Second Order Perdicting-Error Sorting for Reversible Data Hiding

Jiajia Xu, Hang Zhou, Weiming Zhang [*], Ruiqi Jiang, Guoli Ma, and
Nenghai Yu

University of Science and Technology of China
No.96, Baohe District,Hefei, 230026,P.R.China
{xujiajia,jrq123}@mail.ustc.edu.cn
{zhangwm,ynh}@ustc.edu.cn
{hangzhou_ryan}@163.com

Abstract. Reversible data hiding (RDH) schemes compete against each
other for a sharply distributed prediction error histogram, which is of-
ten realized by utilizing prediction strategies together with sorting tech-
niques. The sorting technique aims to estimate the local context com-
plexity for each pixel to optimize the embedding order. In this paper,
we propose a novel second order perdicting and sorting technique for
reversible data hiding. Firstly, the prediction error is obtained by an in-
terchannel secondary prediction using the prediction errors of current
channel and reference channel. Experiments show that this prediction
method can produce a shaper second order prediction-error histogram.
Then, we will introduce a novel second order perdicting-error sorting
(SOPS) algorithm, which make full use of the feature of the edge infor-
mation obtained from another color channel and high correlation between
adjacent pixels. So it will reflect the texture complexity of current pixel
better. Experimental results demonstrate that our proposed method out-
performs the previous state-of-arts counterparts significantly in terms of
both the prediction accuracy and the overall embedding performance.

Keywords: Reversible Data Hiding, Channel Correlation, Second Or-
der Perdicting-Error.

1 Introduction

Reversible data hiding (RDH) [1],as being a special branch of information hiding,
has received a lot of attention in the past few years. It is not only concerned about
the user's embedding data, but also pays attention to the carriers themselves.
RDH ensures that the cover data and the embedded message can be extracted
from the marked content precisely and losslessly. This important data hiding

technique provides valuable functions in many fields, such as medical image protection, authentication and tamper carrier recovery, digital media copyright protection, military imagery and legal, where the cover can not be damaged during data extraction. A framework of RDH for digital images is illustrated in Fig. 1.
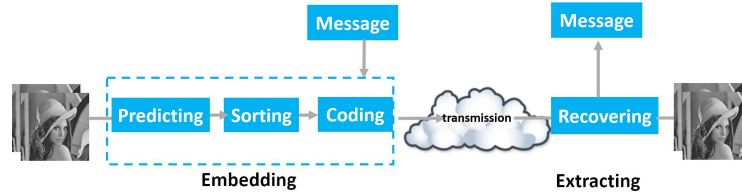

Fig. 1: Framework of RDH embedding/extraction.

In order to improve the efficiency of RDH, researchers have proposed many methods in the past decades. Generally speaking, prediction and sorting play an important role in the process of reversible information hiding. The prediction focuses on how to better exploit inter-pixel correlations to derive a sharply distributed one. And the emphasis of sorting technique is exploiting the correlation between neighboring pixels for optimizing embedding order.

The improvement of the prediction is important for both histogram shifting and difference expansion based on RDH schemes. Some predictors [4] have proposed such as median edge detection (MED), gradient adjusted prediction (GAP), and differential adaptive run coding (DARC). The MED predictor tends to select the lower vertical neighbor in the cases where a vertical edge exists right to the current location, the right neighbor in cases of a horizontal edge below it, or a linear combination of the context pixels if no edge is detected. The GAP algorithm weighs the neighboring pixels according to local gradient and classifies the edges to three classes namely, sharp, normal, and weak, which uses seven neighboring pixels to estimate unknown pixel value. The DARC is a non-linear adaptive predictor that uses three neighboring pixels to estimate the unknown pixels. C. Dragoi and D. Coltuc [3] have proposed extended gradient-based selection and weighting (EGBSW) for RDH. EGBSW [3] algorithm uses four linear predictors, computing the output value as a weighted sum between the predicted values corresponding to the selected gradients, then predicted value is obtained.

Sorting[5, 6] is a fundamental step to exploit the correlation between neighboring pixels for optimizing embedding order, hence sorting is a fundamental step to enhance the embedding capacity and visual quality. Kamstra and Heijmans' [7] use of sorting introduced a significant performance advantage over previous methods.They reduced the location map size through arranging the pairs of pixels in order. Sachnev et al [5] used local variance values to sort the predicted errors. They sort the cells in ascending order of the local variance values, which first embeds the smoother cells with lower local variance values.

