A High-capacity HE Interpolation-based Reversible Data Hiding Method using Difference Interval-based Substitution

Chia-Chen Lin[1], En-Ru Chen[2], Chin-Chen Chang[2]*

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taiwan
[2] Department of Information Engineering and Computer Science, Feng Chia University, Taiwan
ally.cclin@ncut.edu.tw, erchen.phd@gmail.com, alan3c@gmail.com

Abstract

With the growing emphasis on Internet data transmission security, an increasing number of scholars are exploring the field of data hiding. In the spatial domain, various techniques exist, such as pixel expansion, histogram shifting, and the application of interpolation techniques (IT). IT has garnered attention as a novel design approach, particularly in IT-based RDH (reversible data hiding), known for its ability to maintain relatively high image quality while enhancing hiding capacity compared to other methods.

However, previous research has predominantly focused on crafting interpolation images that closely mirror the original image quality, followed by proposing error prediction-based data hiding strategies. Departing from this conventional path, this study introduces the concept of single reference point diffusion to propose the **h**omogeneous **e**xpansion-based **i**nterpolation (HEI) algorithm and then applies the HEI algorithm to generate interpolation images less prone to offset pixel effects. Subsequently, the difference interval substitution (DIS) strategy is employed to maximize hiding capacity while preserving favorable image quality. Experimental results validate that our approach significantly boosts hiding capacity, nearly tripling that of the Zhang et al.'s method under comparable computational complexity, while maintaining an average image quality of 31.54 dB.

Keywords: Interpolation image, Homogeneous expansion, Difference interval substitution, Hiding capacity, Reversible data hiding

1 Introduction

As the range of multimedia data transmission applications on the Internet expands, they are no longer limited solely to forging personal social connections; instead, they are also being utilized for remote medical diagnosis, business data sharing, and even defense purposes. Consequently, the significance of data security is receiving heightened attention from both industry and academia. In response to users' increasing demand for information security, new methods, such as conventional cryptography [1], image encryption [2-5], or steganography also called image hiding [6-13] are constantly being designed by different research groups.

Among the above approaches, conventional cryptographic algorithms or image encryption transformed multimedia media into a meaningless format to protect the content. Conversely, image hiding conceals secret data in multimedia media to prevent the hidden secret data from being accessed by unauthorized users. Conventional data hiding methods typically involve irreversible processes, meaning that the original cover images cannot be completely restored even after extracting the hidden secrets. Consequently, their applications are restricted. In contrast, reversible data hiding (RDH) is more versatile, allowing for both data extraction and recovery, ensuring retrieval of both the extracted secret data and the original cover images. This reversibility feature expands the applicability of RDH to substantive domains like military, medical imagery, and telemedical applications. RDH research can be categorized into spatial-based algorithms, frequency-based algorithms, compression-based algorithms, and compression-based algorithms. Different from the rest categories, spatial-based RDH algorithms modify the pixel values of a cover image to conceal secret data. There are four representative data hiding strategies can be concluded in the last two decades regarding the spatial-based RDH algorithms: difference expansion (DE) [14-16], histogram shifting (HS) [17-21], pixel value ordering (PVO) [22-26] and interpolation technology (IT).

DE was proposed by Tian [14], and 1-bit of secret data is embedded by expanding the difference between two neighboring pixels in a cover image. The auxiliary data also called the location map is required to solve the overflow and underflow problem and ensure reversibility. In 2019, Wang et al. [16] designed a bidirectional DE. In their scheme, a cover image was scanned in a Z-shaped and transformed into a 1-D array. Then, the difference between two neighboring pixels in the 1-D array was expanded to two directions so that 1-bit of secret data could be concealed into the left pixel. Experimental results confirm the payload offered by the scheme of Wang et al. outperforms the schemes of Tian and other DE-based existing works [14]. The HS-based RDH was first designed by Ni et al. [17] to conceal secret data by considering a cover image from the perspective point of pixel distribution. By choosing the combination of peak and zero point pixels from the histogram, the secret data can be embedded into peak pixels

*Corresponding Author: Chia-Chen Lin; Email: ally.cclin@ncut.edu.tw
DOI: https://doi.org/10.70003/160792642026012701001

after shifting non-zero point pixels either by adding 1 or subtracting 1. Being inspired by Ni et al.'s idea, other researchers conducted different preprocessing, to ensure the highest peak of the histogram is as high as possible. Wu et al. [20-21] designed a strategy that shifts the peak pixels of the histogram to emphasize the image contrast of the stego image so that better visual quality can remain. The PVO-based RDH was proposed by Li et al. [22] in 2013. In their scheme, the data hiding process begins by dividing a cover image into predetermined non-overlapping blocks of equal size. Within each block, the pixels are sorted to determine the maximum and minimum values. Subsequently, the differences between the neighboring largest pixels and the neighboring smallest pixels are computed. This allows the second largest pixel to forecast the largest pixel, and similarly, the second smallest pixel can forecast the smallest pixel. Following this principle, a prediction-error expansion (PEE) method is devised to hide a single bit of secret data into the two largest or two smallest pixel pairs that can accommodate the embedding. The location map is constructed to handle the caused overflow and underflow after data hiding. Later, Qu et al. [23] focused on developing a predictor based on pixel value ordering. Their scheme tried to predict the pixel values within an image based on their ordering and successfully facilitated reversible embedding while maintaining high fidelity. Later, various improved PVOs were designed [24-26], such as the generalized directional PVO (DPVO) of Meikap et al.'s [24] or complexity adaptive PVO [25].

In addition to the three RDH data hiding strategies mentioned above, scholars have begun focusing on interpolation-based DRH since 2009. The first interpolation-based RDH was presented by Jung et al. [27]. They initially developed the neighboring mean interpolation (NMI) method, followed by the creation of a data concealment strategy to safeguard sensitive data while preserving the high visual fidelity of the steganographic image. Three years later, Lee et al. designed the image neighboring pixels (INP) method to improve Jung et al.'s scheme and enlarge the hiding capacity [28]. After that, numerous interpolation methods were proposed, such as enhanced NMI (ENMI) [29] and modified NMI (MNMI) [30]. Based on different interpolation methods, various interpolation-based RDH schemes were proposed [27-44]. For instance, in 2011, Hong et al. [42] crafted an RDH scheme relying on image interpolation and identifying smooth and complex regions within the cover images. To create a binary image that not only corresponds to the local image activity's smooth and complex regions but also indicates the position of reference pixels, they introduced a reference pixel distribution mechanism (RPDM). With the reconstructed binary image, the interpolated image can then be generated while maintaining better image quality of the interpolated image comparing with existing schemes. Later, 1-bit of secret data can be embedded into the interpolation error during data embedding.

In 2023, Zhang et al. [45] utilized circular predictor interpolation (CPI) to dynamically adjust the predictive radius, thereby enhancing prediction accuracy across various image textures during the generation of interpolated images. Within CPI, consideration is given to the distance between the predicted pixel and its reference pixel when determining the corresponding weights for various reference pixels. Subsequently, during the data embedding phase, to maintain superior visual quality in steganographic images, particularly at low payload levels, they initially calculated and arranged the complexity of non-reference pixels in ascending order. This prioritized the concealment of secret data into pixels with lower complexity. Finally, secret data was embedded into predicted pixels by adjusting the decimal value of the secret bits according to established data hiding rules. Experimental findings affirmed that Zhang et al.'s interpolation-based RDH scheme yielded a payload of approximately 620,000 bits while maintaining an image quality of around 30 dB, surpassing Bai et al.'s [34] offering by 10,000 bits. In essence, Zhang et al.'s scheme outperformed other existing schemes [29, 34-35, 37-38] in terms of image quality at equivalent payload levels.

2 Proposed Scheme

Upon careful examination of Zhang et al.'s scheme and other existing interpolation technology (IT)-based RDH schemes, it is apparent that the primary objective of the interpolation technique is to generate predicted pixels based on the corresponding reference pixels, intending to make the interpolated image closely resemble the original. Assessing the performance of an IT involves down-sampling an original image (*OI*) of size $W \times H$ to obtain a down-sampled image of size $W/2 \times H/2$, followed by computing an interpolated image (*II*) of size $W \times H$ using interpolation methods such as NMI [31], among others. The aim is to minimize the Mean Squared Error (MSE) between the *II* and *OI*, thereby maximizing the image quality of *II*. In simpler terms, the better the interpolation technique performs, the closer the image quality of the interpolated image is to that of the original.

