J Supercomput (2013) 66:1093–1110
DOI 10.1007/s11227-013-1016-6

Image interpolating based data hiding in conjunction with pixel-shifting of histogram

Ya-Ting Chang · Cheng-Ta Huang · Chin-Feng Lee · Shiuh-Jeng Wang

Published online: 28 September 2013

Abstract Information hiding is an important research issue in digital life. In this paper, we propose a two-stage data hiding method with high capacity and good visual quality based on image interpolation and histogram modification techniques. At the first stage, we first generate a high-quality cover image using the developed enhanced neighbor mean interpolation and then take the difference values from input and cover pixels as a carrier to embed secret data. In this stage, our proposed scheme raises the image quality a lot due to the ENMI method. At the second stage, a histogram modification method is applied on the difference image to further increase the embedding capacity and preserve the image quality without distortion. Experimental results indicate that the proposed method have better PSNR value of stego-image with improving 43 % on the average when compared the past key-studies.

Keywords Data hiding · Steganography · Image interpolation · Histogram modification

1 Introduction

As the progressive development of personal computers, smart phones and network technologies, data generation, transmission, and storage become easier. Browsing

Y.-T. Chang · S.-J. Wang (✉)
Department of Information Management, Central Police University, Taoyuan, 333, Taiwan
e-mail: dopwang@gmail.com

C.-T. Huang
Graduate Institute of Biomedical Engineering, National Taiwan University of Science and Technology, Taipei, 106, Taiwan

C.-F. Lee
Department of Information Management, Chaoyang University of Technology, Taichung, 413, Taiwan

Web pages, sending and receiving E-mails, and instant messaging are common applications of data transmission. It is an important issue to build up a thorough image visual mechanism to protect the security of data when they are transmitted via the Internet. Data hiding reaches the invisibility of hidden data because it embeds the confidential message into a visual cover media such that the secret data cannot be conscious of the existence of message itself by human sense. The cover media of reversible data hiding techniques can be inverted back to the original media without any distortion after the hidden data have been extracted out. It is a critical requirement in some applications, such as medical diagnosis and law enforcement for some legal considerations.

Depending on different image features, data hiding can be classified into three domains: spatial domain [1, 3, 8], frequency domain [2, 5], and compression domain [6, 11]. The main idea of image processing in spatial domain is to handle the pixels directly by modifying pixel values. The most common technique is the least significant bit (LSB) substitution [1, 10]. Secret messages are embedded by modifying the last few significant bits of a pixel value. Human's eyes are not sensitive in the kind of pixel fine tuning. Histogram modification method shifts the pixels between the peak and zero or points of the histogram of an image to slightly modify the pixel grayscale values to embed data into the image. It is able to embed about 5–80 K bits into a 512×512 grayscale image while the PSNR of the marked image versus the original image is kept to be 48 dB. A novel techniques in [8, 12] constructed a histogram of difference values between original pixels and predicted pixels to enhance embedding capacity and reserve the image quality. Secret messages are hidden in the histogram. Multilevel histogram modification [12] is able to enlarge the payload of secret message in a more flexible and adaptive way. Image interpolation [3, 9] is usually used to generate a high-resolution image from its low-resolution, that is, to scale up an image. The interpolated pixels result to some difference values comparing with its original image and data hiding can be applied by taking this advantage. There are some simple interpolation methods, such as nearest neighbor, bilinear, and nearest neighbor mean interpolation will be introduced later in Sect. 2.

The technique of data hiding in frequency domain transforms pixel values into a set of coefficients first and then uses the coefficients to carry secret data. Some well-known transformation functions such as discrete cosine transform (DCT), discrete wavelet transform (DWT), discrete Fourier transform (DFT) and so on are usually employed. A Haar digital wavelet transform (HDWT)-based reversible data hiding method [2] was developed to embed data in the high frequency band since the high frequency band incorporates less energy than other bands of an image. Discarding high frequency DCT coefficients in certain image regions, a watermarking technique [5] modifies the coefficients in the middle frequency to carry a secret message.

In the compression domain, the cover image is an index format that is generally encoded by vector quantization (VQ) [6, 11]. The principle of VQ-based data hiding first trains a representative codebook and then divides the codebook into sub-codebooks in which n codewords with stronger similarity formed a n-member sub-codebooks. The number of codewords in sub-codebook determines the amount of secret embedding at a time when the secret are hidden. One of the codewords in the sub-codebook is selected to substitute the original codeword and to embed the secret message.

In this paper, we propose a two-stage data hiding method on digital images. At the first stage, we first generate a cover image using the developed enhanced neighbor mean interpolation (ENMI for short) and then take the difference values from input and cover pixels as carrier to embed secret data. In this stage, our proposed scheme raises the image quality a lot due to the ENMI method. At the second stage, a histogram modification method applied on the difference image to further enlarges the embedding capacity and preserves the image quality without distortion. Our technique achieves the requirement of high embedding capacity and image quality with low-computation complexity.

The rest of this paper is organized as follows. In Sect. 2, two reversible data hiding methods upon image interpolations and histogram modification are introduced. The proposed scheme is presented in Sect. 3. Experiments are performed and the experimental results are shown in Sect. 4. The performance in terms of embedding capacity and image quality are also discussed. The conclusion is stated in Sect. 5.

2 Related works

Data hiding schemes upon image interpolations and histogram modification are reviewed in this section.

2.1 Data hiding upon interpolations

Interpolation is the process by which a small image is resized or remapped to a larger one. Interpolation algorithms stretch the size of an image and generate pixels to fill in the blanks. Image interpolation is widely used in medical imaging, digital photos, film-scanned images, and so forth. For example, image reconstruction in computed tomography (CT) or magnetic resonance imaging (MRI) often employs image interpolation. In 2009, Jung and Yoo [4] first proposed the use of image interpolation in the spatial domain of data hiding. Data hiding in combination with image interpolation makes the use of imagery more diverse.

