International Journal of Security and Its Applications
Vol.8, No.6 (2014), pp.173-184
http://dx.doi.org/10.14257/ijsia.2014.8.6.16

A Difference Expansion Based Reversible Data Hiding Algorithm Using Edge-Oriented Prediction

Wen-Chao Yang[1], Ling-Hwei Chen[1] and Chang-Hsing Lee[2]

[1]*Department of Computer Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu, Taiwan, 300, R.O.C.*
[2]*Department of Computer Science and Information Engineering, Chung Hua University, 707, Sec.2, WuFu Rd., Hsinchu, Taiwan 30012, R.O.C.*
wchy@debut.cis.nctu.edu.tw lhchen@cc.nctu.edu.tw chlee@chu.edu.tw

Abstract

In this paper, a difference expansion based reversible data hiding approach which achieves the low distortion requirement under the same embedding capacity is proposed. An edge-oriented prediction method and a modified overflow/underflow prevention method are proposed to achieve the requirement. For each pixel, the prediction method uses edge information to get a better predicted value in an attempt to reduce the image distortion. Experimental results have demonstrated that the proposed reversible data hiding method yields lower distortion than several well-known expansion-based reversible data hiding methods.

Keywords: *Reversible data hiding, Difference expansion*

1. Introduction

Data hiding techniques have been widely studied in security community. According to the applications, data hiding techniques can be divided into two categories: watermarking and steganography. Watermarking is usually applied to copyright protection whereas steganography is exploited for privacy communication [1, 2]. In steganography, a secret message is embedded in a carrier, called cover media, and results in a stego media. A variety of digital media has been used to hide privacy massage [3-15]. Chou and Ramchandrad imperceptibly embedded data in a digital audio recording [3]. Image is the most popular cover media in steganography and the simplest method for image steganography is the least significant bit (LSB) technique [4-5] in which secret message is embedded by modifying the LSBs of some image pixels. However, LSB modification is vulnerable to lossy image compression. To resist lossy/JPEG image compression, Lee and Chen embedded secret message in objects of images [6] or in the quantized DCT coefficients [7]. Note that all JPEG compression based methods are vulnerable to lossy image re-compression or format conversion [2]. Although these steganographic methods only distort the cover image slightly, the original cover image cannot be perfectly recovered anymore.

Reversible data hiding can be used to exactly extract the embedded message and exactly recover the original cover image without any distortion. This property is important for some applications where the cover image should not be distorted, including medical images, military images, or forensic images (provided by law) [8].

In the past decade, several reversible data hiding methods [8-15] have been proposed to meet the above mentioned requirements. Ni, *et al*., proposed a histogram-based reversible data hiding method [9]. First, the pixel values that occur the most frequently

ISSN: 1738-9976 IJSIA

(called peak) and the least frequently (called zero) are detected. In the embedding process, the pixels with their pixel values bounded by the peak and the zero will be modified according to the embedded data value. Tian proposed a difference expansion-based data hiding method in which the difference between a pair of pixels is computed [10]. Then, for each pair of pixels, one bit of data is embedded by expanding the difference value. The main drawback is that not every pair of pixels can be used for data hiding because the expanded pixel value may be larger/smaller than the maximum/minimum value. That is, only those pairs of pixels that do not incur any overflow or underflow errors can be expanded and used for secret data embedding. Therefore, a location map, which indicates whether a pair of pixels is embeddable or not, has to be constructed in the embedding process. The overhead for representing such a map is 0.5 bits per pixel. To reduce the overhead, Lee, *et al*., proposed a 2×2 block-based reversible steganographic method [11]. Without loss of generality, let g_1, g_2, g_3, and g_4 denote the pixel values of the pixels in a 2×2 block, sorted in a non-decreasing order. That is, $g_1 \le g_2 \le g_3 \le g_4$. The differences between g_3 and the other g_i (i = 1, 2, 4) are then expanded for embedding one or two bits of secret message according to the difference values. Thodi and Rodríguez used JPEG-LS prediction method to reduce the prediction error [12]. The experimental results have shown that their method can achieve a higher capacity and lower distortion. Hong proposed a new histogram shifting based method which embedded data in both positive and negative values of the prediction-error histogram, rather than in positive/negative value of the prediction-error histogram, to increase the embedding capacity [13]. Further, an error energy control approach was developed to maintain the image quality. The method exploits the image redundancy to get a better performance. Li, *et al*., summarized the previous histogram-shifting-based reversible data hiding methods to construct a general reversible data hiding framework [14]. They proposed two new algorithms for different embedding capacity requirements based on the general framework.

