Lu et al., Int J Comput Softw Eng 2016, 1: 102
http://dx.doi.org/10.15344/ijcse/2016/102

International Journal of
Computer & Software Engineering

Research Article Open Access

Reversible Data Hiding Based on Image Interpolation with a Secret Message Reduction Strategy

Tzu-Chuen Lu*, Mei-Chen Lin, Chun-Chih Huang and Kuang-Mao Deng

Department of Information Management, Chaoyang University of Technology, No. 168, Jifeng E Rd, Wufeng District, Taichung City, Taiwan

Abstract

In 2012, Lee et al. proposed an interpolation enlargement technique using neighboring pixels as the base on which secret messages are embedded in enlarged interpolated pixels. This method can effectively predict the pixel between two adjacent pixels so as to reduce the distortion caused by the hidden secret messages. However, in this study we found that while Lee et al.'s neighboring pixel prediction method can successfully achieve a predictive value, embedded secret messages of different sizes will cause great distortion when a large secret message is hidden in the predicted value. Therefore, the purpose of our research is to determine how to process secret messages in advance to reduce the harm caused by embedding. We propose a secret message reduction strategy that changes positive secret messages to both positive and negative to decrease their values and thereby reduce stego-image distortion.

Publication History:

Received: January 12, 2016
Accepted: May 17, 2016
Published: May 19, 2016

Keywords:

Information Hiding, Reversible Data Embedding, Interpolation

Introduction

The information age has recently witnessed an unprecedented increase in digital media transfer. However, messages transferred over the Internet are vulnerable to illegal interception by a third party [15]. Therefore, researchers have developed data hiding techniques to embed secret messages in media to successfully transfer and protect confidential information. Data hiding has been widely used in many applications, including secret message sharing, image authentication, ownership assertion, tamper detection, and watermarking. This study focuses on secret message sharing.

Data hiding is divided into non-reversible data hiding (Non-RHD) and reversible data hiding (RDH), depending on whether a stego-image can be restored to the original image. When a stego-image cannot be restored to the original image after the extraction of secret messages, it is known as a non-RHD technique. In contrast, an RDH technique can restore a stego-image to the original image, enabling the reuse of images. Therefore RDH techniques are often used in high-value images such as military maps, medical imaging, and works of art [11, 16, 17]. However, the reuse of images comes at the expense of embedding capacity. Therefore, a major focus of RDH research is how to improve the embedding capacity while maintaining image restorability.

The most commonly used RDH methods are the histogram-shifting technique, the difference expansion method, the dual-image-based method, and image-interpolation-based schemes. The histogram-shift method, first proposed by Ni et al. in 2006, analyzes the pixel (or error value) distribution of an image to build a statistical histogram [9]. Then, the scheme shifts the pixel values, and embeds the secret bit into frequently occurring, or peak point values. A weakness of this method is its low-hiding capacity.

The difference expansion method embeds secret messages using the differences between pixels (or between the original and predicted pixels). For example, the difference expansion prototype, as proposed by Tian in 2003, doubles two adjacent pixels to embed one bit of secret message [13]. In 2004, Alatter extended Tian's method by embedding three bits of secret messages in the doubled difference distance between four neighboring pixels [1]. In 2009, Lou et al. proposed a multi-level data hiding technique, which takes two horizontal pixels as an embedding unit, using a column-based strategy with an odd number of rounds, and two vertical pixels as the embedding unit, using a field-based strategy with an even number of rounds. Then, Tian's method is employed for embedding when the difference distance declines after multiple embedding [8]. In 2012, Qin et al. proposed a method that embeds two bits of secret messages in a difference distance, calculated using the original pixels and the average of three adjacent pixels [12]. In 2013, Ou et al. doubled the error threshold, based on the length of the secret messages to be embedded and using the partial differential equation (PDE) as a predictor [10]. In 2013, Li et al. applied difference expansion to embedding secret message in color images, using Tian's method after converting the color images into RGB color charts and then choosing a threshold-based prediction chart [6]. In 2014, Gui calculated the complexity of each pixel using a gradient-adjusted predictor (GAP), which was then used as a basis for difference expansion classification and secret message embedding [3]. In 2014, Lu et al. [7] proposed a reversible method based on difference expansion, histogram shifting and interpolation strategies to conceal secret data in the reference pixels for increasing the hiding payload. After that, Govind and Wilscy proposed a enhanced reversible hiding scheme based on directional interpolation and difference expansion. In their scheme interpolation method is used to get more accurate prediction value for reducing the prediction error and finding more embeddable number of pixels in 2015 [2].

The disadvantage of the difference expansion method is that if the difference distance is too large, the image quality will be severely degraded after expansion. Image quality damage after expansion is less for small difference distanced. Therefore, technical discussions of this technique focus on how to reduce the difference.

Another emerging RDH technique is image enlargement, which embeds secret messages while inserting fictional pixels into existing pixels. Examples of this method include that by Jung and Yoo in 2009, who proposed the neighbor mean interpolation (NMI) [4],

***Corresponding Author:** Dr. Tzu-Chuen Lu, Department of Information Management, Chaoyang University of Technology, No. 168, Jifeng E Rd, Wufeng District, Taichung City, Taiwan; E-mail: tclu@cyut.edu.tw

Citation: Lu TC, Lin MC, Huang CC, Deng KM (2016) Reversible Data Hiding Based on Image Interpolation with a Secret Message Reduction Strategy. Int J Comput Softw Eng 1: 102. doi: http://dx.doi.org/10.15344/ijcse/2016/102

Lee and Huang's proposed interpolation by neighboring pixels (INP) in 2012 [5], and Tang and Song's proposed high-capacity reversible steganography (CRS) in 2014 [14].

In contrast to difference expansion methods, the size of the stego-image from image-enlargement-based hiding methods is larger than that from either difference-expansion-based hiding schemes or histogram-based methods. The sizes of stego-images produced by image enlargement methods are 1-2 times larger than the size of the original image. The secret messages must be extracted from the fictional pixels in the extraction phase, and then the image is reduced to obtain the original image. However, the stego-images produced by such methods are likely to have a jagged blur, and the embedding capacity is also low. Therefore, in this study we propose a reversible data hiding method to improve embedding capacity and image quality.

In this study, we examine the performance of the INP, NMI, and CRS image enlargement methods. According to our experimental results, the INP method proposed by Lee and Huang outperforms the other two. Hence, our proposed method applies INP image enlargement to calculate a virtual predictive value and then embeds a fixed secret message length to obtain the highest embedding capacity while yielding image qualities similar to those of other methods. We use a secret message reduction strategy, as proposed in our previous research [19], in which a secret message is preprocessed before embedding to reduce the image quality damage caused by large-value bits. Based on our experimental results in [19], this secret message reduction strategy can indeed improve image quality. In this paper, we advance this strategy and apply it to general and nature images. We describe the three image enlargement methods in section two. In section three, we describe our research process and provide examples. In section four, we discuss our experimental results. Finally, in section five we offer our analysis and conclusions.

