Efficient Pixel Prediction Algorithm for Reversible Data Hiding

K. Bharanitharan[1], Chin-Chen Chang[2,3], Hai-Rui Yang[4], and Zhi-Hui Wang[4]
(Corresponding author: Chin-Chen Chang)

Department of Electrical Engineering, Feng Chia University[1]
Department of Information Engineering and Computer Science, Feng Chia University[2]
Taichung 40724, Taiwan
Department of Computer Science and Information Engineering, Asia University[3]
Taichung 41354, Taiwan
School of Software, Dalian University of Technology[4]
Dalian 116600, P. R. China
(Email: alan3c@gmail.com)
(Received Apr. 9, 2014; revised and accepted Mar. 20 & May 22, 2015)

Abstract

Prediction-error expansion (PEE) is an important reversible data hiding technique, which can hide large messages into digital media with little distortion. In this paper, we propose a nearest neighborhood pixel prediction algorithm (NNP^2) for reversible data hiding algorithms based on Chinese Remainder Theorem (CRT), in which a rhombus prediction is applied in NNP^2, and prediction errors, the difference between pixels and predictions, are modified to embed data. Further, CRT is utilized to adjust the modification size, thus embedding several bits into one embeddable pixel. Laplacian-like distribution of prediction errors is exploited to achieve a trade-off between embedding capacity and visual quality. Experimental results demonstrate that the NNP^2 achieves better embedding capacity with the same stego image quality than the conventional methods.

Keywords: Chinese remainder theorem (CRT), lossless watermarking, reversible data hiding (RDH)

1 Introduction

Reversible data hiding (RDH) techniques embed data into cover media, and, unlike most data hiding, the hidden message, as well as the cover data, can be completely recovered from the output data [3, 7, 16]. Reversibility of RDH offers a solution for lossless embedding in some sensitive media, such as medical images, military images, or artwork images, where even slight modification of these images is unacceptable due to the risk of a wrong explanation.

Generally, the existing RDH schemes are categorized into the following domains: compression domain, histogram shifting (HS) domain, difference expansion (DE) domain, and prediction-error expansion (PEE) domain. For compression domain, Fridrich et al. [6] compressed and encrypted bit-planes to make space for data embedding. However, large payloads would incur a greater level of distortion as a result of this method. Later, proposed generalized LSB (G-LSB) modifies the lowest level of the host image and achieves capacity-distortion control by modifying only a small portion of signal samples [14]. Although RDH schemes in compression domain attain high visual quality, their hiding capacity is relatively limited.

To lower the level of image distortion, the histogram shifting (HS) technique is employed within reversible data hiding. For HS domain, proposed histogram-based RDH scheme that uses pairs of peak points and zero points to lower image distortion, but lower embedding capacity [14]. In 2009, Tsai et al. [7] used linear prediction to calculate a residual image and embed data into the residual values, thus increasing the embedding capacity. Although this scheme achieves a higher embedding capacity, it is suitable only for medical images. Later, the proposed synchronization method solves the problem of communicating peak points for all histogram modification techniques by selecting certain peak points of the pixel difference histogram [17]. Also, the characteristics of the human visual system (HVS) are applied to histogram modifications to improve the visual quality of the embedded images, but this limited overall embedding capacity [11]. Afterwards, Al-Qershi and Khoo proposed a two-dimensional difference expansion (2D-DE) scheme and achieved a high hiding capacity of approximately 1 bit per pixel (bpp) [1]. Hereafter, histogram modification is extended to a pyramidal structure by utilizing global and local characteristics of host images [9]. This method

achieves a higher embedding capacity with acceptable visual quality. Further, Chang and Tai [4] improved histogram modification by sorting the prediction. However, this method of embedding for the White set expanded the prediction error, thus reducing the number of embeddable pixels of the Gray set. Note that histogram-based RDH schemes can achieve good visual quality and adequate embedding capacity, but they need to send pairs of peak and zero points to the receiver.