In image interpolation, limited information is used to calculate and predict other pixel values. The limited information, or the known pixels, are called reference pixels and will leave unchanged while interpolating. In the following, some common interpolation algorithms including nearest neighbor interpolation and bilinear interpolation are introduced.

2.1.1 Nearest neighbor interpolation

Nearest neighbor interpolation (NNI for short) is a simple method, which selects the value of the nearest neighboring point to yield an interpolant. An example is shown in Fig. 1.

2.1.2 Bilinear interpolation

To the estimated the point value, bilinear interpolation (also called BI) uses the 4 nearest neighboring reference (known) pixels, which are located in diagonal directions from the given pixel. Figure 2 shows an example of the estimation by bilinear

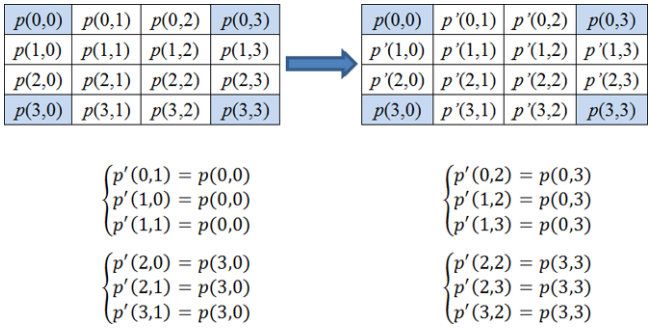
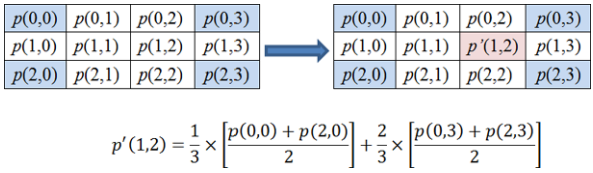


Fig. 1 An example of nearest neighbor interpolation, which estimates new data points by locating the nearest known data points

Fig. 2 An example of bilinear interpolation



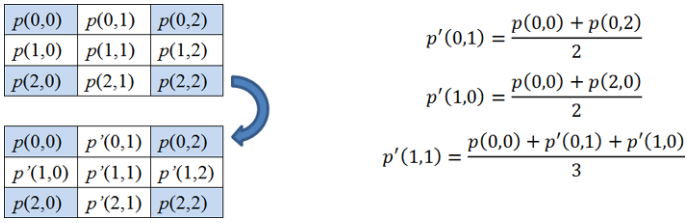


Fig. 3 An example of neighbor mean interpolation method

interpolation, where the pixel value of $p'(1, 2)$ is calculated by weighting its reference neighbors which are pixels $p(0, 0)$, $p(2, 0)$, $p(0, 3)$, and $p(2, 3)$.

2.1.3 Neighbor mean interpolation

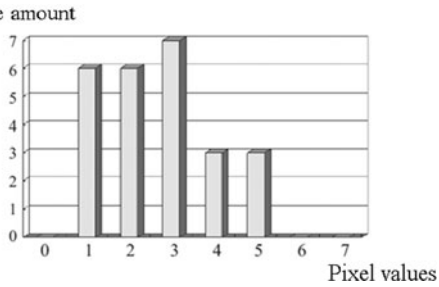
In 2009, Jung and Yoo [4] proposed a new interpolation which is "Neighbor mean interpolation (NMI for short)" with a low-time complexity and high-calculation speed. An example of NMI is shown in Fig. 3 as follows.

The data hiding scheme by Jung and Yoo scales down an input image to 1/4 of its initial size, which then becomes the original image. The scheme then uses NMI to enlarge this original image into a cover image, which has the same size as the input image. Secret data are hidden in the scaling-up cover image. The authorized receiver can extract embedded secret message from the stego-image and restore the cover image to the original image.

Fig. 4 A cover image of simple example

1	3	2	4	5
2	1	3	5	2
4	3	3	1	5
4	2	1	3	3
2	1	3	2	1

Fig. 5 The histogram corresponding to the cover image in Fig. 4



2.2 Data hiding on histogram modification

In the traditional histogram modification data hiding, the amount of every pixel values of the cover image is counted and a histogram is constructed. This method was first applied by Ni et al. in 2006 [7]. For a grayscale cover image C with size of $M \times N$, let a grayscale value with the maximum number of pixels in C as the peak point denoted as P; let a grayscale value with no pixels or the minimum number of pixels in C as the zero point denoted as Z. Assume $P < Z$, histogram modification method shifts the values within the range $[P+1, Z-1]$ of the histogram to the right-hand side by one unit, leaving the grayscale value $(P+1)$ empty. A bit b which is extracted from the secret data will be embedded, and the output stego-image is S. The rule to hide the secret bit b is as shown in Eq. (2).

$$C'(i,j) = \begin{cases} C(i,j) & \text{if } C(i,j) \le P \\ C(i,j)+1 & \text{if } Z > C(i,j) > P \end{cases}, \text{ for } 0 \le i \le M-1,\ 0 \le j \le N-1, \tag{1}$$

$$S(i,j) = \begin{cases} C'(i.j) & \text{if } C'(i,j) = P \text{ and } b = 0 \\ C'(i.j)+1 & \text{if } C'(i,j) = P \text{ and } b = 1 \\ C'(i,j) & \text{otherwise} \end{cases}. \tag{2}$$

A simple example considers a matrix shown in Fig. 4 as a cover image. Assume a secret message be a series of bits "1010111". The histogram of the cover image is shown in Fig. 5. The peak value is $P = 3$, so all pixels larger than three are increased by one. The image and its histogram after pixel-shifting are shown in Figs. 6 and 7. According to the embedding rule, when a pixel value equivalent to three, the pixel is modified to be four as we embed the bit $b = 1$; otherwise, we keep the pixel value unchanged. After the embedding process has been done, the stego-image is shown in Fig. 8.

