels in the downscaled image, calculating the non-reference pixels, and creating the CI (P).
Step 3: Blocking the interpolated image into 2×2 blocks and embedding the data in non-reference pixels for each 2×2 block. The order of embedding is formed based on Eq. (16) when $S = (b_1 b_2 b_3 \ldots\ldots b_n)_{sd}$, resulting in creating the SI (P').

3.2.2. The process of data extraction and OI recovery in the proposed scheme

The steps of extracting the data and restoring the OI are as follows.
Input: SI (P')
Output: Initial CI (P) and SD (S)

Step 1: Replacing the number zero instead of the non-reference pixels embedded in the CI of Eq. (17).
Step 2: Re-interpolating the reference pixels and reconstructing the original CI
Step 3: The difference between the SI and interpolated image, the result of the difference is the signed digit based on Eq. (18), which turns the data stream of the signed digit into a binary one eventually.

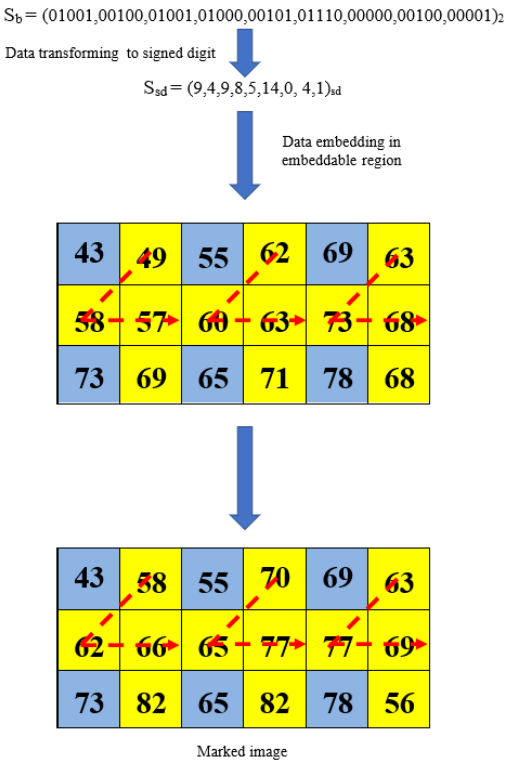
$$P'(0,1) = 0$$
$$P'(1,0) = 0 \quad (17)$$
$$P'(1,1) = 0$$

$$P'(0,1) - P(0,1) = b_1$$
$$P'(1,0) - P(1,0) = b_2 \quad (18)$$
$$P'(1,1) - P(1,1) = b_3$$

3.2.3. Presenting an example for embedding and extracting the data and OI recovery

To understand the issue better, an example is presented for embedding the data in the interpolated image, extracting the data, and restoring the original CI from the covered image. Fig. 4 demonstrates the method of interpolating the reference pixels in the image. The type of interpolation is selected by the NMI method. The reference pixels which cannot be embedded and non-reference pixels which can be embedded are displayed in blue and yellow, respectively. In addition, the path and order of embedding the SD in the non-reference pixels in the CI are shown with a red dashed line.

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025



The SD stream (S) considered here for (n=5) is embedded in the interpolated image after image interpolation, resulting in creating the SI, see Fig. 5.

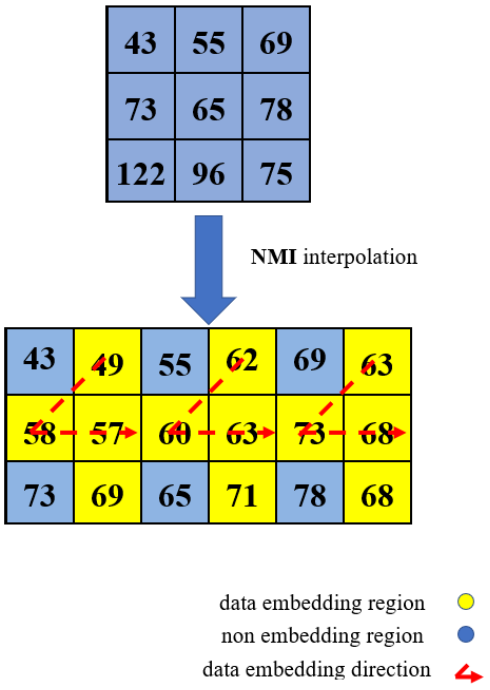


Fig. 4 Location and direction for data embedding in embeddable pixels

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

$S_{sd} = (9,4,9,8,5,14,0,4,1)_{sd}$

Transform to binary digit

$S_b = (01001,00100,01001,01000,00101,01110,00000,00100,00001)_2$

Fig.8 Converting the signed digit stream into a binary

According to step 1 and step 2 related to The process of data extraction and OI recovery in the proposed scheme in section 3.2.2, the number zero is used instead of the non-reference pixels embedded in the CI in order to extract the SD. The interpolation operation is performed again utilizing the reference pixels in the image, resulting in creating the original CI, see Fig. 6. The SD can be extracted by the difference between the SI and recovered CI, see Fig. 7. The extracted data includes a stream of signed digit, which are converted into a binary SD stream, see Fig. 8.