For DE domain, Tian [10] proposed a lossless DE algorithm based on the 1-D Haar wavelet transformation from 2003. The embedding capacity of this DE algorithm ranges from 0.15 to 1.97 bpp, which is notably higher than other previous schemes. Kamstra et al. [12] improved Tians technique by sorting the discrete wavelet transformation (DWT) coefficients in the low-pass band to make appropriate difference expansion in the high-pass band. These aforementioned DE methods achieve high embedding capacity but result in low image quality.

PEE domain is actually a specific extension of DE domain. In 2007, Thodi and Rodriguez [18] first proposed a PEE method and embedded messages by expanding the difference of pixels and predictions. Due to the Laplacian-like distribution of the prediction-error histogram,

PEE outperforms DE and HS. Afterwards, Hu et al. [8] proposed efficient compression of location map, and thus achieved an overall higher capacity. Sachnev et al. [15] embedded data in prediction errors when sorting by local variance. This method combines predicting and sorting and incurs less distortion compared with previous schemes. Li et al. [13] adaptively embedded one bit in rough pixels and two bits in smooth pixels to increase the hiding capacity. This method avoids large prediction-error expansion, thus achieving better image quality than conventional PEE. Coltuc [5] embedded data into the current pixel as well as its prediction context. This scheme obtains higher visual quality than classical PEE based on median-edge-detector (MED) or gradient-adjusted-predictor (GAP), since modifying four pixels minimizes the distortion introduced by data embedding. Bo et al. [2] generated a sequence consisting of prediction-error pairs and embedded data by expanding or shifting the 2D prediction-error histogram based on the sequence. Embedding in the correlations among prediction-errors rather than individual prediction-errors reduces the distortion.

The prior PEE schemes usually embed one bit, or two bits at most, into an expandable pixel. In this paper, a novel PEE algorithm based on Chinese Remainder Theorem is proposed. Compared with the previous PEE schemes, we first propose to embed six bits at most into one expandable pixel. In addition, we can adaptively achieve a trade-off of embedding capacity and visual quality. Experimental results show that the superiority of NNP^2 over other existing methods.

The rest of this paper is organized as follows. Section 1 introduces the Chinese Remainder Theorem briefly. Section 2 presents the proposed NNP^2 algorithm. Section 4 presents the experimental results and Section 5 concludes the paper.

2 Proposed Nearest Neighborhood Pixel Prediction (NNP^2) Algorithm

In this section we briefly explain the Chinese Remainder Theorem. Following that, our proposed algorithm is explained in detail.

2.1 Brief Introduction to Chinese Remainder Theorem

Chinese Remainder Theorem (CRT) provides a solution for a congruence system. A defined modulus set, $\{n_1, n_2, ..., n_m\}$, where m is a positive integer, and GCD(n_i, n_j) is equal to 1 for $i \neq j, i, j \in [0, m]$. For a positive integer X, there exists equations $x_i = X \bmod n_i$, where $i = 1, 2, ..., m$. The m-tuple $x_1, x_2, ..., x_m$ is unique for any $X \in [0, \prod n_i]$. A simple demonstration of CRT is briefly introduced here.

$$N = n_1 * n_2 * ... * n_m = \prod_{i=1}^{m} n_i \quad (1)$$

$$N_i = N/n_i \quad (2)$$

$$a_i = N_i^{-1} \bmod n_i \quad (3)$$

$$X = (\sum_{i=1}^{m} x_i * N_i * a_i) \bmod N. \quad (4)$$

Equations (1)-(4) above demonstrate the unique solution for the defined congruence system. X in Equation (4) tallies the m-tuple $\{x_1, x_2, ..., x_m\}$. The following example provides the detail implementation of Chinese Reminder Theorem In order to find an integer X for the congruence system, $\{n_1, n_2, n_3\} = \{3, 5, 7\}$ with a 3-tuple $x_1, x_2, x_3 = 2, 3, 2$. The computation process by using CRT is as follows.