Fig. 6 The image after pixel-shifting applied

1	3	2	5	6
2	1	3	6	2
5	3	3	1	6
5	2	1	3	3
2	1	3	2	1

Fig. 7 The histogram corresponding to the image in Fig. 6

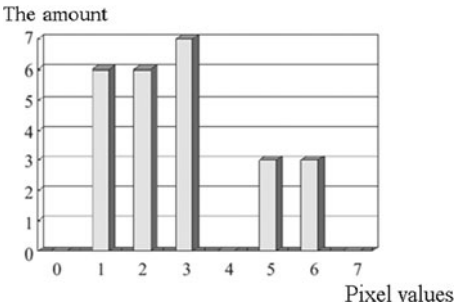


Fig. 8 The stego-image after secret data have been embedded

1	**4**	2	5	6
2	1	**3**	6	2
5	**4**	**3**	1	6
5	2	1	**4**	**4**
2	1	**4**	2	1

3 The proposed scheme

In this section, we propose a high visual quality data hiding scheme. In Sect. 3.1, we design an enhanced neighbor mean interpolation (also called ENMI) to generate a cover image. Second, the proposed embedding procedure in two stages is illustrated in Sect. 3.2. The procedure of secret data extraction and image restoration is presented in Sect. 3.3. To give readers a better understanding of our proposed scheme, we give a simple example to illustrate the method in Sect. 3.4.

3.1 Enhanced neighbor mean interpolation

The proposed enhanced neighbor mean interpolation (ENMI for short) is inspired from the Jung–Yoo-scheme [4]. Figure 9 illustrates the procedure of our proposed scheme. For an input image I sized of 512×512 pixels, we first define an image block, which is composed of four adjacent pixels, i.e., $I(i, j)$, $I(i+1, j)$, $I(i, j+1)$, and $I(i+1, j+1)$. For each pixel $I(i, j)$, the corresponding cover pixel, $C(i, j)$ of a cover image C is defined by the algorithm of ENMI. Figure 10 shows an example of how to process the enhanced neighbor mean interpolation.

Algorithm of ENMI
Input: Input image I with sized $N \times N$
Output: Cover image C
for $i = 0 \sim N - 1$ **do**

for $j = 0 \sim N-1$ **do**

$$C(i,j) = \begin{cases} I(i,j) & \text{if } i \bmod 2 = 0 \text{ and } j \bmod 2 = 0 \\ I(i,j-1) & \text{if } j = N-1 \\ I(i-1,j) & \text{if } i = N-1 \\ \frac{I(i,j-1)+I(i,j+1)}{2} & \text{if } i \bmod 2 = 0 \text{ and } j \bmod 2 = 1 \\ \frac{I(i-1,j)+I(i+1,j)}{2} & \text{if } i \bmod 2 = 1 \text{ and } j \bmod 2 = 0 \\ \frac{I(i-1,j-1)+I(i-1,j+1)+I(i+1,j-1)+I(i+1,j+1)}{4} & \text{otherwise} \end{cases}. \quad (3)$$

end for
end for

3.2 The embedding phase

The proposed data hiding algorithm, including two stages with interpolation data hiding and histogram modification, are shown next in details.

Step 1. For an input image I, the ENMI algorithm is employed to generate the corresponding cover image C. Afterward, a difference image D_0 can be computed by Eq. (4).

$$D_0(i,j) = C(i,j) - I(i,j). \quad (4)$$

Step 2. The number of bits, says n is calculated by Eq. (5).

$$n = trunc\left(\log_2 \left| D_0(i,j) \right|\right) - 1, \quad (5)$$

where "trunc" denotes the function which returns a number truncated to an integer.

Step 3. The first segment of binary secret data W_1 needs to be partitioned into sub-streams before they are hidden. Each sub-stream, say w with n bits denoted by Eq. (6) can be extracted out from secret segment W_1. A bit "1" called the leading bit is padded to the left-hand side of sub-stream w. An integer value d is converted from the sub-stream w and its leading bit.

$$w = w_n w_{n-1} \cdots w_2 w_1 \quad \text{if } n > 0, \quad (6)$$

$$d = \begin{cases} 1 \times 2^n + w_n \times 2^{n-1} + w_{n-1} \times 2^{n-2} + \cdots + w_1 \times 2^0 & \text{if } n > 0 \\ 1 & \text{if } n = 0 \end{cases}. \quad (7)$$

Step 4. At this point in the first stage of embedding procedure, we can embed d to a cover pixel $C(i,j)$. A stego-image S_1 is generated by adding d to or subtracting d from the cover pixel $C(i,j)$ in order to make the stego-pixel value closer to the input pixel $I(i,j)$. The embedding rule is shown as follows:

$$S_1(i,j) = \begin{cases} C(i,j) & \text{if } i \bmod 2 = 0 \text{ and } j \bmod 2 = 0 \\ C(i,j) + d & \text{if } I(i,j) \geq C(i,j) \\ C(i,j) - d & \text{if } I(i,j) < C(i,j) \end{cases}. \quad (8)$$