Fig.5 Secret data embedding in the interpolated image

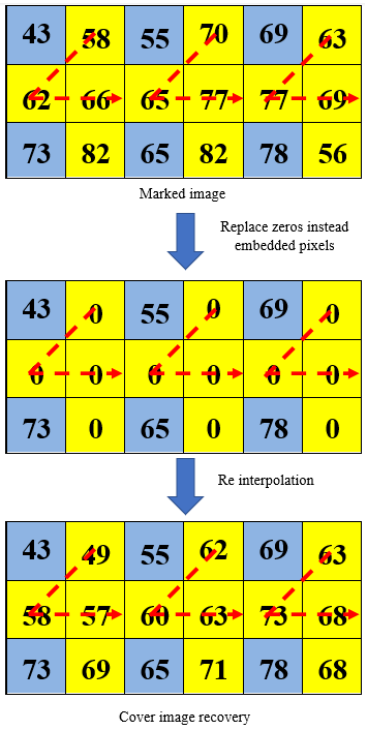


Fig.6 Steps for recovering the original CI

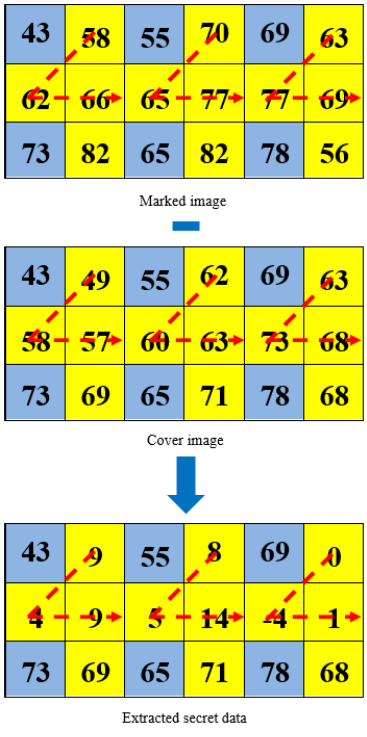


Fig.7 Steps for secret data extraction

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

4. Experimental results

All of the experiments were implemented on Matlab R2016a on Windows 10. Standard images with various textures were selected from [40] and [41] to conduct the experiments. "Lena", "Goldhill", "Baboon", "Couple", "Peppers", "Man", "Elaine", and "Boat", as shown in Fig. 9. were selected as test images. All images have dimensions of 512×512 with gray scale. The SD includes a set of binary numbers randomly generated by MATLAB. The length of the binary data stream was 786432 and 983040 bits, considering that of binary numbers in the experiments. Peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) were applied to compare the performance quality of the proposed algorithm with related methods, whose formulas are shown in Eq. (19) and Eq. (20) and Eq. (21) and Eq. (22) and Eq. (23) and Eq. (24).

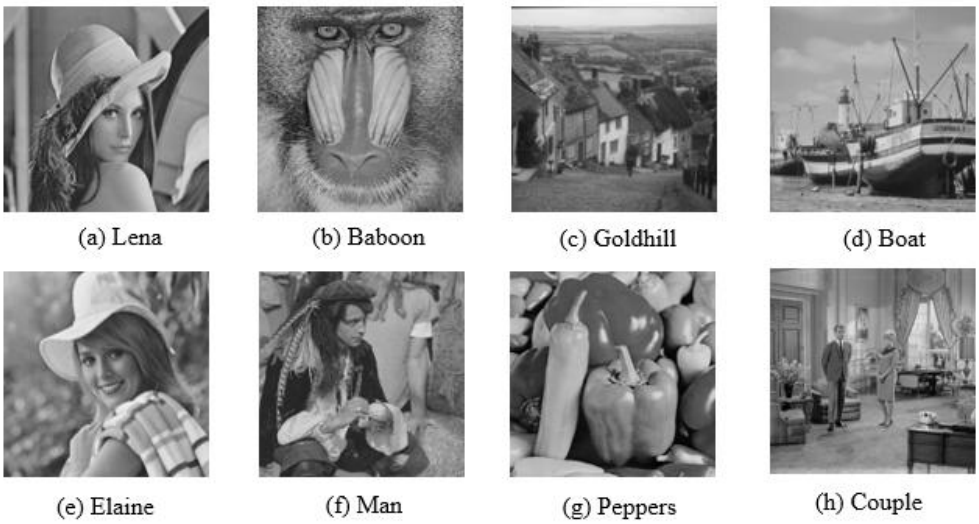


Fig.9 Standard images used in experiments

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \quad (dB) \tag{19}$$

$$MSE = \frac{1}{H \times W} \times \sum_{i=1}^{H} \sum_{j=1}^{W} (p(i.j) - p'(i.j))^2 \tag{20}$$

$$SSIM\ (P,P') = [\ l(P,P')]^{\alpha} \times [\ C(P,P')]^{\beta} \times [\ S(P,P')]^{\gamma} \tag{21}$$

where $\alpha = \beta = \gamma = 1$

$$l(p,p') = \frac{2 \times \bar{p} \times \bar{p}' + c}{\bar{p}^2 \times (\bar{p}')^2 + c} \tag{22}$$

$$c(p,p') = \frac{2\delta_p \delta_{p'} + c}{\delta^2{}_p + \delta^2{}_p + c} \tag{23}$$