$$\begin{aligned} N &= n_1 * n_2 * n_3 = 3 * 5 * 7 = 105. \\ N_1 &= N/n_1 = 105/3 = 35, \\ N_2 &= N/n_2 = 105/5 = 21, \\ N_3 &= N/n_3 = 105/7 = 15. \\ a_1 &= 2, a_2 = 1, a_3 = 1. \\ X &= (\sum_{i=1}^{m} x_i * N_i * a_i) \bmod N \\ &= (2 * 35 * 2 + 3 * 21 * 1 + 2 * 15 * 1) \bmod 105 \\ &= 23. \end{aligned}$$

Verification:

$$\begin{aligned} 23 \bmod 3 &= 2, \\ 23 \bmod 5 &= 3, \\ 23 \bmod 7 &= 2. \end{aligned}$$

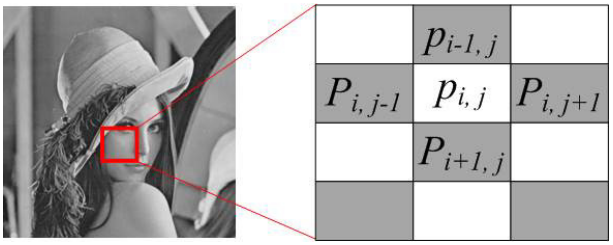

Figure 1: Rhombus prediction

The above example illustrated the Chinese remainder theorem.

2.2 Proposed Nearest Neighborhood Pixel Prediction (NNP^2) Algorithm

In this sub section, we explain the proposed NNP^2 algorithm, which uses CRT to embed data into digital images. NNP^2 can embed six bits, at most, into one expandable pixel by adjusting the modification size, whereas classical PEE schemes can embed a maximum of two bits in an expandable pixel. Laplacian-like distribution of prediction errors is employed to achieve a trade-off of embedding capacity and visual quality. Meanwhile, the rhombus prediction is exploited in order to find more expandable pixels. Simultaneously, we adopt the histogram shifting method to prevent overflow and underflow. In addition, we use a two-phase hiding strategy to transmit the side information to the receiver. Following sections outline the details of NNP^2.

2.2.1 Rhombus Prediction

In NNP^2, we exploit the rhombus prediction to express the high correlation of neighboring pixels. To calculate the prediction value of pixel $p_{i,j}$ in Figure 1, the rhombus prediction considers four neighboring pixels, $p_{i,j-1}$, $p_{i-1,j}$, $p_{i,j+1}$, and $p_{i+1,j}$. In Figure 1, all pixels of the host image are partitioned into two categories: "White" pixels and "Black" pixels. A "White" pixel is predicted by four neighboring "Black" pixels. Note that "White" and "Black" pixels are independent and the modification of "White" pixels do not influence any "Black" pixels, and vice versa. The central pixel, $p_{i,j}$ in Figure 1, can be predicted by its left, upper, right and lower pixels, $p_{i,j-1}$, $p_{i-1,j}$, $p_{i,j+1}$, and $p_{i+1,j}$. The prediction value $p'_{i,j}$ is calculated by Equation (5).

$$p'_{i,j} = \lfloor \frac{p_{i,j-1} + p_{i-1,j} + p_{i,j+1} + p_{i+1,j}}{4} \rfloor \quad (5)$$

2.2.2 Prediction Error Expansion

The data-embedding algorithm for "White" pixels is described as follows. We assume that two primes are p and q, $N = \log_2(q)$, a threshold $T = p*(q-1)$ and the secret file is SF.

The following steps demonstrate the prediction error expansion:

1) Calculate the prediction value $p'_{i,j}$ of "White" pixel $p_{i,j}$ using Equation (6).

2) Calculate the absolute value D of the difference between and using the following equation.

$$D = |p_{i,j} - p'_{i,j}| \quad (6)$$

3) If $0 \le D < p$, read N bits from SF and its decimal value is S, calculate an integer C by using CRT which satisfies that $D = C \bmod p$ and $S = C \bmod q$, and modify $p_{i,j}$ according to those N bits.

$$w_{i,j} = \begin{cases} p'_{i,j} + C \ , \mathrm{p_{i,j}} \ge \mathrm{p'_{i,j}} \\ p'_{i,j} - C \ , \mathrm{p_{i,j}} < \mathrm{p'_{i,j}} \end{cases} \quad (7)$$

where $w_{i,j}$ is the watermarked pixel value of $p_{i,j}$.