But in some cases, it does not work directly accurate. For example, Mahsa Af-sharizadeh [6] extended Sachnev et al.s work [5] by using a new proposed efficient sorting technique, which is a new sorting measure resulted from more accurate sorting procedures. Ou [8] proposed a simple and efficient sorting method by calculating its local complexity, which is the sum of absolute differences between diagonal blank pixels in the 4×4 sized neighborhood. A small local complexity indicates that the pair is located in a smooth image region and should be used preferentially for data embedding. However, the above algorithms did not consider the characteristic of prediction-error distribution.

However, we notice that the existing research about RDH is commonly focused on gray-scale images. In real life, it is the color images that are widely used. In recent days, reversible data hiding for color images is a rarely studied topic in [9, 10]. Considering that the color channels correlate with each other, J. Li *et al.* [11] propose a RDH algorithm based on prediction-error expansion that can enhance the prediction accuracy in one color channel through exploiting the edge information from another channel. By doing so, the entropy of the prediction-error is decreased statistically. Based on the inter-channel correlation of color image J. Li *et al.* [11], a novel inter-channel prediction method is examined and a corresponding reversible data hiding algorithm is proposed.

In our proposed method, the prediction error is obtained by an interchannel secondary prediction using the prediction errors of current channel and reference channel. Experiments show that this prediction method can produce a shaper second order prediction-error histogram. Then, we will introduce a novel second order perdicting-error sorting (SOPS) algorithm, which make full use of the feature of the edge information obtained from another color channel and high correlation between adjacent pixels. So it will reflect the texture complexity of current pixel better. Experimental results demonstrate that our proposed method outperforms the previous state-of-arts counterparts significantly in terms of both the prediction accuracy and the overall embedding performance.

The paper is organized as follows. Section 2 describes the proposed second order perdicting-error sorting algorithm for reversible data hiding. The simulations done using the proposed technique and the obtained results are presented in section 3. In Section 4, conclusions are briefly drawn based on the results.

2 Second Order Perdicting-Error Sorting (SOPS) for Color Image

2.1 Second Order Perdicting-Error Based on Correlation Among Color Channels

Naturally, the edges of images play a critical role in human visual system (HVS), which is revealing with jump in intensity. How to use the property to predict pixel is very important. Some predictors have proposed for RDH such as median edge detection (MED) and gradient adjusted prediction (GAP) [2] may be ineffective for the rough region. The perdicting-error in this situation usually differs

greatly from the pixel because of the large difference between the pixels along the gradient direction in rough region. However, more precise perdicting-error can be obtained by the pixels, which is along the edge direction. Especially for color image, we can make full use of the feature of the edge information obtained from another color channel and high correlation between adjacent pixels. In [11], J. Li *et al.* pointed out that the edge information drawn from different color channels is similar to each other. Hence, if an edge is detected in one color channel, there would be an identical edge in the same position in the other channels.

In each channel, we use double-layered embedding method proposed by Sachnev *et al.* [13], with all pixels divided into the shadow pixel set and the blank set (see fig.2). In the first round, the shadow set is used for embedding data and blank set for computing predictions. While in the second round, the blank set is used for embedding and shadow set for computing predictions. Since the two layers embedding processes are similar in nature, we only take the shadow layer for illustration. Let p_c and p_r denote the sample of a pixel p in the current
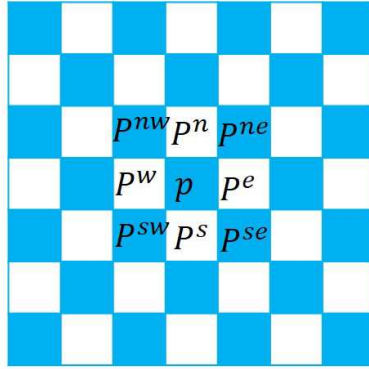

Fig. 2: The current pixel and its neighboring pixels

channel and reference channel. In order to determine whether the pixel is located on the edge of the image, we need to calculate the two parameters. The average distance D_{avg} and direction distance D_{dir} can be given by

$$D_{avg} = \left| \sum_{k=1}^{8} \alpha_{avg}^k p_r^k - p_r \right| \tag{1}$$

where α_{avg}^k (k=1,2,...8) are coefficients of eight neighbors $p_r^1 = p_r^{nw}$, $p_r^2 = p_r^n$, $p_r^3 = p_r^{ne}$, $p_r^4 = p_r^w$, $p_r^5 = p_r^e$, $p_r^6 = p_r^{sw}$, $p_r^7 = p_r^s$ and $p_r^8 = p_r^{se}$. If p_r locate at

smooth region, the average distance D_{avg} should be small. On the contrary, if p_r locate at rough region, the D_{avg} is very large. However, how to determine the direction of the edge in the image. Next, we determine the direction by calculating the direction distance D_{dir} as follow

