



# Article Trace Concealment Histogram-Shifting-Based Reversible Data Hiding with Improved Skipping Embedding and High-Precision Edge Predictor (ChinaMFS 2022)

Hui Shi <sup>1,2,3,\*</sup>, Baoyue Hu<sup>1</sup>, Jianing Geng<sup>1</sup>, Yonggong Ren<sup>1,\*</sup> and Mingchu Li<sup>2</sup>

- <sup>1</sup> School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China
- <sup>2</sup> School of Software Technology, Dalian University of Technology, Dalian 116029, China
- <sup>3</sup> Dalian Yongjia Electronic Technology Co., Ltd., Dalian 116029, China
- \* Correspondence: shihui\_jiayou@126.com (H.S.); ryg@lnnu.edu.cn (Y.R.); Tel.: +86-13604119127 (H.S.)

**Abstract:** Reversible data hiding (RDH) is a special class of steganography, in which the cover image can be perfectly recovered upon the extraction of the secret data. However, most image-based RDH scheme sfocus on improving capacity–distortion performance. In this paper, we propose a novel RDH scheme which not only effectively conceals the traces left by HS but also improves capacity–distortion performance. First, high-precision edge predictor LS-ET (Least Square predictor with Edge Type) is proposed, and the predictor divides pixels into five types, i.e., weak edge, horizontal edge, vertical edge, positive diagonal edge, and negative diagonal edge. Different types of target pixels utilize different training pixels with stronger local consistency to improve accuracy. Then, a novel prediction-based histogram-shifting (HS) framework is designed to conceal embedding traces in the stego images. Finally, we improve both the data-coding method and the skipping embedding strategy to improve the image quality. Experimental results demonstrate that the capacity–distortion performance of the proposed scheme outperforms the other trace concealment schemes and is comparable to the state-of-the-art schemes utilizing sorting technique, multiple histogram modification, and excellent LS-based predictors. Moreover, it can conceal the embedding traces left by the traditional HS schemes to a certain extent, reducing the risk of being steganalyzed.

**Keywords:** embedding trace concealment; LS-ET predictor; histogram shifting; skipping embedding; reversible data hiding

MSC: 68

# 1. Introduction

Data hiding realizes copyright protection and content authentication by hiding secret data into digital media covers, which is an important technique in the field of information security. Due to the requirement of high fidelity of the media cover on some occasions, any little distortion is intolerable, and reversible data hiding (RDH) technology comes into being. RDH offers the ability of the exact recovery of both secret message and original cover media without any distortion. Because of this feature of RDH, many algorithms have been proposed in the last few decades. Existing RDH algorithms based on image space domain mainly include lossless compression, difference extension (DE), and histogram shifting (HS).

The RDH algorithm based on HS was originally proposed by Ni et al. [1], which created embedding space by manipulating the image histograms. However, the embedding capacity depended on the number of pixels belonging to the peak bin in the histogram. A few years later, the idea of prediction was used in HS algorithms to increase the embedding capacity. In this case, the prediction errors histogram is used, which is sharply distributed



Citation: Shi, H.; Hu, B.; Geng, J.; Ren, Y.; Li, M. Trace Concealment Histogram-Shifting-Based Reversible Data Hiding with Improved Skipping Embedding and High-Precision Edge Predictor (ChinaMFS 2022). *Mathematics* 2022, 10, 4249. https:// doi.org/10.3390/math10224249

Academic Editors: Zhongyun Hua and Yushu Zhang

Received: 13 October 2022 Accepted: 11 November 2022 Published: 13 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). and peaks near zero. The algorithm in [2] used median edge detector (MED) and modified prediction error values of 0 and -1 in order to embed data.

Generally, the accuracy of the predictor affects the embedding capacity and the quality of the stego image, and many predictors have been proposed in recent years. Gradient adjusted predictor (GAP) was designed by Fallahpour [3]; it used more neighboring pixels to obtain the prediction values, so it was more suitable for images with complex textures. An improvement over this algorithm was proposed by Coltuc [4]; a simplified GAP (SGAP) was designed to reduce the time complexity. Sachnev et al. [5] designed a rhombus predictor, calculating the average of the four neighboring pixels as the prediction values, which showed the highest prediction accuracy among all fixed predictors.

However, the weights of the above-mentioned predictors are fixed. In recent years, adaptive predictors have been proposed to improve the capacity–distortion performance of HS algorithms. Dragoi and Coltuc [6] proposed an HS algorithm on the basis of local prediction (LP). The prediction value was weighted on a square block centered on the target pixel, and weights were adaptively calculated through the least square (LS) method. Hwang et al. [7] proposed a predictor using least absolute shrinkage and selection operator (LASSO). The LASSO predictor penalized and removed the pixels in the prediction context that did not affect the target pixels. Wang et al. [8] used weighted prediction and watermarking simulation (LP-WPWS) to improve LP. It increased the number of trained pixels in LP predictor, divided the prediction context into watermarked region and original region, and applied different weights on the pixels in the two regions to further improve the accuracy of prediction, but it did not take into account the characteristics of the pixel itself in the prediction process. Wang et al. [9] proposed the ridge regression predictor to solve the overfitting problem of LS and was able to obtain smaller prediction context.

In addition to designing high precision predictors, another way to improve the ability of the HS algorithm is to investigate new embedding methods. Wang et al. [10] proposed a novel RDH general framework using multiple histogram shifting (MH\_RDH), which formulated the rate allocation among multiple histograms as a rate distortion optimization and solved it using an evolutionary algorithms. Experimental results showed that the method could considerably increase the payload. Kim et al. [11] used a pair of extreme predictions to generate two skewed histograms. Only the pixels from the peak and the short tail were used for embedding in the skewed histogram, which decreased the distortion from the lesser number of pixels being shifted. Choi et al. [12] proposed a novel HS method by skipping some pixels between the peak and the zero, which decreased the shifting distortion.

Recently, some algorithms have increased the occurrence of '0' in the secret binary stream by coding secret data, thereby improving the performance of HS algorithms. Yang et al. [13] proposed a message sparse representation to decrease the embedding distortion. It would decrease the number of '1's in the binary stream. In 2020, Xie et al. [14] proposed a signed-digit representation, and Peng et al. [15] proposed a remainder-storage-based EMD (RSBEMD) method. Those representation methods would not cause message expansion, viz., the size of the message was enlarged. However, the coded message stream would appear as new bits, such as '-1', '2', etc., which would cause additional shifting distortion.

However, these algorithms would leave quite obvious traces after embedding, suffering the risks of exposing the data-hiding action. In particular, there is a steganalysis algorithm [16] specifically designed for HS. Attackers can easily detect the existence of secret data from stego images, then intercept those images and decipher secret data. Various previous works have also noticed the detectability issue of RDH [17,18]. In 2020, Dong et al. [19] proposed a method that could effectively conceal the traces left by HS. However, it caused a serious loss of image quality.