Given that current interpolation methods primarily rely on neighboring reference pixels for calculating predicted pixel values, the design core concept of these interpolation algorithms lies in determining the appropriate reference pixels and their corresponding weights. To simplify the optimization calculation of predicted pixels, making this IT-based RDH scheme more suitable for real-time applications, while also aiming to ensure that the generated interpolated image closely approximates the pixel values of the original image, in this paper, we propose a predicted pixels calculation method centered around reference pixels, using the concept of homogeneous expansion. The homogeneous expansion is different from the heterogeneous expansion, in the former, a predicted pixel only refers to a single reference pixel; and in the latter, a predicted pixel may refer to multiple reference pixels under specific circumstances, such as [45]. For ease of explanation, the proposed **h**omogeneous **e**xpansion-based **i**nterpolation method will be referred to as HEI in the subsequent text and the detailed description will be given in Subsection 2.1. As for the data embedding, extraction and recovery will be presented in Subsections 2.2 and 2.3, respectively.

With our proposed HEI algorithm, the framework of HEI-based RDH scheme is presented in Figure 1.

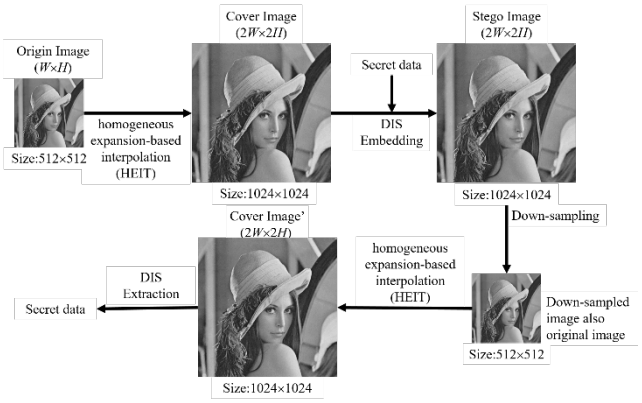

Figure 1. Framework of our proposed HEI-based RDH scheme

As depicted in Figure 1, our proposed RDH scheme based on HEI is structured into two phases: data embedding and data extraction. In the data embedding phase, the original image (*OI*) with dimensions $W\times H$ undergoes expansion into a cover image of size $2W\times 2H$ via our HEI algorithm. Subsequently, our devised difference interval-based substitution method (DIS) is employed to embed secret data into predicted pixels within the cover image. This process yields the stego image, which can then be transmitted to the recipient. In the data extraction phase, the recipient down samples the received stego image and applies the HEI algorithm to expand it back into the re-expanded cover image. Finally, DIS extraction is executed to retrieve the concealed secret data.

2.1 HEI Interpolation Method

The core concept of our proposed HEI algorithm is homogeneous expansion, which means each predicted pixel only has a single reference pixel marked in the gray cells as shown in Figure 2(a). An original image (*OI*) is divided into a non-overlapping block size of 2×2, and each 2×2 block in *OI* is then expanded to an interpolation block with our HEI algorithm. Take a 2×2 *OI* for example, it is expanded to a 4×4 interpolation image which will serve as a cover image (*CI*) with our proposed HEI algorithm in our HEI-based RDH scheme. In other words, a 4×4 interpolation image contains 4 reference pixels (RPs) and 12 interpolation pixels (IPs) as shown in Figure 2(b).

units, and each pixel located at the right-top position in a unit is defined as a reference pixel (*RP*) and is mapped to the corresponding original pixel in *OI* according to Eq. (1).

$$\begin{cases} CI(x+1,y)=OI(i,j) \\ CI(x+3,y)=OI(i+1,j) \\ CI(x+1,y+2)=OI(i,j+1) \\ CI(x+3,y+2)=OI(i+1,j+1) \end{cases}, \quad (1)$$

where i and j are coordinates in *OI*, x and y are coordinates in *CI*. Refer to the equation specified in Eq. (2), $CI(2,1)=OI(1,1)=10$, $CI(4,1)=OI(2,1)=16$, $CI(2,3)=OI(1,2)=25$ and $I(4,3)=OI(2,2)=118$. The value of a reference pixel is scattered to three neighboring interpolation pixels, with three different weights. Three weights are computed according to 2^L where $L=\log_2^{RP}-1$ and *RP* indicates the reference pixel without considering the neighboring pixels in the *OI*. For example, in Figure 2, $CI(2,1)=OI(1,1)=10$ serves as the reference pixel $RP=10$ and the $L=\log_2^{RP=10}-1=2$. Finally, three weights can be calculated as 0, 2, and 4 for three interpolation pixels located in $CI(2,1)$, $CI(1,2)$ and $CI(2,2)$, respectively. After working with three predetermined weights, three interpolation pixels can be derived according to Eq. (2) when $CI(x,y)$ serves as the reference pixel (*RP*).

$$\begin{cases} CI(x,y)=CI(x,y)=RP \\ CI(x-1,y)=CI(x,y)-2^{L-2} \\ CI(x-1,y-1)=CI(x,y)-2^{L-1} \\ CI(x,y-1)=CI(x,y)-2^{L} \end{cases}, \quad (2)$$

where x and y are coordinates in *CI* and $L=\log_2^{RP}-1$ serves as the scattering weight of interpolation pixel for a given reference pixel in *CI*. Take $CI(4,3)=OI(2,2)=118$ served as refence pixel $RP=118$ for another example. $L=\log_2^{RP=118}-1=5$. Therefore, the three weights for the interpolation pixels located in $CI(3,3)$, $CI(3,4)$ and $CI(4,4)$ becomes 8, 16, and 32, respectively. After working with three predetermined weights, three interpolation pixels can be derived according to Eq. (2). $CI(3,3)=118-2^{5-2}=110$, $CI(3,4)$ $118-2^{5-1}=102$ and $CI(4,4)$ $118-2^{5}=86$. By employing Eqs. (1) and (2), finally, the interpolation image also called cover image (*CI*) in our HEI-based RDH scheme is obtained.

However, the interpolation equation defined in Eq. (2) may cause underflow or overflow when the values of RP are either close to 0 or 255. To prevent underflow or overflow from being occurred during the interpolation procedure. Four additional cases in which Eq. (2) can not be applied to generate interpolation pixels (*IP*s) are listed below:

Case 1. If $RP=0$, Eq. (2) is redefined as Eq. (3).

$$\begin{cases} CI(x,y)=CI(x,y)=RP \\ CI(x-1,y)=CI(x,y)-TH_1 \\ CI(x-1,y-1)=CI(x,y)-TH_2 \\ CI(x,y-1)=CI(x,y)-TH_3 \end{cases}, \quad (3)$$

where TH_1, TH_2 and TH_3 are set as 0, 0, and 1, respectively.