$$D_{dir} = min\left\{\left|\frac{p_r^w + p_r^e}{2} - p_r\right|, \left|\frac{p_r^n + p_r^s}{2} - p_r\right|, \right.\tag{2}$$
$$\left.\left|\frac{p_r^{nw} + p_r^{se}}{2} - p_r\right|, \left|\frac{p_r^{ne} + p_r^{sw}}{2} - p_r\right|\right\}$$

where $|(p_r^w + p_r^e)/2 - p_r|, |(p_r^n + p_r^s)/2 - p_r|, |(p_r^{nw} + p_r^{se})/2 - p_r|$, and $|(p_r^{ne} + p_r^{sw})/2 - p_r|$ represent the four edge directions, which are horizontal, vertical, diagonal and antidiagonal. Taking the smallest one as D_{dir}. Edges are revealing with jump in intensity, for instance, the smallest means the pixels are locate on the edges of image.

The $|D_{avg} - D_{dir}|$ can be used to indicate whether the reference sample is located on an edge region. Considering that all the color channels have similar edge distribution in [11], the prediction of the pixels should be taken into account the edge information obtained from another channel. Therefore, we can employ $|D_{avg} - D_{dir}|$ to classify the location of the current sample p_c. If the $|D_{avg} - D_{dir}|$ is close to zero or very small, which means the current sample p_c locate on the smooth region of image. In this case, the edge information obtained from another channel is useless, we can make full use of the eight neighbors of p_c. On the other hand, if $|D_{avg} - D_{dir}|$ is larger than a predefined threshold ρ, we think that p_c is located at or near to an image edge area at a high possibility. Under the circumstances, the edge information obtained from another channel should be taken into account. So, we can get

$$\hat{p}_c = \begin{cases} \lfloor(p_c^w + p_c^e + p_c^n + p_c^s)/4 + 0.5\rfloor & |D_{avg} - D_{dir}| \leq \rho, \\ \lfloor P(p_c^k|D_{dir}) + 0.5\rfloor & |D_{avg} - D_{dir}| > \rho. \end{cases}$$

where $P(p_c^k|D_{dir})$ according to D_{dir}. For instance, when $D_{dir} = |(p_r^w + p_r^e)/2 - p_r|$, then $P(p_c^k|D_{dir}) = (p_r^w + p_r^e)/2$. When $D_{dir} = |(p_r^n + p_r^s)/2 - p_r|$, then $P(p_c^k|D_{dir}) = (p_r^n + p_r^s)/2$. When $D_{dir} = |(p_r^{nw} + p_r^{se})/2 - p_r|$, then $P(p_c^k|D_{dir}) = (p_r^{nw} + p_r^{se})/2$. When $D_{dir} = |(p_r^{ne} + p_r^{sw})/2 - p_r|$, then $P(p_c^k|D_{dir}) = (p_r^{ne} + p_r^{sw})/2$.

Then we can get the first order perdicting-error based on correlation among color channels as follow

$$\Delta e_c = p_c - \hat{p}_c \tag{3}$$

$$\Delta e_r = p_r - \hat{p}_r \tag{4}$$

where Δe_c is first order perdicting-error in the current channel and Δe_r is first order perdicting-error reference channel. Next, the second order prediction-error is computed by

$$\Delta^2 e = \Delta e_c - \Delta e_r \tag{5}$$

when the pixels in the smooth region of image, the pixels are similar to each other and the first order perdicting-errors Δe_c and Δe_r are close to zero. Therefore,

the second order prediction-errors are also close to zero. On the other hand, when the pixels are located at rough region, the first order perdicting-errors Δe_c and Δe_r relatively large. However, considering that all the color channels have similar edge distribution and take into account the edge information obtained from another channel. the second order prediction-errors become smaller. So, the second order prediction-error sequence $\Delta^2 e = (\Delta^2 e_1, \cdots, \Delta^2 e_N)$ is derived.

How to evaluate the prediction method? The entropy value of the prediction-error can be used to evaluate the performance of the proposed prediction method. If the entropy is smaller , the performance of prediction is better. On the contrary, if the entropy is larger , the performance of prediction is worse. In this paper, the UCID (Uncompressed color image database) is employed in our experiment, which has over 1300 uncompressed color images. In the fig.3, we can observe that the entropy value of our proposed method is smaller than that corresponding to other tradition methods such as MED, rhombus and first order perdicting-error based on correlation among color channels.
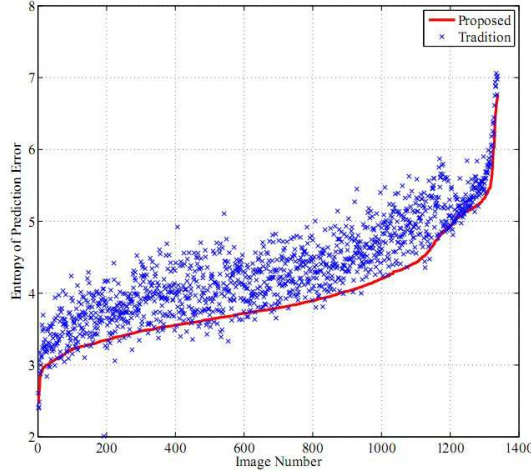

