2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

Optimizing Pixel Predictors Based on Self-similarities for Reversible Data Hiding

Xiaocheng Hu, Weiming Zhang*, Nenghai Yu†
School of Information Science and Technology
University of Science and Technology of China, Hefei, China
e-mail: hxc@mail.ustc.edu.cn
**e-mail: zhangwm@ustc.edu.cn*
†*e-mail: ynh@ustc.edu.cn*

***Abstract*—This paper presents a clustering and optimizing pixel prediction method for reversible data hiding, which exploits self-similarities and group structural information of image patches. Pixel predictors plays an important role for current prediction-error expansion (PEE) based reversible data hiding schemes. Instead of using a fixed or a content-adaptive predictor for each pixel independently, we first employ pixel clustering according to the structural similarities of image patches, and then for all the pixels assigned to each cluster, an optimized pixel predictor is estimated from the group context. Experimental results demonstrate that the proposed method outperforms state-of-art counterparts such as the simple rhombus neighborhood, the median edge detector, and the gradient-adjusted predictor et al.**

***Keywords*-reversible data hiding; pixel prediction; self-similarities; clustering; $l1$-norm approximation**

I. Introduction

Reversible data hiding (RDH), as a special type of information hiding technique, has received much attention from the information community [1] in the last decade. Specifically, RDH ensures not only the embedded messages shall be extracted precisely, but also the cover itself should be restored losslessly. This property is important in some special scenarios such as medical imagery, military imagery and law forensics. In these applications, the cover is too precious or too important to be damaged [2].

Classical RDH methods roughly fall into three categories. The first class follows the idea of compression-embedding framework of Fridrich [3]. In these algorithms, a two-value feature is calculated for each pixel group, the sequence is losslessly compressible and messages can be embedded in the extra space. The send class of techniques is based on difference expansion (DE) [4], [5], in which the difference of each pixel group are expanded, e.g., multiplied by 2, and thus the least significant bits (LSBs) of the differences are all-zeros and can be used for embedding. The last RDH schemes use histogram shift (HS) [6]. The histogram of one special feature(for example, gray-scale value) of the nature image is quite uneven, which implies that the histogram can be modified for embedding data.

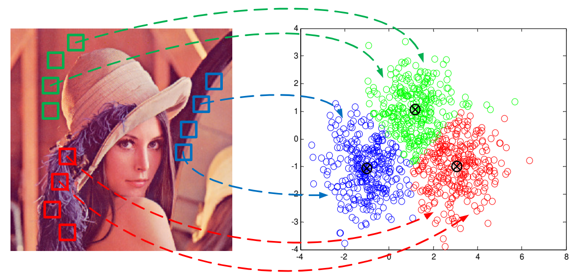


Fig. 1. Pixel clustering based on self-similarities of image patches.

In fact, by applying DE or HS to the residual parts of images instead, e.g., the prediction errors (PE) [7], better performance can be achieved. This extended method is called prediction-error expansion (PEE), which is currently a research hot spot and the most powerful technique of RDH. Recent PEE schemes first generate a PE sequence with a sharp histogram by using some prediction strategies, and then messages are reversibly embedded into the PE sequence by modifying its histogram with methods like HS or DE.

Typical prediction methods either use a fixed average model [8]–[10], or a content adaptive predictor such as the median edge detector (MED) [5], the simplified gradient-adjusted predictor (SGAP) [11] and the global optimal predictor [12]. They treats each pixel independently while structural self-similarities of non-local image patches are rarely considered. In this paper, we propose to first divide all pixels into several clusters according to the structural similarities of image patches, as shown in Fig. 1. Afterwards, for all the pixels assigned to each cluster, an optimized pixel predictor is estimated from the group context. Each pixel is predicted by the linear combination of its nearest eight neighbors, and a quad-layered embedding scheme is proposed to traverse all the pixels in the cover image.

The rest of the paper is organized as follows. Section II briefly reviews the PEE method. The proposed clustering and optimizing scheme is presented in Section III. Experimental comparison results are demonstrated in Section IV. And finally, Section V concludes this paper.