Sachnev, *et al*., [15] proposed a two-pass testing technique to reduce the amount of overhead required for the expansion method based on average prediction. In each pass testing, the maximum (or minimum) value of the data bit is embedded to determine whether the embedded pixel value will produce any overflow/underflow error or not. Those pixels which fail to pass the first testing operation are referred to as non-embeddable pixels (NEPs). Those pixels which pass the first testing operation but fail to pass the second one will be treated as one-time embeddable pixels (OTEPs). The pixels which do not incur any overflow/underflow errors in both testing will be called two-time embeddable pixels (TTEPs). In fact, TTEPs can be easily detected in the extraction process by performing one more testing operation on the embedded pixels. Note that a location map is needed to distinguish NEPs from OTEPs. Since a large proportion of the pixels are TTEPs for most natural images, the overhead (*i.e.*, the location map) is reduced greatly. However, the method suffers from a problem for some OTEPs. Specifically, in the embedding process, if the prediction error d = 0 and the secret data bit b = 0, the embedded pixel can pass the testing operation during the extraction process. That is, they will be viewed as TTEPs in the extraction process. As a result, the location map constructed in the embedding process will not be consistent with that detected in the extraction process. The main reason for causing such inconsistency is that the first testing operation embeds the maximum/minimum value of the data bit, which is different from the actual data value in the embedding process. To solve the problem, a modified overflow/underflow prevention method will be proposed in this paper. Further, an edge-oriented prediction method will be designed to achieve

the low distortion requirement. The rest of the paper is organized as follows. Section 2 describes the proposed edge-oriented prediction-based reversible data hiding method. Comparison of experimental results in terms of embedding capacity and image distortion will be presented in Section 3. Finally, conclusions are given in Section 4.

2. Proposed Edge-oriented Prediction-based Reversible Data Hiding Method (EPRDD)

In EPRDD, an edge-oriented predictor is designed to get a smaller prediction error for each processing pixel. As a result, the embedded error can be reduced and thus a better image quality can be obtained. First, we will describe the embedding and extraction procedures. Secondly, the approach designed to preventing from overflow/underflow errors will be given.

2.1. Embedding Procedure

Figure 1 shows the block diagram of the embedding procedure. For each embedding pixel, the variance of its four neighboring pixels is computed first. Secondly, the image pixels are sorted in a non-decreasing order of the computed variances. The secret data will then be embedded pixel by pixel according to the sorted sequence. For each embedding pixel, its prediction value is computed using its neighboring reference pixels. Then, the difference between the pixel value and prediction value, called prediction error, can be obtained. Finally, the secret data bit will be embedded into the image pixels sequentially by expanding the prediction error accordingly.

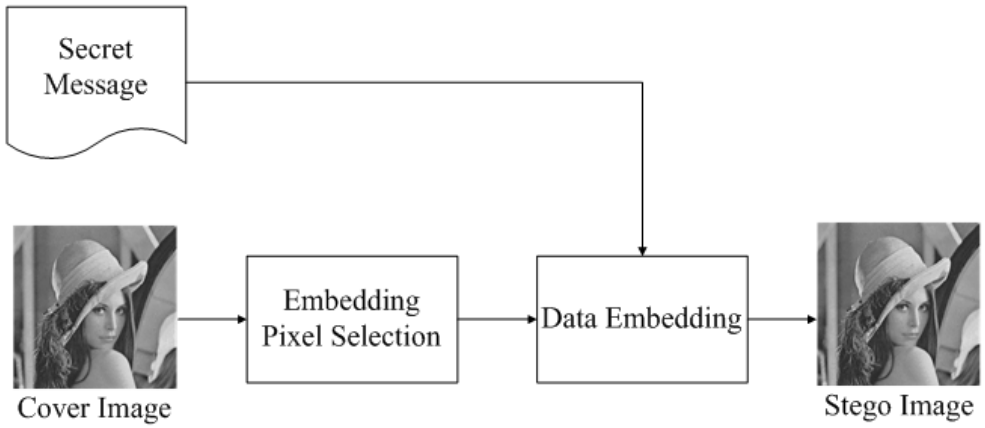


Figure 1. The Block Diagram of the Embedding Procedure

2.1.1. Embedding Pixel Selection: In this paper, the pixels in the cover image are divided into two groups: "Dot" group and "Cross" group (please see Figure 2), which is a two-stage embedding mechanism proposed by Sachnev, *et al*., [15]. In the first stage, only "Dot" group is processed. After "Dot" group is processed, we will embed secret bits in those "Cross" group in the second stage. For each processed pixel, it will be treated as the embedding pixel and the other pixels are treated as the reference pixels. In each stage, all pixels belonging to the same group are sorted in a non-decreasing order of the variance computed from the pixel values of its neighboring four reference pixels. Secret data bit is then sequentially embedded by expanding the prediction error

between the embedding pixel value and its prediction value computed from the reference pixels.