In this work, an embedding trace concealment histogram-shifting-based reversible data-hiding algorithm is proposed. In addition, we propose a novel HS framework using

the improved data-coding method and the improved skipping embedding strategy. The histograms of the stego images maintain a shape similar to that of the original histograms. Therefore, the embedding traces are concealing, to a certain extent. Moreover, we first propose a LS-ET (Least Square predictor with Edge Type), in which the local texture characteristic of the target pixel can be fully utilized. The predictor uses only half-context of the target pixel for the flexibility of the embedding method and low computation, but the prediction accuracy is comparable to the full-context predictors. Meanwhile, the algorithm has a good capacity–distortion performance.

The main contributions of this paper are as follows:

(1) It proposes a novel HS framework, which combines the improved data coding method and the improved skipping embedding method. It aims to conceal the embedding traces, still enabling the stego image histograms to conform to the Laplacian distribution.

(2) In the framework, two distortions are considered in the HS-based data-hiding algorithm: expanding distortion and shifting distortion, resulting in the algorithm's good capacity–distortion performance.

(3) It proposes LS-ET to improve prediction accuracy. Unlike traditional LS-based predictors that use fixed neighbors as the training pixels, the proposed LS-ET predictor utilizes unfixed pixels with stronger local consistency, with the target pixel as training pixels.

(4) It improves the message sparse representation method; the improved data-coding method is contrary to the judgment condition of the original method, which can obtain higher stego image quality when secret data is in image form.

(5) It improves the skipping embedding method to achieve the aim of concealing the embedding traces by adaptively determining the embedding position and skipping position on the basis of the prediction error histogram and the given payload.

The remainder of this paper is organized as follows. Section 2 analyzes the embedding traces left by traditional histogram-shifting algorithms. In Section 3, the proposed LS-ET predictor is described in detail. The framework of the proposed schemes is given in Section 4. Section 5 presents the experimental results and comparisons, and Section 6 concludes this paper.

#### 2. The Analysis of Embedding Traces Left by HS Algorithms

For prediction-based HS algorithms, the following steps achieve data hiding. First, calculate the prediction value p(i,j) for each pixel x(i,j) of the original image using a predictor. Then, compute the prediction error e(i,j) by

$$e(i,j) = x(i,j) - p(i,j) \tag{1}$$

Finally, the prediction error is modified according to the secret data, and the modified prediction error e'(i,j) is calculated by

$$e'(i,j) = \begin{cases} e(i,j) + a - 1 & \text{if } e(i,j) < a \\ 2 \times e(i,j) - data & \text{if } a \le e(i,j) \le 0 \\ 2 \times e(i,j) - 1 + data & \text{if } 0 < e(i,j) \le b \\ e(i,j) + b & \text{if } b < e(i,j) \end{cases}$$
(2)

where data is 1 bit of data to be embedded in the secret binary stream. The prediction errors that belong to [a,b] are called expandable errors, which will be expanded to embed 1 secret data bit. The distortion produced is called expanding distortion. It can be seen that '1's in the secret binary stream are the cause of expanding distortion. Other prediction errors will be shifted to the left or the right to make room for expansion, and the caused distortion is called shifting distortion.

As a known fact, the prediction error can be effectively modeled with a Laplacian distribution with zero mean [20]. Assume that the length of the secret data is the same as the number of expandable errors; Figure 1 gives an example of the HS method. As shown in Figure 1, the original histogram conforms to a Laplacian distribution, while the after-embedding histogram becomes irregular. There are some consecutive bins that are lower in height than the bins on either side of them in the after-embedding histogram. Such obvious embedding traces are easily noticed by attackers. In addition, the secret data is usually encrypted before embedding, and bits '0' and '1' are equally distributed. In this case, those adjacent bins with secret data have almost the same height, which is considered to be an embedding trace (this phenomenon is termed 'flat ground'). With this trace, attackers can not only detect the existence of the secret data from the stego images but also can even decipher the content of secret data. Wang et al. [16] designed an effective steganalysis method for HS based on whether the histogram of the given image exhibits the 'flat ground' phenomenon. The detection accuracy of the steganalysis method at the payload = 0.1 bpp is up to 97.67%. In Section 4, we propose a novel HS framework to conceal the embedding trace mentioned above.



**Figure 1.** Histograms shifting: (**a**) Before embedding; (**b**) After embedding with expandable errors; and (**c**) After embedding with expandable errors [-2,2].

#### 3. Proposed LS-ET Predictor

The least square predictor (LS) is an adaptive prediction technique that exploits local image patterns. The predictor employed an adapted linear combination of the prediction coefficients on the basis of training set of causal neighbors in order to compute the prediction value using

$$p = w(1) \times p(i, j-1) + w(2) \times p(i-1, j) + w(3) \times p(i-1, j-1) + w(4) \times p(i-1, j+1)$$
(3)

where w(k) is the corresponding weights of neighbors. If we use the previous *m* causal pixels for the adaptation of the *k*-th order predictor coefficients, the vector of adapted coefficients *w* is calculated by solving the overdetermined linear system of equations using

$$w = \left(A^T A\right)^{-1} A^T y \tag{4}$$

where the size of the training matrix *A* is  $m \times k$ , and *y* is an  $m \times 1$  vector containing the previous m causal pixels.

Traditional LS-based predictors use fixed neighbors as the training pixels. Figure 2 shows the training pixels used by LP [6] and ridge regression predictor [9], and the shadow pixels are the training pixels of the target pixel x(i,j). Training pixels are fixed, which means that they are not fully utilizing the target pixel's characteristics.

x( <i>i</i> -2, <i>j</i> -2)	x( <i>i</i> -2, <i>j</i> -1)	x(i=2,j)	x( <i>i</i> -2 <i>j</i> +1)	x( <i>i</i> -2, <i>j</i> +2)
x(i-1,j-2)	x(i=1,j=1)	x(i=1,j)	x(i-1,j+1)	x(i-1,j+2)
x(ij-2)	x(i,j=1)	x(i,j)	x(ij+1)	x(ij+2)
x(i+1,j-2)	x(i+1,j-1)	x(i+1,j)	x( <i>i</i> +1, <i>j</i> +1)	x(i+1,j+2)
x(i+2,j-2)	x(i+2,j-1)	x(i+2,j)	x(i+2,j+1)	x(i+2,j+2)
		(a)		

**Figure 2.** Training pixels of the target pixel x(i,j): (a) LP and (b) Ridge regression predictor.

According to the prediction rule of the LS predictor, the prediction accuracy depends on the applicability of the trained prediction weights against the target pixel. Therefore, the training pixel should have strong local consistency with the target pixel. The proposed LS-ET predictor uses different training pixels based on the characteristic of the target pixel.