Related Work

Neighbor Mean Interpolation (NMI)

Neighbor mean interpolation (NMI), a data hiding method proposed by Jung and Yoo in 2009 [4], first produces a predictive image by enlarging an original image, and then embeds into it a secret messages to generate a stego-image. Once received, the receiver extracts the secret messages from the stego-image. Pixel enlargements are shown in Figure 1. Figure 1(a) shows the original pixels, which are then inserted with a predicted pixel into any two horizontal or vertical pixels of the four. Figure 1(b) shows the enlarged result. is calculated using NMI by taking the average value of the adjacent pixels as the virtual predictive value, as follows:

$$P_{(i,j)}^{\mathrm{NMI}} = \begin{cases} \left\lfloor \frac{\left(I_{(i,j-1)} + I_{(i,j+1)}\right)}{2} \right\rfloor, & \text{if} \quad i = 2m, j = 2n+1, \\ \left\lfloor \frac{\left(I_{(i-1,j)} + I_{(i+1,j)}\right)}{2} \right\rfloor, & \text{if} \quad i = 2m+1, j = 2n, \\ \left\lfloor \frac{\left(I_{(i-1,j-1)} + I_{(i-1,j)} + I_{(i,j-1)}\right)}{3} \right\rfloor, & \text{otherwise.} \end{cases}$$

In Formula 1, m and n are the height and width of the original image block, and i and j are the pixel positions. Taking Figure 2 as an example, the original image is shown in Figure 2(a), where $I_{(0,0)}$=229, $I_{(0,1)}$=232, $I_{(1,0)}$=233, and $I_{(1,1)}$=231. Assume that the secret message is $s = (10011)_2$. The virtual predictive values $P_{(0,1)}^{\mathrm{NMI}}$, $P_{(1,0)}^{\mathrm{NMI}}$, and $P_{(1,1)}^{\mathrm{NMI}}$ are then calculated using Formula 1, where $P_{(0,1)}^{\mathrm{NMI}} = \frac{(229+232)}{2} = 230$

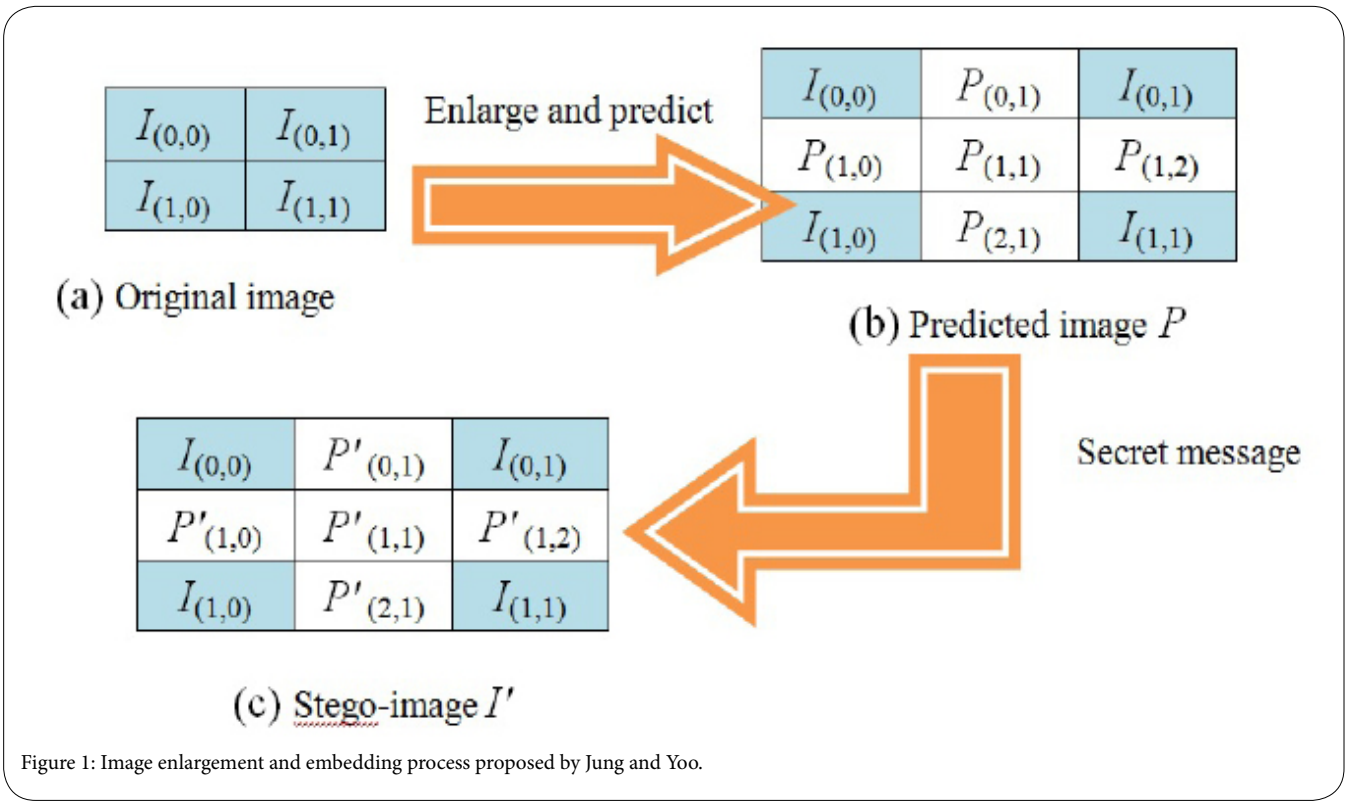

Figure 1: Image enlargement and embedding process proposed by Jung and Yoo.

$P_{(1,0)}^{\mathrm{NMI}} = \frac{(229+233)}{2} = 231$ and the virtual predictive value of the intermediate pixel is the average of $I_{(0,0)}$, $P_{(0,1)}^{\mathrm{NMI}}$, and $P_{(1,0)}^{\mathrm{NMI}}$, namely $P_{(1,1)}^{\mathrm{NMI}} = \frac{(229+230+231)}{3} = 230$. The virtual predictive values are shown in Figure 2(b). The differences d_1^{NMI}, d_2^{NMI}, and d_3^{NMI}, are obtained by subtracting the original pixel $I_{(0,0)}$ from the three virtual predictive values, $P_{(0,1)}^{\mathrm{NMI}}$, and $P_{(1,0)}^{\mathrm{NMI}}$, and $P_{(1,1)}^{\mathrm{NMI}}$ as follows:

$$\begin{cases} d_1^{\mathrm{NMI}} = \left| P_{(0,1)}^{\mathrm{NMI}} - I_{(0,0)} \right|, \\ d_2^{\mathrm{NMI}} = \left| P_{(1,0)}^{\mathrm{NMI}} - I_{(0,0)} \right|, \\ d_3^{\mathrm{NMI}} = \left| P_{(1,1)}^{\mathrm{NMI}} - I_{(0,0)} \right|. \end{cases} \quad (2)$$

Continuing with the example above: d_1^{NMI} = |230 - 229| = 1, d_2^{NMI} = |231 - 229| = 2, and d_3^{NMI} = |230 - 229| = 1. We then take the log of difference to calculate the length of the secret message n^{NMI}, as follows:

$$\begin{cases} n_1^{\mathrm{NMI}} = \log_2\left(d_1^{\mathrm{NMI}}\right), \\ n_2^{\mathrm{NMI}} = \log_2\left(d_2^{\mathrm{NMI}}\right), \\ n_3^{\mathrm{NMI}} = \log_2\left(d_3^{\mathrm{NMI}}\right). \end{cases} \quad (3)$$

Substitute the calculated value d^{NMI} into Formula 3 to get $n_1^{NMI} = \lfloor \log_2(1) \rfloor = 0$, $n_2^{NMI} = \lfloor \log_2(2) \rfloor = 1$, where $n_2^{NMI} = 0$ and $n_3^{NMI} = 0$ indicate that the virtual predictive $P_{(0,1)}^{\mathrm{NMI}}$ values $P_{(1,1)}^{\mathrm{NMI}}$ and are not to be embedded, and $n_2^{NMI} = 1$ indicates that one bit of the secret message *s* is cut from the left. The cut message is then converted into the decimal secret symbol *b*, and embedded into the virtual predictive value $P_{(1,0)}^{\mathrm{NMI}}$. For example, the secret message cut from *s* for n_2^{NMI} is $(1)_2$, $b = (1)_{10}$ after converting it to a decimal number, and *b* is embedded in $P_{(1,0)}^{\mathrm{NMI}}$ to obtain $P'^{\mathrm{NMI}}_{(1,0)} = P_{(1,0)}^{\mathrm{NMI}} + b = 231 + 1 = 232$. The final result is shown in Figure 2(c).