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

$$s(p,p') = \frac{\delta_{pp'}+c}{\delta_p \delta_{p'}+c} \quad (24)$$

4.1. Efficiency of the proposed scheme

The maximum EC was calculated for all standard images under equal conditions. Interpolation was applied to each image based on four primary methods. The results obtained by measuring the EC and PSNR were almost the same for all images. The EC was calculated as 786432 bits for all images employing the stream of signed digit with (n=4). In addition, PSNR was the same in the above-mentioned capacity for all images under four basic interpolation methods with an average size of 36.05 dB. The EC increased by 983040 bits for signed digits with (n=5). Further, PSNR was the same in the aforementioned capacity for all images under four interpolation methods with an average size of 30.05 dB. Table 3 and Table 4 represent the complete results. PSNR and mean-square error (MSE) were measured for the image quality after embedding the data, and the efficiency of the stream in the signed digit was investigated for each of the situations presented in Table 1 and Table 2. PSNR and MSE were 30.48 dB and 58.12 for a "baboon" image, considering the signed digit (n=4) and modes 1 and 2 in Table 1. In addition, PSNR and MSE were 36.05 dB and 16.13, considering modes 3 and 4, as well as the states of the signed digit in Table 1, indicating the growth of PSNR by 18.2% compared to modes 1 and 2. Further, the amount of MSE drop was -72.2%. Based on the results, using modes 3 and 4 in Table 1 increased PSNR and improved image quality more tangibly compared to modes 1 and 2 in Table 1. Further, PSNR and MSE were 24.25 dB and 244.2 on the same sample of the image, considering the signed digit (n=5) in modes 1 and 2 in Table 2. Furthermore, PSNR and MSE were 30.05 dB and 64.2, considering modes 3 and 4 in Table 2, indicating the growth of PSNR by 23.9% compared to modes 1 and 2. The amount of MSE drop was -73.7%. Based on the results, utilizing modes 3 and 4 in Table 2 increased PSNR and improved the image quality more tangibly compared to modes 1 and 2 in Table 2, like the previous case. All of the efforts in data hiding are spent to improve the image quality even a little. Thus, applying the signed digit in modes 3 and 4 and embedding the stream of signed digit generated with the help of those in modes 3 and 4 in Table 1 and Table 2 provide better PSNR in CI. Decoding the arrangement of the embedded data on the receiver's side plays a critical role in accessing the data, considering the method of implementing this process in the CI and the arrangement of such data on the sender's side. In addition, the arrangement of the data stream of signed digit can be accessed only by their conversion into equivalent n-bit ones after extraction in receiver. The key to converting the signed digit into equivalent n-bit ones can be provided to the trusted receiver through a secure channel.

4.2. Comparing the results of the proposed scheme with other related methods

Here, the results of the experiments are analyzed precisely based on the criteria for measuring the effectiveness of image hiding compared to other related methods. Table 5 compares the results of the experiments with related methods in terms of EC, PSNR, and SSIM.

The average EC employing the stream of signed digits with (n=4) on the tested images increases by 203.3% in the proposed method compared to [23]. In addition, the EC increases by 108.9, 34.3, and 22.3% in the proposed method compared to [34], [38], and [39], respectively. The average EC in the tested images using the stream of signed digits with (n=5) increases by 279, 161, 67.9, and 52.9% compared to [23], [34], [38], and [39], respectively, indicating the significant growth of data EC in the proposed method in

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

comparison with other related ones. Fig. 10 illustrates the results of comparing the EC in the proposed method with other related ones. The proposed method was compared with other related ones in terms of the PSNR, which expresses the image quality after embedding the data, as shown in Fig. 11. To this aim, the data stream of signed digits with (n=4) was utilized in the experiments. PSNR size was estimated to be the same for all tested images (36.05 dB) under the maximum EC. The average calculated PSNR in the proposed method increased by 28.7, 15.8, 32.4, and 15.8% compared to [23], [34], [38], and [39], respectively. Based on the results, the image quality was maintained in the proposed method, despite the increase in capacity.

Then, the SSIM was measured on the images. Fig. 12 demonstrates the comparison curve between the proposed method and other related ones. As displayed, the size of the above-mentioned criterion increases in images with complex texture compared to other methods. The size of the aforementioned criterion fluctuates compared to other methods in other images, depending on the complexity of the image texture. In the next step, the calculated size of the above-mentioned criterion for two types of image texture was reviewed and compared to the competitor method, considering the SSIM comparative curve [39]. For instance, the size of SSIM under the EC of 786432 bits was calculated as 0.9289 for the "baboon" image, which exhibits a complex texture. However, the aforementioned value was calculated as 0.8220 under the EC of 721443 bits [39]. The SSIM in the proposed method grew by 13% compared to [39]. The SSIM size under the same EC was 0.8717 for the "Lena" image, which has a soft texture. However, the above-mentioned value was calculated as 0.9225 under the EC of 622812 bits [39]. The proposed method shows -5.8% decline compared to the competing method in terms of SSIM in "Lena" image due to its high EC.