Fig. 3: The entropy value of the prediction-error obtained by our proposed prediction method and other tradition methods

2.2   Second Order Perdicting-Error Sorting Based on Generalized Normal Distribution

The generalized error distribution is a generalized form of the normal, it possesses a natural multivariate form, and has a parametric kurtosis that is unbounded above and possesses special cases that are identical to the Normal and the double exponential distributions [21]. Given that the probability density function (PDF) of prediction-error follows generalized normal distribution or gaussian

distribution, we consider using this model to describe the prediction-error in fig.3. Generalized normal distribution density function is defined by Nadarajah [9].

$$f(\Delta e|u, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(\frac{1}{\beta})}exp\left\{-\left|\frac{(\Delta e - u)}{\alpha}\right|^{\beta}\right\}, \tag{6}$$

where Δe is prediction-error with mean u and variance σ^2. $\alpha = \sqrt{\sigma^2\Gamma(1/\beta)/\Gamma(3/\beta)}$ is a scale parameter, playing the role of a variance that determines the width of the PDF, while $\beta > 0$, called the shape parameter, controls the fall-off rate in the vicinity of the mode (the higher β, the lower the fall-off rate). $\Gamma(.)$ denotes the Gamma function such that $\Gamma(t) = \int_0^\infty x^{t-1}exp(-x)dx$. It is easy to see that the Eq. (6) reduces to the normal distribution for $\beta = 2$, and Laplacian distribution for $\beta = 1$.
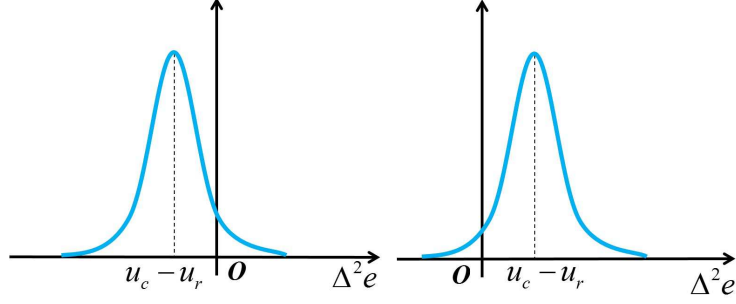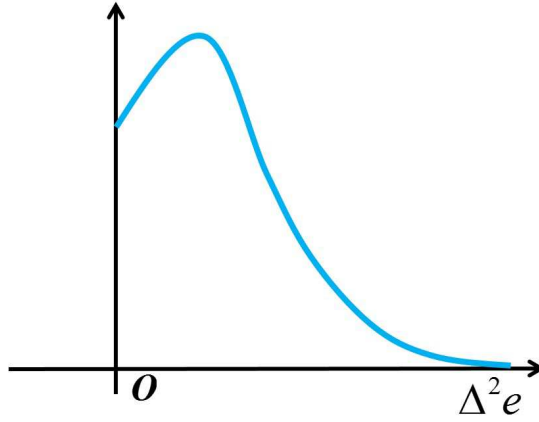
Based on Eq. (6), we can easily get $\Delta e_c \sim GND(u_c, \alpha_c, \beta_c)$ and $\Delta e_r \sim GND(u_r, \alpha_r, \beta_r)$. Then $\Delta^2 e = \Delta e_c - \Delta e_r$ can be given by

$$\begin{aligned} f(\Delta^2 e) &= -\int_{-\infty}^{+\infty} f_{\Delta e_c - \Delta e_r}(\Delta e_c, \Delta^2 e - \Delta e_c)d\Delta e_c \\ &= -\int_{-\infty}^{+\infty} \frac{\beta_c\beta_r}{4\alpha_c\alpha_r\Gamma(1/\beta_c)\Gamma(1/\beta_r)} \\ &\times exp\left(-\left|\frac{\Delta e_c - u_c}{\alpha_c}\right|^{\beta_c} - \left|\frac{\Delta^2 e - \Delta e_c - u_r}{\alpha_r}\right|^{\beta_r}\right)d\Delta e_c \end{aligned} \tag{7}$$