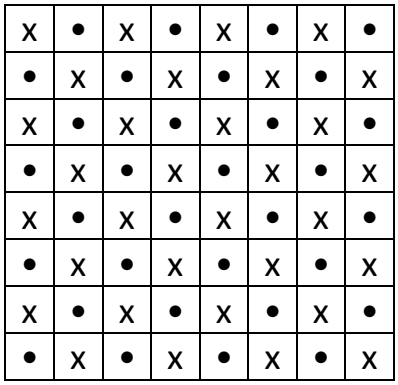


Figure 2. Division of Image Pixels into "Dot" Group (Marked with "•") and "Cross" Group (Marked with "x")

2.1.2. Secret Data Embedding: The secret data bits will be embedded into the sorted embedding pixels by expanding their corresponding prediction errors. The expanded prediction errors must be different for any two distinct secret data bit such that the embedded secret data can be correctly extracted. Assuming that a secret data bit b (b = 0 or 1) will be embedded in pixel p_3 with its four neighboring pixels p_1, p_2, p_4, and p_5 (please see Figure 3). Let g_i denote the pixel value of p_i (i = 1, 2, …, 5). First, the edge gradient values, notated by e_0, e_{45}, e_{90}, and e_{135}, corresponding to four different edge directions (0°, 45°, 90°, and 135°) are estimated from the neighboring pixels:

$$e_0 = |g_2-g_4|, \quad (1)$$

$$e_{45} = (|g_1-g_2|+|g_4-g_5|)/2, \quad (2)$$

$$e_{90} = |g_1-g_5|, \quad (3)$$

$$e_{135} = (|g_1-g_4|+|g_2-g_5|)/2. \quad (4)$$

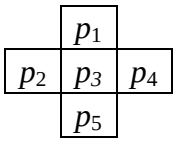


Figure 3. The Embedding Pixel and its Four Neighboring Pixels

The estimated edge mode, *EM*, will then be determined from the edge direction having the smallest estimated edge gradient value:

$$EM = \arg \min_{k \in \{0,45,90,135\}} e_k. \quad (5)$$

For different edge directions, the edge prediction value $\hat{p}_3$ for the processing pixel p_3 is given as follows:

$$\hat{p}_3 = \begin{cases} floor((g_2 + g_4)/2), & \text{if } EM = 0, \\ floor((g_1 + g_2 + g_4 + g_5)/4), & \text{if } EM = 45, \\ floor((g_1 + g_5)/2), & \text{if } EM = 90, \\ floor((g_1 + g_2 + g_4 + g_5)/4), & \text{if } EM = 135, \end{cases} \quad (6)$$

where the function $floor(x)$ returns the largest integer smaller than x.

Note that not every processing pixel has obvious edges. As a result, the proposed edge-oriented predictor may not work well for those pixels in smoothing areas. To deal with this problem, the prediction value of p_3, $\hat{g}_3$, is given as follows:

$$\hat{g}_3 = \begin{cases} \hat{p}_3, & \text{if } |e_{EM} - e_{EM_2}| \geq T_e, \\ floor((g_1 + g_2 + g_4 + g_5)/4), & \text{otherwise,} \end{cases} \quad (7)$$

where EM_2 is defined as follows:

$$EM_2 = \arg \min_{k \in \{0,45,90,135\} - \{EM\}} e_k. \quad (8)$$

Note that e_{EM} and e_{EM_2} respectively denote the smallest and the second smallest gradient values of all edge directions, T_e is a threshold used to control the edge strength of the proposed edge-oriented predictor. Then, the prediction error e_3 between $\hat{g}_3$ and the original pixel value g_3 is given as follows:

$$e_3 = g_3 - \hat{g}_3. \quad (9)$$

To embed a data bit b (b = 0 or 1), the prediction error e_3 is expanded using the equation proposed by Sachnev *et al.*'s [15]:

$$ee_3 = \begin{cases} e_3 + T_p + 1, & \text{if } e_3 > T_p, \\ 2 \times e_3 + b, & \text{if } T_n \leq e_3 \leq T_p, \\ e_3 + T_n, & \text{if } e_3 < T_n, \end{cases} \quad (10)$$