In the next step, the average size of SSIM in the proposed method compared to [23], [34], [38], and [39] was calculated as 14.9, 0.15, 14.5, and -0.6%, respectively. The results represent a growth in SSIM compared to other related methods, except for [39], which shows a slight drop. Finally, the size of PSNR under different ECs was calculated separately for each image (Fig. 13). In the proposed method, the behavior of PSNR to the increase of EC is the same for all of the tested images due to the exact size of PSNR and EC. For example, the competing method [39] exhibits a higher PSNR than the proposed one under the shared EC in images with soft texture such as "Lena" and "Pepper". The proposed method shows a significant superiority over the competing one [39] in terms of PSNR to EC ratio for some other images such as "baboon" and "couple". In addition, the proposed method exhibits an appropriate superiority over the compared ones, especially the competing method, for other images in terms of PSNR to EC ratio under different ECs.

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

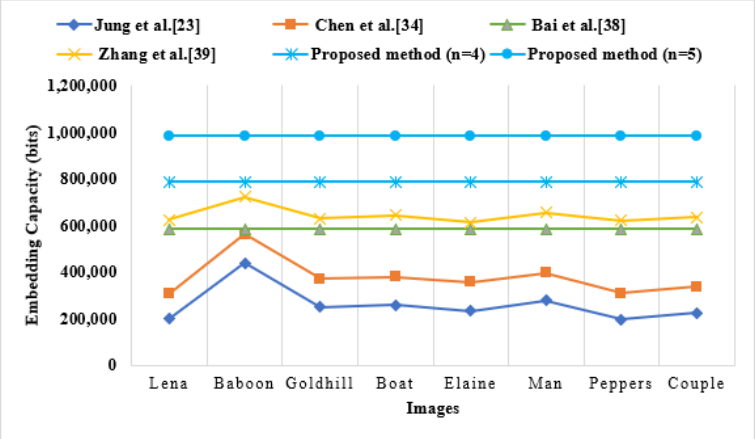


Fig.10 Comparison of EC

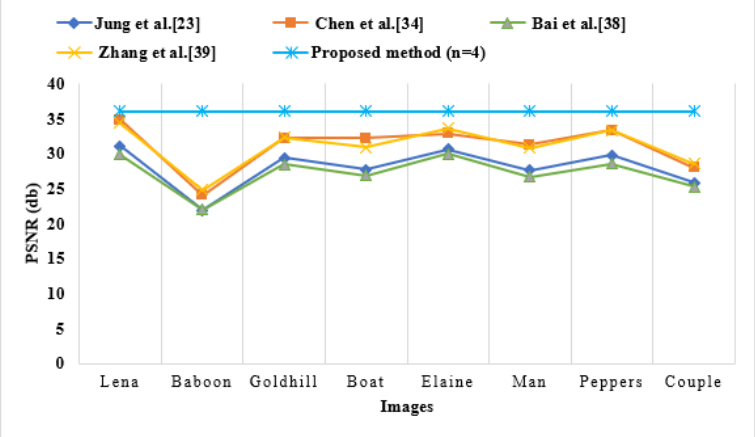


Fig.11 Comparison of PSNR

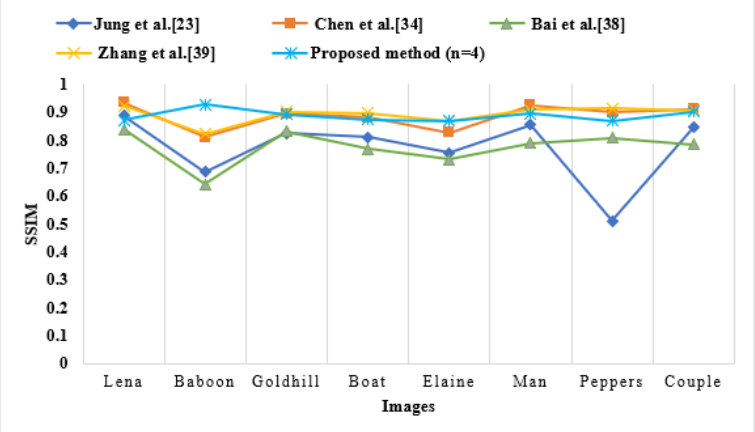


Fig.12 Comparison of SSIM

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

Table 3 - Overall comparisons of HC (bits) and PSNR (dB) in NMI & INP & ENMI & NIE for (n=4)