According to Eq. (7), there does not appear to exist a closed form expression. However, in order to simplify and reduce the complexity of the problem, we take $\beta = 1$ as a particular case of generalized normal. When $\beta = 1$ is corresponding to the Laplace distribution as follows

$$\begin{aligned} f(\Delta^2 e) &= -\int_{-\infty}^{+\infty} f_c(\Delta e_c)f_r(\Delta^2 e - \Delta e_c)d\Delta e_c \\ &= \frac{\alpha_c}{2(\alpha_c^2 - \alpha_r^2)}exp(-\frac{|\Delta^2 e - (u_c - u_r)|}{\alpha_c}) \\ &- \frac{\alpha_r}{2(\alpha_c^2 - \alpha_r^2)}exp(-\frac{|\Delta^2 e - (u_c - u_r)|}{\alpha_r}) \end{aligned} \tag{8}$$

It can easily be seen that the $f(\Delta^2 e)$ is the difference between the two Laplace distributions with the same mean $u_c - u_r$ under different weights. The probability density function (PDF) of second order prediction-error can be seen in fig.4. As shown in fig.4, the distance from $u_c - u_r$ to y axis can be used to reflect the accuracy of the second order prediction error. The smaller the distance is, the better the accuracy of prediction is. Since the distance is used to measure the

Fig. 4: The distribution of $f(\Delta^2 e)$

Fig. 5: The distribution of $\Phi(\Delta^2 e)$

accuracy of the prediction error, we can get a new function $\Phi(\Delta^2 e) = |f(\Delta^2 e)|$. The distribution of $\Phi(\Delta^2 e)$ can be given by It can easily be seen that the expectation of function $E[\Phi(\Delta^2 e)]$ has positive correlation to $u_c - u_r$. So it can be used to characterize the accuracy of the second order prediction error. For example, if the $E[\Phi(\Delta^2 e)]$ is higher, which means the pixels in image region are more random or unpredictable. Consequently, the pixels are hard to predict accurately in this region. Thus, the prediction-errors can be rearranged by sorting according to $E[\Phi(\Delta^2 e)]$. Let $u = u_c - u_r$, $a = \frac{\alpha_c}{2(\alpha_c^2 - \alpha_r^2)}$ and $b = \frac{\alpha_r}{2(\alpha_c^2 - \alpha_r^2)}$. The $\Phi(\Delta^2 e)$ can be given by

$$
\begin{aligned}
\Phi(\Delta^2 e) = {} & \frac{a}{2\alpha_c}(exp(-\frac{|u - \Delta^2 e|}{\alpha_c}) + exp(-\frac{u + \Delta^2 e}{\alpha_c})) \\
& - \frac{b}{2\alpha_r}(exp(-\frac{|u - \Delta^2 e|}{\alpha_r}) + exp(-\frac{u + \Delta^2 e}{\alpha_r}))
\end{aligned}
\tag{9}
$$

where $\Delta^2 e, u \geq 0$. Then we can get the $E[\Phi(\Delta^2 e)]$ as follow

$$E(\Phi(\Delta^2 e)) = -\int_0^{+\infty} \Delta^2 e \Phi(\Delta^2 e) d\Delta^2 e$$
$$= a \times exp(-\frac{u}{\alpha_c}) - b \times exp(-\frac{u}{\alpha_r}) + u \tag{10}$$

As shown in fig.2, the the parameters u and α can be estimated by

$$u_c = min\{(d_1 + d_3)/2, (d_2 + d_4)/2\} \tag{11}$$

$$\alpha_c = \sqrt{\frac{1}{4 \times 2} \sum_4^{k=1} (d_k - u_c)^2} \tag{12}$$

where $d_1 = p_c^w - p_c^n$, $d_2 = p_c^n - p_c^e$, $d_3 = p_c^e - p_c^s$ and $d_4 = p_c^s - p_c^w$.

Observed form above, $E[\Phi(\Delta^2 e)]$ is an increasing function of u_c, u_r, and α_c, α_r. The smaller u_c, u_r, and α_c, α_r is, the better the accuracy of prediction is. So, the $E[\Phi(\Delta^2 e)]$ can well characterize local context complexity for pixel and prediction accuracy of prediction-error.

By setting a threshold λ, the entropy satisfying $E[\Phi(\Delta^2 e)] \leq \lambda$ are utilized in data embedding while the others are skipped. For a specic payload R, λ is determined as the smallest value such that it can ensure the enough payload. Thus, the embedding process starts from the prediction-error with the smallest $E[\Phi(\Delta^2 e)]$ value in the sorted row, and moves on to the next prediction-error until the last bit of data is embedded. As shown in fig.6, the left is the prediction-error of the lena image before sorting and the error margin is very high. The right is sorted by our method and the results can be clearly seen that both error and entropy are small being sorted in front. The image quality can be improved significantly, because the message is embedded in the appropriate prediction-error.
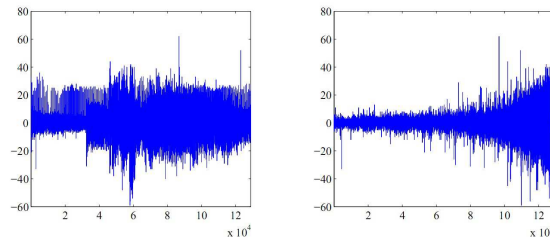