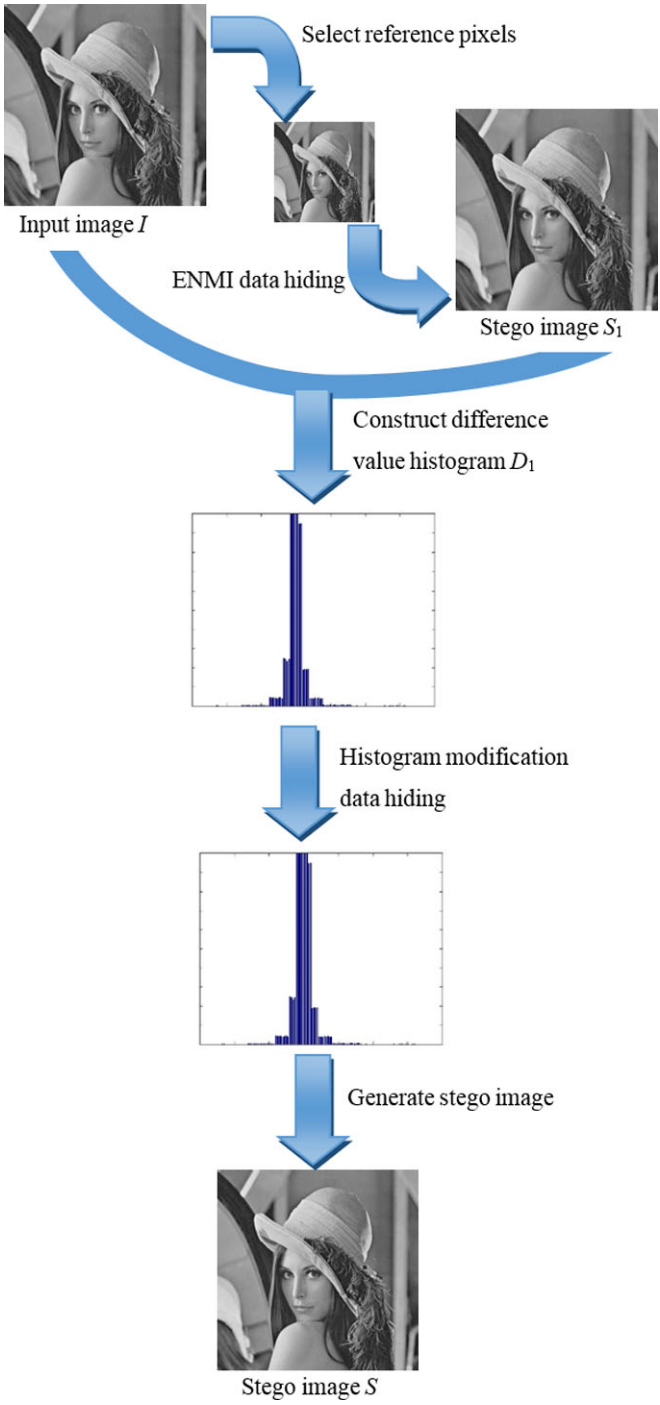


Fig. 9 The flowchart of proposed data hiding method

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

Fig. 10 An example of ENMI method

Step 5. For increasing the embedding capacity, the second stage of embedding procedure produces another difference image D_1 in by Eq. (9).

$$D_1(i,j) = S_1(i,j) - C(i,j). \tag{9}$$

Step 6. A histogram is constructed based on the difference values D_1 and two peak points are chosen where $P_2 < 0 < P_1$. It is worth mentioning that we do not choose 0 as a peak value because we want to present the changes of reference pixels when the pixel-shifting in the histogram modification of D_1 are carried out. That is, the secret data cannot be embedded on reference pixels after histogram modification method is applied on D_1, or we cannot recover D_1 on the recovery phase. The rule for shifting values in the histogram of D_1 is listed as the following Eq. (10):

$$D_1'(i,j) = \begin{cases} D_1(i,j)+1 & \text{if } D_1(i,j) > P_1 \\ D_1(i,j)-1 & \text{if } D_1(i,j) < P_2 \\ D_1(i,j) & \text{otherwise} \end{cases}. \tag{10}$$

Step 7. Sequentially scan the difference image D_1' from left-to-right and up-to-down. If the value $D_1'(i,j)$ equals to one of the peak values, then we can extract one bit w' from the second segment of binary secret data W_2 to be hidden in $D_1'(i,j)$. The second stage of embedding rule is listed as the following Eq. (11):

$$D_1''(i,j) = \begin{cases} D_1'(i,j)+w' & \text{if } D_1'(i,j) = P_1 \\ D_1'(i,j)-w' & \text{if } D_1'(i,j) = P_2 \\ D_1'(i,j) & \text{otherwise} \end{cases}. \tag{11}$$

Step 8. Equation (12) shows that the difference value $D_1''(i, j)$ can be added to the cover pixel $C(i, j)$ to generate the stego-pixel $S(i, j)$.

$$S(i, j) = C(i, j) + D_1''(i, j). \tag{12}$$

The following pseudo-code demonstrates our proposed embedding phase:

Input: Input image I with sized $N \times N$, secret data $W = W_1 \| W_2$ where "$\|$" means string concatenation operator.
Output: Stego-image S, two peak points P_1, P_2

```
Apply ENMI on I to generate C;
for i = 0 ~ N − 1 do
  for j = 0 ~ N − 1 do
    D0(i, j) = C(i, j) − −I(i, j);
    The number of bits n is calculated from D0(i, j);
    Extract n-bit stream w from W1;
    Convert 1‖w to decimal presentation d;
    If I(i, j) > C(i, j)
      S1(i, j) = C(i, j) + d;
    Else
      S1(i, j) = C(i, j) − d;
  end for
end for

for i = 0 ~ N − 1 do
  for j = 0 ~ N − 1 do
    D1(i, j) = S1(i, j) − C(i, j);
  end for
end for

Generate histogram of D1;
Select two peak points from D1 where P2 < 0 < P1;
Apply histogram shifting on D1 resulting D1';
Embed secret bits from W2 on P2 and P1 of D1' resulting D1'';

for i = 0 ~ N − 1 do
  for j = 0 ~ N − 1 do
    S(i, j) = C(i, j) + D1''(i, j);
  end for
end for
```

3.3 The phase of secret extraction and image recovery

The proposed algorithms of secret extraction and image recovery are described in the following.