Firstly, calculate four direction gradients of the target pixel x(i,j): horizontal gradient  $d_h$ , vertical gradient  $d_v$ , positive diagonal gradient  $d_{pd}$ , and negative diagonal gradient  $d_{np}$  by Equations (5)–(8):

$$d_{h} = \frac{\left[|x(i,j-2) - x(i,j-1)| + 0.5 \times (|x(i,j-3) - x(i,j-2)| + |x(i-1,j-1) - x(i-1,j)| + |x(i-1,j-1) - x(i-1,j+1)| + |x(i-1,j-2) - x(i-1,j-1)| + |x(i-1,j-1) - x(i,j+2)|)\right]}{(1+5 \times 0.5)}$$
(5)  
$$d_{v} = \frac{\left[|x(i-2,j) - x(i-1,j)| + 0.5 \times (|x(i-3,j) - x(i-2,j)| + |x(i-1,j-1) - x(i,j-1)| + |x(i-2,j-1) - x(i-1,j-1)| + |x(i-2,j+1) - x(i-1,j+1)|)\right]}{(1+4 \times 0.5)}$$
(6)

$$d_{pd} = \frac{\left[|x(i-2,j-2) - x(i-1,j-1)| + 0.5 \times (|x(i-3,j-3) - x(i-2,j-2)| + |x(i-2,j-1) - x(i-1,j)| + |x(i-1,j-2) - x(i,j-1)|)\right]}{(1+3\times0.5)}$$
(7)

$$d_{nd} = \frac{\left[|x(i-2,j+2) - x(i-1,j+1)| + 0.5 \times (|x(i-3,j+3) - x(i-2,j+2)| + |x(i-1,j) - x(i,j-1)| + |x(i-2,j+1) - x(i-1,j)|)\right]}{(1+3\times0.5)}$$
(8)

According to the four directional gradients of the target pixel x(i,j), the pixel is divided into five types, namely weak edge, horizontal edge, vertical edge, positive diagonal edge, and negative diagonal edge, by Equations (9) and (10), where *Th* is the threshold for dividing types, which is discussed in Section 5.1.

weak edge, if 
$$\max(d_h, d_v, d_{pd}, d_{nd}) - \min(d_h, d_v, d_{pd}, d_{nd}) < Th$$
 (9)

$$\begin{cases} \text{horizontal edge,} & \text{if } \min\left(d_h, d_v, d_{pd}, d_{nd}\right) = d_h \\ \text{vertical edge,} & \text{if } \min\left(d_h, d_v, d_{pd}, d_{nd}\right) = d_v \\ \text{positive diagonal edge,} & \text{if } \min\left(d_h, d_v, d_{pd}, d_{nd}\right) = d_{pd} \\ \text{negative diagonal edge,} & \text{if } \min\left(d_h, d_v, d_{pd}, d_{nd}\right) = d_{nd} \end{cases}$$
(10)

In LS-ET, different training pixels are selected according to the gradient values, which have similar edge characteristics with the target pixel.

Consider a  $B \times B$  block centered on the target pixel as the prediction block. The optimal *B* will be determined by experiments, which will be presented in Section 5.1. In the following, consider B = 7 as an example.

Type I: The target pixel belongs to the weak edge; then, in the  $B \times B$  prediction block, select the top left  $\left|\frac{B \times B}{2}\right|$  neighboring pixels of the target pixel as the training pixels.

Type II: The target pixel belongs to the horizontal edge; then, in the  $B \times B$  prediction block, select the left-side horizontal direction  $\left\lfloor \frac{B}{2} \right\rfloor$  neighboring pixels of the target pixel as the training pixels.

Type III: The target pixel belongs to the vertical edge; then, in the  $B \times B$  prediction block, select  $\left\lfloor \frac{B}{2} \right\rfloor$  neighboring pixels located vertically above the target pixel as the training pixels.

Type IV: The target pixel belongs to the positive diagonal edge; then, in the  $B \times B$  prediction block, select  $\left\lfloor \frac{B}{2} \right\rfloor$  neighboring pixels in the positive diagonal direction above the target pixel as the training pixels.

Type V: The target pixel belongs to the negative diagonal edge; then, in the  $B \times B$  prediction block, select  $\left\lfloor \frac{B}{2} \right\rfloor$  neighboring pixels in the negative diagonal direction above the target pixel as the training pixels.

Limited to the skipping embedding method used in this paper—which needs to generate the entire image's prediction error histogram first to determine the skipping position and then perform the embedding operation—our scheme cannot embed in a pixel immediately after predicting it, as in other schemes. This means that the prediction context cannot contain pixels of embedded data in the prediction process. Similarly, in the extraction and recovery process, the prediction context also needs to be the original pixel, so as to ensure the complete extraction of the data and the lossless recovery of the image. Therefore, the predictor uses only half-context as the prediction context, namely x(i - 1, j - 1), x(i - 1, j), x(i - 1, j + 1), and x(i, j - 1), not 8 neighboring (full-context).

Figure 3a–e respectively illustrate the training pixel and prediction context of target pixel which belongs to the weak edge, horizontal edge, vertical edge, positive diagonal edge, and negative diagonal edge. The shadow pixels are the training pixels of the target pixel, and the 4 neighboring pixels in the green border are the prediction context of the target pixel.

Matrix Y is the training matrix composed of training pixels. Using Equation (11), the predicted matrix  $\mathbf{P}^{\mathbf{Y}}$  of  $\mathbf{Y}$  is obtained:

$$\mathbf{P}^{\mathbf{Y}} = \mathbf{C}^{\mathbf{Y}} \times \mathbf{W} \tag{11}$$

In Equation (11),  $C^{Y}$  is the prediction matrix, which is composed of the prediction context of all training pixels, and **W** is the weight matrix.

For example, if the target pixel x(i,j) belongs to the horizontal edge, the training matrix **Y** and the prediction matrix **C**<sup>Y</sup> are as shown in Equations (12) and (13):

$$\mathbf{Y} = [x(i, j-3), \ x(i, j-2), \ x(i, j-1)]^{\mathrm{T}}$$
(12)

$$\mathbf{C}^{\mathbf{Y}} = \begin{bmatrix} 1, \ x(i-1,j-4), \ x(i-1,j-3), \ x(i-1,j-2), \ x(i,j-4) \\ 1, \ x(i-1,j-3), \ x(i-1,j-2), \ x(i-1,j-1), \ x(i,j-3) \\ 1, \ x(i-1,j-2), \ x(i-1,j-1), \ x(i-1,j), \ x(i,j-2) \end{bmatrix}$$
(13)

The optimized weight matrix **W** is obtained by least square approximation, and the loss function of the least square method is shown in Equation (14):

$$loss(\mathbf{W}) = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_{k=1}^{4} \left( \mathbf{Y}(k) - \mathbf{P}^{\mathbf{Y}}(k) \right)^{2}$$
(14)

Calculate the predicted value p(i,j) using the weight matrix **W** and the prediction context matrix **C**, as shown in Equation (15):

$$p(i,j) = \mathbf{C} \times \mathbf{W} \tag{15}$$



where the prediction context matrix **C** consists of 4 neighboring pixels and the constant 1, as shown in Equation (16):

$$\mathbf{C} = [1, x(i-1, j-1), x(i-1, j), x(i-1, j+1), x(i, j-1)]$$
(16)

**Figure 3.** Training pixels and prediction context of the target pixel x(i,j): (a) Weak edge; (b) Horizontal edge; (c) Vertical edge; (d) Positive diagonal edge; and (e) Negative diagonal edge.