4) If $D \ge p$, shift T unit

$$w_{i,j} = \begin{cases} p_{i,j} + T \ , \mathrm{p_{i,j}} \ge \mathrm{p'_{i,j}} \\ p_{i,j} - T \ , \mathrm{p_{i,j}} < \mathrm{p'_{i,j}} \end{cases} \quad (8)$$

where $w_{i,j}$ is the watermarked pixel value of $p_{i,j}$.

The embedding algorithm calculates the prediction value of "White" pixels using "Black" pixels and embeds data into a "White" pixel. Therefore, after embedding in "White" pixels, "Black" pixels are unchanged and "White" pixels are modified as the watermarked pixels. "Black" pixels are considered to embed data by using "White" pixels to calculate their prediction value in the same way. Therefore, the usage of both "White" and "Black" pixels almost double the hiding capacity.

After receiving the watermarked image, the receiver can extract the hidden message by using the same scanning method as used during the embedding algorithm. The receiver calculates the absolute value of the difference C' between the watermarked pixel, $w_{i,j}$, and the prediction value, $p'_{i,j}$. Then, the hidden message can be extracted by

$$S = C' \bmod (q), 0 \le C' < p \times q \quad (9)$$

where S is the decimal value of original N secret bits. Then the original pixel value of $w_{i,j}$ can be recovered by implementing the following equations,

$$D = \begin{cases} C' \bmod (p) \ , 0 \le \mathrm{C'} < \mathrm{p} \times \mathrm{q} \\ C' - T \ , \mathrm{p} \times \mathrm{q} \le \mathrm{C'} \end{cases} \quad (10)$$

$$p_{i,j} = \begin{cases} p'_{i,j} + D \ , \mathrm{w_{i,j}} \ge \mathrm{p'_{i,j}} \\ p'_{i,j} - D \ , \mathrm{w_{i,j}} < \mathrm{p'_{i,j}} \end{cases} \quad (11)$$

Consequently, the receiver can acquire the exact copy of the original image together with the hidden message.

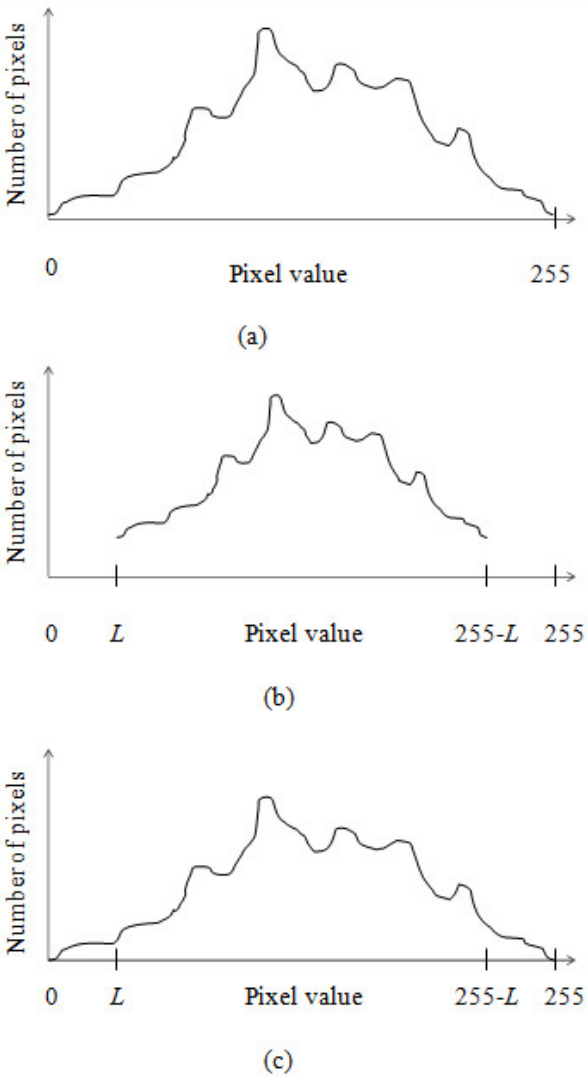

Figure 2: Histograms: (a) histogram of host image, (b) histogram after shifting, and (c) histogram after embedding.