where T_p and T_n are positive and negative thresholds that control the range of the expanded prediction errors. Note that the data bit b can be embedded in the embedding pixel only if $T_n \leq e_3 \leq T_p$. Therefore, if nothing is embedded in the embedding pixel, ee_3 will be larger than $2T_p + 1$ or small than $2T_n$ (*i.e.*, $ee_3 > 2 \times T_p + 1$ or $ee_3 < 2 \times T_n$). Finally, the embedded pixel value u_3 for p_3 is given as follows:

$$u_3 = \hat{g}_3 + ee_3. \quad (11)$$

In this paper, all parameters (T_n, T_p, T_e, $|M|$), where $|M|$ denotes the size of the original secret data, have to be transmitted to the receiver for correctly extracting the embedded data and recovering the original cover image. In this paper, we embed these parameters into the least significant bits (LSBs) of some pre-selected pixels. The original LSBs of these selected pixels will be collected to form a set, denoted by S_{LSB}. S_{LSB} together with the original message M, called payload, will be embedded in the cover image and have to be recovered in the extraction procedure.

2.2. Embedded Data Extraction and Original Image Recovering

The extraction procedure is a reverse order of the embedding procedure. The block diagram is illustrated in Figure 4.

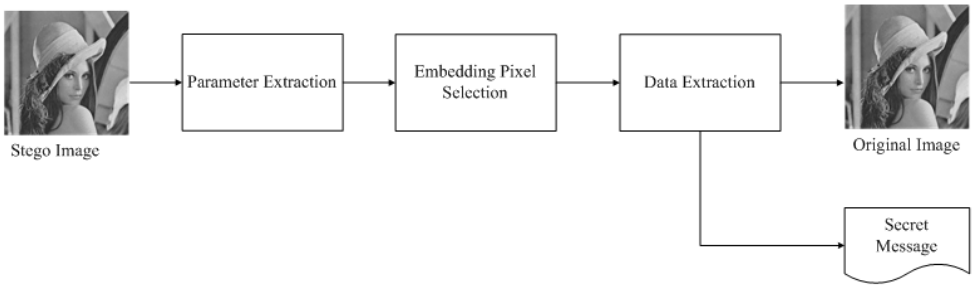


Figure 4. The Block Diagram of the Extraction Process

First, the embedding parameters (T_n , T_p, T_e, $|M|$) are extracted from the LSBs of those pre-selected pixels. Secondly, image pixels are sequentially selected according to the variance of its four neighboring pixels. Then, each data bit is extracted from the embedded pixel to get the payload, including S_{LSB} and M. The detailed procedure for extracting the data bit b is given as follows.

Step 1. Determine the estimated edge mode, EM, of the embedding pixel p_3 according to (1) - (8).

Step 2. Compute the prediction value $\hat{g}_3$ of p_3 and then compute the expanded prediction error $ee_3 = u_3 - \hat{g}_3$.

Step 3. If $ee_3 < 2 \times T_n$, set $e_3 = ee_3 - T_n$ and goto Step 6.

Step 4. If $ee_3 > 2 \times T_p + 1$, set $e_3 = ee_3 - T_p - 1$ and goto Step 6.

Step 5. Derive the embedded data bit b, $b = ee_3 \bmod 2$. Set $e_3 = (ee_3 - b) / 2$.

Step 6. Recover the original pixel value g_3 of p_3, $g_3 = \hat{g}_3 + e_3$.

After all embedded data has been extracted; the secret message M and S_{LSB} can be recovered accordingly. Meanwhile, the original cover image can be reconstructed by replacing the LSBs of the selected pixels with those in S_{LSB}.

2.3. Prevention of Overflow/Underflow Problem

In the embedding process, the modified pixel value u_3, obtained by adding the expanded prediction error ee_3 and the predicted value $\hat{g}_3$ together, may result in overflow/underflow error. That is, the modified pixel value may exceed its maximal value (255 for 8-bit gray level image) or may be smaller than its minimal value (0 for 8-bit gray level image). Most reversible data hiding methods tried to solve this problem by preventing from embedding secret data in those pixels that incur any overflow/underflow errors. Therefore, a location map is needed to indicate whether a pixel is embedded into some secret data or not. This location map will increase the payload for reversible data hiding systems.

Sachnev, *et al*., [15] proposed a two-pass testing operation to reduce the size of the location map. However, as described in Section 1, Sachnev's method will cause the location map inconsistency problem. As a result, the secret message cannot be correctly extracted. To avoid this problem, a modified overflow/underflow prevention method will be proposed.

