Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **2**, March **2018**

Prediction-Based Reversible Data Hiding Using Energy Deviation Strategy

Chih-Cheng Chen[1,*], Ching-Yen Lee[2], Gwoboa Horng[1],
Lee-Jang Yang[2] and Ying-Hsuan Huang[2]

[1]Department of Computer Science and Engineering
National Chung Hsing University, Taichung 40227, Taiwan
salu.chen@gmail.com; gbhorng@cs.nchu.edu.tw

[2]Aeronautical Systems Research Division
National Chung-Shan Institute of Science and Technology, Taichung 40722, Taiwan, R.O.C.
chingyen@ms2.hinet.net; calsasrd@hotmail.com; ying.hsuan0909@gmail.com

Received January, 2017; revised October, 2017

Abstract. *In recent years, many reversible data hiding methods have been proposed to avoid hacker's interception attacks, e.g., histogram shifting and prediction error expansion. The histogram shifting method only modified the pixels between the peak point and the zero point, so the visual quality of the image is satisfactory. However, only some pixels that are equal to the peak point can be used to embed secret data. The prediction error expansion method has a higher hiding capacity during an embedding round than histogram shifting. However, expanding the prediction error between the cover image and its prediction image causes serious distortion of the image. In this paper, first, we proposed an energy deviation method to reduce the error in the prediction, thereby enhancing the number of embeddable pixels and decreasing the level of expansion. Then, we combined our method with the histogram shifting method to reduce the number of modified pixels. The experimental results showed that the proposed method can embed more secret data than previous methods. In addition, the proposed method has a greater peak signal-to-noise ratio value than previous methods.*

Keywords: histogram shifting, Prediction error expansion, Energy deviation, Embedding rate, PSNR value

1. Introduction.

Reversible data hiding is used to embed specific messages into multimedia, e.g., images [1-12], videos, and audio files. The multimedia can be recovered losslessly after all of the specific messages have been extracted. This property makes it possible for reversible data hiding to be used in many applications, e.g., steganography and watermarking.

The difference expansion technique was developed in 2003 [1], in which the difference of a pixel-pair is doubled to embed a secret bit. However, difference expansion may cause serious visual distortion of the stego image as well as overflow and underflow problems. In order to improve the quality of the stego image, Thodi and Rodriguez [2] applied an inherent edge-detection method that can derive a prediction value by three neighboring pixels of the current hiding pixel. Since the prediction error between the current hiding pixel and the prediction value is small, expanding these smaller values does not cause serious distortion of the image. However, there are a few overflow and underflow problems in the boundary's pixels.

In order to avoid this problem, Ni et al. [3] proposed a histogram shifting method in which the pixels between the peak point and the zero point are shifted by one unit, and the secret data are embedded into the peak point. The embedding way controls the maximum modification level of pixels within 1, thus the visual quality of the stego image is better than that of other reported methods. In addition, the method does not modify the boundary pixels to skillfully avoid overflow and underflow problems.

Thodi and Rodriguez [4] combined the prediction error expansion method [2] with histogram shifting method [3] to embed a large amount of secret data and maintain satisfactory quality of the stego image. In 2008, Fallahpour [5] used the gradient-adjusted prediction (GAP) method to generate more exact prediction value, thereby reducing the level of prediction error expansion. Consequently, the method retained better image quality than the previous methods. However, there are some overflow and underflow problems in Fallahpour's method.

In order to solve the overflow and underflow problems, Tai et al. [7] modified the boundary pixels as the embeddable pixels. As a result, the hiding capacity of Tai et al.'s method is higher than that of previous methods. However, their method requires a location map to record whether or not the pixel is a boundary pixel. The location map decreases the hiding capacity and the image quality. In order to reduce the size of the location map, Tseng and Hsieh [6] only recorded the least significant bits (LSBs) of the boundary pixels and those of a few pixels overlapped with the modified boundary pixels. In other words, they only recorded a few LSBs as the reference information for recovering the original image. However, their hiding method only embeds secret data into the prediction errors ranged from half of threshold and threshold, limiting the hiding capacity.

Therefore, Lee et al. [8] used the difference expansion method to improve Tseng and Hsieh's method, thereby enhancing the hiding capacity. However, both the hiding capacity and the image quality can be improved further by adding our energy deviation.

The organization of the paper is as follows. First of all, we introduced several classical reversible data hiding methods in Section 1. Then, Sections 2 and 3 described Thodi and Rodriguez's method and the proposed method. Finally, experimental results and conclusions were given in Sections 4 and 5, respectively.