Images	Metrics	NMI	INP	ENMI	NIE
Lena	HC	for 786,432 bit	for 786,432 bit	for 786,432 bit	for 786,432 bit
	PSNR	36.05 (db)	36.05 (db)	36.05 (db)	36.05 (db)
Baboon	HC	for 786,432 bit	for 786,432 bit	for 786,432 bit	for 786,432 bit
	PSNR	36.05 (db)	36.05 (db)	36.05 (db)	36.05 (db)
Goldhill	HC	for 786,432 bit	for 786,432 bit	for 786,432 bit	for 786,432 bit
	PSNR	36.05 (db)	36.05 (db)	36.05 (db)	36.05 (db)
Boat	HC	for 786,432 bit	for 786,432 bit	for 786,432 bit	for 786,432 bit
	PSNR	36.05 (db)	36.05 (db)	36.05 (db)	36.05 (db)
Elaine	HC	for 786,432 bit	for 786,432 bit	for 786,432 bit	for 786,432 bit
	PSNR	36.05 (db)	36.05 (db)	36.05 (db)	36.05 (db)
Man	HC	for 786,432 bit	for 786,432 bit	for 786,432 bit	for 786,432 bit
	PSNR	36.05 (db)	36.05 (db)	36.05 (db)	36.05 (db)
Peppers	HC	for 786,432 bit	for 786,432 bit	for 786,432 bit	for 786,432 bit
	PSNR	36.05 (db)	36.05 (db)	36.05 (db)	36.05 (db)
Couple	HC	for 786,432 bit	for 786,432 bit	for 786,432 bit	for 786,432 bit
	PSNR	36.05 (db)	36.05 (db)	36.05 (db)	36.05 (db)
Average	HC	**for 786,432 bit**	**for 786,432 bit**	**for 786,432 bit**	**for 786,432 bit**
	PSNR	**36.05 (db)**	**36.05 (db)**	**36.05 (db)**	**36.05 (db)**

Table 4 - Overall comparisons of HC (bits) and PSNR (dB) in NMI & INP & ENMI & NIE for (n=5)

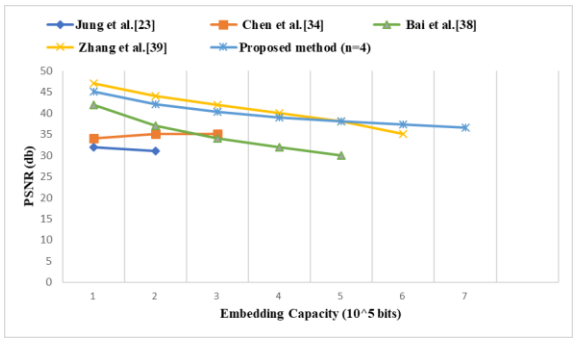
Images	Metrics	NMI	INP	ENMI	NIE
Lena	HC	for 983,040 bit	for 983,040 bit	for 983,040 bit	for 983,040 bit
	PSNR	30.05 (db)	30.05 (db)	30.05 (db)	30.05 (db)
Baboon	HC	for 983,040 bit	for 983,040 bit	for 983,040 bit	for 983,040 bit
	PSNR	30.05 (db)	30.05 (db)	30.05 (db)	30.05 (db)
Goldhill	HC	for 983,040 bit	for 983,040 bit	for 983,040 bit	for 983,040 bit
	PSNR	30.05 (db)	30.05 (db)	30.05 (db)	30.05 (db)
Boat	HC	for 983,040 bit	for 983,040 bit	for 983,040 bit	for 983,040 bit
	PSNR	30.05 (db)	30.05 (db)	30.05 (db)	30.05 (db)
Elaine	HC	for 983,040 bit	for 983,040 bit	for 983,040 bit	for 983,040 bit
	PSNR	30.05 (db)	30.05 (db)	30.05 (db)	30.05 (db)
Man	HC	for 983,040 bit	for 983,040 bit	for 983,040 bit	for 983,040 bit
	PSNR	30.05 (db)	30.05 (db)	30.05 (db)	30.05 (db)
Peppers	HC	for 983,040 bit	for 983,040 bit	for 983,040 bit	for 983,040 bit
	PSNR	30.08 (db)	30.08 (db)	30.08 (db)	30.08 (db)
Couple	HC	for 983,040 bit	for 983,040 bit	for 983,040 bit	for 983,040 bit
	PSNR	30.06 (db)	30.06 (db)	30.06 (db)	30.06 (db)
Average	**HC**	**for 983,040 bit**	**for 983,040 bit**	**for 983,040 bit**	**for 983,040 bit**
	PSNR	**30. 05 (db)**	**30.05 (db)**	**30.05 (db)**	**30.05 (db)**

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

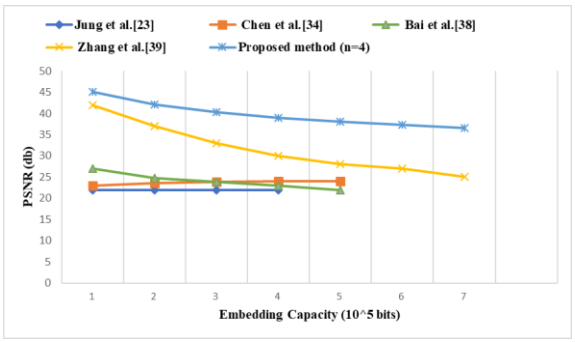
Table 5 - Comparison of PSNR and SSIM between the proposed method and the four related