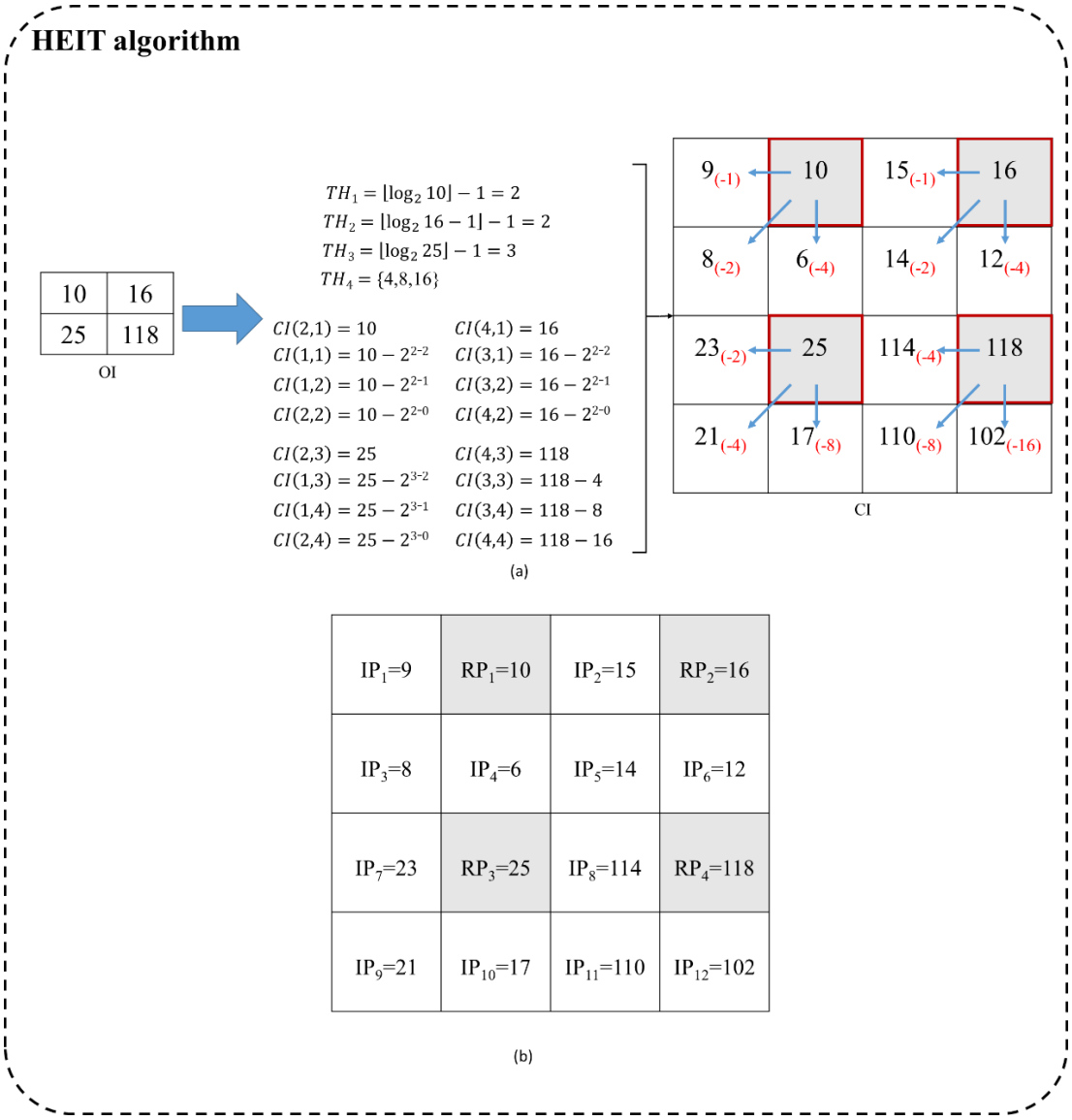

Figure 2. Example of image interpolation with our proposed HEI algorithm

Case 2. If $0 < RP \leq 4$, Eq. (2) is redefined as Eq. (4).

$$\begin{cases} CI(x,y) = CI(x,y) = RP \\ CI(x-1,y) = CI(x,y) - TH_1 \\ CI(x-1,y-1) = CI(x,y) - TH_2 \\ CI(x,y-1) = CI(x,y) - TH_3 \end{cases}, \quad \textbf{(4)}$$

where TH_1, TH_2 and TH_3 are three distinct values of {0, 1, 2}, respectively.

Case 3. If $4 < RP \leq 8$, Eq. (2) is redefined as Eq. (5).

$$\begin{cases} CI(x,y) = CI(x,y) = RP \\ CI(x-1,y) = CI(x,y) - TH_1 \\ CI(x-1,y-1) = CI(x,y) - TH_2 \\ CI(x,y-1) = CI(x,y) - TH_3 \end{cases}, \quad \textbf{(5)}$$

where TH_1, TH_2 and TH_3 are three minimal values of {0, 2^1, 2^2}, respectively.

Case 4. If $64 < RP$, Eq. (2) is redefined as Eq. (6).

$$\begin{cases} CI(x,y) = CI(x,y) = RP \\ CI(x-1,y) = CI(x,y) - TH_1 \\ CI(x-1,y-1) = CI(x,y) - TH_2 \\ CI(x,y-1) = CI(x,y) - TH_3 \end{cases}, \quad \textbf{(6)}$$

where TH_1, TH_2 and TH are three distinct values of {4, 8, 16}, respectively. From the four aforementioned special cases, it's evident that Eq. (2) exclusively governs the generation of interpolation pixels when the condition $8<RP\leq64$ holds true.

2.2 Data Embedding

Once the interpolation image is obtained based on our proposed HEI algorithm, the interpolation image becomes

a cover image (*CI*) in our HEI-based RDH scheme. To guarantee the reversibility of the *OI*, all reference pixels in *CI* remain unchanged during data embedding. The secret data only are concealed in the interpolation pixels in the *CI*.

In order to augment the concealment capacity while preserving the image quality of the stego image, we have developed a method named Difference Interval-based Substitution (DIS) for embedding secret data. Below, we outline the detailed steps for embedding using DIS:

Algorithm. Data embedding

Input: Cover image *CI* and secret data *SD*.

Output: Stego image *SI*.

Step 1. Divide *CI* into 4×4 non-overlapping blocks.

Step 2. For a 4×4 sized block, read pixels from the left-top to the right-bottom with Zig-Zag scan.

Step 3. If the current pixel is a reference pixel *RP*, output it and move to the next pixel.

Step 4. If the current pixel is an interpolation pixel IP_l, where l is the label of *IP* and ranged between 1 and 12 for a block size of 4×4 as shown in Figure 2(b), check if it is larger than 249. If it is, set n_l = 0 and go to Step 8. If it is not, calculate its total difference $D_l = \Sigma_{k=1}^{n} \left| IP_l - RP_k \right|$ where l is the label of *IP* and k is the number of its neighboring *RP* for the given IP_l.

Step 5. Compute the upper bound $n_0 = \log_2^{\lfloor D_0/k \rfloor}$ of the hiding bits for the given IP_l, where l is the label of *IP* and $D_0 = |IP_l - RP_{l_0}|$. It is noted that for IP_l may have k neighboring *RP*s surrounding it but one and only one *RP* is used to generate the value of IP_l and such *RP* is denoted as IP_{l_0}.

Step 6. Compute the amount of the hiding bits $n_l = \log_2^{\lfloor D_l/k \rfloor}$ for the given IP_l where l is the label of *IP* and ranged between 1 and 12 for a block size of 4×4.

Step 7. If n_l is larger than n_0, then only read n_o secret bits from *SD*.

Step 8. Convert the read n_o secret bits into the decimal value denoted as M_l, where l is the label of *IP* and ranged between 1 and 12 for a block size of 4×4.

Step 9. Compute the stego $IP_l' = IP_l + M_l$ and output the stego IP_l', where l is the label of *IP* and ranged between 1 and 12 for a block size of 4×4.

Step 10. Check if the current *IP* is the last pixel within the current block. If it is, go to Step 12. Otherwise, move to the next pixel and go to Step 11.

Step 11. Repeat Steps 2 to 10 until all pixels have been proceeded in the current block size of 4×4.

Step 12. Check if the current block is the last block in *CI*. If it is not, move to the next block and go to Step 2. Otherwise, go to Step 12.

Step 13. Collect all outputted stego *IPs* and finally form a stego image *SI*.

It is noted that in the 4×4 non-overlapping blocks in *CI*, there are four reference pixels marked with red frames and 12 interpolation pixels as shown in Figure 2. Only 12 interpolation pixels generated with our HEI algorithm are used in our DIS hiding algorithm. Once our DIS hiding is completed and the stego image *SI* is obtained, the data owner sends the generated stego image to the recipient via the Internet to share the hidden secret data with the recipient.

2.3 Data Extraction and Recovery

Upon receiving the steganographic image, the recipient can first restore the *OI* using the following steps. Subsequently, they can employ the data extraction algorithm to retrieve the concealed *SD*.

Algorithm. Recovery of the original image

Input: Stego image *SI*

Output: Original image *OI′*

Step 1. Divide *SI* into 4×4 non-overlapping blocks.

Step 2. For a 4×4 sized block, read pixels from the left-top to the right-bottom with Zig-Zag scan.

Step 3. Divide each 4×4 sized block into 4 non-overlapping 2×2 sized units and only output the pixel located at the right-top position from a 2×2 sized unit.

Step 4. Repeat Step 3 until all 2×2 sized units have been proceeded.

Step 5. Repeat Steps 2 to 4 until all 4×4 sized blocks have been proceeded.

Step 6. Collect all outputted pixels and form the restored orginal image *OI′*.

Once the original image *OI*'s is restored with the recovery of the original image algorithm, the hidden secret data can be extracted according to the following steps:

Algorithm. Data extraction

Input: Stego image *SI*, restored original image *OI′*

Output: Secret data *SD*

Step 1. Conduct HEI Interpolation method described in Subsection 2.1 on the restored original image *OI′* and then generate the reconstructed cover image *CI′*.

Step 2. Divide the reconstructed cover image *CI*'s into non-overlapping 4×4 sized blocks.

Step 3. Divide the stego image *SI* into non-overlapping 4×4 sized blocks.

Step 4. Read the stego interpolation pixel IP_l' with Zig-Zag scan from a 4×4 sized block of the *SI* where l is the label of *IP′* and ranged between 1 and 12 for a block size of 4×4.

Step 5. For the stego interpolation pixel IP_l', check if IP_l' is larger than 249. If it is, set n_l = 0 and go to Step 9. If it is not, calculate its total difference D_l

$= \Sigma_{k=1}^{n} \left| IP'_l - RP_k \right|$ where l is the label of IP'_l and k is the number of its neighboring RP for the given IP_l.