2. **Related Work.**

In 2004, Thodi and Rodriguez applied an inherent edge-detection method to embed secret data into the prediction error between the cover pixel and its prediction value, where the prediction value $\hat{I}_{x,y}$ of the cover pixel $I_{x,y}$, was derived by the three neighboring pixels $\{I_{x-1,y}, I_{x,y-1}, I_{x-1,y-1}\}$, i.e.,

$$\hat{I}_{x,y} = \begin{cases} min(I_{x-1,y}, I_{x,y-1}), & \text{if } I_{x-1,y-1} \geq max(I_{x-1,y}, I_{x,y-1}), \\ max(I_{x-1,y}, I_{x,y-1}), & \text{if } I_{x-1,y-1} \leq min(I_{x-1,y}, I_{x,y-1}), \\ I_{x-1,y} + I_{x,y-1} - I_{x-1,y-1}, & \text{otherwise,} \end{cases}$$

where x and y represent coordinates in the horizontal axis and vertical axis, respectively. Fig. 1 shows the cover pixel and its three neighboring pixels. Afterwards, the prediction error $E_{x,y}$ between $I_{x,y}$ and $\hat{I}_{x,y}$ was calculated, i.e., $E_{x,y} = I_{x,y} - \hat{I}_{x,y}$. The prediction error $E_{x,y}$ can be used to embed one secret bit s, i.e., $E'_{x,y} = 2 \times E_{x,y} + s$. Finally, the prediction error $E'_{x,y}$ is increased by the prediction value $\hat{I}_{x,y}$ to obtain the stego pixel $I'_{x,y}$, i.e., $I'_{x,y} = E'_{x,y} + \hat{I}_{x,y}$.

An example was used to illustrate the above procedure. Fig. 2 shows that the current hiding pixel was 105 and its three neighboring pixels were 112, 110, and 107. First, the prediction value of the current hiding pixel was generated by Eq. (1), i.e., $\hat{I}_{2,2} =$

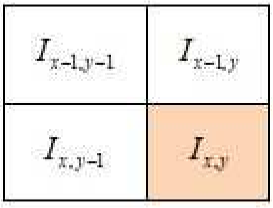


FIGURE 1. Current hiding pixel and its three neighboring pixels

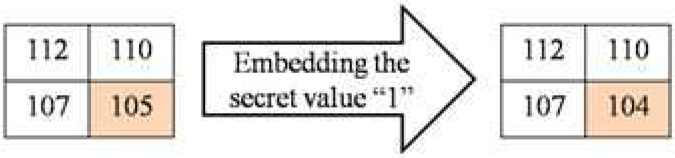


FIGURE 2. Example of embedding data

$min(110, 107) = 107$. Then, the prediction error between the current hiding pixel and the prediction value can be obtained, i.e., $E_{2,2} = I_{2,2} - \hat{I}_{2,2} = 105 - 107 = -2$. The prediction error was used effectively to embed one secret bit s, i.e., $E'_{2,2} = 2 \times E_{2,2} + s = -2 \times 2 + 1 = -3$. Finally, the prediction error having the secret data was increased by the prediction value to obtain the stego pixel, i.e., $I'_{2,2} = E'_{2,2} + \hat{I}_{2,2} = -3 + 107 = 104$.

3. **Proposed Method.**

In our data embedding procedure, the maximum modification level of cover pixels is 1. The modification may cause overflow and underflow problems in the extreme pixels, i.e., 0 or 255. In order to avoid the overflow and underflow problems, the extreme pixels are shrunk before the data embedding procedure, with the shrinking strategy being inspired by Lee et al.'s method [8]. Our method only records the LSBs of the extreme pixels, the values of which are {0, 1, 254, 255}, i.e.,

$$LSB = I_{x,y} \text{mod} 2, \text{if} I_{x,y} \in \{0, 1, 254, 255\}$$

Note that our method does not record the LSB of the non-extreme pixels. The recorded LSBs also are embedded into the cover image with the secret data. The recorded LSBs hidden in the image will be extracted as auxiliary information when the image is recovered. Afterwards, the extreme pixels are shrunk, i.e.,

$$\tilde{I}_{x,y} = \begin{cases} 1, & \text{if } I_{x,y} \leq 1, \\ 254, & \text{if } I_{x,y} \geq 254, \\ I_{x,y}, & \text{otherwise.} \end{cases}$$

The above equation limits the range of pixels within [1, 254] to avoid the overflow and underflow problems.

Fig. 3 shows the flowchart of data hiding of the proposed method. First, the cover image is divided into an embeddable region and a non-embeddable region, as shown in Fig. 4, where H and W denote the height and width of the cover image, respectively. The pixels in the non-embeddable region are used as the reference information in the data extraction and image recovery phases, thus they remain unchanged.

After dividing the image, the prediction value $\hat{I}_{x,y}$ of the embeddable pixel $\tilde{I}_{x,y}$ is generated by

$$\hat{I}_{x,y} = \begin{cases} min(\tilde{I}_{x-1,y}, \tilde{I}_{x,y-1}) - w \times min(\tilde{I}_{x-2,y-1}, \tilde{I}_{x-2,y}, \tilde{I}_{x-1,y-2}, \tilde{I}_{x,y-2}), & \text{if } \tilde{I}_{x-1,y-1} \geq max(\tilde{I}_{x-1,y}, \tilde{I}_{x,y-1}), \\ max(\tilde{I}_{x-1,y}, \tilde{I}_{x,y-1}) + w \times max(\tilde{I}_{x-2,y-1}, \tilde{I}_{x-2,y}, \tilde{I}_{x-1,y-2}, \tilde{I}_{x,y-2}), & \text{if } \tilde{I}_{x-1,y-1} \leq min(\tilde{I}_{x-1,y}, \tilde{I}_{x,y-1}), \\ \tilde{I}_{x-1,y} + \tilde{I}_{x,y-1} - \tilde{I}_{x-1,y-1}, & \text{otherwise.} \end{cases}$$

where w is the weight of the energy deviation. There is a high probability that the current pixel is smaller than the three neighboring pixels when $\tilde{I}_{x-1,y-1} \geq max(\tilde{I}_{x-1,y}, \tilde{I}_{x,y-1})$, and the weight of the minimum neighboring value should be increased. The appropriate weight w can increase the accuracy of the prediction.