First, we embed all data bits in the cover image as described previously. Given a pixel, if its modified pixel value u_3 is greater/smaller than 255/0, it will be viewed as NEP. Otherwise, we perform the overflow/underflow testing operation on the modified pixel value u_3 ($0 \le u_3 \le 255$) by embedding the data bit 1/0 to get the testing pixel value u_t. If u_t satisfies $0 \le u_t \le 255$, the pixel will be treated as TTEP. Otherwise, it will be regarded as OTEP. Figure 5 illustrates these scenarios. In this paper, only those OTEPs and TTEPs are used for secret data embedding.

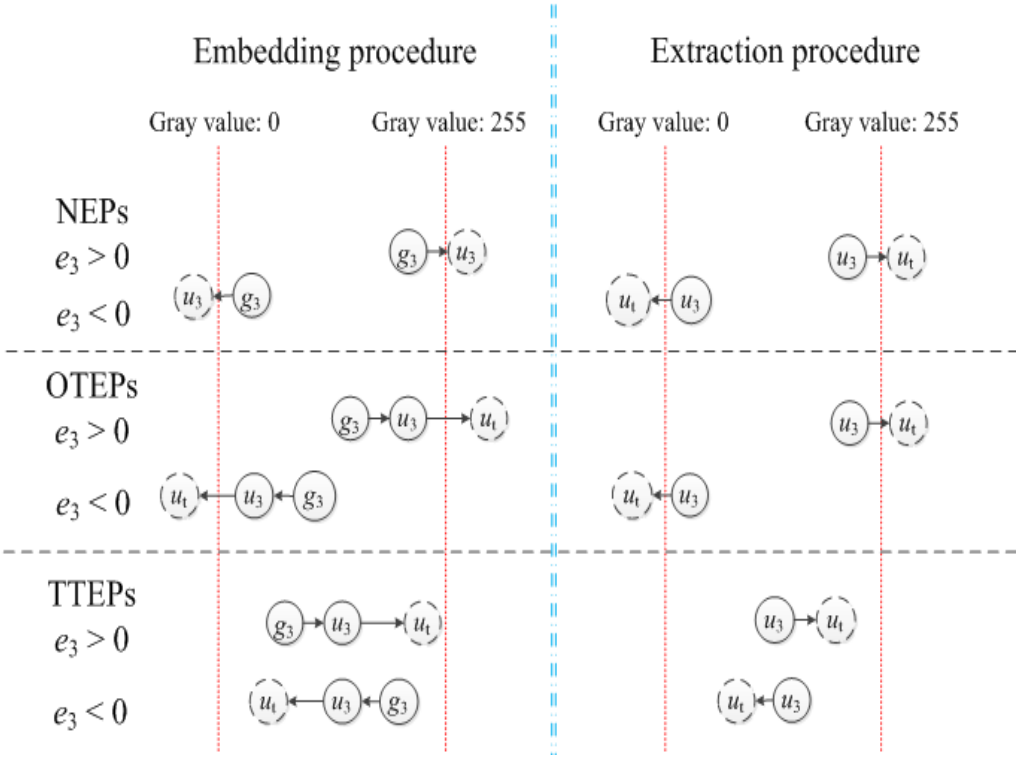


Figure 5. Classification of Image Pixels into NEPs, OTEPs, and TTEPs, where g_3 is the Gray Value of the Processing Pixel p_3, u_3 Denotes the Modified Gray Value of p_3 by Embedding the Secret Data, and u_t is the Obtained Testing Pixel Value by Performing the Overflow/Underflow Testing Operation on u_3

In the extraction process, given an embedded pixel value u_3, if we embed 1/0 on the modified pixel value u_3 of p_3 to get u_t satisfying $0 \le u_t \le 255$, the pixel p_3 will belong to TTEPs. That is, TTEPs can be identified in the extraction process without referring to any indication bit. On the other hand, the pixel may be NEPs or OTEPs, which cannot be distinguished in the extraction process. Therefore, an indication bit is needed to distinguish NEPs (indicated by bit "1") from OTEPs (indicated by bit "0"). The memory space for storing these indication bits is an overhead for the proposed reversible data hiding method.

3. Experimental Results

The embedding and extraction algorithms were developed using Matlab 7. The testing platform is Microsoft Windows 7, Intel Core Duo 1.66 GHz with 2 GB memory. In our experiments, typical 512×512 grayscale images (*i.e.*, Airplane, Lena, and Mandrill) downloaded from the SIPI database [16] and 1000 512×512 grayscale images downloaded from the BOSS Rank database [17] were served as the test images.