$$P_{(i,j)}^{\mathrm{INP}} = \begin{cases} \left\lfloor \frac{\left(I_{(i,j-1)} + \left(I_{(i,j-1)} + I_{(i,j+1)}\right)/2\right)}{2} \right\rfloor, & \text{if } i = 2m, j = 2n+1, \\ \left\lfloor \frac{\left(I_{(i-1,j)} + \left(I_{(i-1,j)} + I_{(i+1,j)}\right)/2\right)}{2} \right\rfloor, & \text{if } i = 2m+1, j = 2n, \\ \left\lfloor \frac{\left(P_{(i-1,j)}^{\mathrm{INP}} + P_{(i,j-1)}^{\mathrm{INP}}\right)}{2} \right\rfloor, & \text{otherwise.} \end{cases} \quad (4)$$

The example here uses the same original pixels as those in section 2.1. Assume the reduced pixels are $I_{(0,0)}$ = 229, $I_{(0,1)}$ = 232, $I_{(1,0)}$ = 233, and $I_{(1,1)}$ = 231. We calculate the virtual predictive values $P_{(0,1)}^{\mathrm{INP}}$, $P_{(1,0)}^{\mathrm{INP}}$, and $P_{(1,1)}^{\mathrm{INP}}$ using Formula 4:

$$P_{(0,1)}^{\mathrm{INP}} = \left(I_{(0,0)} + \frac{(I_{(0,0)} + I_{(0,1)})}{2}\right)/2 = \left(229 + \frac{(229+232)}{2}\right)/2 = 229$$

$$P_{(1,0)}^{\mathrm{INP}} = \left(I_{(0,0)} + \frac{(I_{(0,0)} + I_{(1,0)})}{2}\right)/2 = \left(229 + \frac{(229+233)}{2}\right)/2 = 230$$

and the middle virtual predictive value is the average of $P_{(0,1)}^{\mathrm{INP}}$ and $P_{(1,0)}^{\mathrm{INP}}$, namely, $P_{(1,1)}^{INP} = \left\lfloor \frac{(229+230)}{2} \right\rfloor = 229$. Figure 3(a) shows the predicted results using INP. Before calculating differences, d_1^{INP}, d_2^{INP} and d_3^{INP} we first a calculate the fixed value B, which takes the maximum pixel as the base value for subtracting the virtual predictive value, thereby improving the embedding capacity, as follows:

$$B = \max\{I_{(0,0)}, I_{(0,1)}, I_{(1,0)}, I_{(1,1)}\}. \quad (5)$$

Next, we select the maximum value, using Formula 5, from the four corners as the basis for subtracting the virtual predictive value. For example, if *B* = max(129,132,133,131) = 133, we subtract the virtual predictive value from the calculated value of B to get the difference, as follows:

$$\begin{cases} d_1^{\mathrm{INP}} = \left| B - P_{(0,1)}^{\mathrm{INP}} \right|, \\ d_2^{\mathrm{INP}} = \left| B - P_{(1,0)}^{\mathrm{INP}} \right|, \\ d_3^{\mathrm{INP}} = \left| B - P_{(1,1)}^{\mathrm{INP}} \right|. \end{cases} \quad (6)$$

229	232
233	231

(a) Original image

229	230	232
231	230	
233		231

(b) Predicted result of NMI

229	230	232
232	230	
233		231

(c) Stego-image

Figure 2: NMI embedding example (Jung and Yoo).

Interpolation by Neighboring Pixels (INP)

Interpolation by neighboring pixels (INP), proposed by Lee and Huang in 2012, also produces a predictive image first by enlarging an original image, and then embeds secret messages to obtain a stego-image [5]. The virtual predictive value P^{INP} is calculated using INP, which takes the average value of the original pixel $I_{(0,0)}$ and the average of the two adjacent pixels, as the virtual predictive value. The INP formula is as follows:

Continuing with the above example, $d_1^{\mathrm{INP}} = |233 - 229| = 4$, $d_2^{\mathrm{INP}} = |233 - 230|$, and $d_3^{\mathrm{INP}} = |233 - 229|$ We then calculate the log difference d^{INP} to obtain the secret message length n^{INP}, as follows:

$$\begin{cases} n_1^{\mathrm{INP}} = \log_2(d_1^{\mathrm{INP}}), \\ n_2^{\mathrm{INP}} = \log_2(d_2^{\mathrm{INP}}), \\ n_3^{\mathrm{INP}} = \log_2(d_3^{\mathrm{INP}}). \end{cases} \quad (7)$$

Next, substitute the calculated value d^{INP} into Formula 7 to get $n_1^{INP} = \lfloor \log_2(4) \rfloor = 2$, $n_2^{INP} = \lfloor \log_2(3) \rfloor = 1$ $n_3^{INP} = \lfloor \log_2(4) \rfloor = 2$ and Since n_1^{INP}=2, we cut two bits of secret message *s*, $(10)_2$ from the left, convert it into the decimal secret symbol $b(2)_{10}$, and then embed it in the virtual predictive value $P_{(0,1)}^{INP}$, $P'^{INP}_{(0,1)} = P^{INP}_{(0,1)} + b$ = 229 + 2 = 231. By embedding the secret messages in this way, we obtain the final results shown in Figure 3(b).

229	229	232
230	229	
233		231

(a) Virtual predictive

229	231	232
230	232	
233		231

(b) Embedding result

Figure 3: INP embedding example.

High Capacity Reversible Steganography (CRS)

High-capacity reversible steganography (CRS), proposed by Tang and Song in 2014 [14], selects the maximum I_{max} and minimum I_{min} values from pixels at the four corners before calculating the virtual predictive value . The formula is as follows:

$$I_{Max} = \max\{I_{(0,0)}, I_{(0,1)}, I_{(1,0)}, I_{(1,1)}\},$$

$$I_{Min} = \min\{I_{(0,0)}, I_{(0,1)}, I_{(1,0)}, I_{(1,1)}\}. \quad (8)$$

The reference value AD, calculated using the maximum I_{max} and minimum I_{min}, refers to the maximum and minimum pixels at the four corners, so the calculated virtual predictive value is more accurate. The formula for calculating *AD* is as follows:

$$AD = (3 \times I_{min} + I_{max})/4. \quad (9)$$

Next, we use the values from the same example used in Formulas 8 and 9 to get I_{max} = max{229,232,233,231} = 233, I_{min} = min{229,232,233,231} = 229, and *AD* = (3×229+233)/4 = 230. The average of the calculated reference value AD and the average of the two adjacent pixels is, P^{CRS} calculated as follows:

$$P_{(i,j)}^{CRS} \text{ üüüüüüü } \begin{cases} \left\lfloor \frac{\left(AD + \left(I_{(i,j-1)} + I_{(i,j+1)}\right)/2\right)}{2} \right\rfloor, & \text{if } i = 2m, j = 2n+1, \\ \left\lfloor \frac{\left(AD + \left(I_{(i-1,j)} + I_{(i+1,j)}\right)/2\right)}{2} \right\rfloor & i \quad m \quad n \\ \left\lfloor \frac{\left(I_{(i-1,j-1)} + P_{(i-1,j)}^{CRS} + P_{(i,j-1)}^{CRS}\right)}{3} \right\rfloor, & \text{otherwise.} \end{cases} \quad (10)$$