Images	Metrics	Jung et al.[23]	Chen et al.[34]	Bai et al.[38]	Zhang et al.[39]	Proposed method
Lena	Capacity	199,091	305,644	585,225	622,812	786,432
	PSNR	31.1455	34.8739	29.9431	34.3648	36.05
	SSIM	0.8884	0.9354	0.8389	0.9225	0.8717
Baboon	Capacity	437,311	561,306	585,225	721,443	786,432
	PSNR	21.9165	24.0876	22.0269	24.8436	36.05
	SSIM	0.6871	0.8104	0.6424	0.8220	0.9289
Goldhill	Capacity	248,664	369,792	585,225	629,495	786,432
	PSNR	29.3897	32.2438	28.4892	32.3406	36.05
	SSIM	0.8244	0.8954	0.8325	0.9024	0.8910
Boat	Capacity	257,201	377,764	585,225	643,888	786,432
	PSNR	27.7151	32.2438	26.8833	30.9272	36.05
	SSIM	0.8108	0.8815	0.7677	0.8981	0.8727
Elaine	Capacity	234,020	355,971	585,225	612,120	786,432
	PSNR	30.6612	32.8834	29.9978	33.6922	36.05
	SSIM	0.7545	0.8282	0.7313	0.8687	0.8695
Man	Capacity	276,525	394,606	585,225	655,050	786,432
	PSNR	27.6780	31.3166	26.7049	30.8785	36.05
	SSIM	0.8548	0.9254	0.7903	0.9098	0.8943
Peppers	Capacity	196,094	307,957	585,225	621,170	786,432
	PSNR	29.7686	33.3563	28.5597	33.4011	36.05
	SSIM	0.511	0.9004	0.8095	0.9142	0.8691
Couple	Capacity	225,338	338,031	585,225	635,743	786,432
	PSNR	25.8746	28.0466	25.3406	28.5732	36.05
	SSIM	0.8459	0.9119	0.7851	0.9072	0.9022

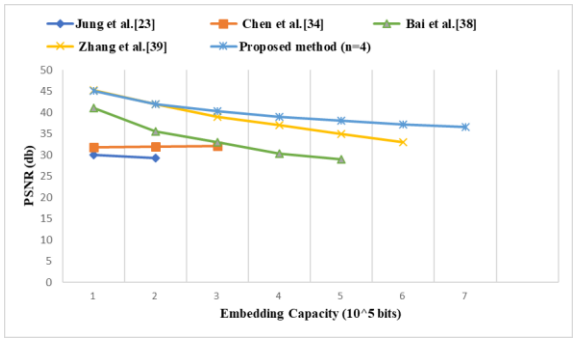
International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025