Step 1. The cover image C can be retrieved from the stego-image S. The procedure first scans the stego-image S and introduces the ENMI algorithm to obtain C by the following rule as shown in Eq. (13):

$$C(i,j) = \begin{cases} S(i,j) & \text{if } i \bmod 2 = 0 \text{ and } j \bmod 2 = 0 \\ \frac{S(i,j-1)+S(i,j+1)}{2} & \text{if } i \bmod 2 = 0 \text{ and } j \bmod 2 = 1 \\ \frac{S(i-1,j)+S(i+1,j)}{2} & \text{if } i \bmod 2 = 1 \text{ and } j \bmod 2 = 0 \\ \frac{S(i-1,j-1)+S(i-1,j+1)+S(i+1,j-1)+S(i+1,j+1)}{4} & \text{otherwise} \end{cases}. \tag{13}$$

Step 2. We can construct a histogram by subtracting C from S as Eq. (14) shown

$$D_1''(i,j) = S(i,j) - C(i,j). \tag{14}$$

Afterward, the extraction procedure sequentially scans the difference image D_1'' from left-to-right and up-to-down for extracting a segment of secret message by referencing to the values around the peak points.

Step 3. There are two stages need to be performed for extracting the whole hidden data and restoring cover pixels. The first stage initializes W_2 as an empty set in advanced. If the difference value $D_1''(i,j)$ equals to P_1, P_1+1, P_2, or P_2-1, then a secret bit has been hidden among them. We can extract the secret bit w' according to Eq. (15).

$$w' = \begin{cases} 0 & \text{if } D_1''(i,j) = P_1 \text{ or } D_1''(i,j) = P_2 \\ 1 & \text{if } D_1''(i,j) = P_1 + 1 \text{ or } D_1''(i,j) = P_2 - 1 \end{cases}. \tag{15}$$

Append the secret bits to W_2 in the way of $W_2 = W_2 \| w'$. Afterward, the histogram modification method is applied on the difference image D_1'' by Eq. (16).

$$D_1(i,j) = \begin{cases} D_1''(i,j) - 1 & \text{if } D_1''(i,j) > P_1 \\ D_1''(i,j) + 1 & \text{if } D_1''(i,j) < P_2 \\ D_1''(i,j) & \text{otherwise} \end{cases}. \tag{16}$$

Step 4. The next is to convert the integer $D_1(i,j)$ to its binary representation, say b. The first bit which must have been the bit "1" that denotes the leading bit of the string b. What we want to find is the secret sub-stream, say w, is to discard the leading bit and take the rest of the string b. For example, if $D_1(i,j) = 9$, convert it to binary, that is "1001". Discard the leading bit and we obtain the secret sub-stream "001". However, when $D_1(i,j) = 1$ is encountered, it means no secret data are embedded.

Step 5. At last, all the secret sub-streams are appended to obtain the secret segment W_1. Append W_2 to W_1 and the whole secret message $W = W_1 \| W_2$ can be completely extracted out.

The pseudo-code demonstrates our proposed extraction and image recovery phase as follows:

Input: Stego-image S with sized $N \times N$; two peak points P_1 and P_2
Output: Secret data $W = W_1 \| W_2$ and cover image C
Apply ENMI on S to generate C;
Let W_1 and W_2 be EMPTY;
for $i = 0 \sim N - 1$ **do**
  for $j = 0 \sim N - 1$ **do**
    $D_1''(i, j) = S(i, j) - C(i, j)$;
    If $D_1''(i, j) = P_1$ **or** P_2
      $W_2 = W_2 \| 0$;
    Else if $D_1''(i, j) = P_1 + 1$ **or** $P_2 - 1$
      $W_2 = W_2 \| 1$;
  end for
end for

for $i = 0 \sim N - 1$ **do**
  for $j = 0 \sim N - 1$ **do**
    If $D_1''(i, j) > P_1$
      $D_1(i, j) = D_1''(i, j) - 1$;
    Else if $D_1''(i, j) < P_2$
      $D_1(i, j) = D_1''(i, j) + 1$;
    Else
      $D_1(i, j) = D_1''(i, j)$;
    Convert $D_1(i, j)$ to binary presentation b;
    Discard the first bit of b and the rest bit stream is w;
    $W_1 = W_1 \| w$;
    end for
end for
$W = W_1 \| W_2$;

3.4 An example of the proposed scheme

We give a simple example to illustrate the proposed scheme. For an input image I sized of 3×3 pixels, the reference pixels are 35, 101, 199, and 79. The cover pixel, $C(i, j)$ of a cover image C can be calculated by the algorithm of ENMI. Figure 11 shows an example of how to process the enhanced neighbor mean interpolation to obtain a cover image C.

A difference image D_0 can be computed by Eq. (4) and shown in Fig. 12 by subtracting the pixel values of the input image from that of the cover image.

Apply Eq. (5) to calculate the number of bits which have to be extracted out from secret segment W_1 and embedded onto a cover pixel. The embedded process is shown as Fig. 13 and Fig. 14.

Another difference image D_1 can be calculated by subtracting each pixel values in the cover image C from which in the stego-image S_1. Figure 15 shows the result of D_1 in our simple example. A histogram can be constructed from D_1 and we choose -6 and 9 as two peak values. Then the histogram is shifted and secret message $W_2 =$ "1001" is embedded continuously.