Continuing with the above example, and using Formula 10, $P_{(0,1)}^{CRS} = \left\lfloor \left(230 + \frac{(229+232)}{2}\right)/2 \right\rfloor = 230$, and $P_{(1,0)}^{CRS} = \left(230 + \frac{(229+233)}{2}\right)/2 = 230$

The calculation of $P_{(1,1)}^{CRS}$ differs from that of $P_{(0,1)}^{CRS}$ and $P_{(1,0)}^{CRS}$. The virtual predictive value is the average of the above two values and the original pixel, $P_{(1,1)}^{CRS} = \frac{(229+230+230)}{3} = 229$, Figure 4(a) shows the predicted result. With the CRS method, the difference is determined by comparing the virtual predictive value and the average of maximum I_{max} and minimum I_{min} values, as follows:

$$d_1^{CRS} = \begin{cases} I_{max} - P_{(0,1)}^{CRS}, & \text{if } P_{(0,1)}^{CRS} < (I_{min} + I_{max})/2, \\ P_{(0,1)}^{CRS} - I_{min}, & \text{if } P_{(0,1)}^{CRS} \geq (I_{min} + I_{max})/2. \end{cases}$$

$$d_2^{CRS} = \begin{cases} I_{max} - P_{(1,0)}^{CRS}, & \text{if } P_{(1,0)}^{CRS} < (I_{min} + I_{max})/2, \\ P_{(1,0)}^{CRS} - I_{min}, & \text{if } P_{(1,0)}^{CRS} \geq (I_{min} + I_{max})/2. \end{cases} \quad (11)$$

$$d_3^{CRS} = \begin{cases} I_{max} - P_{(1,1)}^{CRS}, & \text{if } P_{(1,1)}^{CRS} < (I_{min} + I_{max})/2, \\ P_{(1,1)}^{CRS} - I_{min}, & \text{if } P_{(1,1)}^{CRS} \geq (I_{min} + I_{max})/2. \end{cases}$$

Entering these values into Formula 11, we obtain d_1^{CRS} = 233-230=3, d_2^{CRS} = 233-230 = 3, and d_3^{CRS} = 233-229 = 4 . We then calculate the log of the three differences to determine the secret message lengths, n_1^{CRS} = 1, n_2^{CRS} = 1, and n_3^{CRS} = 2. The three secret message sections cut out are then $b_1 = (1)_2 = (1)_{10}$, $b_2 = (0)_2 = (0)1_0$, and $b_3 = (01)_2 = (1)_{10}$, respectively. Finally, to complete the embedding procedure, the secret message is embedded by subtracting the three virtual predictive values. The final results are shown in Figure 4(b).

229	230	232
230	229	
233		231

(a) Predicted result

229	229	232
230	228	
233		231

(b) Embedding result

Figure 4: CRS embedding example.

Proposed Scheme

In this study, for image enlargement and prediction, we employ the embedding rules of the INP prediction method described in section 2.3, whereby secret messages are hidden between virtual predictive values $P_{(i,j)}$. The reasons we chose to use INP for image expansion are shown in Figure 5 and Table 1.

The diagram in Figure 5 demonstrates our comparison of the interpolated and original images. The proposed scheme reduces a 256 × 256-pixel original image to a 128 × 128-pixel reduced image, which is then used to produce an enlarged interpolated image using the NMI, INP, and CRS techniques. Then, we compared the image qualities of the three interpolated images and the original image. Table 1 shows that Lee and Huang's virtual predictive value yielded a higher accuracy and PSNR value. Therefore, we employed the INP technique to enlarge the original image in this study.

Figure 6 is a diagram of the embedding procedure we used. First, the original image is enlarged using INP. Next, the secret message is folded using the proposed message reduction strategy. The folded message is then embedded into the interpolated image to generate a stego-image. The embedding process is described in detail in section 3.1 below.

Image Method	Lena	Baboon	Jet	Peppers	Scene	Tiffany
NMI [4]	30.85	22.88	28.70	29.95	27.53	29.76
INP [5]	31.79	23.50	29.28	30.42	28.14	30.15
CRS [14]	29.13	22.34	27.19	28.44	26.20	29.05

Table 1: Comparison of PSNR values (in dB) of NMI-, INP- and CRS-interpolated and original images.

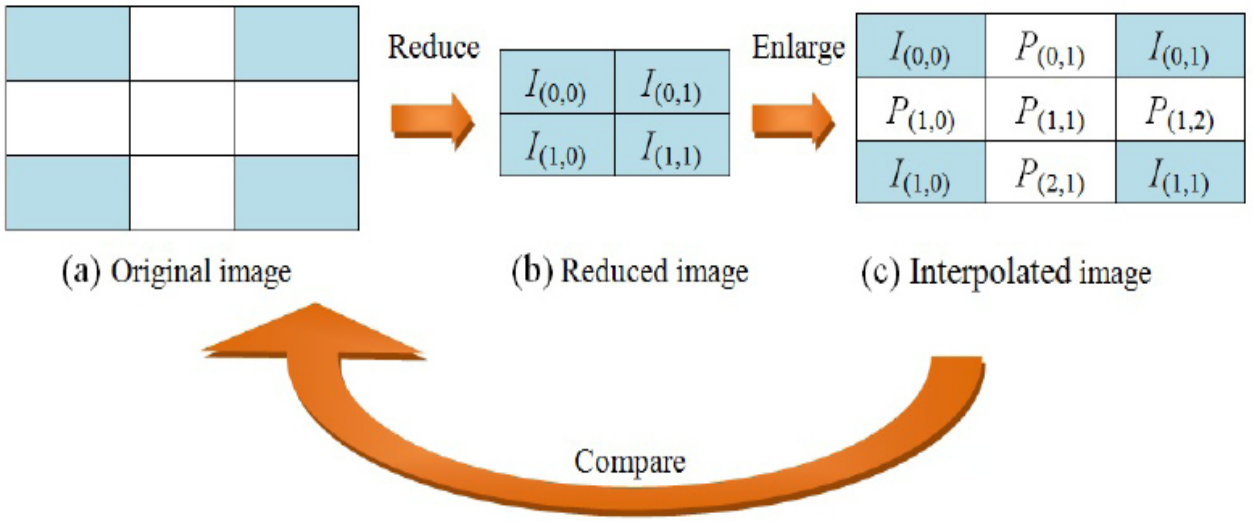

Figure 5: Diagram comparing the interpolated and original images.