## 4. Proposed Framework for Concealing Embedding Traces

Almost all state-of-the-art prediction-based HS algorithms focus only on improving the capacity-distortion performance. However, the obvious embedding traces are left in the after-embedding histogram. Thus, we propose an HS framework toward concealing the embedding traces by keeping the after-embedding histogram in a similar shape to the original histogram. The framework combines the improved data-coding method and the improved skipping embedding strategy. It can flexibly select the predictors and encoding methods to obtain the high quality stego images, while reducing the risk of being detected by attackers. The detailed process is as follows.

# 4.1. Data Hiding

The data-hiding process consists of three parts: (1) prediction based on proposed LS-ET; (2) improved data coding; and (3) improved histogram-shifting embedding with skipping. Figure 4 illustrates the data-hiding process of the framework.



Figure 4. The data-hiding process.

#### 4.1.1. Prediction

Scan the pixels in a raster scan order and calculate the predicted value p by the proposed LS-ET predictor. Any prediction method can be used in this section. We use the proposed LS-ET predictor in the subsequent experiments to improve prediction accuracy.

Compute the prediction error by:

$$e(i,j) = x(i,j) - p(i,j)$$
 (17)

Then, generate a prediction error histogram of the original image.

#### 4.1.2. Improved Data Coding

Yang et al. [13] proposed a message sparse representation method to reduce the number of '1's in the secret data, which will lead to the expanding distortion. We make a simple improvement to the sparse representation for higher image quality. The detailed procedure for encoding the data is as follows.

Set the parameter r ( $r \ge 0$ ) to represent the sparse rate. The coding rate is defined as

$$R = \frac{L}{CL}$$
(18)

where *L* is the length of the original data, *CL* is the length of the sparse representation data, i.e., coded data. Experimental results show that the parameter *r* is inversely proportional to coding rate *R*.

1

Initialize pos1 = 0 and pos2 = 0. pos1 is used to record the last cover symbol that has been coded, and pos2 is used to record the number of message bits that have been coded.

Read the secret data (*Data*) for the (pos2 + 1)-th bit, i.e., the *Data*(pos2 + 1). There are two coding cases according to the value of *Data*(pos2 + 1).

Case 1. If Data(pos2 + 1) = 1, set  $pos1 = pos1 + 2^r$  and pos2 = pos2 + 1, and one-bit Data(pos2 + 1) is coded. The coded data (*CData*) are as follows:

$$CData(pos1 - 2^r + 1: pos1) = 0$$
(19)

Case 2. If Data(pos2 + 1) = 0, read the next *r* bits Data(pos2 + 2:pos2 + r + 1), which can be represented by a decimal integer belonging to  $[0, 2^r - 1]$ , denoted by  $(Data(pos2 + 2:pos2 + r + 1))_{dec}$ . Set  $pos1 = pos1 + (Data(pos2 + 2:pos2 + r + 1))_{dec} + 1$ , pos2 = pos2 + r + 1, and flip CData(pos1) from '0' to '1'. Thus, r + 1 bits Data(pos2 + 2:pos2 + r + 1) are coded. The coded data CData are as follows:

$$CData(pos1 - (Data(pos2 - r + 1 : pos2))_{dec} : pos1 - 1) = 0$$

$$CData(pos1) = 1$$
(20)

Repeat the above judgment until all the secret data Data is coded as CData.

It is observed that secret data may be in the form of images, with a high probability of a consecutive '0' or '1' in the data. If the content of Data(pos2 + 2:pos2 + r + 1) contains more '0's in Case 2, it means that the value of  $(Data(pos2 + 2:pos2 + r + 1))_{dec}$  is smaller, that is, the length of the coded data is shorter, and fewer pixels will be used for embedding data in the embedding process. If that happens, it is more likely that the value of Data(pos2 + 1) is

0. Therefore, we set the condition for Case2 to Data(pos2 + 1) = 0, which is contrary to the condition in the original coding method of [13]. In Yang's method [13], if Data(pos2 + 1) = 0, execute the above Case 1; and if Data(pos2 + 1) = 1, execute Case 2. The experimental results in Section 5 demonstrate the effectiveness of this change. The improvement in this paper can obtain higher stego image quality where secret data is in image form. In addition, it does not affect the experimental results where the '1's and '0's are randomly distributed in the secret data.

Consider Data = [0,0,1,0,1,0,1] as a simple example to show the sparse representation process. Set the parameter r = 2. Initialize pos1 = 0 and pos2 = 0.

Firstly, read the secret data bit Data(1) = 0 and the next *r* bits Data(2:3) = [0,1], which is interpreted as a decimal integer  $(0,1)_{dec} = 1$ . Set  $pos1 = 0 + (01)_{dec} + 1 = 2$  and pos2 = 0 + 2 + 1 = 3. Calculate the CData(1:2) = [0,1] by Equation (20).

Next, read the secret data bit Data(4) = 0 and the next *r* bits Data(5:6) = [1,0], which is interpreted as a decimal integer  $(1,0)_{dec} = 2$ . Set  $pos1 = 2 + (10)_{dec} + 1=5$  and pos2 = 6. Calculate the CData(3:5) = [0,0,1] by Equation (20).

Then, read the secret data bit Data(7) = 1. Set pos1 = 5 + 4 = 9 and pos2 = 6 + 1 = 7. Calculate the CData(6:9) = [0,0,0,0] by Equation (19).

Finally, get the coded data CData = [0,1,0,0,1,0,0,0,0]. The coded data CData is the to-be-embedded data in the subsequent embedding process.

#### 4.1.3. Improved Skipping Embedding

Skipping embedding is a novel HS method that decreases the shifting distortion to improve image quality. The embedding process for Choi's method [12] is outlined as follows. Firstly, set the peak and minimum (zero) histogram indices as the (IdxP, IdxZ). Then, set the skipping index as the IdxSkip, shifting the pixels between [IdxP + IdxSkip + 1, IdxZ - 1]. Finally, modify the pixels of the IdxP position for the data hiding. If it embeds a 0, the pixel does not change; if it embeds a 1, the pixel of the IdxP is increased to IdxP + IdxSkip + 1.

Choi's method [12] considered the skipping position to obtain higher image quality, but it did not focus on the deformation of the histogram. In this paper, we improve the skipping embedding method with the aim of concealing the embedding traces of existing HS methods.

First, the embedding position and skipping position are determined in the prediction error histogram of the original image.

Search the embedding position  $e_e$  starting from the sides and progressing to the middle of the histogram; stop searching when the following is satisfied:

$$h(e_e) \ge \text{length}(CData) \tag{21}$$

where h(x) denotes the number of pixels whose prediction errors value is x,  $e_e$  denotes embedding position, i.e., the value of expandable error, and length (*CData*) is the length of the given payload.