Afterwards, the prediction error $E_{x,y}$ between $\tilde{I}_{x,y}$ and $\hat{I}_{x,y}$ is calculated, i.e., $E_{x,y} = \tilde{I}_{x,y} - \hat{I}_{x,y}$. These prediction errors are compiled to generate a histogram, where the value of the selected peak point P must be smaller than that of the selected zero point Z, i.e., $P < Z$. If the number of the embedded bits is more than that of the frequency of the peak point, the above method is repeated. However, the value of the next peak point must be greater the next zero point, i.e., $P > Z$. This rule can reduce the number of repeated modifications of the pixel, thereby maintaining good image quality. After determining the peak point and the zero point, and the hiding space, the secret bits and the recorded LSBs can be embedded into the prediction error, i.e.,

$$E'_{x,y} = \begin{cases} E_{x,y} + 1, & \text{if } P \leq E_{x,y} \leq Z \text{and} P < Z, \\ E_{x,y} + s, & \text{if } E_{x,y} = P \text{and} P < Z, \\ E_{x,y} - 1, & \text{if } Z \leq E_{x,y} \leq P \text{and} P > Z, \\ E_{x,y} - s, & \text{if } E_{x,y} = P \text{and} P > Z. \end{cases}$$

Finally, the embedded prediction error is increased by the prediction value to obtain the stego pixel, i.e., $I'_{x,y} = \hat{I}_{x,y} + E'_{x,y}$.

We used an example to illustrate the above procedure. Fig. 5 shows a cover image. The cover pixels in the two top-most columns and the left-most rows remain unchanged because they are used as the reference pixels in the extraction and recovery phases. Then, the prediction values of the other pixels are generated. For example, the prediction value of the pixel in the coordinates $\{3, 3\}$ is derived by $\hat{I}_{3,3} = \tilde{I}_{2,3} + \tilde{I}_{3,2} - \tilde{I}_{2,2} = 181 + 185 - 184 = 182$. Then, the prediction value of the next pixel is generated by $\hat{I}_{3,4} = \tilde{I}_{2,4} + \tilde{I}_{3,3} - \tilde{I}_{2,3} = 179 + 182 - 181 = 180$.

After deriving the prediction value, the prediction errors between the cover pixels and their corresponding prediction values are calculated. Fig. 5 shows the prediction errors. The frequencies with which these prediction errors occur are counted to generate a histogram, where the peak point is 0 and the zero point is 2. Consequently, the prediction errors between 0 and 2 are increased by 1 to create an embedding space. For example, the prediction error with the coordinate $\{4, 5\}$ is 1, and it is increased by 1, i.e., $E'_{4,5} = E_{4,5} + 1 = 1 + 1 = 2$.

Moreover, the prediction errors that have values equal to the peak point "0" can be used to embed secret data. Assume that three secret bits are $(101)_2$. The first embeddable prediction error in the coordinates $\{3, 3\}$ is equal to 0, hence it can be used to embed the first secret bit "1," i.e., $E'_{3,3} = E_{3,3} + s_1 = 0 + 1 = 1$. Then, the second secret bit "0" is embedded into the second embeddable prediction error, i.e., $E'_{3,4} = E_{3,4} + s_2 = 0 + 0 = 0$. The procedure of embedding the third secret bit is the same as the above procedure, and the prediction error are increased by the prediction value to obtain the stego pixel.

We used the following procedure to extract secret data and recover the cover image. The stego pixels in the two top-most columns and the left-most rows are used as the reference information of data extraction and image recovery. Then, the prediction values of the other pixels are generated. Then, the prediction error between the stego pixel and

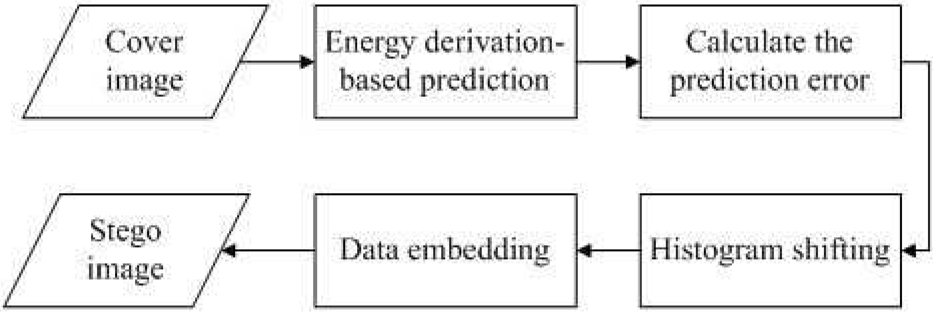

Cover image
Energy derivation-based prediction
Calculate the prediction error
Histogram shifting
Data embedding
Stego image