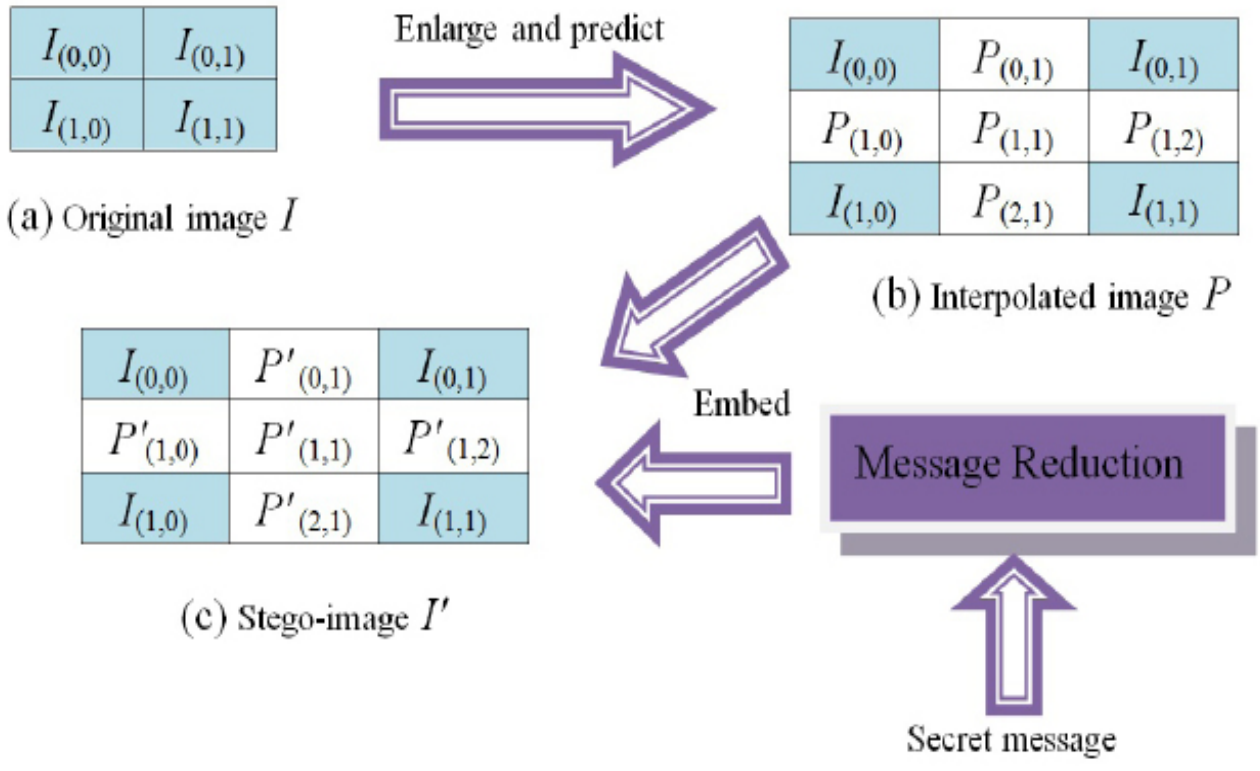

Figure 6: Embedding procedure of the proposed scheme.

Embedding process

To enhance the security of the proposed scheme, a cryptography algorithm can be used first to encrypt the secret message. After being encrypted, the secret message is folded using the proposed message reduction strategy to conceal the interpolated image.

The virtual predictive value $P_{(i,j)}$, as described in section 2.2, is calculated using two adjacent pixels. A 4-bit fixed-length secret message s is embedded into each virtual predictive value. The secret message is transformed into a decimal number b. The secret symbol b, from 0 to 15, is obtained after converting the secret message into a decimal number. In this study, we set four bits of a secret message to be hidden in each predicted value, so the secret symbol value range was [0, 15]. If a secret symbol is embedded directly into a predicted value, it can cause large stego-pixel distortion. Therefore, we first changed the secret symbols to positive and negative values by subtracting eight, the intermediate value of the range, from the secret messages. In this way, the range of positive numbers changed to [-8, 7]. The reduction formula is as follows:

$$b'=b-8 \tag{12}$$

For example, if we suppose the secret message is b = 3, we subtract from it the intermediate value eight to obtain b' = 3-8= -5 which is the reduced secret message. The proposed scheme replaces b with b' to embed it into the predicted value. We then add the reduced secret message b' and the predicted values to get stego-pixel P', as follows:

$$P'_{(i,j)} = P_{(i,j)} + b'.$$

When the predicted value is less than (or equal to) 7, however, the possibility that the reduced secret message may be equal to $b' = -8$ will result in stego-pixel P' being less than 0, therefore causing an underflow problem. Also, when the predicted value is greater than (or equal to) 249 and $b' = 7$, the stego-pixel $P' = 249 + 7 = 256$ will cause an overflow problem. Therefore, if the predicted value is less than (or equal to) 7 or greater than (or equal to) 248, extra judgment is required. If the predicted value $P_{(i,j)}$ is less than (or equal to) 7 or greater than (equal to) 248, the secret message will not be reduced, and we amend Formula 12 as follows:

$$b' = \begin{cases} b, & if P_{(i,j)} \le 7 or P_{(i,j)} \ge 249, \\ b-8, & otherwise. \end{cases} \tag{14}$$

Formula 13, used to produce a stego-pixel, must also be slightly amended as follows:

$$P'_{(i,j)} = \begin{cases} P_{(i,j)} - b', & if P_{(i,j)} \ge 249, \\ P_{(i,j)} + b', & \text{otherwise.} \end{cases} \tag{15}$$

When the predicted value is greater than (or equal to) 249, messages must not be hidden by addition as this may cause overflow. Instead, subtraction is used for message embedding.

The pseudo code of the embedding procedure for each embedding pixel is shown below:

Input: Predictive value $P_{(i,j)}$, secret message s

Output: Stego pixel $P'_{((i,j)}$

Step 1: Transform the secret message s into a decimal secret symbol b

Step 2: If $P_{(i,j)}$ <= 7 or $P_{(i,j)}$ >= 249) then b is reduced by using reduction strategy where $b'= b$ -8. Otherwise, $b' = b$.

Step 3: Embed the reduced symbol b' into $P_{(i,j)}$. If ($P_{(i,j)}$ >= 249) then $P'_{((i,j)} = P_{(i,j)} - b'$, otherwise, $P_{(i,j)} = P_{(i,j)} + b'$.

The embedding order begins with the first virtual predictive value and ends with the last virtual predictive value. For example, in Figure 7(a), the virtual predictive values $P_{(0,1)}$ = 129, $P_{(1,0)}$ = 130, and $P_{(1,1)}$ = 129 are first calculated using INP, and the secret message $s = (100110110101)_2$ is assumed. Four bits of secret message are embedded into each virtual predictive value, $s_1 = (1001)_2$, and are then converted into a secret symbol to get b_1 = (9)10. As such, the predicted value $P_{(0,1)}$ is neither less than (or equal to) 7 nor more than (or equal to) 249. We use the reduction formula to then get b' = 9-8 = 1. Next, Formula 15 is used to obtain: $P'_{(0,1)} = 229 + 1 = 230$. In this way, secret messages are embedded in all the virtual predictive values, as shown in Figure 7(b).

The pixels that may cause underflow/overflow are shown in Figure 7(c). Their predicted values are $P_{(0,1)}$ = 251, $P_{(1,0)}$ = 250, and $P_{(1,1)}$ = 250. If we assume that the secret message is $s = (110010110011)_2$, then $s_1 = (1100)_2$, we can convert it into a secret symbol to get b_1 = (12)10. The predicted value $P_{(0,1)}$ is greater than 249, so we do not reduce b_1, and $b'_1 = b_1 = 12$. Using Formula 15, we obtain: $P'_{(0,1)} = 251–12 = 239$, and continue the calculation in this way. The embedding results are shown in Figure 7(d).

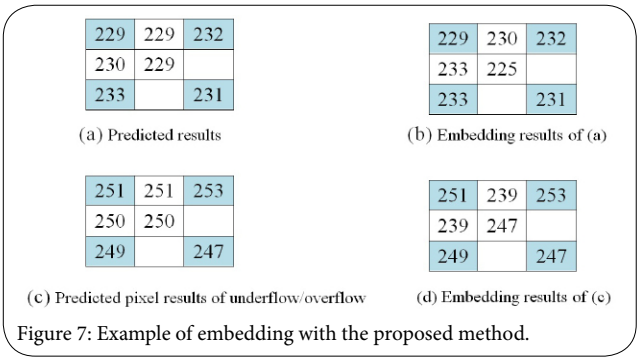

Figure 7: Example of embedding with the proposed method.