Because the prediction error histogram conforms to the Laplacian distribution, that is, the number of prediction errors near both sides of the histogram is small, the number of prediction errors near the middle of the histogram will be large. The embedding position is searched from both sides to the middle of the histogram to shorten the distance between the embedding position and the skipping position and to decrease the expanding distortion in the subsequent embedding process.

Search the skipping position  $e_s$  starting from the embedding position  $e_e$  and progressing to one side of the histogram. If  $e_e$  is negative, search to the left side of the histogram, otherwise, to the right side. Stop searching when the following is satisfied

$$h(e_s) \le \text{length}(CData_1) \tag{22}$$

where length (*CData*\_1) is the number of '1's in the given payload.

Then, modify the prediction error according to the coded data cdata by

$$e'(i,j) = \begin{cases} e_e & \text{if } e(i,j) = e_e \& cdata = 0\\ e_s & \text{if } e(i,j) = e_e \& cdata = 1\\ e(i,j) + 1 & \text{if } e(i,j) \ge e_s \& e_e < e_s\\ e(i,j) - 1 & \text{if } e(i,j) \le e_s \& e_e > e_s\\ e(i,j) & \text{else} \end{cases}$$
(23)

For the prediction errors which fulfill  $e(i,j) = e_e$ , if the *cdata* is '0', no shifting; otherwise, shift errors to the skipping position. For the prediction errors that fulfill  $e(i,j) \ge e_s > e_e$  or  $e(i,j) \le e_s < e_e$ , shift errors to the left or the right to make room. Repeat the above until the secret data is fully embedded.

Compared with the original skipping embedding method [12], which uses only the peak bin for embedding and the skipping position is fixed, the improved skipping embedding method carefully designs the embedding position and the skipping position selection. The embedding position and skipping position are determined adaptively according to the prediction error histogram and the given payload to achieve the aim of concealing the embedding traces, ensuring that the after-embedding histogram still maintains the general shape as the original histogram, avoiding the 'flat ground' phenomenon and, thus, being able to resist steganalysis to a certain extent, especially for those that are HS-based, while also focusing on reducing the distortion generated in the embedding process to improve image quality.

Figure 5 demonstrates the comparison of the traditional HS methods with the improved HS method that combines the improved data-coding method and the improved skipping embedding strategy. Compared with traditional HS schemes where all prediction errors between the embedding position and the zero bin need to be shifted, the prediction errors in the proposed scheme between the embedding position and the skipping position are not modified. Hence, the proposed scheme causes less shifting distortion. Because of the improved data coding, the number of prediction errors that are skipped is increased, which further highlights the advantage of skipping embedding in reduce shifting distortion. Meanwhile, the after-embedding histogram maintains a similar shape to the original histogram.



Figure 5. The comparison of the traditional HS methods with the proposed framework.

With the predicted value p and the modified prediction error e', the stego pixel x' can be obtained by:

$$x'(i,j) = p(i,j) + e'(i,j)$$
(24)

It is worth mentioning that there are several rows and columns outside the image as reference pixels in the prediction process, which are excluded from secret data embedding. In order to ensure blind extraction, side information is embedded into the first row by LSB replacement, and the least significant bits in the first few pixels are reserved and appended to the secret binary stream. Since overflows or underflows may occur after embedding, we use location map *LM* to record those overflow and underflow locations and process the image by:

$$x'(i,j) = \begin{cases} 255 - (x'(i,j) - 255) & \text{if } x'(i,j) > 255\\ -x'(i,j) & \text{if } x'(i,j) < 0\\ x'(i,j) & \text{else} \end{cases}$$
(25)

If one pixel is modified to 0 or 255, we label it with '1'; otherwise, we label it with '0'. In addition, *LM* is losslessly compressed into *CLM* using the algorithm in [21] to reduce its size. In this paper, side information includes:

- 1. The length of side information: *L<sub>side</sub>*.
- 2. The number of embedding times: *ET*.
- 3. All embedding positions  $\{e_e\}$  and skipping positions  $\{e_s\}$ .
- 4. Size of the original secret data.
- 5. The length of CLM:  $L_{CLM}$ .
- 6. The compressed location map: *CLM*.

Finally, the stego image I' is generated.

#### 4.2. Data Extraction and Image Recovery

For a received stego image I', the LSBs of the first  $L_{aux}$  pixels are extracted to retrieve the side information. According to the side information, LM is decompressed to restore the pixels. Figure 6 illustrates the data extraction and image recovery process.



Figure 6. The data extraction and image recovery process.

The predicted value *p* for each pixel is obtained by using the proposed LS-ET predictor as in the hiding process.

The prediction error e'(i,j) is obtained by Equation (26):

$$e'(i,j) = x'(i,j) - p(i,j).$$
 (26)

According to the received embedding position  $e_e$  and the skipping position  $e_s$ , a bit of data cdata that was hidden in the pixel can be extracted by

$$cdata = \begin{cases} 0 & \text{if } e'(i,j) = e_e \\ 1 & \text{if } e'(i,j) = e_s \end{cases}$$
(27)

The original prediction error e(i,j) is obtained by

$$e(i,j) = \begin{cases} e_e & \text{if } e'(i,j) = e_s \\ e'(i,j) - 1 & \text{if } e'(i,j) > e_s \& e_e < e_s \\ e'(i,j) + 1 & \text{if } e'(i,j) < e_s \& e_e > e_s \\ e'(i,j) & \text{else} \end{cases}$$
(28)

The original pixel x(i,j) can be recovered by

$$x(i,j) = p(i,j) + e(i,j)$$
 (29)

After performing the above processes for all pixels in the stego image, the data *CData* hidden in the image is extracted and the original image *I* is recovered.

Decode the extracted data *CData* to obtain the original secret data *Data*. The detailed procedure for decoding is as follows.

Initialize pos1' = 0 and pos2' = 0. pos1' is used to record the last cover symbol that has been decoded, and pos2' is used to record the number of message bits that have been decoded.

Read the secret data bits  $CData(pos2' + 1:pos2' + 2^r)$ , and obtain the decoded data in two cases:

Case 1. If Cdata( $pos2' + 1:pos2' + 2^r$ ) = 0, set pos1' = pos1' + 1,  $pos2' = pos2' + 2^r$ . The decoded data Data are obtained by

$$Data(pos1') = 0. (30)$$

Case 2. If there is a '1' in the  $CData(pos2 + 1:pos2' + 2^r)$ , count the number of consecutive '0's until finding the first '1', regarded as *i*. Then set pos1' = pos1' + r + 1 and pos2' = pos2' + i + 1, and the decoded data Data are obtained by

$$\begin{cases} Data(pos1' - r : pos1 - length((i)_{bin})) = 0\\ Data(pos1' - length((i)_{bin}) + 1 : pos1') = (i)_{bin} \end{cases}$$
(31)

where  $(\cdot)_{bin}$  is a function that converts decimal to binary. All original secret data *Data* are decoded by repeating above procedure.