Fig. 6: the prediction-error of lena

3   Application, Experiment and Analysis

In this section, we apply second order prediction-error sorting (SOPS) algorithms to the J. Li *et al.* [11] method. It is stressed that the embedding and extraction procedures are the same with the algorithms in [11]. We just replace or add the prediction and sorting algorithm in experiments. Then frameworks of the proposed SOPS for RDH scheme are presented in fig.7. As shown in Fig.7, we first hide data into the red and red channels by taking the green one as the reference channel. When hiding data into the green one itself, the reference channel is the marked red one. The embedding procedures and extracting procedures of second
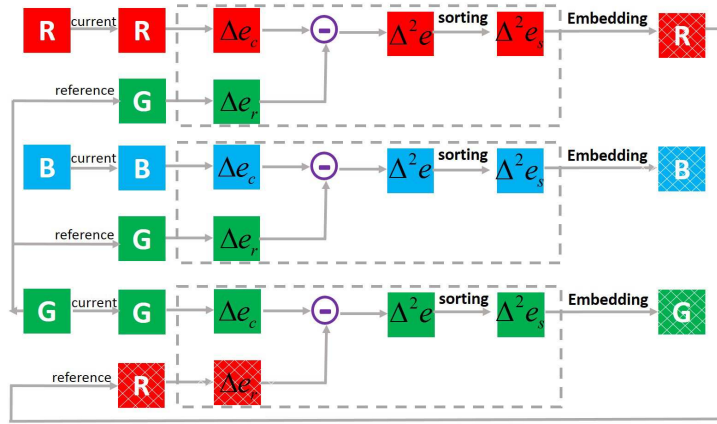

Fig. 7: The framework of data hiding for the three channels.

order perdicting-error sorting for color image are as follows:

We implemented these methods on the computer with Intel core i3 and 4GB RAM. The program developing environment is MATLAB R2011b based on Microsoft Windows 7 operating system. In the experiment, we employ four color images (refer to Fig.8) to test the performance of our proposed RDH algorithm via embedding capacitydistorting curves. In the fig.8, the first two standard images are saved in TIFF format, with size 512×512. And the two Kodak images are the first and the last ones saved in the database (http://r0k.us/graphics/kodak/), with PNG format and 512×768 in size. Our method is evaluated by comparing with the other six recent works of Li *et al.* [11], Li *et al.* [12], Sachnev *et al.* [13], Alattar [14], Hu *et al.* [15], Yang and Huang [16]. For our method, we vary the embedding from 100,000 bits to 300,000 bits or 600,000 bits with step size 50,000 bits.

Observing form fig.9, Our method is evaluated by comparing with the other six recent works. The comparison results are shown in fig.9 (a),(b),(c) and (d). According to the experimental results, one can see that the proposed method

---

Algorithm 1 The Embedding Procedures of Second Order Perdicting-error Sorting for Color Image RDH.

Input:

   The first order perdicting-error in current channel Δe_c.

   The first order perdicting-error in reference channel Δe_r.

   The embedding rate R.

Output:

1: Calculate the second order prediction-error by $\Delta^2 e = \Delta e_c - \Delta e_r$.

2: Sort the second order prediction-errors in ascending order according to their corresponding $E[\Phi(\Delta^2 e)]$. Process the prediction-errors satisfying $E[\Phi(\Delta^2 e)] \leq \lambda$ to embed the payload.

3: Hide the input data into the sorted sequence by the proposed method in J. Li *et al.* [11]. Using LSB replacement, embed the values of λ, the compressed location map size and the message size into LSBs of some first-line pixels.

4: **return** After this step, the shadow layer embedding is completed. The marked second order prediction-error sequence $\Delta^2 e' = (\Delta^2 e'_1, \cdots, \Delta^2 e'_N)$.

Algorithm 2 The Extracting Procedures of Second Order Perdicting-error Sorting for Color Image RDH.

Input:

   The marked second order prediction-error by $\Delta^2 e$.

   The first order perdicting-error in reference channel Δe_r.

Output:

1: By reading LSBs of some first-line pixels, determine the values of the values of λ, the compressed location map size and the message size.

2: Use the same prediction and scan order to obtain the marked second order prediction-error sequence $\Delta^2 e' = (\Delta^2 e'_1, \cdots, \Delta^2 e'_N)$.