FIGURE 3. Flowchart of data embedding in the proposed method

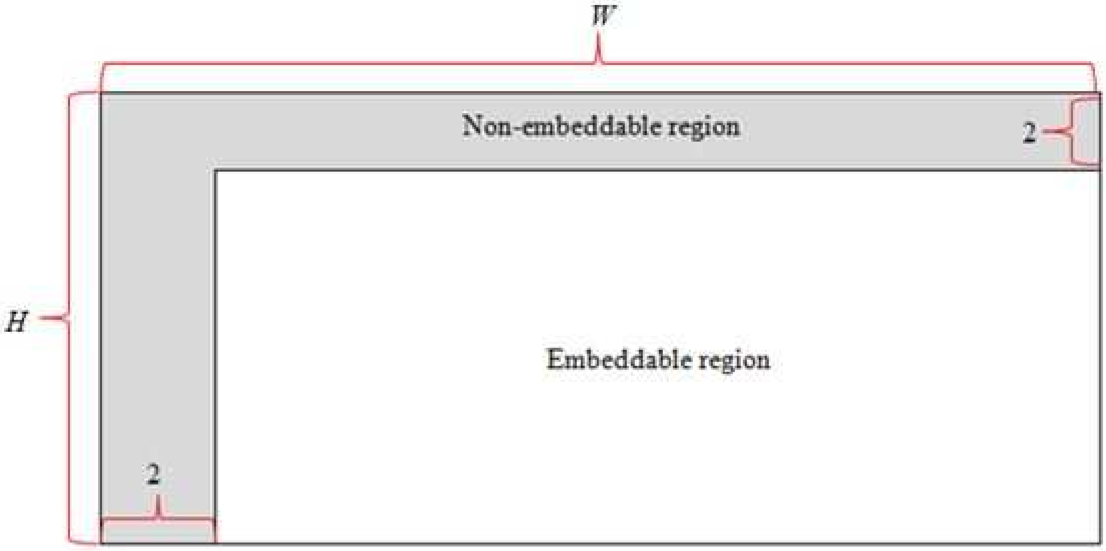


FIGURE 4. Cover image divided into the embeddable region and the non-embeddable region

its prediction value is calculated, i.e., $E'_{x,y} = I'_{x,y} - \hat{I}_{x,y}$. If the prediction error does not belongs the range between the peak point and the zero point, there is no secret bit or the recorded LSB in the prediction error. In addition, the stego pixel is the same as the cover pixel. Otherwise, the embedded bits can be extracted correctly by

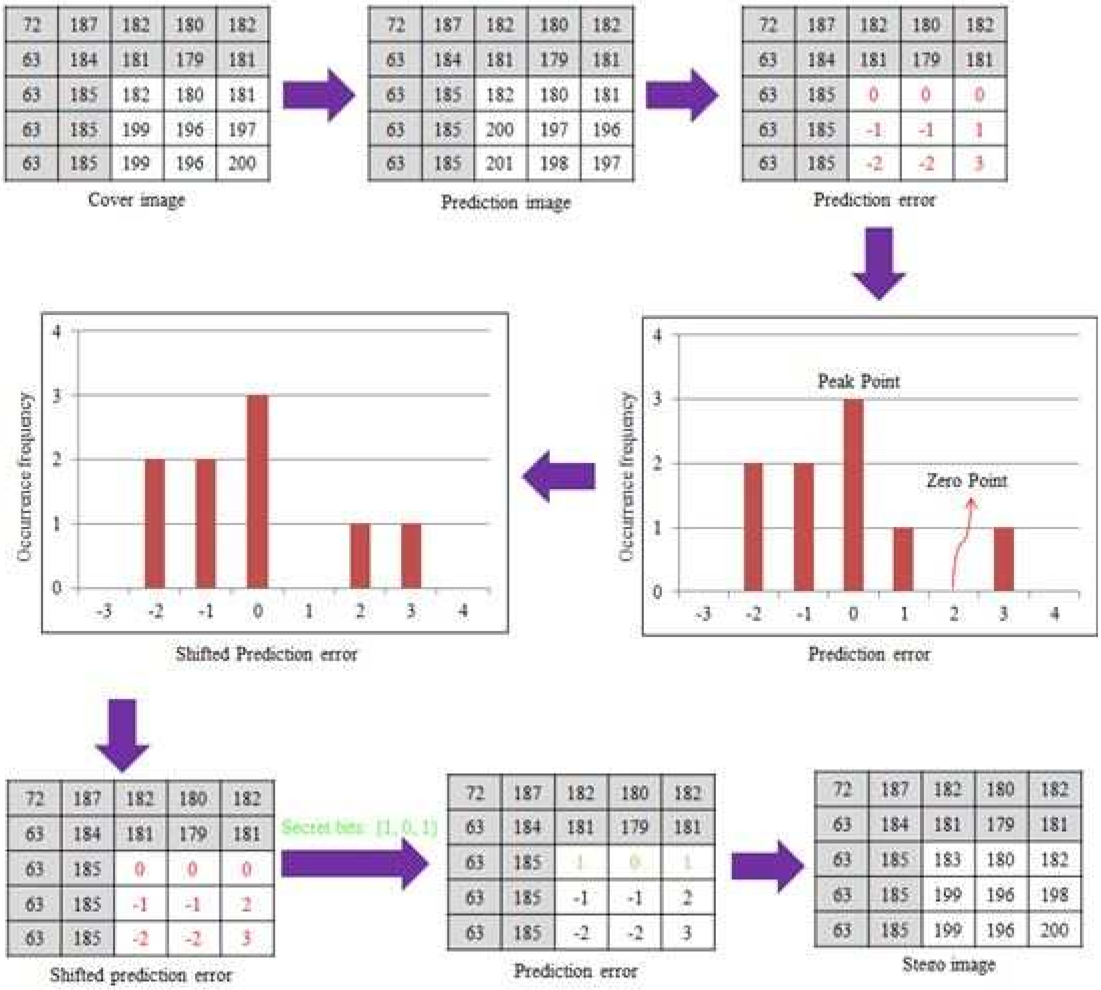
$$s = \begin{cases} E'_{x,y}, & \text{if } E'_{x,y} \in \{P, P+1\} \text{and} P > Z, \\ |E'_{x,y}|, & \text{if } E'_{x,y} \in \{P, P-1\} \text{and} P < Z. \end{cases}$$