978-1-4799-5390-5/14 $31.00 © 2014 IEEE
DOI 10.1109/IIH-MSP.2014.126

II. Prediction-error expansion (PEE)

Typical PEE based schemes divide cover image pixels into different parts, while a pixel of one part is predicted by its neighboring pixels in other parts. In Sachnev et al.'s double-layered embedding method [8], all pixels are divided into two sets: the Cross set and the Dot set (see Fig. 2). Here we consider the Cross layer to illustrate the procedures.

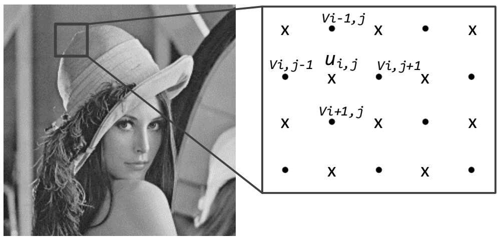


Fig. 2. Rhombus prediction pattern. The pixel value of u of the Cross set is predicted by using its four neighboring pixels in the Dot set.

As shown in Fig. 2, the Cross pixels $u_{i,j}$ in the cover image are collected into a sequence $\mathbf{u} = (u_1, u_2, \cdots, u_n)$ from left to right and from top to bottom. For each Cross pixel $u_{i,j}$, the rhombus predicted value $\hat{u}_{i,j}$ is computed by averaging its four nearest Dot pixels $\hat{u}_{i,j} = \lfloor \frac{v_{i,j-1}+v_{i+1,j}+v_{i,j+1}+v_{i-1,j}}{4} \rfloor$. Then, by subtracting the predicted value $\hat{u}_{i,j}$ from the original pixel value $u_{i,j}$, we obtain the prediction-error sequence $\mathbf{e} = (e_1, e_2, \cdots, e_n)$. Afterwards, secret data are embedded into e_i through expanding and shifting as

$$e'_i = \begin{cases} 2e_i + m, & \text{if } e_i \in [T_n, T_p] \\ e_i + T_p + 1, & \text{if } e_i \in (T_p, +\infty) \\ e_i + T_n, & \text{if } e_i \in (-\infty, T_n) \end{cases} \quad (1)$$

where $T_n < 0$ and $T_p \geq 0$ are threshold parameters, and $m \in \{0, 1\}$ is a to-be-embedded message bit. Here, the bins in $[T_n, T_p]$ are expanded to embed data, and those in $(-\infty, T_n) \cup (T_p, +\infty)$ are shifted outwards to create vacancies. Finally, each pixel value u_i is modified to $u'_i = \hat{u}_i + e'_i$.

In PEE extraction process, the original prediction-error e_i is recovered from the marked prediction-error e'_i as

$$e_i = \begin{cases} \lfloor e'_i/2 \rfloor, & \text{if } e'_i \in [2T_n, 2T_p + 1] \\ e'_i - T_p - 1, & \text{if } e'_i \in (2T_p + 1, +\infty) \\ e'_i - T_n, & \text{if } e'_i \in (-\infty, 2T_n) \end{cases} \quad (2)$$

and the embedded message bits are extracted as the LSBs of those prediction-errors $e'_i \in [2T_n, 2T_p+1]$. Finally, the cover image is restored using the recovered prediction-errors.

III. Quad-layered embedding scheme

For each pixel b in the cover image, instead of predicted by averaging its four nearest neighbors, we propose to compute its predicted value $\hat{b}$ through the linear combinations of its eight nearest neighbors $\mathbf{a} = (a_1, a_2, \cdots, a_8)$

$$\hat{b} = \mathbf{a}\mathbf{x}^T \quad (3)$$

where $\mathbf{x} = (x_1, x_2, \cdots, x_8)$ is the coefficients vector satisfies $\sum_{i=1}^{8} x_q = 1$ and $0 \leq x_q \leq 1, q = 1, 2, \cdots, 8$.

As depicted in Fig. 3, all the pixels in the cover image are divided into four sets: the Square set, the Star set, the Triangle set, and the Circle set. A pixel in each set is predicted by its eight neighbors from the other three sets. And in order to traverse all the pixels, a consecutive quad-layered embedding scheme is developed, each layer covers a type of set. Without loss of generality, we take the Square layer for instance to elaborate our embedding scheme.