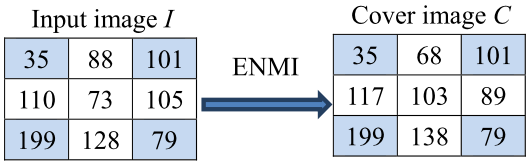


Fig. 11 Example of a cover image from an input image by ENMI algorithm

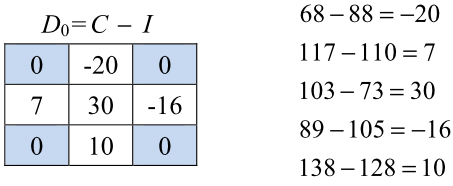


Fig. 12 An example of difference image D_0

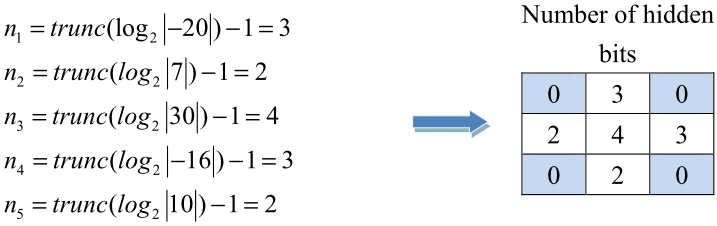


Fig. 13 An example of calculating the number of hidden bits

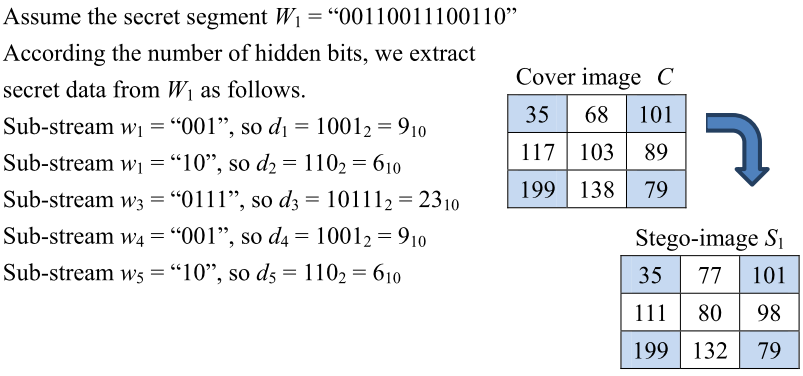


Fig. 14 An example of embedding secret messages by ENMI method

After histogram modification, adding up the modified difference values D_1'' to cover image. A stego-image S is generated as Fig. 16.

When the receiver gets a stego-image S and two peak values, the cover image C can be recovered by ENMI. Thus, the histogram can be constructed by the difference value of S and C. While scanning D_1'' and the difference value -6 or 9 is encountered, a secret bit "0" can be extracted; otherwise, if the difference value is -7 or 10, the secret bit "1" can be extracted, as shown in Fig. 17.

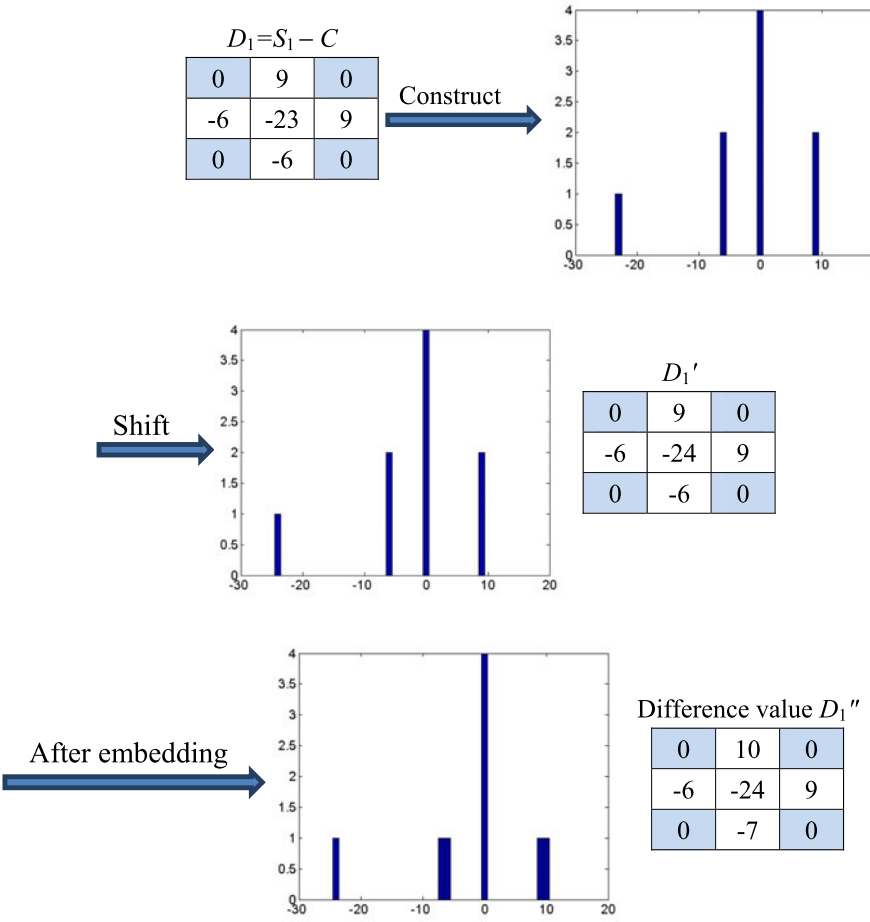


Fig. 15 An example of embedding secret messages by histogram modification

By applying the inverse process of histogram modification, the difference image D_1 can be recovered. Convert each absolute value of $D_1(i, j)$ to its binary string b and discard the leading bit "1"of string b to obtain the hidden secret sub-bitstream w. The extraction process is shown in Fig. 18. A complete secret message $W = W_1 \| W_2$ comes from appending the secret segment W_2 in Fig. 17 to the secret segment W_1 in Fig. 18.

