Reversible Image Watermarking Based on Neural Network and Parity Property

Rongrong Ni, H. D. Cheng, Yao Zhao, Zhitong Zhang and Rui Liu

Abstract Reversible watermarking can recover the original cover after watermark extraction, which is an important technique in the applications requiring high image quality. In this paper, a novel image reversible watermarking is proposed based on neural network and parity property. The retesting strategy utilizing the parity detection increases the capacity of the algorithm. Furthermore, the neural network is considered to calculate the prediction errors. Experimental results show that this algorithm can obtain higher capacity and preserve good visual quality.

Keywords Reversible watermarking · Neural network · Parity property · Retesting strategy

1 Introduction

Reversible watermarking can completely restore the original digital contents after data extraction. For this characteristic, reversible watermarking is very useful for some applications where the availability of the original data is essential, such as military image processing and medical image sharing.

Early reversible watermarking algorithms mainly focus on lossless compression until the difference expansion algorithm is proposed by Tian [1]. The method divides the image into pairs of pixels and uses each legitimate pair for hiding one bit of information. It has high embedding capacity and high quality, and becomes the basic idea of some reversible watermarking methods. Later, prediction error

R. Ni (✉) · Y. Zhao · Z. Zhang · R. Liu
Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
e-mail: rrni@bjtu.edu.cn

H. D. Cheng
Department of Computer Science, Utah State University, Logan, UT, USA

J. J. (Jong Hyuk) Park et al. (eds.), *Multimedia and Ubiquitous Engineering*, Lecture Notes in Electrical Engineering 240, DOI: 10.1007/978-94-007-6738-6_36,

© Springer Science+Business Media Dordrecht(Outside the USA) 2013

expansion (PEE) method is proposed by Thodi and Rodriguez [2]. Their method uses PEE to embed data, and suggests incorporating expansion embedding with histogram shifting to reduce the location map. Then, several PEE-based methods have been proposed [3–6]. In [6], Sachnev et al. propose a method which combines sorting and two-pass-testing with prediction error expansion method. The algorithm has higher capacity and lower distortions than most of other existing reversible watermarking methods.

In this paper, a novel image reversible watermarking is proposed based on neural network and parity property. Because the real embedded data is not always identical with the testing bit, some ambiguous pixel cells are generated. A retesting strategy utilizing the parity detection activates the capacity of the ambiguous pixel cells. As a result, the capacity is increased. Furthermore, considering the global feature, the neural network is used to predict the prediction errors. The experimental results show that the proposed algorithm can obtain higher capacity and preserve good visual quality.

2 Proposed Algorithm Based on Neural Network and Parity Detection

In the proposed algorithm, all pixels of the image are divided into two sets: the "Cross" set and the "Dot" set (Fig. 1) as suggested in [6]. The watermark bits are embedded in the "Cross" set first, and then embedded in the "Dot" set.

2.1 Prediction Based on Neural Network

During "Cross" embedding, the "Cross" set is used for embedding data while the "Dot" set works as the reference signals. And vice versa.

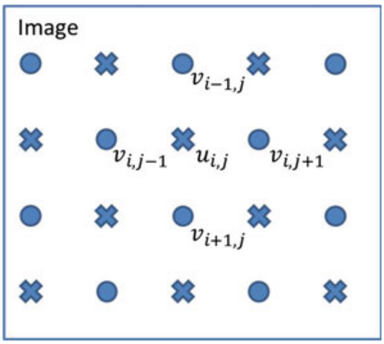

Fig. 1 "*Cross*" set and "*Dot*" set

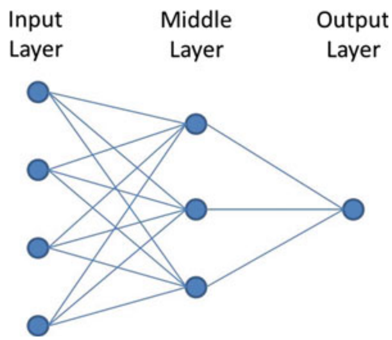

Fig. 2 Structure of neural network

A center pixel of a cell is predicted by the four neighboring pixels. In this paper, neural network is used to predict the pixel values considering the global feature. Since four pixels in the neighboring region are utilized to calculate the prediction values, a neural network with four inputs is designed here. As shown in Fig. 2, the input layer has four neurons and the middle layer is created with three neurons. The output layer has one neuron which refers to the central pixel value.