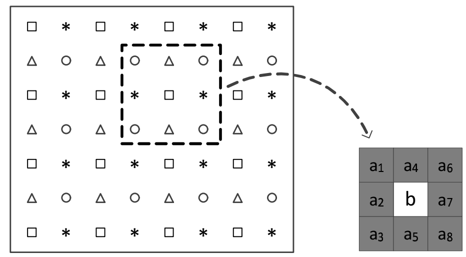


Fig. 3. Quad-layered prediction pattern. A pixel in each set is predicted by its nearest eight neighboring pixels from the other three set.

A. Optimizing pixel predictors with clustering

Firstly, for each pixel $b_i, i = 1, 2, \cdots, N_s$ in the Square set, a patch-level structural feature $\mathbf{f}_i$ is calculated form its prediction context vector $\mathbf{a}_i = (a_{i1}, a_{i2}, \cdots, a_{i8})$ as

$$\mathbf{f}_i = \mathbf{a}_i - \tfrac{a_{i1}+a_{i2}+\cdots+a_{i8}}{8} \quad (4)$$

where N_s is the total number of pixels in the Square set.

According to the extracted features $\mathbf{f}_i, i = 1, 2, \cdots, N_s$, we use the K-means clustering algorithm to divide all the pixels b_i in the Square set into K clusters as depicted in Fig. 1. Here K is a predefined parameter for the K-means algorithm. Note that the initial cluster centroid pixel indexes for the K-means algorithm are selected every S pixels from the S-th pixel, namely $S, 2S, \cdots, KS$. And the space parameter S is transmitted to the receiver side for the sake of repeating the K-means algorithm.

After that, for all the pixels $b_j, j = c_1, c_2, \cdots, c_{N_c}$ assigned to a specified cluster, a content adaptive pixel predictor is estimated by optimizing the following problem

$$\begin{aligned} &\text{minimize} \quad |A\mathbf{x} - \mathbf{b}|_1 \\ &\text{subject to} \quad \sum_{q=1}^{8} x_q = 1, \; 0 \leq x_q \leq 1 \end{aligned} \quad (5)$$

here N_c is the total number of pixels dispatched to the cluster, and the matrix A and vector $\mathbf{b}$ is given by

$$A = \begin{vmatrix} a_{c_1 1} & a_{c_1 2} & \cdots & a_{c_1 8} \\ a_{c_2 1} & a_{c_2 2} & \cdots & a_{c_2 8} \\ \vdots & \vdots & \ddots & \vdots \\ a_{c_{N_c} 1} & a_{c_{N_c} 2} & \cdots & a_{c_{N_c} 8} \end{vmatrix} \qquad \mathbf{b} = \begin{vmatrix} b_{c_1} \\ b_{c_2} \\ \vdots \\ b_{c_{N_c}} \end{vmatrix} \quad (6)$$

In (5), the l_1-norm is used rather than the l_2-norm due to the fact that we aims to optimize the eight coefficients $\mathbf{x} = (x_1, x_2, \cdots, x_8)$ according to most of the pixels in the cluster, while the l_1-norm is more robust to outliers.

When the coefficients vector $\mathbf{x}$ is estimated by solving (5), the prediction-error e_j is calculated as

$$e_j = b_j - \lfloor \mathbf{a}_j \mathbf{x}^\mathbf{T} \rfloor \tag{7}$$

Then we embedded messages by modifying e_j to e'_j using the expanding and shifting techniques described in (1), thus the original pixels $b_j, j = c_1, c_2, \cdots, c_{N_c}$ are modified to

$$b'_j = e'_j + \lfloor \mathbf{a}_j \mathbf{x}^\mathbf{T} \rfloor \tag{8}$$

B. Compression of the optimized coefficients

The optimized coefficients $\mathbf{x} = (x_1, x_2, \cdots, x_8)$ has to be transmitted to the receiver side to recover the prediction value $\hat{b}_j$. As the l_1-norm used in (5), embedding modifications on vector $\mathbf{b}$ result slight changes for the optimized coefficients $\mathbf{x}$. Using the modified vector $\mathbf{b}'$ after embedding, we can optimize (5) again and get a revised coefficients vector $\mathbf{x}' = (x'_1, x'_2, \cdots, x'_8)$. If we restrain the precision of the coefficients to d decimal places, a coefficients residual vector $\mathbf{r}$ can be derived as

$$\mathbf{r} = (r_1, r_2, \cdots, r_8) = \lfloor \mathbf{x} * 10^d \rfloor - \lfloor \mathbf{x}' * 10^d \rfloor \tag{9}$$