Meanwhile, the original prediction error can be recovered losslessly by

$$\tilde{E}_{x,y} = \begin{cases} I'_{x,y} - 1, & \text{if } P < E'_{x,y} \leq Z \text{and} P < Z, \\ I'_{x,y} + 1, & \text{if } Z \leq E'_{x,y} < P \text{and} P > Z. \end{cases}$$

The above procedure is repeated until all of the embedded bits are extracted.

Fig. 6 shows the example of data extraction and image recovery. The stego pixels in the two top-most columns and the left-most rows are reference pixels, so they does not need to any modification procedure. Then, the prediction value of the stego pixel in the coordinates {3, 3} is generated, i.e., $\hat{I}_{3,3} = I'_{2,3} + I'_{3,2} - I'_{2,2} = 181 + 185 - 184 = 182$. The prediction error between the stego pixel ”183” and the prediction value ”182” is ”1”, which is equal to $P + 1$. Consequently, there is one secret bit in the stego pixel. The secret bit can be extracted by $s = E'_{3,3} = 1$. In addition, the pixel is recovered losslessly



72	187	182	180	182
63	184	181	179	181
63	185	182	180	181
63	185	199	196	197
63	185	199	196	200

Cover image

72	187	182	180	182
63	184	181	179	181
63	185	182	180	181
63	185	200	197	196
63	185	201	198	197

Prediction image

72	187	182	180	182
63	184	181	179	181
63	185	0	0	0
63	185	-1	-1	1
63	185	-2	-2	3

Prediction error

72	187	182	180	182
63	184	181	179	181
63	185	0	0	0
63	185	-1	-1	2
63	185	-2	-2	3

Shifted prediction error

Secret bits: [1, 0, 1]

72	187	182	180	182
63	184	181	179	181
63	185	1	0	1
63	185	-1	-1	2
63	185	-2	-2	3

Prediction error

72	187	182	180	182
63	184	181	179	181
63	185	183	180	182
63	185	199	196	198
63	185	199	196	200

Stego image

FIGURE 5. Example of embedding data using the proposed method

by $\tilde{I}_{3,3} = I'_{3,3} - 1 = 183 - 1 = 182$. The recovered pixel is used as the reference information of the next stego pixel.

The stego pixel in the coordinates {3, 4} is 180 and its prediction value is 180, thus the prediction error is 0. The prediction error "0" is equivalent to the peak point, so the secret bit hidden in the prediction error is extracted by $s = E'_{3,4} = 0$. Moreover, the pixel is recovered losslessly by $\tilde{I}_{3,4} = I'_{3,4} - s = 180$. The other secret data can be extracted by the same procedure mentioned above, and the original image can be derived without any distortion.