3.1. Comparison of Prediction Error for Different Threshold T_e

As described in Section 2, the threshold T_e is used to control the edge strength of the proposed edge-oriented predictor in (7). To assess the impact of different thresholds, we

evaluate the means and standard deviations of all prediction error for the 1000 test image for various thresholds T_e (please see Table 1). In Table 1, we can see that the best prediction error can be obtained when T_e = 10. Thus, we use T_e = 10 in the following experiments.

Table 1. Comparison of Prediction Error for Different Threshold T_e, the Number in Parentheses Denotes the Standard Deviation

T_e	Prediction error
0	3.8583(5.2198)
1	3.8279(5.1772)
2	3.7998(5.1277)
3	3.7833(5.0920)
4	3.7725(5.0643)
5	3.7648(5.0414)
6	3.7592(5.0216)
7	3.7557(5.0058)
8	3.7526(4.9909)
9	3.7512(4.9798)
10	**3.7498**(4.9723)
11	3.7542(4.8758)
12	3.7560(4.9756)
13	3.7557(4.9671)

3.2. Comparison of the Proposed Edge-Oriented Predictors with Different Predictors

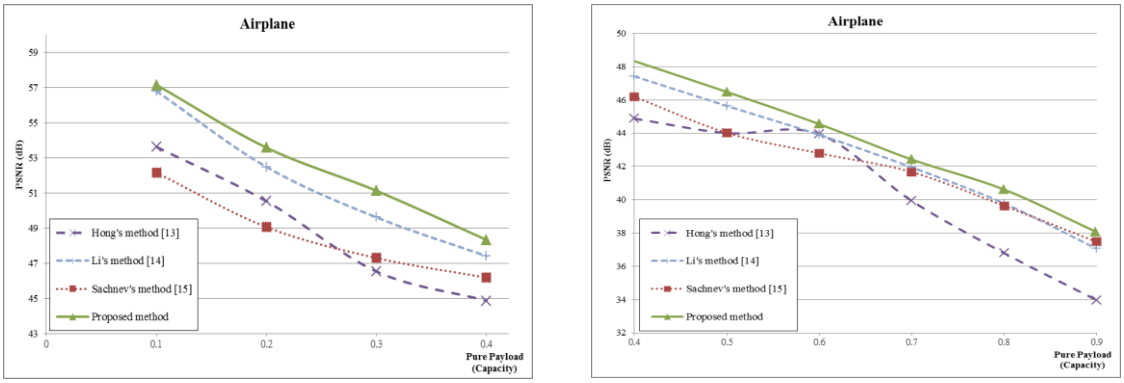
To show the effectiveness of the proposed edge-oriented prediction, we compare the image quality under variant payload for different predictors, including average predictor, Median Edge Detection (MED) predictor [12], Gradient Adjusted Predictor (GAP) [18], and the proposed edge-oriented predictor (EOP) in 1000 test images of the BOSS Rank database. Table 2 shows the experimental results. From Table 2, we can see that the proposed edge-oriented predictor outperforms MED, GAP, and average predictor.

Table 2. Comparison of the Image Quality in Terms of PSNR for Different Predictors, the Number in Parentheses Denotes the Standard Deviation

Payload \ Predictors	MED	GAP	Average Predictor	Proposed EOP
0.1	54.57(4.18)	53.39(4.54)	52.49(3.44)	**55.50**(4.40)
0.2	49.33(4.66)	48.84(4.84)	47.63(3.74)	**50.60**(4.66)
0.3	45.00(5.08)	44.81(5.17)	43.41(3.54)	**46.08**(5.42)
0.4	41.80(4.50)	41.72(4.51)	41.30(3.50)	**43.22**(4.87)
0.5	38.51(3.85)	38.52(3.82)	38.71(3.84)	**39.56**(4.12)
0.6	37.21(4.19)	37.31(4.18)	37.56(4.11)	**39.30**(4.31)
0.7	34.74(4.83)	34.93(4.81)	35.17(4.71)	**36.46**(5.05)
0.8	33.23(5.24)	33.54(5.26)	33.91(4.90)	**36.14**(5.04)