After the construction of the neural network, the corresponding weights and parameters can be determined by training. Considering the global influence and generalization, a great number of pixel cells from many natural images are input to the neural network for obtaining a common model of prediction. This model can be shared by encoder and decoder in advance. Thus, the prediction value $u'_{i,j} = \lfloor nnpredict\ (v_{i,j-1}, v_{i-1,j}, v_{i,j+1}, v_{i+1,j}) \rfloor$. Where, *nnpredict*(.) is the prediction model based on neural network.

2.2 Data Embedding and Extraction

The combination of difference expansion and histogram shifting method [2] is also utilized in this paper.

If the prediction error $e_{i,j} = u_{i,j} - u'_{i,j}$ inside the region $[T_n, T_p]$, $e_{i,j}$ is expanded to $E_{i,j} = 2 \times e_{i,j} + \mathrm{b}$. T_n is the negative threshold and T_p is the positive threshold. Otherwise, the pixel does not carry any data and the prediction error is simply shifted. That is,

$$E_{i,j} = \begin{cases} 2 \times e_{i,j} + b & \text{if } e_{i,j} \in [T_n, T_p] \\ e_{i,j} + T_p + 1 & \text{if } e_{i,j} > T_p \text{ and } T_p \geq 0 \\ e_{i,j} + T_n & \text{if } e_{i,j} < T_n \text{ and } T_n < 0 \end{cases} \tag{1}$$

The watermarked value is computed by $U_{i,j} = u'_{i,j} + E_{i,j}$.

During extraction, if $E_{i,j} \in [2T_n, 2T_p + 1]$, the watermark can be extracted. Otherwise, shifting is used to recover the image. That is,

$$e_{i,j} = \begin{cases} \lfloor E_{i,j}/2 \rfloor & \text{if } E_{i,j} \in [2T_n, 2T_p + 1] \\ E_{i,j} - T_p - 1 & \text{if } E_{i,j} > 2T_p + 1 \\ E_{i,j} - T_n & \text{if } E_{i,j} < 2T_n \end{cases} \tag{2}$$

Then, $u_{i,j} = u'_{i,j} + e_{i,j}$.

2.3 Improved Classification Using Parity Property

To ensure $U_{i,j}$ without overflow or underflow problems, two-pass-testing [6] is used here. If a pixel can be modified twice based on Eq. (1), it belongs to Class A; if the pixel is modifiable once owing to overflow or underflow errors during the second embedding test, it belongs to Class B; and if the pixel cannot be modified even once, the pixel belongs to Class C. During the testing process, bit "1" is used as an embedding bit for positive prediction errors, and bit "0" is for negative prediction errors. The locations of Class B and Class C are marked in a location map, which is also embedded with the payload.

In the decode phase, use once-embedding-test to distinguish Class A, and Class B (or Class C). And further discriminate Class B and Class C using the location map. However, some pixel cells belonging to Class B will be misclassified to Class A if the actually embedded bit does not coincide with the testing bit.

We utilize the parity characteristic and retesting strategy to activate the capacity of Class B. After once-embedding-test during the extraction phase, the pixel cells are assigned into two parts: Part one contains the cells without overflow or underflow, and Part two contains the overflow or underflow cells. As a result, Part one is the set consisting of Class A and partial Class B, while Part two is the set containing Class C and part of Class B. It is obvious that the elements of Class B which are attributed in Part one are problem pixel cells. Since they will cause the wrong localization in the location map, these problem pixel cells should be identified further.

For the cells in Part one, a retesting detection is designed to distinguish the ambiguous cells belonging to Class B. As for the expandable pixel cells, $U_{i,j} = u'_{i,j} + e_{i,j} = u'_{i,j} + 2e_{i,j} + \mathrm{b}$. Thus, $U_{i,j} - u'_{i,j} = 2e_{i,j} + \mathrm{b}$. Due to $2e_{i,j}$ is an even number, $\mathrm{b} = \mathrm{LSB}(U_{i,j} - u'_{i,j})$, here $\mathrm{LSB}(x)$ means the LSB of x. For the positive prediction errors, if $U_{i,j} - u'_{i,j}$ is an even number, the embedded bit is not consistent with the testing bit. Thus, add one to the pixel value $U_{i,j}$ and retest the corresponding prediction error using the testing bit "1". For the negative prediction errors, if $U_{i,j} - u'_{i,j}$ is odd, subtract one from the pixel value $U_{i,j}$ and retest the corresponding prediction error using the testing bit "0". If the retesting result shows the pixel value is overflow, it belongs to Part two. Otherwise, it still belongs to Part one. After the retesting, Part one only contains Class A, and Part two contains Class B and Class C. Further classification is conducted to distinguish Class B and Class C with the help of the location map.