Step 6. Compute the upper bound $n_0 = \log_2 \lfloor D_0 / k \rfloor$ of the hiding bits for the given IP'_l where l is the label of IP' and $D_0 = |IP'_l - RP'_{l_0}|$. It is noted that for IP'_l may have k neighboring RPs surrounding it but one and only one RP is used to generate the value of IP'_l and such RP' is denoted as IP'_{l_0}.

Step 7. Compute the upper bound $n_0 = \log_2 \lfloor D_0 / k \rfloor$ of the hiding bits for the given IP'_l where l is the label of IP' and $D_0 = |IP'_l - RP'_{l_0}|$. It is noted that for IP'_l may have k neighboring RPs surrounding it but one and only one RP is used to generate the value of IP'_l and such RP' is denoted as IP'_{l_0}.

Step 8. Compute the amount of the hiding bits $n_l = \log_2 \lfloor D_l / k \rfloor$ for the given IP'_l where l is the label of IP' and ranged between 1 and 12 for a block size of 4×4.

Step 9. If n_l is larger than n_0, then only n_o secret bits have been concealed in the current IP'_l.

Step 10. Read the reconstructed interpolation pixel IP''_l with Zig-Zag scan from 4×4 sized block of the reconstructed cover image CI' where l is the label of IP'' and ranged between 1 and 12 for a block size of 4×4.

Step 11. Compute the hidden secret message $M'_l = IP'_l - IP''_l$ where l is ranged between 1 and 12.

Step 12. Convert the extracted secret message M'_l into the binary format according to the number of n_l and output the binary bitstream.

Step 13. Repeat Steps 4 to 11 until all non-overlapping 4×4 sized blocks of the SI and CI' have been proceeded, respectively.

Step 14. Collet all outputted binary streams to obtain the extracted secret data SD.

2.4 Example of Data Embedding and Extraction

In this subsection, we elucidate our proposed strategy for concealing data, extracting it, and subsequently recovering it. To illustrate our reversible data hiding scheme, we employ a straightforward example, building upon the 2×2 image block introduced in our previous HEI interpolation example (Figure 3).

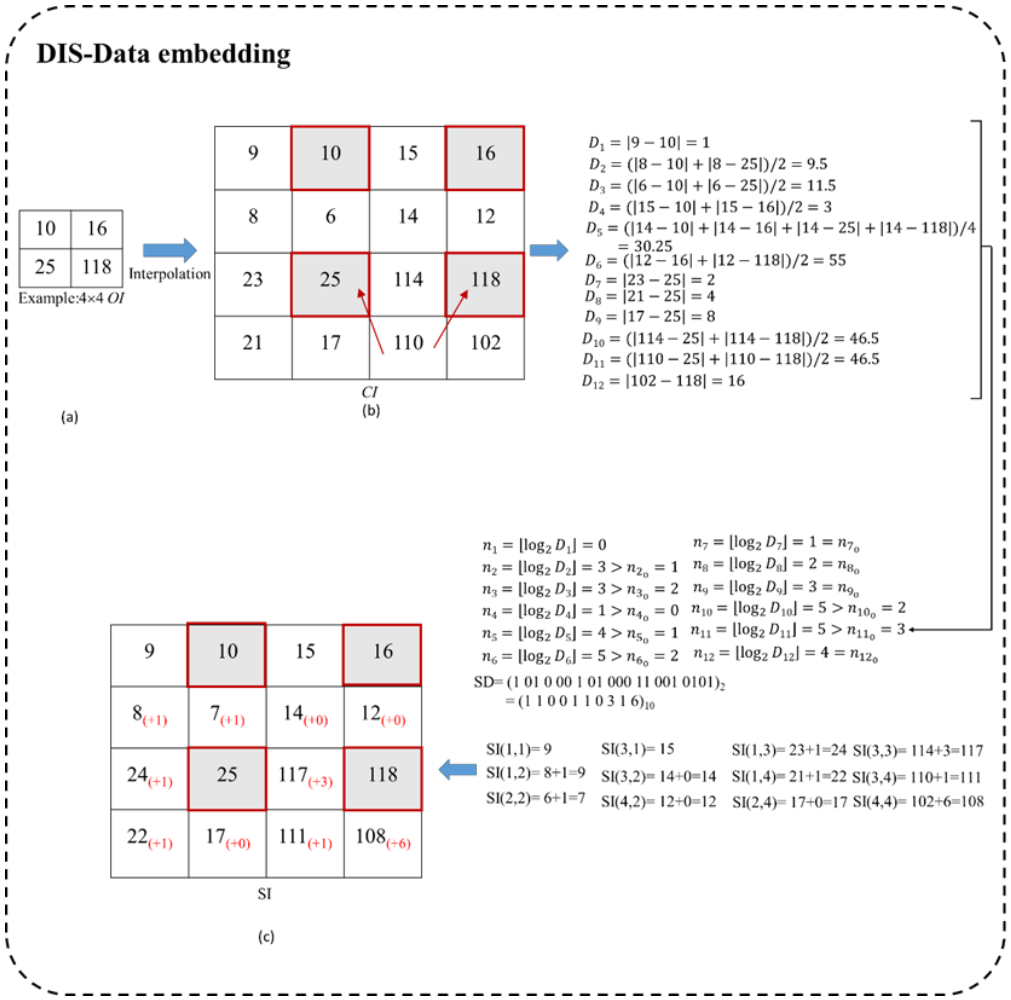

Figure 3. Example of DIS data hiding

Here, we assume an original image (*OI*) is depicted in Figure 3(a), consisting of four pixel values: 10, 16, 25, and 118. Moreover, we consider the secret data (*SD*) represented as (1 01 0 00 1 01 000 11 001 0101)$_2$. Utilizing the 4×4 cover image (*CI*) illustrated in Figure 3(b), we can generate the stego image (*SI*), as displayed in Figure 3(c).

Let's consider the case of RP=118 as an example. With our proposed HEI algorithm, its three interpolated pixels (*IP*s) are derived as 114, 110, and 102, respectively. Following the generation of these *IP*s, the DIS-hiding algorithm is executed. Taking IP_{11}=110 as an instance, it has two neighboring reference pixels (*RP*s), namely RP_3=25 and RP_4=118. Thus, the difference $D_{11} = |IP_{11} - RP_3| + |IP_{11} - RP_4| = |110-25| + |110-118| = 93$ and the average difference is $D_{11}/2 = 46.5$. Consequently, the number of hiding bits for IP_{11}, denoted as n_{11}, is computed as $n_{11} = \left\lfloor \log_2^{D_{11}/2} \right\rfloor = \left\lfloor \log_2^{46} \right\rfloor = 5$. For IP_{11}=110, its original RP is RP_4 and $D_0 = |IP_{11} - RP_{11_0}| = |110-118| = 8$. Thus, $n_{11_0} = \left\lfloor \log_2^{8} \right\rfloor = 3$. In essence, only 3 secret bits $(001)_2$ are extracted from *SD* and converted into a decimal value $(1)_{10}=(001)_2$ to generate the stego *IP* as $=110+1=(111)_{10}$.

Following similar operations, the stego image (*SI*) can finally be obtained. Regarding our data extraction and recovery process, it is rather straightforward. Figure 4 shows the example. Initially, the reconstructed original image (*OI*′) can be restored by gathering the reference pixels from the received stego image (*SI*). Once *OI*′ is reconstructed, the HEI interpolation algorithm is reapplied to generate a cover image (*CI*′) devoid of any embedded secret data.

By calculating the difference between each stego *IP*′ and its corresponding reconstructed *IP*″, and then converting this difference into a binary stream whose length is determined by *n* derived from each stego *IP*′, the process proceeds. For instance, considering IP''_{11}= 111, IP''_{11}= 11, the difference between them is $D_{11} = |IP'_{11} - IP''_{11}|$. Following Steps 5-8 in the data extraction procedure, it becomes evident that n_{11} = 5 is greater than or equal to n_{11_0}= 3. This implies that even though D_{11}= |111−110|=1, it still needs to be converted into 3-bits secret data representation, such as $(001)_2$, for the extracted secret data.