The original image *I* and the original data *Data* are recovered completely losslessly.

#### 5. Experimental Results and Analysis

5.1. Determination of Parameters

5.1.1. Parameters for LS-ET Prediction

Parameters involved in the proposed LS-ET include the prediction block size *B* and the threshold *Th* for dividing pixel types.

We set prediction block size ranges to be  $B \in \{5,7,9,13\}$  and  $Th \in [15,25]$  respectively, which are sufficient in practice.

Figure 7 lists the *PSNR* (peak-signal-to-noise ratio) under different parameters for image 'Lena' with embedding rates of 0.1 bpp to 0.6 bpp. It is observed that a  $9 \times 9$  size block best exploits the local characteristics with as few reference pixels as possible. It is shown that the block size of B = 9 and threshold Th = 21 achieve the desirable performance. Such results can be similarly observed for other test images. Therefore, B = 9 and Th = 21 are adopted as preferable parameters in our scheme, which will be demonstrated in the next experiments.

51.55

51.5

51.45

51.

51.35

51.3

51.25

51.

51.15 L

45.5

 $4^{\circ}$ 

43.5

43 L 15

PSNR(dB)

PSNR(dB)



**Figure 7.** PSNR of different parameters for Lena: (a) *PSNR* under 0.1 bpp; (b) *PSNR* under 0.2 bpp; (c) *PSNR* under 0.3 bpp; (d) *PSNR* under 0.4 bpp; (e) *PSNR* under 0.5 bpp; and (f) *PSNR* under 0.6 bpp.

It is known that *Th* is an important threshold for pixel classification, so the value of *Th* can affect prediction accuracy, and the prediction accuracy should vary slowly with *Th*. When *Th* takes different values, the shape of the prediction error histogram changes accordingly, and it may change the embedding or skipping position in the embedding process, which is adaptively selected according to the shape of the histogram. However, if there is a slight difference in the choice of two positions, even if only a bin difference, it may have a great impact on the stego image quality. Thus, some of the sharp changes in *PSNR* in Figure 7 are caused by changes in the embedding position or skipping position.

## 5.1.2. Parameter r for Data Coding

The parameter *r* represents the sparse rate, which will directly affect the coding rate and coding effect. Figure 8 shows the effect of *r* on the capacity–distortion performance of the algorithm, where  $r \in [1,5]$ . As can be seen from Figure 8, the algorithm performance is acceptable when *r* values are 1, 2, or 3. If the value of *r* is too large, the coded data is too long. Although the number of '1's in the coding data is small, it produces a large amount of shifting distortion and loses great image quality, even the data cannot be fully embedded in the image.

At r = 1 and r = 2, the algorithm performance is similar in the 'Lena', 'Airplane', and 'Boat' images. However, the image quality of the complex image 'Baboon' is deteriorated at r = 2; the reason is that the prediction error histogram of the complex image is flatter (the number of pixels per bin is relatively small). If the coding data is too long, it will require a lot of bins as the expandable bin, resulting in a lot of shifting distortion. Thus, the image quality of complex images is more sensitive to the change of r. Moreover, the smaller the r, the better the capacity–distortion performance of the algorithm.



**Figure 8.** The effect of *r* on the capacity–distortion performance: (**a**) Lena; (**b**) Baboon; (**c**) Airplane; and (**d**) Boat.

Figure 9 shows the prediction error distributions for 'Lena' at r = 1, 2, and 3 under different embedding rates (*ER*) of 0.1 bpp, 0.4 bpp, and 0.7 bpp, respectively. As can be seen, the three prediction error histograms all conform to the Laplacian distribution. Furthermore, the larger the r, the better the performance of concealing the embedding traces. When r = 3, since the number of '1's in the coded data is very small, there is no significant change in either bin in the embedding process, so the after-embedding histogram maintains a similar shape to the original histogram, even if the embedding capacity is very large.



**Figure 9.** The prediction error distributions for Lena at r = 1, 2, 3: (**a**) under 0.1 bpp; (**b**) under 0.4 bpp; and (**c**) under 0.7 bpp.

Considering all of the above, the algorithm uses r = 2 as the preferable parameter and will be demonstrated in the next experiments.

#### 5.2. Analysis of the Simple Improvement on Data Coding

Table 1 shows the comparisons of the proportion of the original data and the coded data. Herein, the original data are generated randomly, resulting in a uniform distribution of '1's and '0's. For the original data, the proportions of '0's and '1's are each approximately

50%. After coding, the proportion of '0's is up to approximately 84.6%, and the proportion of '1's is reduced to approximately 15.4%. The proportion of '1's after coding decreased by 34.6% compared with that before coding. The decrease in the number of '1's will result in the improvement in image quality with the decrease of embedding distortion.

	Original Data			Coded Data		Reduction in the
Number of Bits	Proportion of '0's	Proportion of '1's	Number of Bits	Proportion of '0's	Proportion of '1's	Number of '1's
100,000	49.79%	50.22%	162,293	84.57%	15.43%	25,178
200,000	49.97%	50.04%	325,984	84.68%	15.32%	50,139
300,000	49.94%	50.06%	487,393	84.61%	15.39%	75,170
400,000	50.04%	49.96%	646,512	84.49%	15.51%	99,566
500,000	50.01%	49.99%	811,958	84.60%	15.40%	124,908

Table 1. Comparison between the original data and the coded data.

Figure 10 shows the comparisons of the *bpp-PSNR* performance between the sparse coding in [13] and the improved sparse coding. Figure 10a,b are the *bpp-PSNR* performance comparisons when the secret data is image and pseudo random number, respectively. The secret image uses the classic watermark image 'flower', as shown in Figure 10c, and the original image uses 'Lena'. It is observed that the proposed algorithm with simple improvement on the sparse representation has better *PSNR* performance than that of [13]. The *PSNR* of the proposed algorithm is increased by approximately 4dB. This is because the probability of consecutive '0's in the image data is too high, so that the length of the data coded by the original sparse representation is too long. When the secret data is a randomly distributed in a '0' and '1' binary stream, the *bpp-PSNR* performances of the two algorithms are basically the same, as shown in Figure 10b.



**Figure 10.** Comparisons between the sparse coding in [13] and the improved sparse coding: (**a**) secret data is in image form; (**b**) secret data is generated randomly; and (**c**) flower.

### 5.3. Anti-Steganalysis of the Proposed Framework

In order to demonstrate the trace concealment performance of our proposed framework, it is compared with the traditional HS algorithm and skipping HS. The experiments are conducted on the Bossbase1.01 database. Consider the widely used test image Lena and Baboon as examples; Figure 11 illustrates the results with different embedding rates (*ER*) of 0.1 bpp, 0.4 bpp, and 0.7 bpp, respectively.