Next, we use a variable-length coding scheme to record $\mathbf{r}$ for each cluster. Specifically, we first check that the maximum absolute value of $\mathbf{r}$ is less then 2^M, otherwise we don't embed messages for this cluster. Here M is a predefined bit length. Then for each coefficient residual r_i, if $r_i == 0$, we just add a bit "0" to the coded stream, or else $M + 2$ bits are added. The $M + 2$ bits consists of a bit "1", a sign indicator bit to record the sign of r_i, and M bits to record the absolute value of r_i. And finally, the coded coefficients residual streams for all the K clusters are concatenated together to be transmitted to the receiver side.

C. Embedding and extracting

This section describes the proposed quad-layered embedding process in detail. The embedding and extraction processes are nearly the same for the four sets: Square set, Star set, Triangle set, and Circle set. Next we discuss the Square layer to elaborate the procedures.

Embedding process

1) Extract features f_i from all pixels b_i in the Square set.
2) Select K pixel indexes every S spaced as initial cluster centroids, run the K-means algorithm to divide all pixels in the Square set into K clusters.
3) For $k = 1 : K$
   a) Collect all pixels b_j and vectors $\mathbf{a}_j$ in the cluster to form matrix A and vector $\mathbf{b}$ in (6).
   b) Estimate the optimal coefficients $\mathbf{x}$ by solving (5), then calculate prediction errors e_j using (7).
   c) Embed messages into the prediction errors e_j by (1), then calculate the coefficients residual vector $\mathbf{r}_k$ by (9) and generate the coded residual bit stream $\mathbf{rs}_k$ as discussed in Section III-B.
4) Finally, concatenate all the coded residual bit streams $\mathbf{rs}_k, k = 1, 2, \cdots, K$ together and embedded them into the LSBs of some preserved pixels.

Extracting process

1) Extract all the coded residual bit streams $\mathbf{rs}_k, k = 1, 2, \cdots, K$ from the LSBs of the preserved pixels.
2) Extract features f_i from all pixels b_i in the Square set.
3) Select K pixel indexes every S spaced as initial cluster centroids, run the K-means algorithm to divide all pixels in the Square set into K clusters.
4) For $k = 1 : K$
   a) Collect all pixels b'_j and vectors $\mathbf{a}_j$ in the cluster to form matrix A and vector $\mathbf{b}'$ in (6).
   b) Estimate the revised pixel predictor coefficients $\mathbf{x}' = (x'_1, x'_2, \cdots, x'_8)$ by solving (5), and together with the decoded residual vector $\mathbf{r}_k$, recover the original optimal coefficients $\mathbf{x}$.
   c) Calculate prediction errors e'_j using (8), then extract messages and recover the original prediction errors e_j through (2), then original pixel values b_j can be restored losslessly through (7).

IV. SIMULATIONS

First, the proposed clustering and optimizing scheme is compared to other four prediction methods, namely, MED [5], Sachnev et al.'s method [8], simplified gradient-adjusted predictor (SGAP) [11], and checkerboard based prediction (CBP) [10]. Test grayscale images are shown in Fig. 4. Table I records the prediction comparison results in terms of MAE (mean absolute error), which is defined by

$$MAE = \frac{1}{n} \times \sum_{i=1}^{n} |b_i - \hat{b}_i| \tag{10}$$

Here we only predict the Square set pixels for comparison. It can be seen from Table I that our proposed method provides the best prediction accuracy among all the competitors.

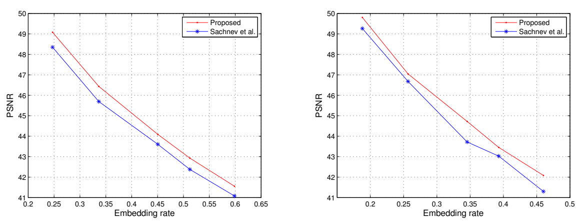
Image	Lena	Barbara	Goldhill	Pepper
MED [5]	4.4168	9.2978	5.5387	5.3624
SGAP [11]	4.0362	8.8888	5.6568	4.8651
Sachnev et al. [8]	3.2330	7.4485	4.5441	4.1278
CBP [10]	3.1837	7.0533	4.5579	3.9778
Proposed	3.0041	5.2685	4.2508	3.6046

Table I
PREDICTION ACCURACY (MAE) COMPARISON RESULTS.