2.2.3 Prevention of Overflow and Underflow

The overflow and underflow problem may occur when watermarked pixels are out of the range of [0, 255]. To prevent these potential issues, we employ a histogram shifting scheme in [8] to shift the histogram from both left and right sides as shown in Figure 2. Assume that two primes used in NNP^2 are p and q, the maximum modification size is $L = p \times q - p$. Therefore, we shift the histogram of host image L units from both left and right sides.

After shifting all potential overflow and underflow pixels, we should construct a one-bit location map to record the histogram shifting information. If a pixel's grayscale value is in the range of [L, 255-L], then we assign a "0" as its location information; otherwise, a "1" is assigned to it. We compress the location map by using the run length-coding algorithm, which can achieve great compression efficiency for overflow and underflow pixels when their numbers are few and consecutive. In NNP^2, the maximum modification size is L. Therefore, shifting the histogram of the host image from both the left and right side by L units can prevent overflow and underflow. Note that the extra information including p, q, and the location map should be transmitted to the receiver to extract the hidden message and recover the host image correctly. For this purpose, we adopt a two-stage embedding strategy to embed extra information together with secret data.

2.2.4 Upper Bound of p and q

NNP^2 uses two primes, p and q, to adjust the range of embeddable pixels and the number of secret bits one embeddable pixel can embed. The larger the p and q values are, the higher the embedding capacity is. However, to prevent overflow and underflow, the histogram of host image is shifted $L = p \times q - p$ units. Pixels with values in [0, L-1] are shifted to [L, 2L-1], and pixels with values in [256-L, 255] are shifted to [256-2L, 255-L]. To distinguish overflow pixels from underflow pixels, p and q must satisfy that $2L - 1 < 256 - 2L$, namely $p \times q - p < 257/4$.

Based on the above condition, while p and q are both integers, we can derive that $p \times q - p < 64$. We set q as a multiple of 2, such as 2, 4, 8, 16, 32 or 64, which means one pixel can embed 6 bits at most.

2.2.5 Two-stage Embedding

Extra information, namely two primes, p and q, and the location map, are sent to the receiver to extract hidden message and recover the host image. Note that $L = p \times q - p$, and there is no need to transmit. Assume that $|LM|$ is the size of the location map. The LSB values of the first $2 \times 6 + |LM|$ pixels of the watermarked image are replaced with $p, q (2 \times 6$ bits) and the location map ($|LM|$ bits). The original $2 \times 6 + |LM|$ LSB values are added to the payload. Therefore, the pure capacity, C, which excludes all extra information, is

$$C = |N_s| - |EI| \tag{12}$$

where $|N_s|$ is the number of secret bits and $|EI|$ is the size of extra information.

3 Experimental Results and Discussion

We have conducted several experiments to evaluate the performance of NNP^2 and compare it with some existing RDH algorithms. Figure 3 shows six typical grayscale images of size 512×512 that were used as test images.

Table 1 shows the pure payload, C, and the PSNR of different test images when p and q are set to be 1 and 2. We can see that the number of expandable pixels, N_e, is highly variable between different images. Smoother images, such as F-16, result in a larger N_e than that of more complex images, like Baboon. We also have observed that

Table 1: Embedding capacity and PSNR for six test images with $p = 1$ and $q = 2$

Host image (512×512)	N_e	**Pure capacity** C **(bits)**	**Extra information** $\|EI\|$ **(bits)**	**PSNR (dB)**	**Bit rate (bpp)**
Lena	34096	34037	32	48.42	0.1298
Baboon	9625	9493	132	48.22	0.0362
Boat	33432	33400	32	48.42	0.1274
F-16	53199	53167	32	48.60	0.2028
Peppers	20250	20218	32	48.30	0.0771
GoldHill	24245	24213	32	48.34	0.0924