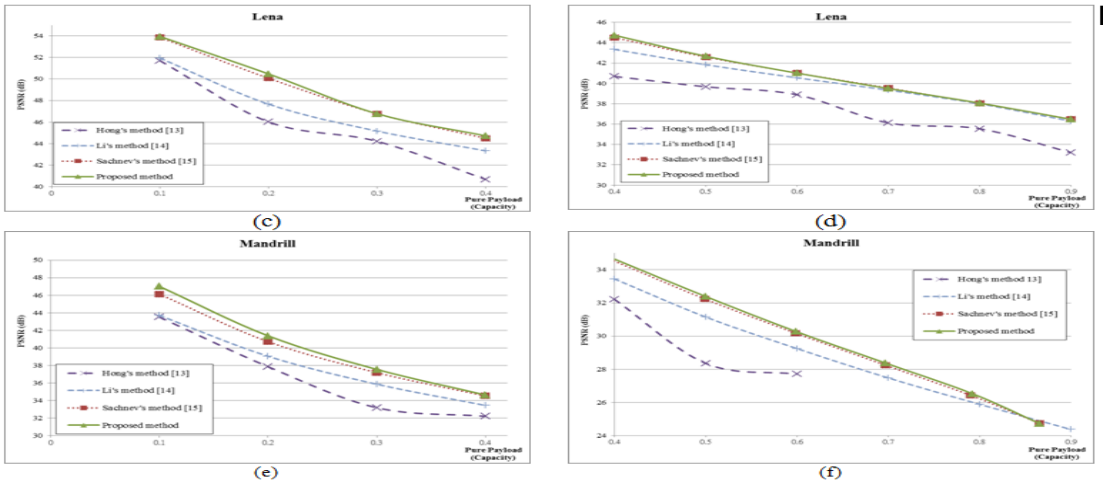
3.3. Image Quality and Embedding Capacity

Figure 6 compares the proposed method with Hong's method [13], Li's method [14] and Sachnev's method [15] in terms of image quality (PSNR) and capacity *C* (bit per pixel, bpp). As shown in Figure 6, we can see that our proposed method yield the best result in terms of image quality/capacity. However, Sachnev's method [15] is comparable to our proposed method for test images Lena and Mandrill. Therefore, we further compare the proposed method with Sachnev's method in terms of the image quality under the same capacity for 1000 test images in BOSS Rank database. Table 3 shows the means and standard deviations of the image quality of the proposed method and Sachnev's method. The result shows that the proposed method can maintain the better image quality under the same capacity in BOSS Rank database.



(a) (b)

Figure 6. Comparison of the Image Quality and Embedding Capacity for Three Test Images, the Left Column Compares the Results for Low Capacities (0.1-0.4 bpp), whereas, the Right Column Compares the Results for High Capacities (0.4-0.9 bpp)



(c) (d)

(e) (f)

Figure 6. Comparison of the Image Quality and Embedding Capacity for Three Test Images, the Left Column Compares the Results for Low Capacities (0.1-0.4 bpp), whereas, the Right Column Compares the Results for High Capacities (0.4-0.9 bpp), (Continued)

Table 3. Comparison of Average Image Quality (in PSNR) for 1000 Test Images in BOSS Rank Database, the Number in Parentheses Denotes the Standard Deviation

Methods	Capacity (bpp)			
	0.2 bpp	0.4 bpp	0.6 bpp	0.8 bpp
Sachnev's method [15]	45.75(4.12)	40.70(3.65)	37.59(4.14)	34.53(5.59)
The proposed method	50.60(4.66)	43.22(4.87)	39.30(4.31)	36.14(5.04)

4. Conclusion

A difference expansion based reversible data hiding method exploiting the edge-oriented prediction is proposed to achieve the low distortion requirement. Experimental results have demonstrated that the proposed edge-oriented prediction method outperforms widely-used Median Edge Detection (MED) predictor, and Gradient Adjusted Predictor (GAP), and average predictor. Further, a modified overflow/underflow prevention method is proposed to reduce the overhead required to store the location map and to correctly extract secret data. Experimental results also demonstrate the proposed method outperforms Hong's method [13], Li's method [14] and Sachnev's method [15] in terms of image quality under the same capacity.

Acknowledgements

This work was supported in part by the Ministry of Science and Technology of R. 0. C. under contract MOST 103-2221-E-009-121-MY2.