3: The recovery of these perdicting-errors are implemented by the inverse mapping of the proposed method in J. Li *et al.* [11].

4: **return** Recover the original second order prediction-error sequence $\Delta^2 e = (\Delta^2 e_1, \cdots, \Delta^2 e_N)$. Then, the original shadow pixels are recovered.

---

outperforms these state-of-the-art works. Our method can provide a larger PSNR whatever the test image or capacity is. Comparing with Li *et al.* [11], experimental results show that our method provides an average increase in PSNR of 0.65dB for Lena, 3.63dB for Barbara, 2.66dB for Kodak-01 and 0.91dB for Kodak-24. our method, an average 1.96dB PSNR gains is earned compared with the Li *et al.* [11] method, and compared with Sachnev *et al.* [5], the gains of PSNR is much more higher.

4   Conclusion

In this paper, we propose a novel second order perdicting and sorting technique for reversible data hiding. Firstly, the prediction error is obtained by an inter-channel secondary prediction using the prediction errors of current channel and

Fig. 8: Test image Lena, Barbara, Kodak-01 and Kodak-24.

reference channel. When the pixels in the smooth region of image, the pixels are similar to each other and the first order perdicting-errors are close to zero. Therefore, the second order prediction-errors are also close to zero. On the other hand, when the pixels are located at rough region, the first order perdicting-errors relatively large. However, considering that all the color channels have similar edge distribution and take into account the edge information obtained from another channel. the second order prediction-errors become smaller. Experiments show that this prediction method can produce shaper second order prediction-error histogram. Then, we will introduce a novel second order perdicting-error sorting (SOPS) algorithm, which make full use of the feature of the edge information obtained from another color channel and high correlation between adjacent pixels. So it will reflect the local context complexity for pixel and prediction accuracy of prediction-error. Experimental results show that the proposed method has better results compared to the other six recent works of Li *et al*. [11], Li *et al*. [12], Sachnev *et al*. [13], Alattar [14], Hu *et al*. [15], Yang and Huang [16].

References

1. J. Tian. Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890-896,Aug. 2003.
2. J. Fridrich and M. Goljan. Lossless data embedding for all image formats. Proc. SPIE, vol. 4675, pp. 572-583, Jan. 2002.
3. C. Dragoi and D. Coltuc. Gradient Based Prediction for Reversible Watermarking by Difference Expansion. IH&MMSec14,pp.35-41,June 2014.
4. R. M. Rad, A. Attar. A predictive algorithm for multimedia data compression. Multimedia Systems. 19(2):103-115, 2013.
5. Z. Ni, Y. Shi, N. Ansari, and S. Wei. Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp.354-362,Mar. 2006.
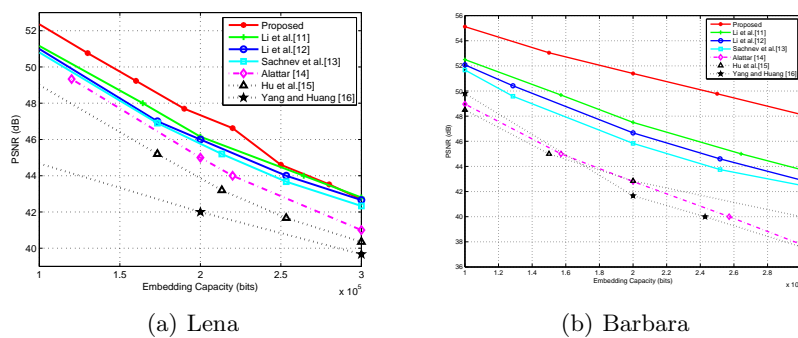
(a) Lena

(b) Barbara

Fig. 9: (a) and (b) is performance comparison between our method and six methods of Li et al.[11], Li et al.[12], Sachnev et al.[13], Alattar [14], Hu et al.[15], Yang and Huang [16].
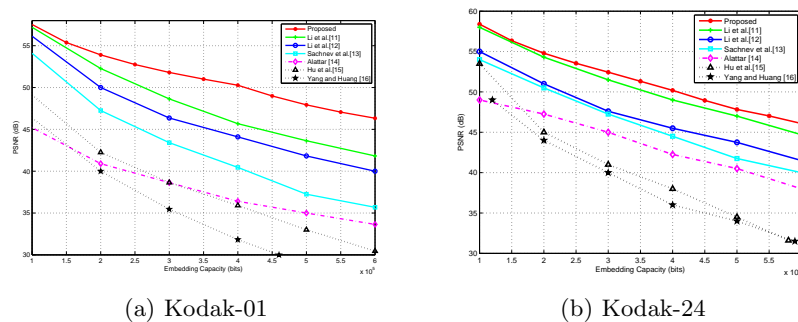

(a) Kodak-01

(b) Kodak-24

Fig. 10: (a) and (b) is performance comparison between our method and six methods of Li et al.[11], Li et al.[12], Sachnev et al.[13], Alattar [14], Hu et al.[15], Yang and Huang [16].