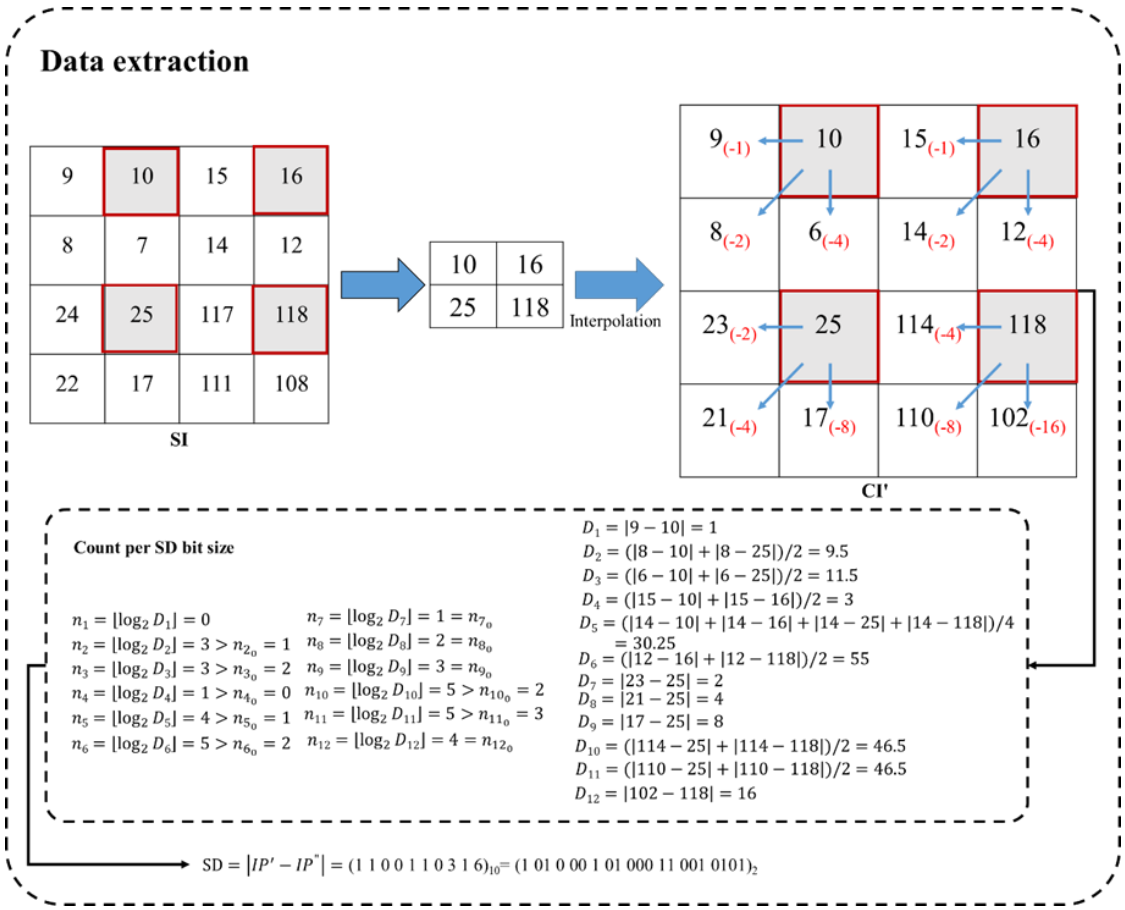

Figure 4. Example of DIS data extraction

3 Experimental Analysis and Discussions

In Section 3, we showcase the effectiveness and resilience of our approach through comprehensive evaluations. We utilized a randomly selected set of 1000 images from BOSSbase [46] as our test dataset. The implementation of our method was carried out using MATLAB R2022b on a Windows 10 laptop, equipped with an Intel i5-8300H 2.3GHz CPU, a 2GB GeForce GTX 1050 GPU with VRAM, and 8GB of RAM.

In our HEI-based reversible data hiding (RDH) scheme, aiming to maintain competitive visual quality in stego images while augmenting hiding capacity, we initially devised a novel interpolation algorithm termed HEI. Subsequently, we developed a data hiding technique named DIS to embed secret data into the cover image produced by HEI. Hence, three experiments were conducted. Firstly, we evaluated the visual quality of our HEI algorithm. Secondly, we assessed the maximum hiding capacity and its associated image quality achievable through the integration of our HEI and DIS algorithms. Finally, in the third experiment, we compared our proposed scheme with existing interpolation-based RDH schemes [28, 32-35, 45].

To ensure impartial evaluation, we employed two distinct datasets across our experiments. The first dataset comprised eight general grayscale images, each sized as 512x512, including Lena, Goldhill, Baboon, Boat, Man, Elaine, Peppers, and Couple, as depicted in Figure 5. The second dataset encompassed 1000 images randomly selected from BOSSbase [46], validating the overall performance of our proposed scheme in terms of hiding capacity and visual quality. Figure 6 illustrates eight images chosen from the 1000 BOSS test images.

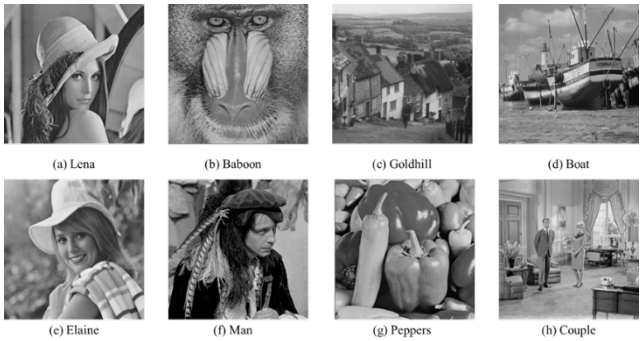

Figure 5. Eight 512×512 grayscale test image

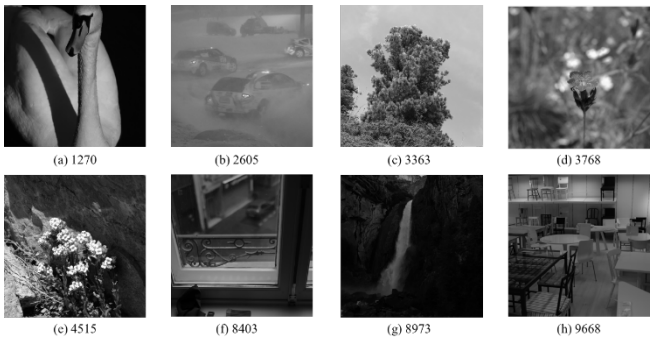

Figure 6. Eight random images selected from the BOSSbase

The assessment of a reversible data hiding (RDH) scheme involves two primary analyses: imperceptibility analysis and hiding capacity evaluation. These analyses are tailored to assess the performance of our proposed scheme in terms of both the visual quality of stego images and the maximum amount of concealed secret data. Peak Signal-to-Noise Ratio (PSNR), defined in Eqs. (7) and (8), serves as the primary metric for evaluating imperceptibility. PSNR acts as an indicator of the visual fidelity of the stego image compared to the original cover image. Typically, a PSNR exceeding 30 dB suggests that the differences between the cover image and its stego counterpart are imperceptible to the human visual system.

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE}, \tag{7}$$

$$MSE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \left(CI_{ij} - SI_{ij} \right)^2, \tag{8}$$

where CI and SI represent the cover image and stego image, respectively, H denotes the height and W denotes the width of the image, respectively.

Besides PSNR, there are three measurements also used to evaluate the similarity between the stego image and the original image: Structural Similarity Index (SSIM), Mean Absolute Error (MAE), and Normalized Cross-Correlation (NCC). Four measurements are defined in the following paragraphs. Structural Similarity Index (SSIM) is the commonly used metric used to evaluate the similarity between the stego image and the original image. As depicted in Eq. (9), the SSIM algorithm meticulously evaluates likeness and divergence by independently evaluating luminance, contrast, and structure among two images, x and y.

$$SSIM(x, y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}, \tag{9}$$

These components are then harmoniously combined using weighted products. Here, μ_x and μ_y correspond to the average luminance, σ_x and σ_y indicate the luminance's standard deviation, and σ_{xy} represents the covariance between images x and y. Furthermore, C serves as a constant to maintain stability. SSIM is ranged between 0 and 1. A higher SSIM value indicates a higher similarity between the two images. Conversely, the lower the SSIM value is, the less similar the two images are.

Mean Absolute Error (MAE), as defined in Eq. (10), calculates using absolute values, enhancing its intuitiveness and reducing sensitivity to outliers, thereby contributing to a more robust computation. The specific range of MAE may vary depending on factors such as image characteristics and the level of similarity under consideration. However, in general, a low MAE indicates a high degree of similarity between the images. Normalized Cross-Correlation (NCC), defined in Eq. (11), offers computational flexibility as it remains unaffected by brightness and can accommodate various shapes and sizes, making it more adaptable compared to other metrics. NCC values range between -1 and 1, with an NCC value close to 1 indicating a higher degree of similarity between the two images.