References

[1] S. Katzenbeisser and F. A. P. Petitcolas, "Information Hiding Techniques for Steganography and Digital Watermark", Massachusetts: Artech House, Inc., **(2000)**, pp. 3.
[2] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich and T. Kalker, "Digital watermarking and steganography", Morgan Kaufmann Publisher, **(2008)**, pp. 429-495.
[3] J. Chou and K. Ramchandrad, "Next generation techniques for robust and imperceptible audio data hiding," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, **(2001)** May, pp. 1349-1352, Salt Lake City, UT.
[4] S. H. Liu, T. H. Chen, H. X. Yao and W. Gao, "A variable depth LSB data hiding technique in images," Proc. International Conference on Machine Learning and Cybernetics, vol. 7, **(2004)** August, pp. 3990-3994, Shanghai, China.
[5] Y. K. Lee and L. H. Chen, "High capacity image steganographic model," Proc. IEE Conference on Vision, Image, and Signal Processing, vol. 147, issue. 3, **(2000)** June, pp. 288-294.
[6] Y. K. Lee and L. H. Chen, "Object-based Image Steganography Using Affine Transformation," International Journal of Pattern Recognition and Artificial Intelligence, vol. 16, no. 6, **(2002)** September, pp. 681-696.
[7] Y. K. Lee and L. H. Chen, "Secure Error-Free Steganography for JPEG images," International Journal of Pattern Recognition and Artificial Intelligence, vol. 17, no. 6, **(2003)** September, pp. 967-981.
[8] J. Fridrich, M. Goljan and R. Du, "Lossless data embedding for all image formats," Proc. SPIE, vol. 4675, **(2002),** pp. 572-583.
[9] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, **(2006)** March, pp. 354-362.
[10] J. Tian, "Reversible data embedding using a difference expansion," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, **(2003)** August, pp. 890-896.
[11] C. C. Lee, H. C. Wu, C. S. Tsai and Y. P. Chu, "Adaptive lossless steganographic scheme with centralized difference expansion," Pattern Recognition, vol. 41, no. 6, **(2008)** June, pp. 2097-2106.

[12] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," IEEE Trans. Image Process, vol. 16, no. 3, **(2007)** March, pp. 721-730.
[13] W. Hong, "Adaptive reversible data hiding method based on error energy control and histogram shifting," Opt. Comm., vol. 285, no. 2, **(2012),** pp. 101-108.
[14] X. Li, B. Li, B. Yang and T. Zeng, "General Framework to Histogram-Shifting-Based Reversible Data Hiding," IEEE Trans. Image Process, vol. 22, no. 6, **(2013)** June, pp. 2181-2191.
[15] V. Sachnev, H. J. Kim, J. Nam, S. Suresh and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," IEEE Trans. Circuits Syst. Video Technol., vol. 19 , no. 7, **(2009)** July, pp. 989-999.
[16] "University of Southern California", SIPI database [Online], Available: http://sipi.usc.edu/database/.
[17] P. Bas, T. Filler and T. Pevny, "Break Our Steganographic System --- the ins and outs of organizing BOSS," Proc. Information Hiding, Prague, **(2011)**.
[18] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," IEEE Trans. Comm., vol. 45 , no. 4, **(2007),** pp. 437-444.

Authors

Wen-Chao Yang, he was born in Taoyuan, Taiwan, Republic of China on June 27, 1975. He received the B.S. degree in Department of Criminal Investigation from Central Police University, Taoyuan, Taiwan in 1997 and M.B.A. degree in Department of Information Management from National Central University, Taoyuan, Taiwan in 2002. He is now a Ph.D. Candidate of College of Computer Science at National Chiao Tung University, Hsinchu, Taiwan. His major research interests include image processing, information hiding, and forensic sciences.

Ling-Hwei Chen, she was born in Changhua, Taiwan, in 1954. She received the B.S. degree in Mathematics and the M.S. degree in Applied Mathematics from National Tsing Hua University, Hsinchu, Taiwan in 1975 and 1977, respectively, and the Ph.D. degree in Computer Engineering from National Chiao Tung University, Hsinchu, Taiwan in 1987.

From August 1977 to April 1979 she worked as a research assistant in the Chung-Shan Institute of Science and Technology, Taoyan, Taiwan, From May 1979 to February 1981 she worked as a research associate in the Electronic Research and Service Organization, Industry Technology Research Institute, Hsinchu, Taiwan. From March 1981 to August 1983 she worked as an engineer in the Institute of Information Industry, Taipei, Taiwan. She is now a Professor of the Department of Computer Science at the National Chiao Tung University. Her current research interests include image processing, pattern recognition, Multimedia compression, content-based retrieval and Multimedia Steganography.

Chang-Hsing Lee, he was born on July 24, 1968 in Tainan, Taiwan. He received the B.S. and Ph.D. degrees in Computer and Information Science from National Chiao Tung University, Hsinchu, Taiwan in 1991 and 1995, respectively. He is currently an Associate Professor in the Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan. His main research interests include audio/sound classification, multimedia information retrieval, and image processing.