After the above procedure, the proposed method used the recorded LSBs to losslessly recover the extreme pixels. The extreme pixels are equal to 1 or 254, and their LSBs are replaced by the recorded LSBs to recover the original pixels, i.e.,

$$I_{x,y} = \begin{cases} 2 \times \left\lfloor \frac{\tilde{I}_{x,y}}{2} \right\rfloor + LSB, & \text{if } \tilde{I}_{x,y} \in \{1, 254\}, \\ \tilde{I}_{x,y}, & \text{otherwise.} \end{cases}$$

4. **Experimental Results.** We used two measurements to confirm the effectiveness of the proposed method, i.e., the embedding rate (ER) and the PSNR value, where former was calculated by

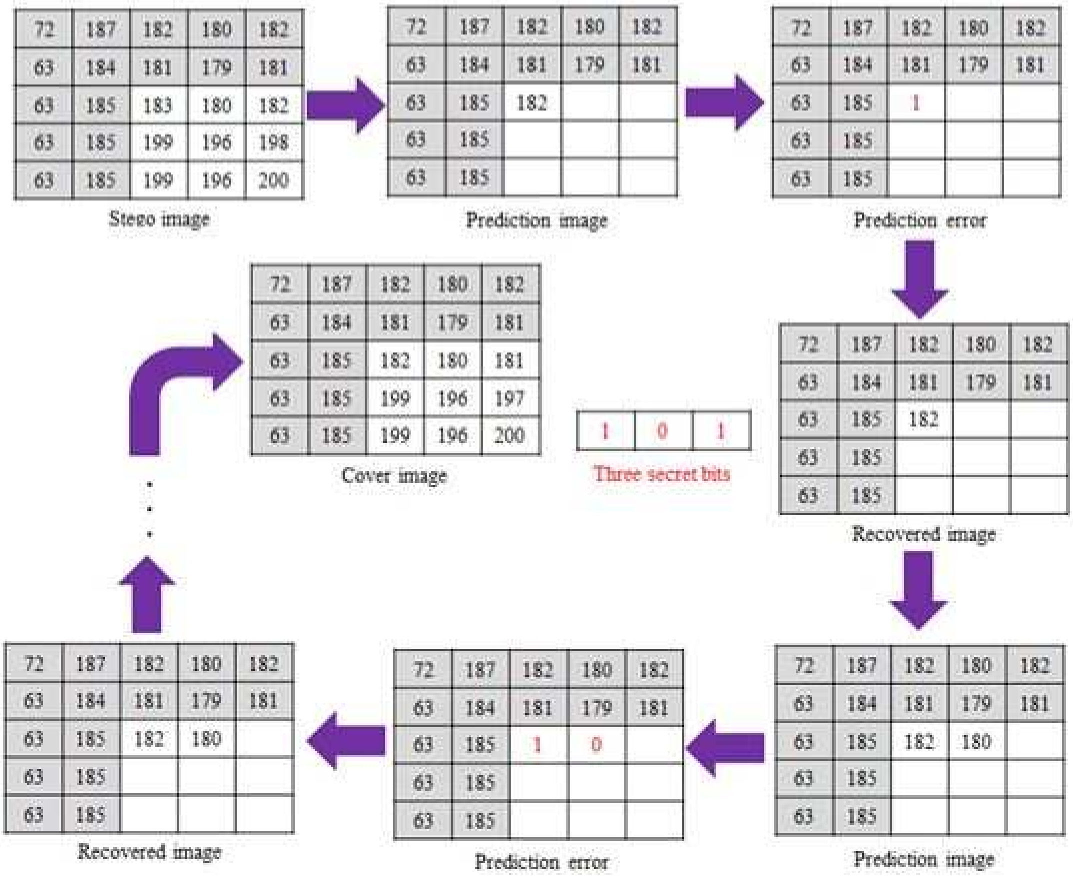


Figure 6. Example of data extraction and image recovery using the proposed method

$$\mathrm{ER} = \frac{\text{Number of secret bits}}{\text{Size of cover image}}(\mathrm{bpp})$$

Obviously, the greater ER value confirms that the method can embed more secret bits into the cover image. The PSNR value was calculated to measure the visual similarity between the cover image and the stego image, i.e.,

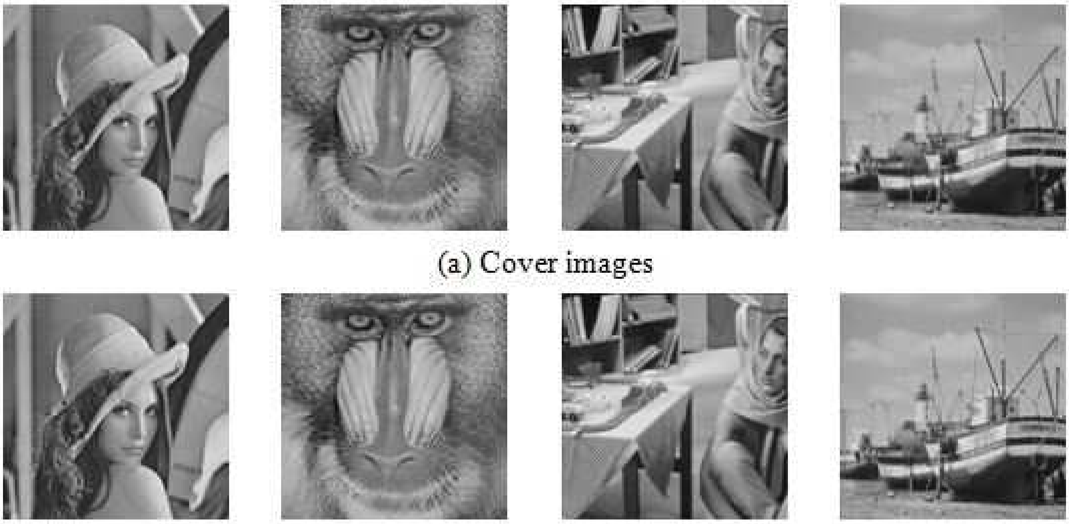
$$\mathrm{PSNR} = 10 \times \log_{10}\frac{255^2}{\mathrm{MSE}}(\mathrm{dB}),$$