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \left| CI_{ij} - SI_{ij} \right|. \tag{10}$$

$$NCC(I_1, I_2) = \frac{\Sigma_{ij}\left(I_1(i,j)-\mu_{I_1}\right)\left(I_2(i,j)-\mu_{I_2}\right)}{\sqrt{\Sigma_{ij}\left(I_1(i,j)-\mu_{I_1}\right)^2 \Sigma_{ij}\left(I_2(i,j)-\mu_{I_2}\right)^2}}. \quad (11)$$

3.1 Experimental Results of Different Interpolation Methods

In order to enhance the adaptability of our proposed interpolation algorithm for real-time applications and allow sufficient space for embedding secret data while preserving comparable visual quality in the interpolated image, we restrict the interpolation pixel generation to referencing only one reference pixel. Within our Homogeneous Expansion Interpolation (HEI) algorithm, each interpolation pixel is exclusively tied to a single reference pixel (RP), ensuring a consistent correlation between the generated interpolation pixels and their reference counterparts. Moreover, it remains unaffected by surrounding reference pixels. Essentially, even if there are few outlines present, the interpolation pixels remain largely unaffected. These characteristics are evident in the visual quality comparison between our HEI algorithm and five existing interpolation algorithms: NMI [27], INP [28], ENMI [29], MNMI [30], and CPI [45].

Table 1. Performance comparisons of PSNRs (dB) for interpolation

Images	NMI	INP	ENMI	MNMI	CPI	Proposed
Lena	31.81	30.70	33.44	31.98	32.84	28.89
Baboon	22.23	21.72	22.67	22.43	22.96	28.92
Goldhill	29.84	29.10	30.68	29.94	30.49	28.95
Boat	28.16	27.39	29.15	28.27	28.82	29.08
Elaine	30.99	30.34	31.80	31.62	31.75	28.89
Man	28.27	27.33	29.43	28.26	28.94	30.37
Peppers	29.88	28.82	31.48	30.48	31.31	29.19
Couple	25.67	25.03	26.56	26.06	26.93	29.05
Average	28.36	27.55	29.40	28.63	29.26	29.17

Considering the PSNR values provided in Table 1, let's use ENMI as an example. Its highest PSNR reaches 33.44 dB, while its lowest is 22.67 dB. In contrast, our proposed HEI exhibits an average PSNR of 29.17 dB, closely resembling that of ENMI (29.40 dB) and CPI (29.26 dB), respectively. Notably, our HEI consistently sustains near-average image quality across various image types and excels beyond other interpolation algorithms, particularly in challenging scenarios where they commonly falter.

Specifically, when evaluating the image quality of other methods across three images—Baboon, Man, and Couple—the performance notably falls below the respective method's average level. For instance, ENMI achieves PSNR values of 22.67 dB, 29.43 dB, and 26.57 dB in Baboon, Man, and Couple, respectively. In contrast, our HEI method surpasses ENMI by 6 dB, 0.9 dB, and 3 dB in Baboon, Man, and Couple, respectively. Additionally, our HEI algorithm's computational complexity is reduced due to its utilization of only a single reference pixel to generate the interpolation pixel, distinguishing it from other existing interpolation algorithms.

3.2 Experimental Results of Our HEI-based RDH Scheme

In Figure 7, it is evident that even with a hiding capacity of up to 1,000,000 bits, the image quality of the seven test images, with the exception of Man, remains consistently above 34 dB. Conversely, when the hiding capacity is limited to 70,000 bits, the image quality of all eight images falls within the range of 34 dB to 38 dB.

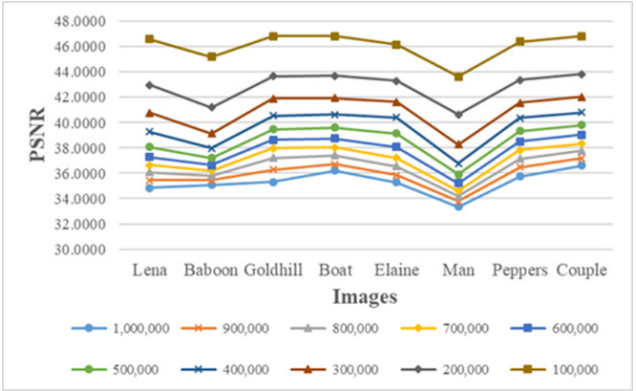

Figure 7. PSNRs of the eight test grayscale images under ten different sizes of the secret data

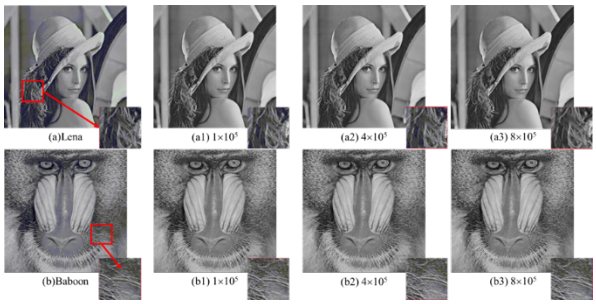

Figure 8. The SI with different embedding capacity

To demonstrate that our DIS-hiding method preserves the textures in a cover image, Figure 8 displays the stego image alongside magnified views of a section of the image texture area at three distinct hiding capacities. It is evident that despite an increase in the number of concealed secret bits, the selected and enlarged image texture area retains a commendable level of visual quality, as depicted in Figure 8(3a) and Figure 8(3b).

Finally, the detailed discussions of our proposed scheme on visual quality (PSNR, MAE, NCC, SSIM) and hiding capacity (ER, capacity, efficiency) with eight general test images are listed in Table 2. From Table 2, we can see the average ER is around 2 bpp while maintaining the average image quality of the stego image as 32.00 dB. Even the average hiding capacity is up to 2,130,723 bits, the NCC is 0.99 and SSIM is also 0.85. The NCC value of 0.99 indicates a strong similarity between the intensity patterns of the interpolation image and stego image. It also means that the pixel values in corresponding locations across the images are highly correlated. In other words, the images have similar brightness and contrast characteristics. As for the SSIM value of 0.85 when the average hiding

capacity is up to 2,130,723 bits, it indicates even though the intensity patterns of the interpolation image and stego image are highly correlated, there might be differences in the finer details, such as textures, sharpness or visual fine features of these two images that affect their SSIM value. However, the SSIM is up to 0.9 when the hiding capacity is the same as that of Zhang et al. [45] shown in Table 4. Efficiency denotes the ratio of payload to the size of the steganographic image in bits. Conversely, ER (Embedding Rate) is the ratio of payload bits to the number of pixels in the stego image. Both metrics serve to assess the effectiveness of data hiding methods. Greater efficiency or a higher ER signifies superior hiding performance of the proposed data method.

To demonstrate the resilience of our proposed scheme across diverse data hiding scenarios and considerations of visual quality, we present experimental findings on hiding capacity, image quality, and SSIM for 1000 randomly selected test images from the BOSS dataset [46] in Table 3. The table reveals an average hiding capacity of approximately 1,855,390 bits, accompanied by an average PSNR of 30.56 dB and an average SSIM of 0.79. Table 3 shows that even with the highest capacity of 2,243,409 bits, the SSIM value remains at 0.42 when tested against 1000 randomly selected images from the BOSS dataset [46]. This outcome can be attributed to the test set's diverse nature, encompassing a wide range of images from the BOSS database. By combining the experimental data outlined in Table 2 and Table 3, we can affirm the overall effectiveness of our proposed DIS-data hiding strategy in terms of both data concealment and image fidelity.

Table 2. Performance comparison for data embedding capacity (bits), PSNR (dB), ER (bpp), MAE, NCC, efficiency, SSIM

Images	PSNR	Capacity	ER	MAE	NCC	Efficiency	SSIM
Lena	30.68	2,166,021	2.07	4.69	0.99	0.26	0.80
Baboon	32.06	2,217,582	2.11	3.98	0.99	0.26	0.92
Goldhill	30.76	2,153,115	2.05	4.66	0.99	0.26	0.82
Boat	31.65	2,129,638	2.03	4.22	0.99	0.25	0.86
Elaine	32.21	2,177,164	2.08	3.97	0.99	0.26	0.85
Man	31.28	1,699,064	1.62	4.14	0.99	0.20	0.87
Peppers	35.50	2,077,051	1.98	4.17	0.99	0.25	0.83
Couple	31.83	2,140,723	2.04	4.12	0.99	0.25	0.86
Average	32.00	2,095,045	2.00	4.24	0.99	0.25	0.85

Table 3. Performance comparison of 1000 random images from BOSS based on hiding capacity (bits), PSNR (dB), and SSIM

Metrics	Embedding capacity	PSNR	SSIM
Max	2,243,409	25.29	0.42
Min	428,485	38.95	0.95
Average	1,855,390	30.56	0.79

3.3 Experimental Results of Different RDH Schemes

To showcase the superior performance of our HEI-based RDH scheme compared to existing methods, we select six representative IT-based RDH schemes. Table 4 presents a performance comparison between these six existing representative methods and ours.