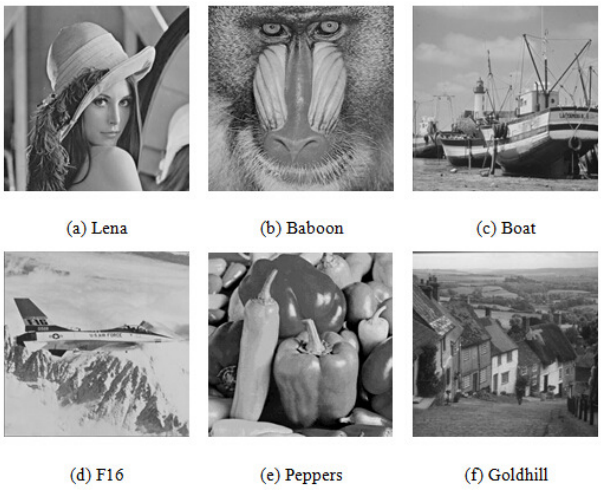

Figure 3: Six test images

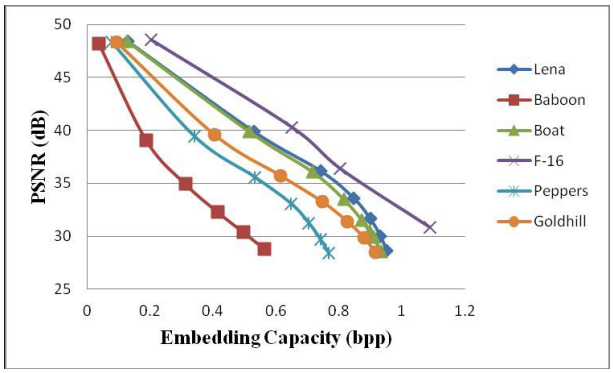

Figure 4: Performance of test images with optimal p and q

all test images except for Baboon have no pixels out of the range [1, 254]. For Lena, the size of location map is $|LM| = 20$ bits when we use the run-length coding algorithm to compress the location map. Thus, the size of extra information is $|EI| = (2 \times 6) + |LM| = 32$ bits.

Table 2 shows several experiments to evaluate the pure embedding capacity versus the PSNR change with different p and same q. An expandable pixel, which can only embed one bit for q, is constantly equal to 2, while p is equal to $2i + 1, i[0, 6]$, which is the threshold of expandable pixels whose absolute value of prediction errors are less than p. We can see that the average PSNR values decrease with increasing p values. However, the growth of the pure embedding capacity is smaller when the size of extra information is growing increasingly larger. In Table 3, we assign p a constant value of 1, and q changes from 2 to 64. This means that only a pixel with a prediction error of 0 is expandable, and one pixel embeds 1 to 6 bits. We observed that the pure embedding capacity will increase and then decrease while q changes from 2 to 64. The embedding capacity increase first for one pixel can embed several bits, but when q is much bigger, the size of extra information increases faster, and the embedding capacity decreases. By adjusting the values of p and q, NNP^2 can adaptively achieve a high embedding capacity with little loss of visual quality.

Figure 4 presents the performance of NNP^2 for test images with optimal p and q values. We can see from the figure that smooth images, like F-16, achieve higher embedding capacity with the same PSNR values, while complex images, like Baboon, will experience severe distortion when the embedding capacity is higher. Therefore, when implementing NNP^2 , images with high correlation perform better than those with low correlation.