where MSE is the mean square error between the two images. The MSE value can be calculated by

$$\mathrm{MSE} = \frac{1}{H \times W}\sum_{x=1}^{H}\sum_{y=1}^{W}(I_{x,y} - I'_{x,y})^2$$

Fig. 7 (a) shows four test images, i.e., Lena, Baboon, Barbara, and Boats, each of which consists of 512 512 pixels. Fig. 7 (b) shows four stego images generated by the proposed method. There is a high similarity between the cover image and the stego image, so a hacker cannot easily detect that there are some secret bits in the stego image.

Since the weight setting influences both the embedding rate and the PSNR value, we determined the appropriate weight during the first experiment. Fig. 8 shows the embedding rates with various weights, where the PSNR values of the images are equal to 35 dB. When the PSNR value is equal to or greater than 35 dB, the hackers cannot



(a) Cover images

(b) Stego images

Figure 7. Test images

efficiently detect the visual difference between the cover image and the stego image. In addition, the appropriate weight can be selected from the maximum embedding rate. In other words, for the same PSNR value, the appropriate weight can embed more secret bits than other weights. Note that the PSNR values of all stego images are smaller than 35 dB when the weight is set to 0.301. Consequently, we set the range of weights from 0 to 0.300. Fig. 8 shows that the embedding rate of the smooth image ”Lena” is higher than that of the complex image ”Baboon,” implying that the proposed method is more suitable for smooth images.

Fig. 9 shows that the proposed method has a greater PSNR value than the related methods [4, 6, 7, 8, 9]. This is because the proposed method uses the energy deviation to reduce the prediction error between the cover pixel and its prediction value, thereby decreasing its expansion level. Consequently, the proposed method can achieve a greater PSNR value than other methods for the same embedding rate.

The experimental results showed that the PSNR values of the proposed method were significantly greater than those of Qin et al.'s method for the complex images, e.g., Baboon. This is because Qin et al.'s method only used the average value of adjacent pixels as the prediction value. When there are many textures in the cover image, e.g., Baboon, the prediction accuracy of Qin et al.'s method is decreased. However, the proposed method analyzed the correlation of adjacent pixels to derive the exact prediction value, so its prediction error is smaller than that of Qin et al.'s method. The advantage decreases the modification frequency of the prediction error to reduce the distortion of the image, thereby achieving a higher PSNR value.

5. **Conclusions.**

In this paper, we proposed an energy deviation strategy to decrease the prediction error between the cover pixel and the prediction value. This advantage can effectively reduce the level of prediction error expansion, thereby controlling the level of modification of pixels. Consequently, the distortion of the stego image of the proposed method is less than that of the previously prediction-based hiding methods. In the future, we will analyze the relationship between the smooth block and the complex block to design a dynamic energy deviation method to further enhance the prediction accuracy.