As can be seen from Figure 11, the blue line shows the hypothesis Laplacian model. The prediction error histogram of the original image is smooth, and the distribution of the prediction error can be well-fitted with Laplacian model. However, the prediction error histograms of the traditional HS algorithm and the skipping HS algorithm become more irregular, with greatly deviation from the hypothesis Laplacian model. At low embedding rates, the histograms can barely maintain a shape similar to the original histogram. However, at higher embedding rates, the middle of the histogram, which should exhibit an obvious peak, becomes a large area of flatness ('flat ground'). The embedding traces can be obviously identified, so it is difficult to ensure the security of the data hidden in the image. Instead, the prediction error histogram of our proposed trace concealment algorithm is relatively smooth, which resembles the original histogram. Even though the prediction error histogram inevitably becomes flatter after data embedding, it still preserves the bellshape resembling the Laplacian distribution, thus achieving the purpose of concealing the embedding traces. The trace concealment effect in Baboon is not as good as that in Lena because the number of adjacent prediction errors in the prediction error histograms for such complex textured images is not much different; thus, it is more difficult to keep the histogram smooth after hiding data, but the proposed trace concealment algorithm is relatively better than the other two algorithms.

Next, demonstrate the performance of the proposed framework against steganalysis for the HS-based method.

Wang et al. [16] designed an effective steganalysis method for HS, which shows excellent steganalysis ability when 'flat ground' phenomenon occurs in the after-embedding histogram and when the histogram conforms to the Laplacian distribution.

Table 2 illustrates the identification error rates of the steganalysis method [16] for traditional HS algorithm, skipping HS algorithm, and proposed algorithm with different embedding rates. The general Bossbase1.01 database is employed in the experiment, and 1000 images with size  $512 \times 512$  are randomly selected from an amount of 10,000 images. The identification error rate is the proportion of stego images that cannot be identified as hidden by the steganalysis method after hiding the data in all the test images. A high identification error rate indicates that the proposed algorithm is well resistant to steganalysis. As one can see from Table 2, the resisting steganalysis performance of proposed trace concealment algorithm is superior to the traditional HS algorithm and skipping HS algorithm.



(a1)



(**b1**)

Figure 11. Cont.



(c1)



(d1)



Figure 11. Cont.



Figure 11. Comparison of the after-embedding prediction error histograms with proposed trace concealment algorithm, traditional HS algorithm, and the skipping HS algorithm: (a1) Original image; (b1) Stego images by the proposed (ER = 0.1 bpp, PSNR = 51.33 dB); (c1) Stego images by the proposed (ER = 0.4 bpp, PSNR = 45.26 dB); (d1) Stego images by the proposed (ER = 0.7 bpp, PSNR = 39.29 dB); (a2) Histogram of (a1); (b2) Histogram of (b1); (c2) Histogram of (c1); (d2) Histogram of (d1); (a3) Original image; (b3) Stego images by traditional HS (ER = 0.1 bpp, PSNR = 52.38 dB); (c3) Stego images by traditional HS (ER = 0.4 bpp, PSNR = 45.80 dB); (d3) Stego images by traditional HS (ER = 0.7 bpp, PSNR = 39.09 dB); (a4) Histogram of (a3); (b4) Histogram of (b3); (c4) Histogram of (c3); (d4) Histogram of (d3); (a5) Original image; (b5) Stego images by skipping HS (ER = 0.1 bpp, PSNR = 51.44 dB); (c5) Stego images by skipping HS (ER = 0.4 bpp, PSNR = 43.67 dB); (d5) Stego images by skipping HS (ER = 0.7 bpp, PSNR = 40.17 dB); (a6) Histogram of (a5); (b6) Histogram of (b5); (c6) Histogram of (c5); (d6) Histogram of (d5); (a7) Original image; (b7) Stego images by the proposed (ER = 0.1 bpp, PSNR = 41.25 dB); (c7) Stego images by the proposed (ER = 0.4 bpp, PSNR = 33.43 dB); (d7) Stego images by the proposed (ER = 0.7 bpp, PSNR = 27.50 dB); (a8) Histogram of (a7); (b8) Histogram of (b7); (c8) Histogram of (c7); (d8) Histogram of (d7); (a9) Original image; (b9) Stego images by traditional HS (ER = 0.1 bpp, PSNR = 45.62 dB); (c9) Stego images by traditional HS (ER = 0.4 bpp, PSNR = 36.71 dB); (d9) Stego images by traditional HS (ER = 0.7 bpp, PSNR = 29.04 dB); (a10) Histogram of (a9); (b10) Histogram of (b9); (c10) Histogram of (c9); (d10) Histogram of (d9); (a11) Original image; (b11) Stego images by skipping HS (ER = 0.1 bpp, PSNR = 43.35 dB); (c11) Stego images by skipping HS (ER = 0.4 bpp, PSNR = 35.49 dB); (d11) Stego images by skipping HS (ER = 0.7 bpp, PSNR = 29.38 dB); (a12) Histogram of (a11); (b12) Histogram of (b11); (c12) Histogram of (c11); and (d12) Histogram of (d11).

Algorithms	Embedding Rates (bpp)	Identification Error Rates
	0.01	13.5%
Traditional US algorithm	0.05	12.4%
fractional 115 algorithm	0.1	10.7%
	0.2	8.8%
	0.01	15.5%
Elemping US algorithm	0.05	11.1%
Skipping H5 algorithm	0.1	11.5%
	0.2	20.4%
	0.01	21.2%
Droposed algorithm	0.05	20.4%
r roposed algorithm	0.1	19.7%
	0.2	13.4%

Table 2. Performance of resisting steganalysis for HS algorithms.

# 5.4. Capacity–Distortion Performance Comparison

The proposed RDH scheme focuses on not only improving anti-steganalysis ability but also on capacity-distortion performance. The proposed scheme considers two embedding distortions that lead to the decreasing image quality of the HS algorithms: reducing the expanding distortion by the improved data coding method and reducing the shifting distortion by the improved skipping embedding strategy. Moreover, due to the application of the improved data-coding method, the skipping distance is increased, which further highlights the advantages of skipping embedding in reducing shifting distortion. Figure 12 shows the capacity-distortion performance comparison results, including the rhombus predictor with sorting technique (Sachnev et al.) [5], LP (Dragoi et al.) [6], ridge regression predictor (Wang et al. (Ridge)) [9], multiple histogram algorithm (Wang et al. (MH)) [10], and trace concealment algorithm (Dong et al.) [19]. The slightly worse capacity-distortion performance of proposed algorithm at 0.1 bpp in the Figure 12 is due to the fact that, at low embedding rates, there is no expandable bin in the prediction error histogram similar to length of the payload. Therefore, the embedding position selected may be closer to the middle of the histogram, resulting in more unnecessary shifting distortion. Thus, the worse capacity-distortion performance at the low embedding rate that does not occur in all images, but usually in smooth images, is unexpected. As can be seen from Figure 12, compared with the state-of-the-art algorithms that focus only on improving capacity-distortion performance, our proposed algorithm achieves similar capacity-distortion performance and superior anti-steganalysis ability. Moreover, compared with the trace concealment algorithm in [19] that loses too much image quality, our algorithm also achieves superior image quality.