We also compare the embedding capacity (bpp) versus image quality (dB) of NNP^2 with that of existing RDH schemes for Lena, as shown in Figure 5. NNP^2 can embed six bits at most, rather than one or two bits, into the prediction error and shift all the non-embeddable pixels. It should be noted that [12] and [8] are histogram-based RDH schemes, which embed data into peak points and just shift the pixels between peak points and zero points rather than all non-embeddable pixels, thus achieving a high embedding capacity while keeping a low distortion rate. Based on the above reasons, the PSNR of NNP^2 is lower than that of the compared methods [12] and [8] when embedding the same number of bits into one expandable pixel. However, proposed algorithm can achieve a trade-off between the embeddable pixels and the number of secret bits that one embeddable pixel can embed. It should be noted that the proposed NNP^2 achieved a higher embedding capacity than those of existing schemes

Table 2: Pure embedding capacity for test images with $q = 2$

Host image (512×512)	$p = 1$	$p = 3$	$p = 5$	$p = 7$	$p = 9$	$p = 11$	$p = 13$
Lena	0.1298	0.5288	0.7424	0.8468	0.8998	0.9316	0.9509
Baboon	0.0362	0.1876	0.3117	0.4140	0.4968	0.5627	0.6151
Boat	0.1247	0.5156	0.7165	0.8152	0.8724	0.9089	0.9332
F-16	0.2028	0.6505	0.8039	0.8729	0.9111	0.9355	0.9522
Peppers	0.0771	0.3405	0.5329	0.6474	0.7045	0.7409	0.7671
GoldHill	0.0924	0.4047	0.6129	0.7461	0.8270	0.8799	0.9143
Average PSNR	48.38	39.69	35.81	33.24	31.30	29.72	28.38

Table 3: Pure embedding capacity for test images with $p = 1$

Host image (512×512)	$q = 2$	$q = 4$	$q = 8$	$q = 16$	$p = 32$	$p = 64$
Lena	0.1298	0.2374	0.3106	0.3680	0.4025	0.0939
Baboon	0.0362	0.0707	0.0914	0.1069	0.0541	-0.8490
Boat	0.1247	0.2298	0.3081	0.3663	0.2778	0.2048
F-16	0.2028	0.3561	0.4536	0.5394	0.5939	0.1090
Peppers	0.0771	0.1347	0.1360	0.0945	0.1098	-0.1639
GoldHill	0.0924	0.1726	0.2338	0.2751	0.1826	-0.1671
Average PSNR	48.38	38.87	31.49	24.84	18.52	12.34

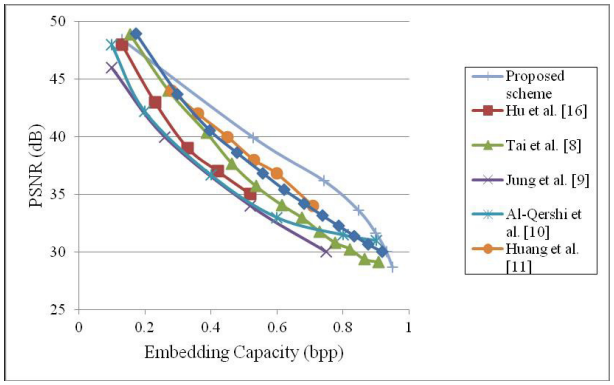

Figure 5: Performance comparison for Lena of NNP^2 and existing RDH schemes

with low distortion. As a result, NNP^2 can obtain better performance in RDH schemes by adjusting the embedding capacity and image quality.

4 Conclusions

In this paper, we proposed NNP^2 to embed data into prediction errors by using CRT to control the modification size. Laplacian-like distribution of prediction errors, which is centered with 0, increases the embedding ability of NNP^2. Also, we control the number of embeddable pixels by adjusting the threshold of prediction errors and embed one to six bits at most into one embeddable pixel that leads to the best performance. We also adopt a two-stage strategy to embed the extra information, rather than sending it to the receiver in an open way. In addition, we use a histogram shifting technique to prevent overflow and underflow. As a result, NNP^2 outperformed the compared algorithms.

Acknowledgments

This work was supported by the National Nature Science Foundation of China under Grant No. 61201385.

References

[1] Q. M. Al-Qershi and B. Ee Khoo, "Two-dimensional difference expansion (2d-de) scheme with a characteristics-based threshold," *Signal Processing*, vol. 93, pp. 154–162, 2013.