In Table 4, Zhang et al. [45] demonstrated superior performance in terms of payload and visual quality of stego images compared to the other five existing schemes. Furthermore, our HEI-based RDH achieved an average payload of 2,095,044 bits, nearly three times higher than Zhang et al. (642,715 bits) [45], while maintaining an average PSNR of 31.54 dB, comparable to Zhang et al. (31.13 dB) [45]. When the hiding capacity is fixed to match that of Zhang et al. [45], our method exhibits an average PSNR 6 dB higher than that of Zhang et al.

4 Conclusions

In this paper, we introduce the HEI algorithm, which produces an interpolation image while preserving the integrity of reference pixels with outlier values, thus ensuring stable image quality. Leveraging the stability of the interpolation image generated by HEI, we develop a dynamic DIS hiding strategy for our HEI-based RDH scheme.

Unlike the HEI algorithm, the DIS hiding strategy considers neighboring reference pixels when determining the amount of secret data to hide for a given interpolation pixel. This approach prevents overflow by considering the disparity between the interpolation pixel and its original reference pixel. Experimental findings confirm that our proposed scheme, integrating the HEI algorithm and DIS hiding strategy, achieves three times the hiding capacity of Zhang et al. [45] under the highest payload scenario while maintaining acceptable image quality. When constrained to the same payload as Zhang et al. [45], our scheme delivers an average image quality 6 dB higher.

Despite offering competitive hiding capacity, our method exhibits relatively lower computational complexity for both interpolation image generation and data embedding operations compared to existing schemes. Consequently, our HEI-based RDH is well-suited for real-time applications. Moving forward, we aim to enhance the usability of received stego images by exploring methods for embedding authentication codes during data embedding.

Table 4. comparison for data embedding capacity (bits) and PSNR (dB) with seven methods

Images	Metrics	Lee et al. [28]	Xiong et al. [32]	Hassan et al. [33]	Chen et al. [34]	Bai et al. [35]	Zhang et al. [45]	Proposed method with the same payload as [45])	Proposed method
Lena	Capacity	345,446	319,712	385,680	305,644	585,225	622,812	622,812	2,166,021
	PSNR	29.53	33.00	29.68	34.87	29.94	34.36	37.11	30.68
	SSIM	0.87	0.91	0.87	0.94	0.84	0.92	0.94	0.80
Baboon	Capacity	616,981	514,112	679,982	561,306	585,225	721,443	721,443	2,217,582
	PSNR	21.18	22.53	21.27	24.09	22.03	24.84	36.13	32.06
	SSIM	0.65	0.71	0.64	0.81	0.64	0.82	0.98	0.92
Goldhill	Capacity	408,008	358,862	469,598	369,792	585,225	629,495	629,495	2,153,115
	PSNR	28.26	30.46	28.34	32.24	28.49	32.34	38.45	30.76
	SSIM	0.80	0.85	0.79	0.90	0.83	0.90	0.96	0.82
Boat	Capacity	412,668	368,284	474,606	377,764	585,225	643,888	643,888	2,129,638
	PSNR	26.49	28.86	26.40	32.24	26.88	30.93	38.43	31.65
	SSIM	0.79	0.84	0.79	0.88	0.77	0.90	0.96	0.858
Elaine	Capacity	383,043	338,430	443,813	355,971	585,225	612,120	612,120	2,177,164
	PSNR	29.76	31.59	29.89	32.88	30.00	33.69	37.97	32.21
	SSIM	0.75	0.77	0.76	0.83	0.73	0.87	0.96	0.85
Man	Capacity	438,344	387,266	494,530	394,606	585,225	655,050	655,050	1,699,064
	PSNR	26.16	29.17	26.10	31.32	26.70	30.88	34.91	31.28
	SSIM	0.81	0.88	0.81	0.92	0.79	0.91	0.94	0.872
Peppers	Capacity	347,610	313,117	390,295	307,957	585,225	621,170	621,170	2,077,051
	PSNR	28.63	31.14	28.08	33.36	28.56	33.40	38.38	31.83
	SSIM	0.83	0.87	0.84	0.90	0.81	0.91	0.96	0.83
Couple	Capacity	375,348	347,097	432,475	338,031	585,225	635,743	635,743	2,140,723
	PSNR	25.46	26.42	25.05	28.05	25.34	28.57	38.80	31.83
	SSIM	0.82	0.87	0.82	0.91	0.79	0.91	0.96	0.86

Acknowledgments

This work was supported in part by the National Science and Technology Council under Grant NSC 111-2410-H-167 -005 -MY2 and 112-2634-F-005 -001 -MBK.

References

[1] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C, 20th Anniversary Edition*, Wiley, 2015.

[2] S. Sabir, V. Guleria, D. C. Mishra, Security of multiple RGB images in the time domain and frequency domain, *Journal of Information Security and Applications*, Vol. 63, Article No. 103005, December, 2021. https://doi.org/10.1016/j.jisa.2021.103005

[3] S. Kumar, Image data security using Quasigroup combined with Fibonacci Q-transformation, *Journal of Information Security and Applications*, Vol. 61, Article No. 102941, September, 2021. https://doi.org/10.1016/j.jisa.2021.102941

[4] O. Kocak, U. Erkan, A. Toktas, S. Gao, PSO-based image encryption scheme using modular integrated logistic exponential map, *Expert Systems with Applications*, Vol 237, Article No. 121452, March, 2024. https://doi.org/10.1016/j.eswa.2023.121452

[5] M. SaberiKamarposhti, M. Sahlabadi, C. C. Lin, R. C. Muniyand, Using 2d hénon map, cycling chaos and dna sequence for new secure color image encryption algorithm, *Arabian Journal for Science and Engineering*, Vol. 49, No. 3, pp. 4125-4137, March, 2024. https://doi.org/10.1007/s13369-023-08298-3

[6] G. Xuan, X. Li, Y.-Q. Shi, Histogram-pair based reversible data hiding via searching for optimal four thresholds, *Journal of Information Security and Applications*, Vol. 39, pp. 58-67, April, 2018. https://doi.org/10.1016/j.jisa.2018.01.006

[7] X. Chen, C. Hong, An efficient dual-image reversible data hiding scheme based on exploiting modification direction, *Journal of Information Security and Applications*, Vol. 58, Article No. 102702, May, 2021. https://doi.org/10.1016/j.jisa.2020.102702

[8] A. Malik, H. Wang, T. Chen, T. Yang, A. N. Khan, H. Wu, Y. Chen, Y. Hu, Reversible data hiding in homomorphically encrypted image using interpolation technique, *Journal of Information Security and Applications*, Vol. 48, Article No. 102374, October, 2019. https://doi.org/10.1016/j.jisa.2019.102374

[9] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, *IBM Systems Journal*, Vol. 35, No. 3.4, pp. 313-336, 1996.
https://doi.org/10.1147/sj.353.0313

[10] Y. Q. Shi, X. Li, X. Zhang, H. T. Wu, B. Ma, Reversible data hiding: Advances in the past two decades, *IEEE Access*, Vol. 4, pp. 3210-3237, May, 2016.
https://doi.org/10.1109/ACCESS.2016.2573308

[11] C.-C. Chang, Y.-C. Hu, A fast LBG codebook training algorithm for vector quantization, *IEEE Transactions on Consumer Electronics*, Vol. 44, No. 4, pp. 1201-1208, November, 1998.
https://doi.org/10.1109/30.735818

[12] P. Tsai, Y.-C. Hu, C.-C. Chang, A color image watermarking scheme based on color quantization, *Signal Processing*, Vol. 84, No. 1, pp. 95-106, January, 2004.
https://doi.org/10.1016/j.sigpro.2003.07.012

[13] C.-C. Chang, T. D. Kieu, A reversible data hiding scheme using complementary embedding strategy, *Information Sciences*, Vol. 180, No. 16, 3045-3058, August, 2010.
https://doi.org/10.1016/j.ins.2010.03.027

[14] J. Tian, Reversible data embedding using a difference expansion, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 8, pp. 890-896, August, 2003.
https://doi.org/10.1109/TCSVT.2003.815962

[15] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, Y. Q. Shi, Reversible watermarking algorithm using sorting and prediction, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 7, pp. 989-999, July, 2009.
https://doi.org/10.1109/TCSVT.2009.2020257

[16] W. Wang, A reversible data hiding algorithm based on bidirectional difference expansion, *Multimedia Tools and Applications*, Vol. 79, No. 9-10, pp. 5965-5988, March, 2020.
https://doi.org/10.1007/s11042-019-08255-z

[17] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, Reversible data hiding, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 3, pp. 354-362, March, 2006.
https://doi.org/10.1109/TCSVT.2006.869964

[18] P. Tsai, Y.-C. Hu, H.-L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, *Signal Processing*, Vol. 89, No. 6, pp. 1129-1143, June, 2009.
https://doi.org/10.1016/j.sigpro.2008.12.017