6. Mahsa Afsharizadeh, Majid Mohammadi. A Reversible Watermarking Prediction Based Scheme using a New Sorting and Technique. 10th International Conference on.Information Security and Cryptology (ISC),pp.98-104.2013.
7. L. H. J. Kamstra and A. M. Heijmans. Reversible data embedding into images using wavelet techniques and sorting. IEEE Trans. Image Process., vol. 14, no. 12, pp. 2082-2090, Dec. 2005.
8. W. Zhang, B. Chen, and N. Yu. Improving various reversible data hiding schemes via optimal codes for binary covers. IEEE Trans. Image Process., vol. 21, no. 6, pp. 2991-3003, Jun. 2012.
9. Athanasios Nikolaidis. Low overhead reversible data hiding for color JPEG images. Multimedia Tools and Applications, vol. 75, no. 4, pp. 1869-1881, 2016.
10. Y Tseng, H Pan. Data Hiding in 2-Color Images. IEEE Transactions on Computers, V.51 No.7, pp 873-880,2002.
11. J Li, X Li, B Yang. Reversible data hiding scheme for color image based on prediction-error expansion and cross-channel correlation. Signal Processing, vol. 93, no. 9, pp. 2748-2758, 2013.
12. J. Li, X. Li, B. Yang. PEE-based reversible watermarking for color image. International Conference on Image Processing, 2012.
13. V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Shi. Reversible watermarking algorithm using sorting and prediction. IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 7, pp. 989-999, Jul. 2009.
14. A.M. Alattar. Reversible watermark using the difference expansion of a generalized integer transform. IEEE Transactions on Image Processing 13 (8) (2004) 1147-1156.
15. Y. Hu, H. Lee, J. Li. DE-based reversible data hiding with improved overflow location map. IEEE Transactions on Circuits and Systems for Video Technology 19 (2) (2009) 250-260.
16. H. Yang, K. Hwang. Reversible data hiding for color BMP image based on block difference histogram. Proceedings of the Fourth International Conference on Ubimedia Computing (U-Media), 2011, pp. 257-260.
17. S. Lee, C. D. Yoo, and T. Kalker. Reversible image watermarking based on integer-to-integer wavelet transform. IEEE Trans. Inform. Forensics Security, vol. 2, no. 3, pp. 321-330, Sep. 2007.
18. D. M. Thodi and J. J. Rodriguez. Expansion embedding techniques for reversible watermarking. IEEE Trans. Image Process., vol. 16, no. 3, pp. 721-730, Mar. 2007.
19. X. Li, B. Yang, and T. Zeng. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. IEEE Trans. Image Process., vol. 20, no. 12, pp. 3524-3533, Dec. 2011.
20. D. Coltuc. Low distortion transform for reversible watermarking. IEEE Trans. Image Process., vol. 21, no. 1, pp. 412-417, Jan. 2012.
21. G. Giller. A Generalized Error Distribution. August 16, 2005. Available at SSRN:http://dx.doi.org/10.2139/ssrn.2265027
22. C. Dragoi and D. Coltuc. Improved rhombus interpolation for reversible watermarking by difference expansion. inProc. EUSIPCO, 2012,pp. 1688-1692.
23. H.-T. Wu and J. Huang. Reversible image watermarking on prediction errors by efficient histogram modification. Signal Process., vol. 92,no. 12, pp. 3000-3009, Dec. 2012.
24. C. Qin, C.-C. Chang, Y.-H. Huang, and L.-T. Liao. An inpaintingassisted reversible steganographic scheme using histogram shifting mechanism,IEEE Trans. Circuits Syst. Video Technol., vol. 23, no.7,pp. 1109-1118, Jul. 2013.

25. W. Zhang, X. Hu,X.Li and N. Yu. Recursive Histogram Modification: Establishing Equivalency Between Reversible Data Hiding and Lossless Data Compression. IEEE Trans. Image Process., vol. 22, no. 7, pp. 2775-2785, July. 2013.
26. W. Zhang, B. Chen, and N. Yu. Capacity-approaching codes for reversible data hiding. Proc. 13th Inf. Hiding Conf., LNCS 6958. 2011, pp. 255-269.
27. G. Schaefer, M. Stich. UCIDan uncompressed colour image database. Proceedings of the SPIE Storage and Retrieval Methods and Applications for Multimedia, San Jose, CA, USA, 2004, pp. 472-480.