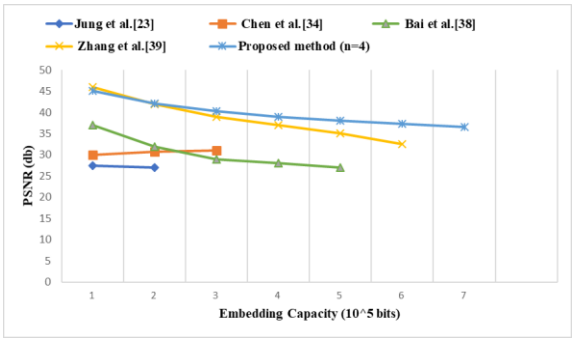
Lena



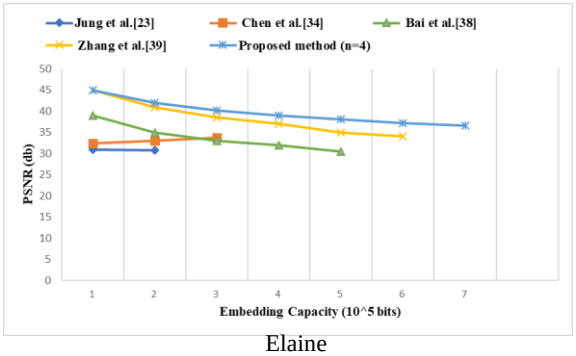
Baboon



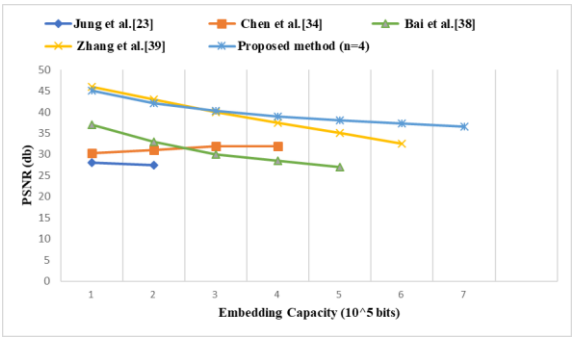
Goldhill



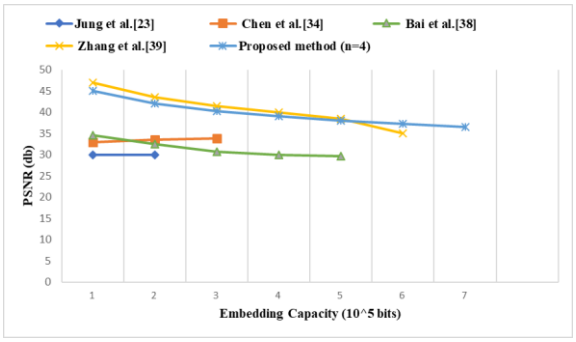
Boat



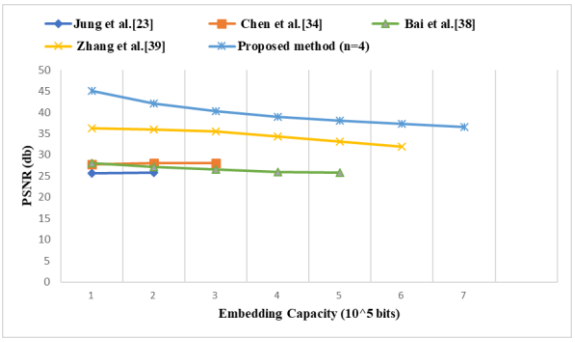
Elaine



Man



Peppers



Couple

Fig.13 Comparison between the proposed method and the four related methods [23, 34, 38, 39] in terms of embedding rates and PSNR values

International Journal of Innovation in Computer Science and Information Technology
Vol.5, NO.1, P:107-129
Received: 01 December 2023
Accepted: 16 March 2025

5. Conclusion

Here, a reversible data hiding scheme based on interpolation of images was proposed applying the stream of decimal signed digit. To this aim, the binary data stream is converted into that of signed digit with one stages of conversion. Creating a high EC, along with maintaining the visual quality in the data hiding process based on image interpolation, is among the advantages proposed method. The proposed method was simulated using eight standard images in the experiments. The same and satisfactory results were achieved for all images under critical interpolation techniques. The aforementioned method is employed to obtain a higher EC with high security. In the future work, we intend to increase the resistance of the stego image against all kinds of the intentional attacks by using the method of this paper and using the Hamming code based on proposed schemes in [42] and [43].

References

[1] I.J. Cox et al., Digital Watermarking and Steganography, 2nd ed., Burlington: Morgan Kaufmann, 2008
[2] Ran-Zan Wang, Chi-Fang Lin, Ja-Chen Lin, Hiding data in images by optimal moderately significant-bit replacement, IEE Electron. Lett. 36 (25) (2000) 2069–2070.
[3] J. Fridrich, R. Du,and M. Long, “Steganalysis of LSB encoding in color images,” in Proc. ICME 2000, New York, NY, USA, 31 July–2 August 2000.
[4] J. Fridrich, M. Goljan, and R. Du, “Reliable detection of LSB steganography in grayscale and color images,” in Proc. ACM Workshop on Multimedia and Security, pp. 27–30, Ottawa, Canada, October 2001.
[5] Jun Tian, “Reversible Data Embedding Using a Difference Expansion,” IEEE Trans. Circuits Syst. Video Technol, vol. 13, no. 8, pp. 890-896, Aug. 2003.
[6] Alattar AM. Reversible watermark using the difference expansion of a generalized integer transform. IEEE Transactions on Image Processing 2004;13(8):1147–56.
[7] Thodi DM, Rodriguez JJ. Expansion embedding techniques for reversible watermarking. IEEE Transactions on Image Processing 2007;16(3):721–30.
[8] Chin-Chen Chang, Ying-Hsuan Huang, Hsin-Yi Tsai, Chuan Qin . Prediction-based reversible data hiding using the difference of neighboring pixels. Int. J. Electron. Commun. (AEÜ) 66 (2012) 758– 766.
[9] Qin C, Chang CC, Liao LT (2012) An adaptive prediction-error expansion oriented reversible information hiding scheme. Pattern Recogn Lett 33(16):2166–2172
[10] Qu X, Kim S, Kim HJ (2015) Reversible watermarking based on compensation. J Electr Eng Technol 10(1):422–428
[11] Chang, C., Huang, Y. & Lu, T. A difference expansion based reversible information hiding scheme with high stego image visual quality. Multimed Tools Appl 76, 12659–12681 (2017).
[12] Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible Data Hiding. IEEE Transactions on Circuits and Systems for Video Technology 16(3), 354–362 (2006)
[13] Tsai PY, Hu YC, Yeh HL (2009) Reversible image hiding scheme using predictive coding and histogram shifting. Signal Process 89(6):1129–1143
[14] Hu YC, Tsai PY, Yeh JS, Chen WL (2015) Residual histogram shifting technique based on cascading prediction for reversible data hiding. Advanced multimedia and ubiquitous engineering, Berlin, Heidelberg, 2015, pp. 105–110
[15] Tang Z, Xu S, Ye D, Wang J, Zhang X, Yu C (2019) Real-time reversible data hiding with shifting block histogram of pixel differences in encrypted image. J Real-Time Image Proc 16(3):709–724
[16] X.-Z. Xie, C.-C. Chang, and Y.-C. Hu, ‘‘An adaptive reversible data hiding scheme