[2] Ou Bo, X. Li, et al., "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5010–5021, 2013.

[3] R. Caldelli, F. Filippini, and R. Becarelli, "Reversible watermarking techniques: an overview and a classification," *EURASIP Journal on Information Security*, vol. 2010, pp. 1–19, 2010.

[4] Ya Fen Chang and Wei Liang Tai, "Histogram-based reversible data hiding based on pixel differences with

prediction and sorting," *KSII Transaction on internet and information systems*, vol. 6, no. 12, pp. 3100–3116, 2012.

[5] D. Coltuc, "Low distortion transform for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 412–417, 2012.

[6] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication watermark for JPEG images," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 412–417, 2001.

[7] Y. Hu and B. Jeon, "Reversible visible watermarking and lossless recovery of original images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 11, pp. 1423–1429, 2006.

[8] Y. Hu, H. K. Lee, et al., "De-based reversible data hiding with improved overflow location map," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 250–260, 2009.

[9] H. C. Huang and F. C. Chang, "Hierarchy-based reversible data hiding," *Expert Systems with Applications*, vol. 40, pp. 34–43, 2013.

[10] T. Jun, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[11] S. W. Jung, S. J. Ko, et al., "A new histogram modification based reversible data hiding algorithm considering the human visual system," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 95–98, 2011.

[12] L. Kamstra and J. A. M. H. Henk, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2082–2090, 2005.

[13] X. Li, B. Yang, et al., "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524–3533, 2011.

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

[15] V. Sachnev, J. K. Hyoung, et al., "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989–999, 2009.

[16] Y. Q. Shi, Z. Ni, D. Zou, C. Liang, and G. Xuan, "Lossless data hiding: fundamentals, algorithms and applications," in *Proceedings of the 2004 IEEE International Symposium on Circuits and Systems (ISCAS'04)*, vol. 2, pp. II–33, 2004.

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

[18] D. M. Thodi, J. R. Jeffrey, et al., "Expansion embedding techniques for reversible watermarking," *IEEE Transaction Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.

K. Bharanitharan (S07CM09) received the PhD degree in Electrical Engineering from the National Cheng Kung University, Tainan, Taiwan, in 2009. In 2005, he won outstanding international student fellowship award at National Cheng Kung University. He serves as a reviewer for IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Very Large Scale Integration Systems, IEEE Transactions on Evolutionary Computation, IEEE Signal processing letter, IEEE Transactions on Very Large Scale Integration Systems since 2009. He has published more than sixteen research papers in highly reputed journals and conferences. His research interests include H.264/AVC video coding, HEVC, scalable video coding, image processing, multiview video coding, and associated VLSI architectures. His research works also include Multi-Core reconfigurable systems, Java based apps development and dynamic power management for advanced video coding.

Chin-Chen Chang obtained his Ph.D. degree in computer engineering from Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in Computer and Decision Sciences. Both were awarded in Tsing Hua University. Dr. Chang served in Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in Chung Hsing University, chair professor in Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang's specialties include, but not limited to, data engineering, database systems, computer cryptography and information security. A researcher of acclaimed and distinguished services and contributions to his country and advancing human knowledge in the field of information science, Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. He also published more than 1200 papers in Information Sciences. In the meantime, he participates actively in international academic organizations and performs advisory work to government agencies and academic organizations.

Hai-Rui Yang received the BS degree in software engineering in 2012 from the Dalian University of Technology,

Dalian, China. Since September 2012, he has been studying for his MS degree in software engineering in Dalian University of Technology, Dalian, China. His current research interests include information hiding and image processing.

Zhi-Hui Wang received the BS degree in software engineering in 2004 from the North Eastern University, Shenyang, China. She received her MS degree in software engineering in 2007 and the PhD degree in software and theory of computer in 2010, both from the Dalian University of Technology, Dalian, China. Since November 2011, she has been an assistant professor of Dalian University of Technology. Her current research interests include information hiding and image processing.