3 Encoder and Decoder

3.1 Data Embedding

We first embed data in "Cross" set, then embed in "Dot" set. For recovering data, threshold values T_n(7 bits), T_p(7 bits), payload size $|P_{cross}|$(17 bits) or payload size $|P_{dot}|$(17 bits), and the length of location map (7 bits) should be known first. We will embed these 38 bits into the first 38 pixels' LSB. The original 38 LSB should be recorded with the payload. The "Cross" embedding method is designed as follows:

Step 1: Calculate the prediction errors. For each pixel $u_{i,j}$, compute the prediction value and the corresponding prediction error $e_{i,j}$ based on the common neural network.

Step 2: Sort the prediction errors. For each pixel $u_{i,j}$, compute the variance $Var_{i,j}$ of the four neighbor pixels which is used as the sorting parameter. Skip the first 38 pixels. Sort the pixel cells according to the ascending order $\{Var_{i,j}\}$ to produce a sorted row of prediction errors e_{sort}.

Step 3: Determine the threshold. According to the two-pass-testing, all pixels are classified in one of classes A, B and C. Although the shiftable pixels can be modified, they cannot carry watermark bits. Only the expandable pixels in Class A and Class B are capable of carrying data. Let set of expandable pixels in class A be EA. Let set of expandable pixels in class B be EB.

In the sorted vector e_{sort}, create the location map L. If a pixel belongs to Class B, the corresponding element in the location map is marked as "0"; while if the pixel belongs to Class C, it is marked as "1". If $|P_{cross}| \leq |EA| + |EB| - |L| - 38$ and $|EA| \geq |L|$ are satisfied, the to-be-embedded bits can be successfully embedded. Otherwise, increase the threshold T_p or decrease T_n, and repeat Step 3.

Step 4: Embed data. The location map L, the true payload P_{cross}, and the first 38 LSBs will be embedded in the image by using the embedding method described in Sect. 2.2. The location map L is first embedded in Class A. The elements belonging to Class A and Class B are all used to improve the capacity. Use the auxiliary data to modify the first 38 LSB values of the pixels by simple binary replacement. If the last to-be-embedded bit is processed, the "Cross" embedding phase is completed.

After 4 steps, the "Cross" embedding process is finished. The "Dot" embedding scheme uses the modified pixels from the "Cross" set for computing the predicted values. The original pixels from the "Dot" set are used for embedding data, and the embedding procedure is similar to the "Cross" embedding. After the "Dot" embedding, the watermarked image is obtained.

3.2 Data Extraction

Double decoding scheme is the inverse of the double encoding scheme. We only describe the "Cross" decoding method.

Step 1: Calculate the prediction values. For each pixel $U_{i,j}$, compute the prediction value based on the common neural network. Then, the prediction errors $E_{i,j}$ are obtained afterwards.

Step 2: Sort the prediction errors. Skip the first 38 pixels. Sort the pixels according to $Var_{i,j}$ to get the set of sorted prediction errors E_{sort}. Read the first 38 LSB values to recover the values of T_n, T_p, payload size P_{cross}, and the length of location map.

Step 3: Extract the watermark. Skip the first 38 sorted cells. Test every pixel cell to classify it into Class A, Class B and Class C according to Sect. 2.3. Extract location map from Class A firstly. Further classification is conducted to distinguish Class B and Class C based on the location map. Then, extract data from Class A and Class B, meanwhile recover the original prediction errors using the method in Sect. 2.2. The extracted data is the cascading of the true payload, and the 38 LSBs.

Step 4: Restore the original image. Computer the original pixel values based on $u_{i,j} = u'_{i,j} + e_{i,j}$.