Extraction and restoration processes

After receiving stego-image I', the receiver first reduces and restores it to the reduced image, and then uses INP to enlarge the image to obtain the virtual predictive value. Next, the receiver subtracts the virtual predictive value from the stego-pixel to get the embedded message b', as follows:

$$b' = P'_{(i,j)} - P_{(i,j)}. \tag{16}$$

If the predicted value is less than (or equal to) 7 or greater than (or equal to) 249, b' is the original secret symbol. Otherwise, we add the intermediate value 8 to b' to restore secret symbol b. The formula is as follows:

Citation: Lu TC, Lin MC, Huang CC, Deng KM (2016) Reversible Data Hiding Based on Image Interpolation with a Secret Message Reduction Strategy. Int J Comput Softw Eng 1: 102. doi: http://dx.doi.org/10.15344/ijcse/2016/102

$$b = \begin{cases} b', & if P_{(i,j)} \le 7 or P_{(i,j)} \ge 249, \\ b' + 8, & otherwise. \end{cases} \quad (17)$$

Next, we convert b into a binary bit to extract the embedded secret messages, as shown in the flow chart in Figure 8. For example, in Figure 7, after receiving a stego-image, the receiver first reduces and then enlarges this image to obtain the virtual predictive values, $P_{(0,1)}$ = 229, $P_{(1,0)}$ = 230, and $P_{(1,1)}$ = 229. Then, the predicted value is subtracted from P' to extract the secret message $b'_1=P_{(0,1)}-P_{(0,1)}$=230-229=1, and 8 is added to b'_1 to get b = 1 + 8 = 9, which is $(1001)_2$ in binary. All secret messages are extracted in this way.

The extraction process from the pixels, after underflow/overflow processing, is the same, as shown in Figure 7(c). First, the predicted value is calculated to get $P_{(0,1)}$ = 251, $P_{(1,0)}$ = 250, and $P_{(1,1)}$ = 250, and then the predicted value is subtracted from the stego-pixel to get b'_1= |239-251|=12. Since the predicted value $P_{(0,1)}$ is greater than 249, b_1 is b'_1, and $b_1 = b'_1$ = 12, which is $(1100)_2$ in binary. The procedure continues in this way to extract the original secret message.

The pseudo code of the extraction procedure for each stego pixel is shown below:

Input: Stego pixel $P'_{(i,j)}$ and predictive pixel $P_{(i,j)}$

Output: secret message s

Step 1: Subtract $P_{(i,j)}$ from $P'_{(i,j)}$ to get embedded message b', where $b'=P'_{(i,j)}-P_{(i,j)}$

Step 2: Get the secret symbol b. If $P_{(i,j)} \le 7$ or $P_{(i,j)} \ge 249$ then $b = b'$. Otherwise, $b=b'+8$.

Step 3: Convert b into a binary bit to extract the embedded secret messages s.

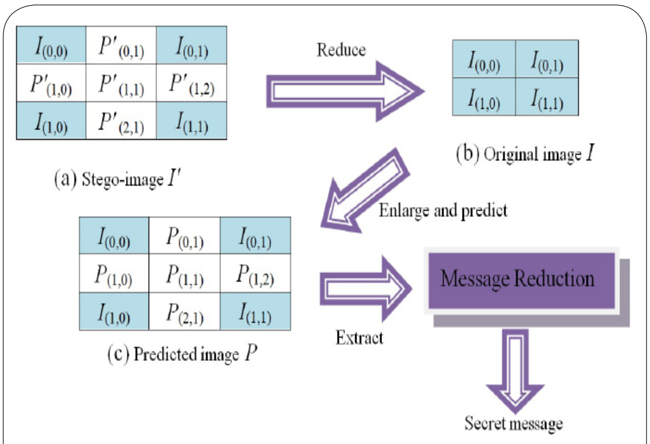

Figure 8: Extraction and recovery procedure of the proposed scheme.

Experimental Results and Discussion

We experimentally compared the method proposed in this paper with the NMI method proposed by Jung et al., the INP method proposed by Lee and Huang, and the CRS method proposed by Tang et al. In the experiments, we used six grayscale 256 × 256-pixel images for testing, as shown in Figure 9. After enlargement, the size of the test image was 512 × 512. We used MATLAB R2012a for the system development environment. Measurements were made using peak signal-to-noise ratio (PSNR), calculated as follows:

$$PSNR = \text{üüü¤§üü}_{10}\left[\frac{255^2}{\frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(x_{(i,j)} - x'_{(i,j)}\right)^2}\right] \quad (18)$$

where m×n is the image size. All the experimental images were 8-bit 256-color grayscale images. The greater the difference between the stego-pixel and the interpolated pixel, the higher the degree of image distortion and the lower the calculated PSNR. The smaller the difference, the lower the degree of distortion and the higher the PSNR.

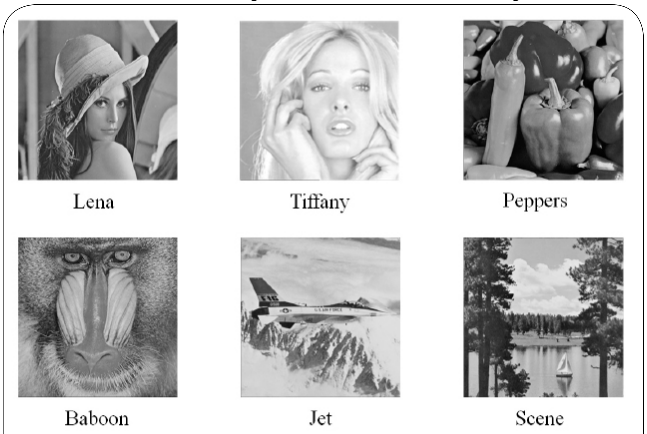

Figure 9: Experimental images.

In the experiments, we generated the secret message using a MATLAB random number generator (RNG). Table 2 shows the comparison results of the proposed and other three methods for the PSNRs and bits per pixel (bpp) of the stego- and interpolated images. We see that, except for being 1 db less than those of the NMI and INP methods for the Tiffany image, the PSNR of the proposed method is more than 2 db better than the other methods on all the other images, and may even be 5 db better on complex images such as the Baboon. Although the PSNR of the proposed scheme is less than that of other methods in some areas, the hiding capacity of the proposed scheme is more than 1.61 bpp higher than the others.

Whether the images are smooth or complex, the proposed method can embed 4-bit fixed-length secret messages. For an enlarged 512 × 512 image, the total number of embedded pixels is (512×512−256×256) = 196,608. Figure 10 shows a diagram of the embedding position. Cells marked +4 indicate that the pixel can be embedded with four pixels. As each predicted pixel can hide 4 bits, the maximum embedding capacity is 196,608 × 4 = 786,432 bits. The maximum bpp is 3786,432/(512×512) bpp, or up to 2.98 bpp, excluding the edge pixels, which cannot be embedded. As we can see in Table 2, the fixed embedding capacity of the proposed method is 2.98 bpp, which is more than 1 bpp better than that of the other methods.

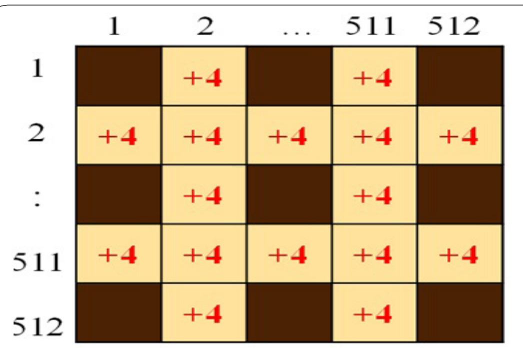

Figure 10: Diagram of the embedding position.