**Figure 12.** The capacity–distortion performance comparison: (**a**) Lena; (**b**) Baboon; (**c**) Airplane; (**d**) and Boat.

## 6. Conclusions

In this paper, we design a LS-ET predictor that innovatively uses the edge characteristics of a target pixel to select training pixels. Then, we propose a novel HS framework to conceal the embedding trace, which adaptively determines the embedding position and skipping position according to the prediction error histogram and the given payload, so that the stego image histogram still conforms to the Laplacian distribution. Moreover, we improve both the data-coding method and the skipping embedding strategy in the framework. Furthermore, to improve the image quality, both shifting distortion and expanding distortion are considered. There is less research presented regarding concealing the embedding traces left by a hiding process in the HS field. This work makes up for the lack of the invisibility of traditional HS algorithms on the histogram level. Several methods in the paper, including LS-ET predictor, improved data-coding method, and improved skipping embedding method, are highly portable and suitable for popularization. In the future, we would like to investigate the use of neural networks to conceal the embedding traces that are left by histogram-shifting algorithms.

**Author Contributions:** Conceptualization, H.S. and B.H.; methodology, H.S.; validation, B.H., and J.G.; formal analysis, Y.R.; investigation, M.L.; resources, H.S.; data curation, B.H.; writing—original draft preparation, B.H.; writing—review and editing, H.S.; visualization, J.G.; supervision, M.L.; project administration, Y.R.; funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by [National Youth Science Foundation of China] grant number [61877007] and The APC was funded by [61877007].

Data Availability Statement: Data will be made available on reasonable request.

Acknowledgments: This work was recommended by ChinaMFS 2022. This work was supported by the Scientific Research Program Funded by Liaoning Provincial Education Department (Grant No. WQ2020014), the National Science Foundation of China (61601214, 61877007), and the Liaoning Planning Office of Philosophy and Social Science (CN) (Grant L19BTQ001).

Conflicts of Interest: The authors declare no conflict of interest.

# References

- 1. Ni, Z.; Shi, Y.Q.; Ansari, N.; Wei, S. Reversible data hiding. *Trans. Circuits Syst. Video Technol.* 2006, 16, 354–362.
- Hong, W.; Chen, T.S.; Shiu, C.W. Reversible data hiding for high quality images using modification of prediction errors. J. Syst. Softw. 2009, 82, 1833–1842. [CrossRef]
- Fallahpour, M. Reversible image data hiding based on gradient adjusted prediction. *IEICE Electron. Express* 2008, 5, 870–876. [CrossRef]
- Coltuc, D. Improved embedding for prediction-based reversible watermarking. *IEEE Trans. Inf. Forensics Secur.* 2011, 6, 873–882. [CrossRef]
- 5. Sachnev, V.; Kim, H.J.; Nam, J.; Suresh, S.; Yun, Q.S. Reversible watermarking algorithm using sorting and prediction. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 989–999. [CrossRef]
- 6. Dragoi, I.C.; Coltuc, D. Local-prediction-based difference expansion reversible watermarking. *IEEE Trans. Image Process.* **2014**, *23*, 1779–1790. [CrossRef]
- Hwang, H.J.; Kim, S.H.; Kim, H.J. Reversible data hiding using least square predictor via the lasso. EURASIP J. Image Video Process. 2016, 2016, 42. [CrossRef]
- 8. Wang, W.L.; Wang, C.T.; Zheng, H.C.; Wang, J.X.; Xiao, D.Q. An improved reversible watermarking scheme using weighted prediction and watermarking simulation. *Signal Process. Image Commun.* **2020**, *81*, 20375–20385. [CrossRef]
- 9. Wang, X.; Wang, X.; Ma, B.; Li, Q.; Shi, Y.Q. High precision error prediction algorithm based on ridge regression predictor for reversible data hiding. *IEEE Signal Process. Lett.* **2021**, *28*, 1125–1129. [CrossRef]
- 10. Wang, J.; Chen, X.; Ni, J.; Mao, N.; Shi, Y. Multiple histograms-based reversible data hiding: Framework and realization. *IEEE Trans. Circuits Syst. Video Technol.* 2020, *30*, 2313–2328. [CrossRef]
- 11. Kim, S.; Qu, X.; Sachnev, V.; Kim, H.J. Skewed histogram shifting for reversible data hiding using a pair of extreme predictions. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 3236–3246. [CrossRef]
- 12. Choi, Y.S.; Hwang, S.O. Improved histogram shifting-based data hiding method with alternative skipping method for image retrieval. *J. Intell. Fuzzy Syst.* **2018**, *35*, 6189–6195. [CrossRef]
- 13. Yang, Y.; Zhang, W.; Liang, D.; Yu, N. Reversible data hiding in medical images with enhanced contrast in texture area. *Digit. Signal Process.* **2016**, 2016, 13–24. [CrossRef]
- 14. Xie, X.Z.; Chang, C.C.; Hu, Y.C. An adaptive reversible data hiding scheme based on prediction error histogram shifting by exploiting signed-digit representation. *Multimed. Tools Appl.* **2020**, *79*, 24329–24346. [CrossRef]
- 15. Fei, P.A.; Yu, Z.A.; Xiang, Z.A.; Long, M.; Pan, W. Reversible data hiding based on RSBEMD coding and adaptive multi-segment left and right histogram shifting. *Signal Process. Image Commun.* **2020**, *81*, 115715.
- 16. Wang, J.; Huang, L.; Zhang, Y.; Zhu, Y.; Ni, J.; Shi, Y. An effective steganalysis algorithm for histogram-shifting based reversible data hiding. *Comput. Mater. Contin.* **2020**, 2020, 325–344. [CrossRef]
- 17. Hong, W.; Chen, T.S.; Chen, J. Reversible data hiding using delaunay triangulation and selective embedment. *Inf. Sci.* 2015, *308*, 140–154. [CrossRef]
- Hou, D.; Zhang, W.; Yang, Y.; Yu, N. Reversible data hiding under inconsistent distortion metrics. *IEEE Trans. Image Process.* 2018, 27, 5087–5099. [CrossRef]
- 19. Dong, L.; Zhou, J.; Sun, W.; Yan, D.; Wang, R. First steps toward concealing the traces left by reversible image data hiding. *IEEE Trans. Circuits Syst. II Express Briefs* 2020, 67, 951–955. [CrossRef]
- 20. Zhou, J.; Au, O.C. Determining the capacity parameters in PEE based reversible image watermarking. *Signal Process. Lett.* **2012**, 19, 287–290. [CrossRef]
- 21. Pan, H.; Siu, W.C.; Law, N.F. Lossless image compression using binary wavelet transform. *Image Process. IET* 2008, *1*, 353–362. [CrossRef]