Step 5: Recover the rest pixels. Replace the first 38 LSB values of the pixels with the extracted 38 LSBs.

When the "Dot" and "Cross" decoding are both finished, the entire watermark is obtained and the original image is restored.

4 Experimental Results

Several 8-bit gray images "Lena", "Baboon" and "Plane" with size 512 × 512 are used in the experiments. Figure 3 shows the watermarked images with payload 50000 bits:

Figure 4 shows the performances of Capacity versus Visual quality in terms of payload and Peak Signal-to-Noise Ratio (PSNR). The horizontal axis represents the capacity in terms of bpp (bits per pixel). The vertical axis represents the corresponding PSNR. The results show that our method has both high visual

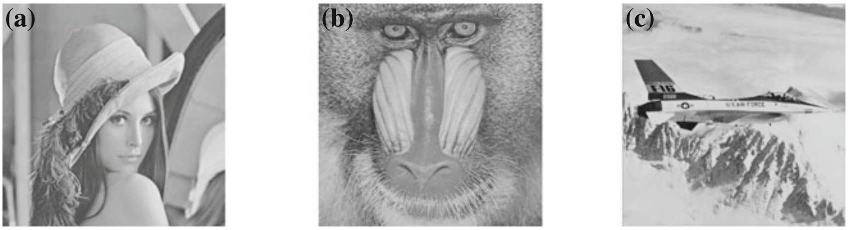

Fig. 3 The watermarked *gray* images. **a** Lena (PSNR = 51.4 dB). **b** Baboon (PSNR = 43.0 dB). **c** Plane (PSNR = 53.5 dB)

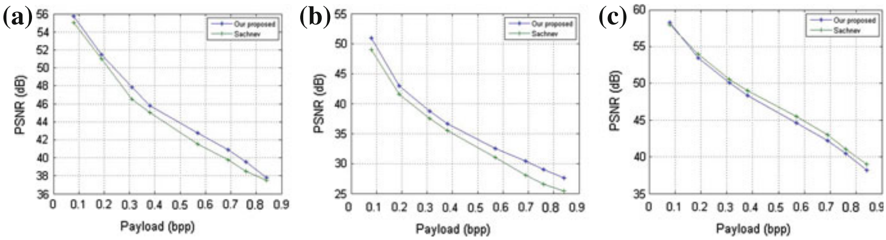

Fig. 4 Capacity versus PSNR for testing images. **a** Results for Lena. **b** Results for Baboon. **c** Results for Plane

quality and high capacity. Compared with [6], our method can achieve better results for "Lena" and "Baboon", and get comparable result for "Plane". The reason is that the neural network is not trained enough.

5 Conclusions

A high capacity image reversible watermarking based on neural network and parity property is proposed. A retesting strategy utilizing the parity detection activates the capacity of the ambiguous pixel cells. In addition, the prediction errors are obtained by using the neural network to consider the global feature. The experimental results show that this algorithm can obtain higher capacity and preserve good visual quality. In the future, we will further research and discuss the effectiveness of the neural network.

Acknowledgment This work was supported in part by 973 Program (2011CB302204), National Natural Science Funds for Distinguished Young Scholar (61025013), National NSF of China (61073159, 61272355), Sino-Singapore JRP (2010DFA11010), Fundamental Research Funds for the Central Universities (2012JBM042).

References

1. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol 8:890–896
2. Thodi DM, Rodriguez JJ (2007) Expansion embedding techniques for reversible watermarking. IEEE Trans Image Process 3:721–730
3. Tsai WL, Yeh CM, Chang CC (2009) Reversible data hiding based on histogram modification of pixel differences. IEEE Trans Circuits Syst Video Technol 6:906–910
4. Tsai PY, Hu C, Yeh HL (2009) Reversible image hiding scheme using predictive coding and histogram shifting. IEEE Signal Process Mag 6:1129–1143
5. Luo LZ, Chen N, Zeng X, Xiong Z (2010) Reversible image watermarking using interpolation technique. IEEE Trans Inf Forensics Secur 1:187–193
6. Sachnev V, Kim HJ, Nam J, Shi YQ, Suresh S (2009) Reversible watermarking algorithm using sorting and prediction. IEEE Trans Circuits Syst Video Technol 7:989–999