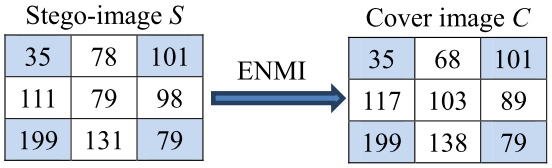
4 Experimental results

In our experiments, six 512 × 512 grayscale images as shown in Fig. 19 are used as input images. A secret data composed of bits "0" and "1" are generated by a random function. Capacity (bits) and PSNR (Peak signal-to-noise ratio) are performance measurements to estimate the embedding capacity and image quality, respectively.

Fig. 16 An example of stego-image

$S=C+D_1''$

35	78	101
111	79	98
199	131	79



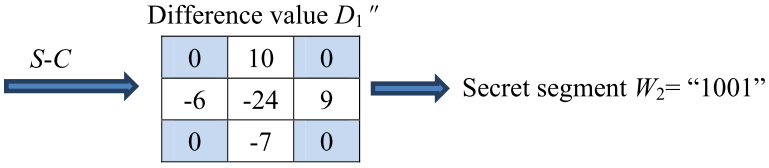
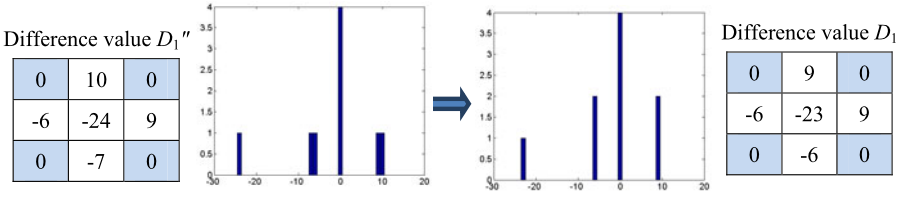


Fig. 17 Obtain D_1'' and secret messages in extraction process



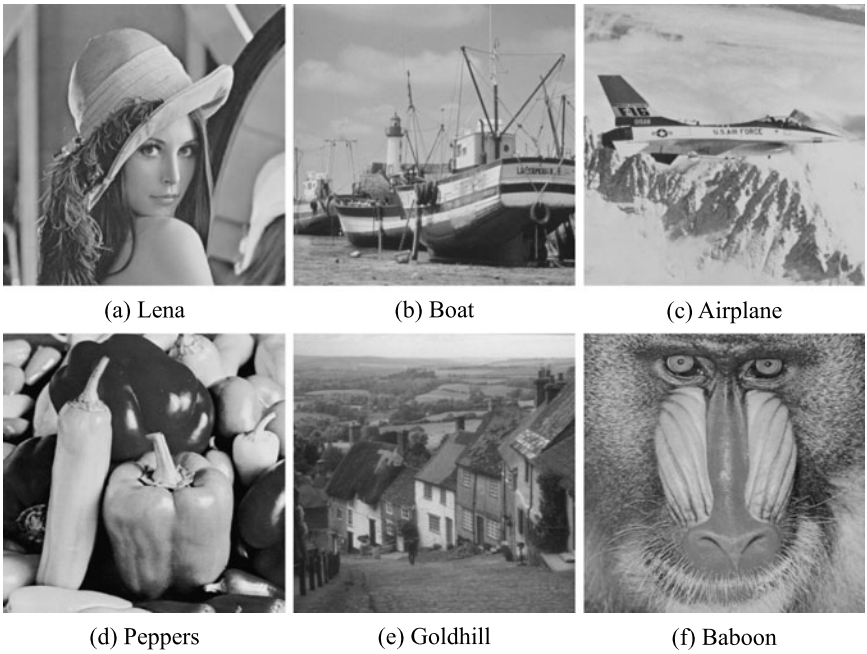
$$b_1 = |9|_{10} = 1001_2 \Longrightarrow w_1 = "001"$$
$$b_2 = |-6|_{10} = 110_2 \Longrightarrow w_2 = "10"$$
$$b_3 = |-23|_{10} = 10111_2 \Longrightarrow w_3 = "0111"$$
$$b_4 = |9|_{10} = 1001_2 \Longrightarrow w_4 = "001"$$
$$b_5 = |-6|_{10} = 110_2 \Longrightarrow w_5 = "10"$$

So, the first secret segment W_1 = “00110011100110”

Fig. 18 An example of extracting secret messages

For two images I and K for both of the size $M \times N$, the PSNR value is computed as follows:

$$\text{MSE (mean squared error)} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[I(i,j) - K(i,j) \right]^2, \qquad (17)$$



(a) Lena (b) Boat (c) Airplane

(d) Peppers (e) Goldhill (f) Baboon

Fig. 19 Six 512 × 512 grayscale input images

Table 1 A comparison of cover image quality by NMI and proposed ENMI

Input image	Nearest neighbor interpolation (NMI) (dB)	Enhanced nearest neighbor mean interpolation (ENMI) (dB)
Lena	31.79	33.47
Boat	29.51	30.83
Airplane	31.45	33.37
Peppers	33.39	35.29
Gold	31.80	33.06
Baboon	23.86	24.47

$$\text{PSNR} = 10 \times \log\left(\frac{255^2}{MSE}\right). \tag{18}$$

Table 1 shows the comparison in terms of PSNR for the cover image quality by the methods of NMI and ENMI. In our proposed ENMI method, we use different prediction equations to calculate the estimation points. In the right-bottom pixel of every block as shown in Fig. 10, the interpolation method refers to four neighbor pixels of the to-be-estimated point so that the prediction can be more correct. Thus, the PSNR of cover images is enhanced in our ENMI method.