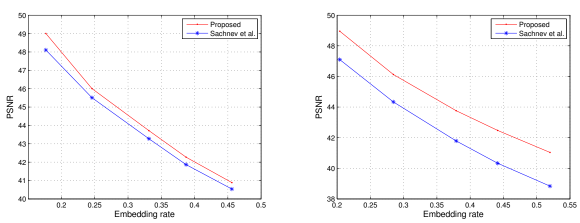
Next, to demonstrate that prediction accuracy directly influence the embedding performance of PEE based RDH methods, we compare our proposed quad-layered embedding scheme with Sachnev et al.'s method [8]. For our proposed scheme, the cluster number K is set to 25, the decimal place d and bit length M in Section III-B are set with $d = 2, M = 3$. Embedding performance comparison results for various embedding rates are demonstrated by Fig. 5.

(a) Lena (b) Barbara (c) Goldhill (d) Peppers

Fig. 4. Test images.



(a) Lena (b) Barbara



(c) Goldhill (d) Peppers

Fig. 5. Comparisons in terms of rate-distortion performance.

Observing from Fig. 5, for the Lena and the Goldhill image, our proposed scheme earns 0.5dB higher PSNR than Sachnev et al.'s method on average. And for the Barbara and especially the Peppers image, the gains of PSNR increase to 2dB on average. As shown in Fig. 4, the Barbara and the Peppers image exhibit better structural self-similarities, and thus better embedding performance gains are obtained.

V. Conclusion

This paper presents a clustering and optimizing prediction method for reversible data hiding, which exploits self-similarities and group structural information of image patches. And a quad-layered embedding scheme is proposed accordingly to traverse all the pixels in the cover image. Compared to other fixed or content adaptive pixel predictors, our proposed method offers the best prediction accuracy. Experimental results imply that structural self-similarities of intra image patches undoubtedly benefits pixel prediction, so structural self-similarities across multiple images or even among a image dataset might be more helpful to reversible data hiding. This could be our future direction to work on.

Acknowledgment

This work was supported in part by the Natural Science Foundation of China under Grant 61170234 and Grant 60803155, by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06030601, and by the Funding of Science and Technology on Information Assurance Laboratory under Grant KJ-13-003.

References

[1] R. Caldelli, F. Filippini, and R. Bacarelli, Reversible watermarking techniques: An overview and a classification. *EURASIP J.Inf. Secur.*, 2010.

[2] J. Feng et al., Reversible watermarking: Current status and key issues, *Int. J. Netw. Security*, vol. 2, no. 3, pp. 161-171, May 2006.

[3] J. Fridrich and M. Goljan, Lossless data embedding for all image formats, In *Proc. SPIE*, vol. 4675, pp. 572-583, Jan. 2002.

[4] J. Tian, Reversible data embedding using a difference expansion, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890-896, Aug. 2003.

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

[6] 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.

[7] F. Peng, X. Li, and B. Yang, Adaptive reversible data hiding scheme based on integer transform, *Signal Process.*, vol. 92, no. 1, pp. 54-62, Jan. 2012.

[8] 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.

[9] H. C. Huang and F. C. Chang, Hierarchy-Based Reversible Data Hiding,*Expert Systems with Applications*, vol. 40, no. 1, pp. 34-43, Jan. 2013.

[10] R. R, X. Hu, X. Li, and J. Guo, A unified data embedding and scrambling method, *IEEE Trans. on Image Processing*. Vol. 23, no. 4, pp. 1463-1475, April 2014.

[11] D. Coltuc. Improved embedding for prediction-based reversible water- marking. *IEEE Trans. Inf. Forens. Security*, 6(3):873-882, Sep. 2011.

[12] S. L. Lin, C. F. Huang, M. H. Liou, and C. Y. Chen, Improving Histogram-based Reversible Information Hiding by an Optimal Weight-based Prediction Scheme, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 4, No. 1, pp. 19-33, January 2013.