Method	NMI[3]		INP[4]		CRS[10]		Proposed Method	
Image	PSNR	BPP	PSNR	BPP	PSNR	BPP	PSNR	BPP
Lena	34.8	1.12	34.27	1.27	32.99	1.62	36.09	2.98
Baboon	30.32	2.08	29.8	2.13	29.72	2.5	35.63	2.98
Jet	33.05	1.05	32.64	1.19	31.58	1.51	36.09	2.98
Peppers	34.23	1.11	33.68	1.26	33.41	1.53	36.09	2.98
Scene	31.35	1.44	30.86	1.55	30.13	1.88	35.68	2.98
Tiffany	37.78	0.93	37.15	1.09	35.99	1.37	36.09	2.98

Table 2: PSNR and bpp comparisons of NMI, INP, CRS and proposed methods on stego- and interpolated images.

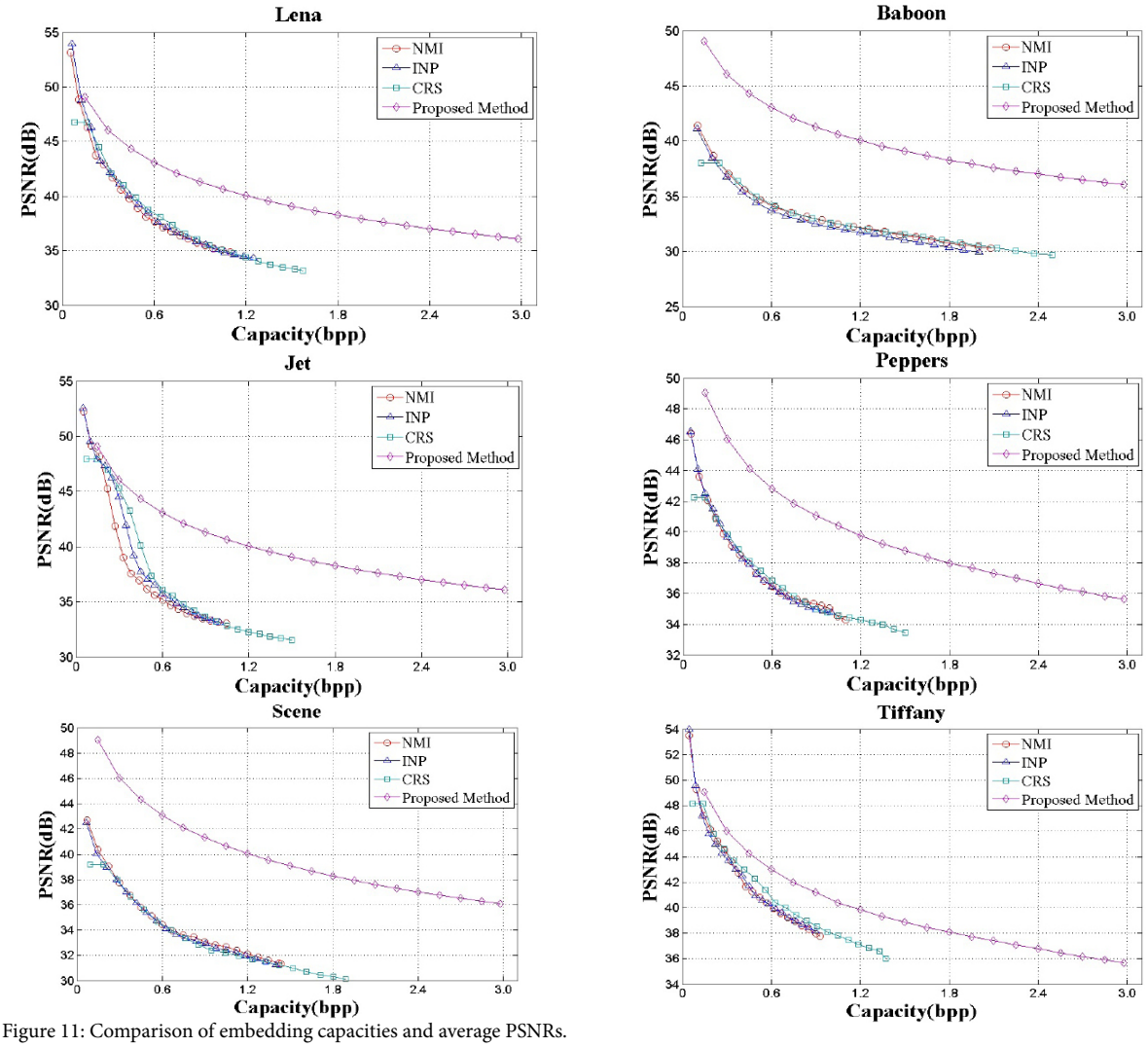

Figure 11: Comparison of embedding capacities and average PSNRs.

Images	Lu et al. [7]		Proposed Method		Govind and Wilscy [2]	
	PSNR	BPP	PSNR	BPP	PSNR	BPP
Lena	33.92	0.95	36.09	2.98	34.09	0.95
Baboon	28.39	0.71	35.63	2.98	31.60	0.89
Jet	33.61	0.94	36.09	2.98	33.77	0.94

Table 3: PSNR and bpp comparisons of Lu, Govind and Wilscy and proposed methods.

Figure 11 shows comparisons of the embedding capacities and PSNRs of the proposed and other three methods. Under the same embedding capacity, the total embedding capacity of the proposed method is approximately 1.5 bpp higher than that of the other methods and it maintains a stego-image quality above 30 dB.

Furthermore, the proposed scheme also compare the experimental results with two latest RDH schemes [2] [7]. Table 3 shows the comparison results. The image qualities and the hiding capacities of the proposed scheme are higher than that of the other two methods.

To prove that the proposed method is not only effective for specific images, we tested it on 1,338 images from the Uncompressed Colour Image Database (UCID) [18] (http://homepages.lboro.ac.uk/~cogs/datasets/ucid/ucid.html). All the UCID images are color images, sowere first converted into grayscale images for this test. Examples are shown in Figure 12.

To facilitate comparison, we sorted the experiment results according to the PSNR and bpp values of the proposed scheme. For example, Tables 4 and 5 show some experimental results for the proposed scheme and the other methods. Both tables are sorted by the PSNR and bpp value of the proposed scheme, in increasing order. In Table 4, the PSNR value using NMI on image number No. 1211 is 28.02 dB, that of INP is 27.67 dB, and that of the proposed scheme is 32.26 dB. The proposed scheme outperformed all the other methods. In Table 5, the hiding capacity using NMI on image number No. 1060 is 0.11 bpp, that of INP is 0.17 bpp, and that of the proposed scheme is 2.97 bpp.

The overall results are shown in Figs. 13 and 14. We see in Fig. 13 that the PSNRs for about 80% of the images by the proposed method are better than those of the other methods, and the stego-image quality of about 20% of the images is worse than in the other methods. A careful analysis of these 20% shows that most are very smooth images such as image Nos. 1060 and 603 in Table 5. The proposed method has an embedding capacity of 2.97 bpp regardless of whether the image is complex or smooth, while the other methods embed fewer messages in smooth areas. As such, the proposed method has a higher degree of distortion compared with the other methods.

From Table 5, we see that the fixed embedding capacity of the proposed method is 2.97 bpp, which is better than that of the other methods. We also see that the other methods are better than the proposed method for two images, Nos. 1030 and 1046 in Table 6, which are complex. The other methods have an embedding capacity of over 3 bpp, but their stego-image quality drops significantly. While the embedding capacity of the proposed method is about 1–2 bpp lower, the PSNR is over 14 db higher.