Table 2 shows the improvement of embedding capacity and image quality after secret data have been hidden by the proposed histogram modification method. Af-

Table 2 A comparison of EMNI and histogram modification embedding on capacity and PSNR

Input image	After ENMI embedding		After histogram modification embedding			
	Capacity (bits)	PSNR (dB)	Capacity (bits)		PSNR (dB)	
Lena	145,302	42.47	247,095	101,793 increased	43.61	1.14 increased
Boat	166,294	40.21	269,431	103,137 increased	41.12	0.91 increased
Airplane	107,581	42.79	239,206	131,625 increased	43.48	0.69 increased
Peppers	106,930	44.22	227,822	120,892 increased	45.15	0.93 increased
Gold	164,815	42.12	258,855	94,040 increased	43.40	1.28 increased
Baboon	348,315	33.97	396,651	48,336 increased	34.63	0.66 increased

Table 3 A comparison of Jung and Yoo's data hiding method on pure payload and PSNR

Input image	Method					
	Jung and Yoo [4]			Proposed		
	Capacity (bits)	Payload (bpp)	PSNR (dB)	Capacity (bits)	Payload (bpp)	PSNR (dB)
Lena	235,460	0.90	30.61	247,095	0.94	43.61
Boat	226,159	0.86	28.62	269,431	1.03	41.12
Airplane	188,939	0.72	30.54	239,206	0.91	43.48
Peppers	202,238	0.77	32.02	227,822	0.87	45.15
Gold	268,389	1.02	30.88	258,855	0.99	43.40
Baboon	460,740	1.76	23.13	396,651	1.51	34.63
Average	263,654	1.00	29.30	273,177	1.04	41.90

ter embedding secret messages using ENMI, a set denoted as D_1 is calculated from the difference values between the stego-image and cover image. A histogram corresponding to the set D_1 is constructed for embedding more secret data by histogram modification method. Two peak points P_1 and P_2 are selected, where $P_2 < P_1$. For the values greater than P_1 of the histogram are shifted to the right-hand side by 1 unit; similarly, for the values smaller than P_2 of the histogram are shifted to the left-hand side by 1 unit. The pixel-shifting makes the difference values of D_1 be increased or decreased at most one such that the stego-image is closer to its corresponding input image. By the way, the image quality is then significantly improved.

Table 3 shows the pure payload and PSNR compared with that of Jung–Yoo-scheme. Only the secret data is considered on computing the capacity and pure payload. The extra information, such as peak value, is ignored on both two schemes. The image quality of proposed scheme is greatly enhanced. PSNR is therefore increased over 10 dB on the average for each input image. Actually, the proposed scheme is significantly better than the Jung and Yoo's scheme in terms of PSNR regardless of whether the test image is smooth or complex. The reason why there is such a good effect in our scheme is that while embedding secret data, pixel values are changed, but we try to make the pixel values not too far from their corresponding pixels in the cover image. Moreover, the embedding capacity of the proposed scheme is enhanced about 10 % in most test images except for "Baboon" and "Gold" images.

Since "Baboon" and "Gold" images are complicated images with irregular and complex patterns so that the amount of peak value is not that huge. Therefore, histogram modification does not perform well in the complicated images.

5 Conclusions

In this paper, we proposed a data hiding method with high capacity and high image quality. In the proposed scheme, we first sample an input image down to a small sized image and then scale it up to the same size as the input image by using proposed enhanced neighbor mean interpolation (ENMI). After embedding secret data, the histogram modification method is applied to increase up the embedding capacity. We also enhance the embedding methods in two stages mentioned above to maintain high image quality. The proposed scheme allows certain changes to hide secret in an image to go undetected by Human Vision System (HVS). As shown in Sect. 4, the proposed scheme performs well in terms of embedding capacity and image quality. The PSNR of proposed scheme are higher than that of the past works. This meets one of the requirements in data hiding with security issue. Also, the secret messages can be extracted completely so the goal of private communication is reached.

Acknowledgements This research was partially supported by the National Science Council of the Republic of China under the Grant NSC 100-2221-E-015-001-MY2, NSC 101-2221-E-324-006-MY2, and NSC 102-2221-E-015 -001.

References

1. Chan CK, Cheng LM (2004) Hiding data in images by simple LSB substitution. Pattern Recognit 37(3):469–474
2. Chan YK, Chen WT, Yu SS, Ho YA, Tsai CS, Chu YP (2009) A HDWT-based reversible data hiding method. J Syst Softw 82(3):411–421
3. Jan SR, Hsu SJ, Chiu CF, Chang SL (2011) An improved data hiding method using image interpolation. In: 2011 seventh international conference on intelligent information hiding and multimedia signal processing, pp 185–188
4. Jung KH, Yoo KY (2009) Data hiding method using image interpolation. Comput Stand Interfaces 31(2):465–470
5. Langelaar GC, Lagendijk RL (2001) Optimal differential energy watermarking of DCT encoded images and video. IEEE Trans Image Process 10(1):148–158
6. Lu ZM, Pan JS, Sun SH (2000) VQ-based digital image watermarking method. Electron Lett 36(14):1201–1202
7. Ni Z, Shi YQ, Ansari N, Su W (2006) Reversible data hiding. IEEE Trans Circuits Syst Video Technol 16(3):354–362
8. Pawar PH, Jondhale KC (2012) Histogram-based reversible data hiding using block division. In: 2012 IEEE international conference on advanced communication control and computing technologies (ICACCCT), pp 295–299
9. Yalman Y, Akar F, Erturk I (2010) An image interpolation based reversible data hiding method using R-weighted coding. In: 2010 13th IEEE international conference on computational science and engineering, pp 346–350
10. Yang CH (2008) Inverted pattern approach to improve image quality of information hiding by LSB substitution. Pattern Recognit 41(8):2674–2683
11. Yang CH, Wang WJ, Huang CT, Wang SJ (2011) Reversible steganography based on side match and hit pattern for VQ-compressed images. Inf Sci 181(11):2218–2230
12. Zhao Z, Luo H, Lu ZM, Pan JS (2011) Reversible data hiding based on multilevel histogram modification and sequential recovery. AEÜ, Int J Electron Commun 65(10):814–826