[19] S. Kim, X. Qu, V. Sachnev, H. J. Kim, Skewed histogram shifting for reversible data hiding using a pair of extreme predictions, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 29, No. 11, pp. 3236-3246, November, 2019.
https://doi.org/10.1109/TCSVT.2018.2878932

[20] H.-T. Wu, J.-L. Dugelay, Y.-Q. Shi, Reversible image data hiding with contrast enhancement, *IEEE Signal Processing Letters*, Vol. 22, No. 1, pp. 81-85, January, 2015.
https://doi.org/10.1109/LSP.2014.2346989

[21] H.-T. Wu, S. Tang, J. Huang, Y.-Q. Shi, A novel reversible data hiding method with image contrast enhancement, *Signal Processing: Image Communication*, Vol. 62, pp. 64-73, March, 2018.
https://doi.org/10.1016/j.image.2017.12.006

[22] X. Li, J. Li, B. Li, B. Yang, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, *Signal Processing*, Vol. 93, No. 1, pp. 198-205, January, 2013.
https://doi.org/10.1016/j.sigpro.2012.07.025

[23] X. Qu, H. J. Kim, Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding, *Signal Processing*, Vol. 111, pp. 249-260, June, 2015.
https://doi.org/10.1016/j.sigpro.2015.01.002

[24] S. Meikap, B. Jana, Directional PVO for reversible data hiding scheme with image interpolation, *Multimedia Tools and Applications*, Vol. 77, No. 23, pp. 31281-31311, December 2018.
https://doi.org/10.1007/s11042-018-6203-2

[25] Z. Pan, X. Gao, E. Gao, G. Fan, Adaptive complexity for pixel-value-ordering based reversible data hiding, *IEEE Signal Processing Letters*, Vol. 27, pp. 915-919, 2020.
https://doi.org/10.1109/LSP.2020.2996507

[26] Y. Niu, S. Shen, A novel pixel value ordering reversible data hiding based on dual-image, *Multimedia Tools and Applications*, Vol. 81, No. 10, pp. 13751-13771, April, 2022.
https://doi.org/10.1007/s11042-022-12149-y

[27] K.-H. Jung, K.-Y. Yoo, Data hiding method using image interpolation, *Computer Standards & Interfaces*, Vol. 31, No. 2, pp. 465-470, February 2009.
https://doi.org/10.1016/j.csi.2008.06.001

[28] C.-F. Lee, Y.-L. Huang, An efficient image interpolation increasing payload in reversible data hiding, *Expert Systems with Applications*, Vol. 39, No. 8, pp. 6712-6719, June, 2012.
https://doi.org/10.1016/j.eswa.2011.12.019

[29] F. S. Hassan, A. Gutub, Efficient reversible data hiding multimedia technique based on smart image interpolation, *Multimedia Tools and Applications*, Vol. 79, No. 39-40, pp. 30087-30109, October, 2020.
https://doi.org/10.1007/s11042-020-09513-1

[30] M. Tang, J. Hu, W. Song, S. Zeng, Reversible and adaptive image steganographic method, *AEU-International Journal of Electronics and Communications*, Vol. 69, No. 12, pp. 1745-1754, December, 2015.
https://doi.org/10.1016/j.aeue.2015.08.011

[31] X. Xiong, L. Wang, Z. Li, C. Ye, Y. Chen, M. Fan, Y. Zhu, An adaptive high capacity reversible data hiding algorithm in interpolation domain, *Signal Processing*, Vol. 194, Article No. 108458, May, 2022.
https://doi.org/10.1016/j.sigpro.2022.108458

[32] X. Xiong, Y. Chen, M. Fan, S. Zhong, Adaptive reversible data hiding algorithm for interpolated images using sorting and coding, *Journal of Information Security and Applications*, Vol. 66, Article No. 103137, May, 2022.
https://doi.org/10.1016/j.jisa.2022.103137

[33] F. S. Hassan, A. Gutub, Efficient image reversible data hiding technique based on interpolation optimization, *Arabian Journal for Science and Engineering*, Vol. 46, No. 9, pp. 8441-8456, September, 2021.
https://doi.org/10.1007/s13369-021-05529-3

[34] Y.-Q. Chen, W.-J. Sun, L.-Y. Li, C.-C. Chang, X. Wang, An efficient general data hiding scheme based on image interpolation, *Journal of Information Security and Applications*, Vol. 54, Article No. 102584, October, 2020.
https://doi.org/10.1016/j.jisa.2020.102584

[35] X. Bai, Y. Chen, G. Duan, C. Feng, W. Zhang, A data hiding scheme based on the difference of image interpolation algorithms, *Journal of Information Security and Applications*, Vol. 65, Article No. 103068, March, 2022.
https://doi.org/10.1016/j.jisa.2021.103068

[36] Y.-T. Chang, C.-T. Huang, C.-F. Lee, S.-J. Wang, Image interpolating based data hiding in conjunction with pixel-

shifting of histogram, *The Journal of Supercomputing*, Vol. 66, No. 2, pp. 1093-1110, November, 2013. https://doi.org/10.1007/s11227-013-1016-6

[37] X. Zhang, Z. Sun, Z. Tang, C. Yu, X. Wang, High capacity data hiding based on interpolated image, *Multimedia Tools and Applications*, Vol. 76, No. 7, pp. 9195-9218, April, 2017. https://doi.org/10.1007/s11042-016-3521-0

[38] X.-T. Wang, C.-C. Chang, T.-S. Nguyen, M.-C. Li, Reversible data hiding for high quality images exploiting interpolation and direction order mechanism, *Digital Signal Processing*, Vol. 23, No. 2, pp. 569-577, March, 2013. https://doi.org/10.1016/j.dsp.2012.06.015

[39] A. Verma, K. K. Saini, Refined neighbor mean interpolation algorithm for reversible data-hiding techniques, *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2019, pp. 774-778.

[40] K.-H. Jung, K.-Y. Yoo, Steganographic method based on interpolation and LSB substitution of digital images, *Multimedia Tools and Applications*, Vol. 74, No. 6, pp. 2143-2155, March, 2015. https://doi.org/10.1007/s11042-013-1832-y

[41] J. Hu, T. Li, Reversible steganography using extended image interpolation technique, *Computers & Electrical Engineering*, Vol. 46, pp. 447-455, August, 2015. https://doi.org/10.1016/j.compeleceng.2015.04.014

[42] W. Hong, T.-S. Chen, Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism, *Journal of Visual Communication and Image Representation*, Vol. 22, No. 2, pp. 131-140, February, 2011. https://doi.org/10.1016/j.jvcir.2010.11.004

[43] A. Malik, G. Sikka, H. K. Verma, An image interpolation based reversible data hiding scheme using pixel value adjusting feature, *Multimedia Tools and Applications*, Vol. 76, No. 11, pp. 13025-13046, June, 2017. https://doi.org/10.1007/s11042-016-3707-5

[44] A. Malik, G. Sikka, H. K. Verma, Image interpolation based high capacity reversible data hiding scheme, *Multimedia Tools and Applications*, Vol. 76, No. 22, pp. 24107-24123, November, 2017. https://doi.org/10.1007/s11042-016-4186-4

[45] H. Zhang, H. Sun, F. Meng, Reversible data hiding scheme based on improved interpolation and three-in-one intelligent strategy, *Journal of Information Security and Applications*, Vol. 77, Article No. 103573, September, 2023. https://doi.org/10.1016/j.jisa.2023.103573

[46] P. Bas, T. Filler, T. Pevný, Break our steganographic system: the ins and outs of organizing BOSS, in: T. Filler, T. Pevný, S. Craver, A. Ker (Eds.), *International Workshop on Information Hiding*, Berlin, Heidelberg, 2011, pp. 59-70.

Biographies

Chia-Chen Lin received the M.S. degree and the Ph.D degree in information management from Chiao Tung University, Hsinchu, Taiwan, in 1994 and 1998, respectively. She is currently a Professor in the Department of Computer and Information Management, Providence University, Sha-Lu, Taiwan. Her research interests include image and signal processing, image data hiding.

En-Ru Chen received the B.S. degree in computer science and engineering from Yuan Ze University, Taoyuan, Taiwan, in 2019. He is currently pursuing the M.S. degree at Feng Chia University. His research interests include image processing and image data hiding.

Chin-Chen Chang received the Ph.D degree in computer engineering from National Chiao Tung University, Hsinchu, in 1982. From July 1998 to June 2000, he was Director of the Advisory Office, Ministry of Education, R.O.C. From 2002 to 2005, he was a Chair Professor at National Chung Cheng University. From February 2005, he has been a Chair Professor at Feng Chia University. In addition, he was severed as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression, and data structures.