Therefore, the proposed method obtains good results regardless of whether the images are smooth or complex.

Figure 12: Eight example images from the UCID database.

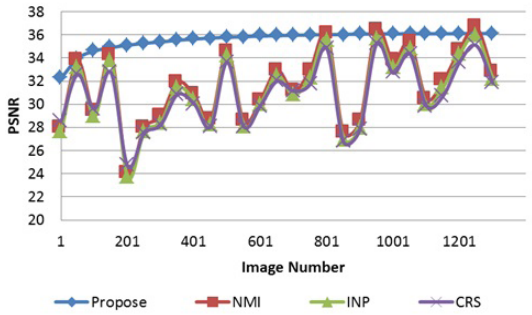

Figure 13: Comparison of PSNRs of UCID database images.

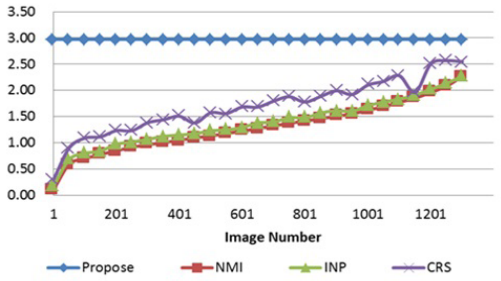

Figure 14: Comparison of bpp values of UCID database images.

Image Number	NMI	INP	CRS	Proposed
1211	28.02	27.67	28.53	32.26
449	33.72	33.27	32.58	33.98
554	29.44	28.96	29.61	34.61
608	34.18	33.81	32.75	34.90
…	…	…	…	…
349	32.72	32.16	31.84	36.11

Table 4: Comparison of PSNR values for different methods.

Image Number	NMI	INP	CRS	Proposed
1060	0.11	0.17	0.30	2.97
1007	0.59	0.68	0.90	2.97
328	0.70	0.80	1.09	2.97
307	0.78	0.85	1.12	2.97
…	…	…	…	…
353	2.27	2.27	2.55	2.97

Table 5: Comparison of bpp values for different methods.

Images	No.1060 (smooth image) 		No.603 (smooth image) 		No.1030 (complex image) 		No.1046 (complex image) 	
Methods	bpp	PSNR	bpp	PSNR	bpp	PSNR	bpp	PSNR
NMI[3]	0.11	45.41	0.11	43.37	3.43	21.10	3.20	21.30
INP[4]	0.17	46.49	0.22	42.65	3.40	20.89	3.15	20.96
CRS[10]	0.30	40.97	0.26	43.29	3.51	22.74	3.71	21.09
Proposed Method	2.97	33.87	2.97	36.11	2.97	36.11	2.97	36.11

Table 6: Comparison of bpp and PSNR values of some example images.

Conclusion

In this study, we proposed a RDH method based on INP image enlargement to shrink secret messages using a reduction strategy that reduces the damage caused by the enlargement process. The proposed method obtains good results regardless of whether the images are smooth or complex that is because the proposed scheme applied the secret message reduction strategy. The value range of the secret message was changed from [0, 15] to [-8, 7] by using secret message reduction strategy. The max distortion between the secret pixel and the original pixel is from $(\max(|0|, |15|))^2 = 225$ reduced to $(\max(|7|, |-8|))^2 = 64$. Hence, the secret message reduction strategy can effectively reduce the image distortion and improve the image quality.

To avoid underflow/overflow problems, the proposed method utilizes the concept of the same predicted value to process underflow/overflow pixels in different embedding directions. The receiver can obtain the same predicted value, and can therefore retrieve the correct information in the opposite direction.

Experimental results show that the embedding capacity of the proposed method is greater than that of the other methods when the image quality is similar. Our results also show that a 4-bit fixed-length secret message can be embedded regardless of whether an image is complex or smooth. Since the underflow/overflow problem is taken into account during the embedding process, there is no need for any additional recording of the underflow/overflow pixels. The more accurate the virtual predictive value, the better the image quality will be.

Competing Interests

The author declare that he has no competing interests.

Funding

This study was financially supported by the Research Grant MOST from Taiwan's Ministry of Science and Technology (MOST 103-2221-E-324 -014 -).

References

1. Alattar AM (2004) Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform. IEEE Transactions on Image Processing 13: 1147-1156.
2. Govind PVS, Wilscy M (2015) A New Reversible Data Hiding Scheme with Improved Capacity Based on Directional Interpolation and Difference Expansion. Procedia Computer Science 46: 491-498.
6. Gui X , Li X, Yang B (2014) A High Capacity Reversible Data Hiding Scheme Based on Generalized Prediction–Error Expansion and Adaptive Embedding. Signal Processing 98: 370-380.
7. Jung KH, Yoo KY (2009) Data Hiding Method Using Image Interpolation. Computer Standards & Interfaces 31: 465-470.
8. Lee CF, Huang YL (2012) An Efficient Image Interpolation Increasing Payload in Reversible Data Hiding. Expert Systems with Applications 39: 6712-6719.
9. Jian Li, Xiaolong Li, Yang B (2013) Reversible Data Hiding Scheme for Color Image Based on Prediction-Error Expansion and Cross-Channel Correlation. Signal Processing 93: 2748-2758.
10. Lu TC, Chang CC, Huang YH (2014) High Capacity Reversible Hiding Scheme Based on Interpolation, Difference Expansion and Histogram Methods. Multimedia Tools and Applications 72: 417-435.
11. Lou DC, Liu CL, Hu MC (2009) Multiple Layer Data Hiding Scheme for Medical Images. Computer Standards and Interfaces 31: 329-335.
12. Ni Z, Shi YQ, Ansari N, Su W (2006) Reversible Data Hiding. IEEE Transactions on Circuits and Systems for Video Technology 16: 354-362.
13. Oua B, Li X, Zhaoa Y, Ni R (2013) Reversible Data Hiding Based on PDE Predictor. The Journal of Systems and Software 86: 2700-2709.
14. Pan Z, Hu S, Ma X, Wang L (2015) Reversible data hiding based on local histogram shifting with multilayer embedding. Journal of Visual Communication and Image Representation. 31: 64-74.
15. Qina C, Chang CC, Li-Ting Liao LT (2012) An Adaptive Prediction-Error Expansion Oriented Reversible Information Hiding Scheme. Pattern Recognition Letters 33: 2166-2172.
16. Tian J (2003) Reversible Data Hiding Using a Difference Expansion. IEEE Transactions on Circuits and Systems for Video Technology 13: 890-896.
17. Tanga M, Hub J, Song W (2014) A High Capacity Image Steganography using Multi-layer Embedding. Optik 125: 3972-3976.
18. Vossberg M, Tolxdorff T, Krefting D (2008) DICOM Image Communication in Globus-Based Medical Grids. IEEE Transactions on Information Technology in Biomedicine 12: 145-153.
19. Jingyang Wen, Jinli Lei, and Yi Wan (2012) Reversible Data Hiding Through Adaptive Prediction and Prediction Error Histogram Modification. International Journal of Fuzzy Systems 14: 244-256.
20. Wua HT, Huangb J, Shi YQ (2015) A reversible data hiding method with contrast enhancement for medical images. Journal of Visual Communication and Image Representation 31: 146-153.
21. Schaefer G, Stich M (2004) UCID - An Uncompressed Colour Image Database. In Storage and Retrieval Methods and Applications for Multimedia pp. 472-480.
22. 呂慈純*、林美辰、黃俊智, "基於影像內插技術之可逆式資訊隱藏使用機密訊息縮減策略", The 9th International Conference on Advanced Information Technologies/Consumer Electronics Forum (AIT/CEF 2015), Taiwan, Taichung, Apr. 2015.