based on prediction error histogram shifting by exploiting signed-digit representation,'' Multimedia Tools Appl., vol. 79, nos. 33–34, pp. 24329–24346, Sep. 2020.
[17] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," Proc. IEEE, vol. 61, pp. 692-702, June 1973.
[18] Hou H, Andrews H. Cubic splines for image interpolation and digital filtering. IEEE Trans Acoust Speech Signal Process 1978;26(6):508–17.
[19] Einar Maeland, On the comparison of interpolation methods, IEEE Transactions on Medical Imaging 7 (3) (1988) 213–217.
[20] Jan Allebach, Ping Wah Wong, Edge-directed interpolation, IEEE International Conference on Image Processing 3 (1996) 707–710.
[21] Xin Li, Michael T. Orchard, New edge-directed interpolation, IEEE Transactions on Image Processing 10 (10) (2001) 1521–1527.

[22] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," IEEE Trans. Image Process., vol. 15, no. 8, pp. 2226–2238, Aug. 2006.
[23] K.H. Jung and K.Y. Yoo, Data hiding method using image interpolation, Computer Standards and Interfaces, 31(2)(2009) 465-470.
[24] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," IEEE Trans. Inf. Forensics Secur., vol. 5, no. 1, pp. 187–193, 2010.
[25] M.A.M. Abadi, H. Danyali, M.S. Helfroush, Reversible watermarking based on interpolation error histogram shifting, In: 5th International Symposium On Telecommunications (IST), Kish Island, Iran, 2010, pp. 840–845.
[26] Lee C-F, Huang Y-L. An efficient image interpolation increasing payload in reversible data hiding. Expert Syst Appl 2012;39:6712–9.
[27] 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 2013;66:1093–110. https://doi.org/10.1007/s11227-013-1016-6.
[28] Malik A, Sikka G, Verma HK. Image interpolation based high capacity reversible data hiding scheme. Multimed Tools Appl 2017;76:24107–23.
[29] Zhang X, Sun Z, Tang Z, et al. High capacity data hiding based on interpolated image. Multimed Tools Appl 2017;76:9195–218. https://doi.org/10.1007/s11042- 016-3521-0.
[30] Mohammad AA, Al-Haj A, Farfoura M. An improved capacity data hiding technique based on image interpolation. Multimed Tools Appl 2019;78(6):7181–205.
[31] Ye, Hanmin, Zhibo Li, and Lili Pu. "Research on Reversible Date Hiding Algorithms Based on Bilinear Interpolation about Watermark." In Proceedings of the 2020 3rd International Conference on E-Business, Information Management and Computer Science, pp. 592-596. 2020.
[32] Hassan FS, Gutub A. Efficient reversible data hiding multimedia technique based on smart image interpolation. Multimed Tools Appl 2020;79:30087–109. https:// doi.org/10.1007/s11042-020-09513-1.
[33] Malik A, Sikka G, Verma HK. A reversible data hiding scheme for interpolated images based on pixel intensity range. Multimed Tools Appl 2020;79:1–27.
[34] Chen Y-Q, Sun W-J, Li L-Y, et al. An efficient general data hiding scheme based on image interpolation. J Inf Secur Appl 2020;54. https://doi.org/10.1016/j. jisa.2020.102584.
[35] Hassan FS, Gutub A. Efficient image reversible data hiding technique based on interpolation optimization. Arabian J Sci Eng 2021;46:8441–56. https://doi.org/ 10.1007/s13369-021-05529-3.
[36] Xiong X, Wang L, Li Z, et al. An adaptive high capacity reversible data hiding algorithm in interpolation domain. Signal Process 2022;194. https://doi.org/ 10.1016/j.sigpro.2022.108458.
[37] Xiong X, Chen Y, Fan M, Zhong S. Adaptive reversible data hiding algorithm for

interpolated images using sorting and coding. J Inf Secur Appl 2022;66. https://doi.org/10.1016/j.jisa.2022.103137.
[38] Bai X, Chen Y, Duan G, et al. A data hiding scheme based on the difference of image interpolation algorithms. J Inf Secur Appl 2022;65. https://doi.org/10.1016/j.jisa.2021.103068.
[39] Zhang H , Sun H, Meng F. Reversible data hiding scheme based on improved interpolation and three-in-one intelligent strategy. J Inf Secur Appl 2023;77. https://doi.org/10.1016/j.jisa.2023.103573
[40] The USC-SIPI Image Database. Available: http://sipi.usc.edu/database/.
[41] Standard dataset images available at https://ccia.ugr.es/cvg/CG/base.htm
[42] Rasouli, Faeze, and Mohammad Taheri. "Perfect Recovery of Small Tampers Using a Novel Fragile Watermarking Technique Based on Distributed Hamming Code." ISeCure 14.2 (2022).
[43] Rasouli, Faeze, Mohammad Taheri, and Reza Rohani Sarvestani. "A Fragile Watermarking by Hamming Code on Distributed Pixels with Perfect Recovery for Small Tampers." ISeCure 15.2 (2023).