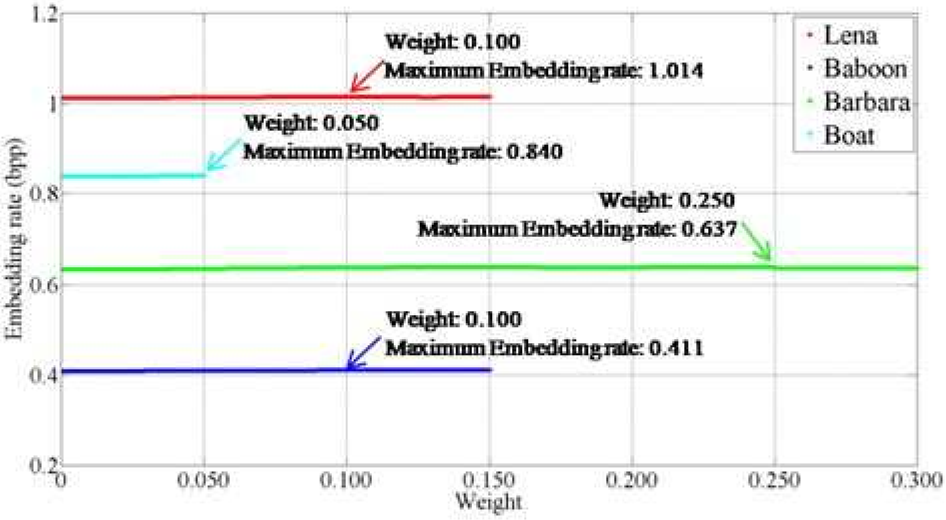
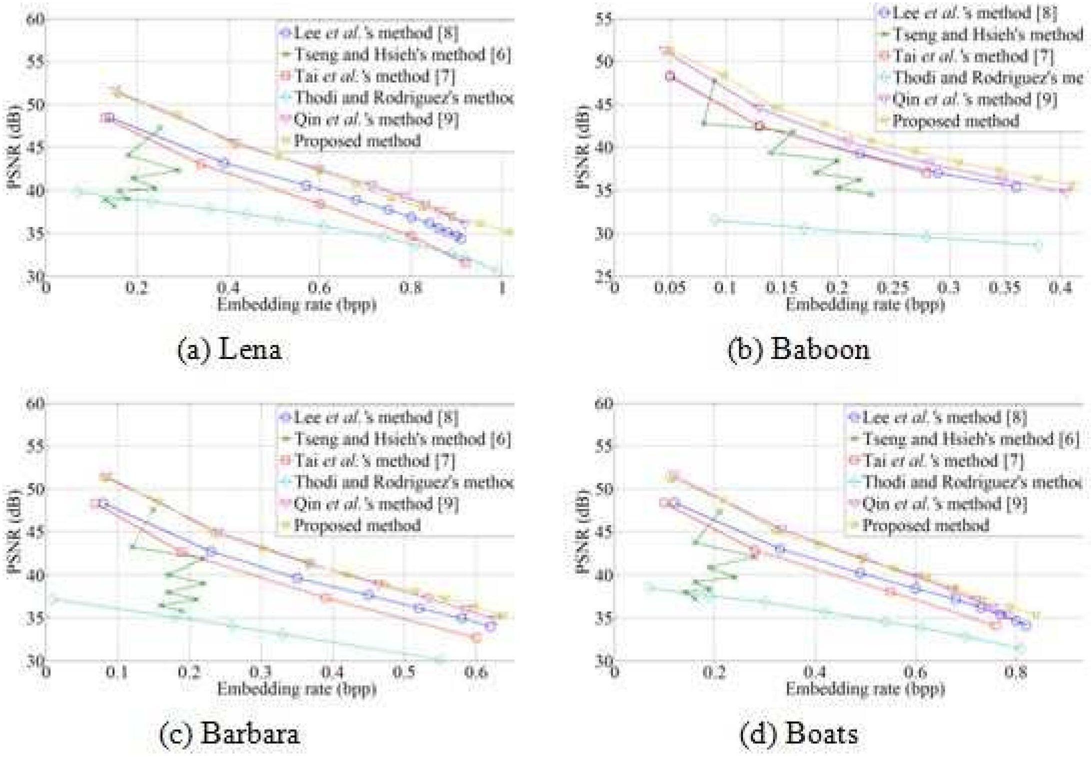


Figure 8. Experimental results of finding an appropriate weight



(a) Lena

(b) Baboon

(c) Barbara

(d) Boats

Figure 9. Comparison among the proposed method and the five related methods in terms of embedding rate and PSNR value

REFERENCES

[1] J. Tian, Reversible data embedding using a difference expansion, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, 2003.

[2] D. M. Thodi, and J. J. Rodriguez, Prediction-error based reversible watermarking, *Proc. of International Conference on Image Processing*, vol. 3, pp. 1549-1552, 2004.

[3] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, Reversible data hiding, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, 2006.

[4] D. M. Thodi, and J. J. Rodriguez, Expansion embedding techniques for reversible watermarking, *IEEE Trans. Image Processing*, vol. 16, no. 3, pp. 721-730, 2007.

[5] M. Fallahpour, Reversible image data hiding based on gradient adjusted prediction, *IEICE Electronics Express*, vol. 5, no. 20, pp. 870-876, 2008.

[6] H. W. Tseng, and C. P. Hsieh, Prediction-based reversible data hiding, *Information Sciences*, vol. 179, no. 14, pp. 2460-2469, 2009.

[7] W. L. Tai, C. M. Yeh, and C. C. Chang, Reversible data hiding based on histogram modification of pixel differences, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906-910, 2009.

[8] C. F. Lee, H. L. Chen, and H. K. Tso, Embedding capacity raising in reversible data hiding based on prediction of difference expansion, *Journal of Systems and Software*, vol. 83, no. 10, pp. 1864-1872, 2010.

[9] C. Qin, C. C. Chang, and L. T. Liao, An adaptive prediction-error expansion oriented reversible information hiding scheme, *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2166-2172, 2012.

[10] S. Weng, Y . Zhao, J. S. Pan, and R. Ni, Reversible watermarking based on invariability and adjustment on pixel pairs, *IEEE Signal Processing Letters*, vol. 15, pp. 721-724, 2008.

[11] S. Weng, Y. J. Liu, J. S. Pan, and N. Cai, Reversible data hiding based on flexible block-partition and adaptive block-modification strategy, *Journal of Visual Communication and Image Representation*, vol. 41, pp. 185-199, 2016.

[12] B Ou, X. Li, Y. Zhao, R. Ni, and Y. Q. Shi, Pairwise prediction-error expansion for efficient reversible data hiding, *IEEE Trans. Image Processing*, vol. 22, no. 12, pp. 5010-5021, 2013.