



Article Reversible Data Hiding Based on Pixel-Value-Ordering and Prediction-Error Triplet Expansion

Heng-Xiao Chi¹, Ji-Hwei Horng^{2,*} and Chin-Chen Chang^{1,*}

- ¹ Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan; hx9704@gmail.com
- ² Department of Electronic Engineering, National Quemoy University, Kinmen 89250, Taiwan
- * Correspondence: horng@email.nqu.edu (J.-H.H.); ccc@o365.fcu.edu.tw (C.-C.C.)

Abstract: Pixel value ordering and prediction error expansion (PVO+PEE) is a very successful reversible data hiding (RDH) scheme. A series of studies were proposed to improve the performance of the PVO-based scheme. However, the embedding capacity of those schemes is quite limited. We propose a two-step prediction-error-triplet expansion RDH scheme based on PVO. A three-dimensional state transition map for the prediction-error triplet is also proposed to guide the embedding of the two-step scheme. By properly designing the state transitions, the proposed scheme can embed secret data or expand without embedding by modifying just a single entry of the triplet. The experimental results show that the proposed scheme significantly enlarges the embedding capacity of the PVO-based scheme and further reduces the distortion due to embedding.

Keywords: reversible data hiding; pixel-value-ordering; prediction-error triplet expansion; prediction-error histogram

1. Introduction

With the rapid development of computer and communication technologies, the privacy and security of data transmission have become more and more important. In order to make data transmission more secure, researchers developed various encryption algorithms to convert data into unrecognizable ciphertext. To make unrecognizable ciphertext appear less conspicuous, researchers developed various data hiding algorithms to embed the secret data into digital media, such as images, for transmission. Since a modified digital medium with confidential information is very similar to the original one, it will not draw the eavesdroppers' attention.

Data hiding methods are mainly divided into two types: reversible and irreversible. Reversible data hiding (RDH) [1–6] can extract secret data and restore the original digital objects without distortion. Irreversible data hiding (IRDH) [7,8] can only extract secret data and cannot restore the original digital objects losslessly.

So far, researchers have developed many efficient and easy-to-implement image reversible data hiding (RDH) technologies, such as differential expansion (DE) [9–11], histogram shift (HS) [12–14], prediction error expansion (PEE) [15–17], and so on. The DE method embeds secret data by extending the pixel difference. In HS technology, the image intensity histogram must be generated first and then modified for data embedding. Data embedding based on the PEE algorithm needs to generate the prediction error histogram (PEH) first, then modify it according to specific rules.

Due to the inherent characteristics of PEE-based technology, the embedding performance is related to the generated PEH, and the degree of image distortion is related to the modification rules. Therefore, work improving on PEE has focused on the design of a more concentrated PEH, such as rhombus predictor [18] and interpolation prediction [19], etc., while other works were dedicated to designing rules for minimum distortion. In addition, there is also a method called pairwise PEE, which uses the joint distribution between two



Citation: Chi, H.-X.; Horng, J.-H.; Chang, C.-C. Reversible Data Hiding Based on Pixel-Value-Ordering and Prediction-Error Triplet Expansion. *Mathematics* **2021**, *9*, 1703. https:// doi.org/10.3390/math9141703

Academic Editors: Ximeng Liu, Yinbin Miao and Zuobin Ying

Received: 22 June 2021 Accepted: 19 July 2021 Published: 20 July 2021

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



Copyright: © 2021 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/). prediction errors to generate a two-dimensional (2D) PEH and embeds secret data into the cover image, the digital image used to carry secret information, by modifying the generated 2D PEH [20].

The pixel value ordering (PVO) algorithm proposed by Li et al. [21] flexibly divides the cover image into scalable non-overlapping blocks to obtain a more desirable prediction error histogram. Then, the algorithm modifies the maximum and minimum values in each block to embed the data. The maximum value (minimum value) in each block is either increased (decreased) by one or unchanged so that the order of pixel values does not change after the data is embedded, thus ensuring reversibility. The improved PVO (IPVO) algorithm proposed by Peng et al. [22] takes the spatial relationship between pixels into consideration to improve the prediction method and PEH modification rules. In order to make better use of the spatial relationship between pixels, flexible spatial location-based PVO (FSL-PVO) was proposed by He et al. [23]. In [24], Ou et al. proposed a pairwise PEE to expand the embedding capability of PVO. In addition, researchers also developed various improved algorithms for PVO [25–28]. An example approach is to simultaneously embed data at the first k maximum values (minimum values) of a pixel block, such as PVO-k [26], k-pass PVO [27], and improved k-pass PVO [28].

PVO-based RDH schemes have the advantage of very low image distortion. That means this series of data hiding schemes can produce a stego-image, the digital image embedded with secret information, with a high peak-signal-to-noise ratio (PSNR) value. However, their embedding capacity is quite limited. As we outline in the next section, the existing PVO-based RDH schemes are all based on the two-dimensional state transition of prediction errors. In order to improve the embedding capacity, the new scheme proposed in this paper extends the state space of the prediction errors into three dimensions. More specifically, a prediction-error-triplet expansion RDH scheme based on PVO is proposed. Through an ingenious design of a three-dimensional state transition map for prediction-error triplets, the embedding of secret data or expansion for vacating rooms can be performed by modifying just a single entry of the triplet. The new scheme can improve the embedding capacity of the traditional PVO-based schemes while preserving the visual quality of the stego images.

The rest of this paper is organized as follows. The second section briefly introduces various typical PVO-based RDH schemes. The third section introduces the proposed scheme in detail. The experimental results and comparisons with related works are provided in Section 4. Finally, this paper is summarized in Section 5.

2. Related Works

In this section, PVO [21], IPVO [22], and Pairwise-PVO [24] are briefly introduced, and the embedding maps of typical PVO-based schemes are summarized. This series of RDH schemes are all based on the same problem formulation.

The PVO-based RDH schemes need to divide the cover image into non-overlapping equal-sized pixel blocks first. Each pixel block contains *n* pixel values $(x_1, x_2, ..., x_n)$. Then, the *n* pixel values are sorted into the sequence $x_{\sigma(1)} \leq x_{\sigma(2)} \leq ... \leq x_{\sigma(n)}$, where $(\sigma(1), \sigma(2), ..., \sigma(n))$ represents the original location as a function of the current order. That is, the function $\sigma : \{1, 2, ..., n\} \rightarrow \{1, 2, ..., n\}$ is the unique one-to-one and onto mapping. In the case of equal value, pixels are ordered by

$$\sigma(i) < \sigma(j), \text{ if } x_{\sigma(i)} = x_{\sigma(j)} \text{ and } i < j.$$
(1)

Finally, the pixels of maximum values (minimum values) are exploited to calculate prediction errors and embed secret data.

2.1. PVO

The original PVO data hiding scheme [21] uses the second largest value $x_{\sigma(n-1)}$ (second smallest value $x_{\sigma(2)}$) to predict the maximum value $x_{\sigma(n)}$ (minimum value $x_{\sigma(1)}$), the prediction errors PE_{max} and PE_{min} can be calculated by

$$\begin{cases}
PE_{max} = x_{\sigma(n)} - x_{\sigma(n-1)} \\
PE_{min} = x_{\sigma(1)} - x_{\sigma(2)}
\end{cases}$$
(2)

According to the histogram distributions of PE_{max} and PE_{min} , the peak bins are exploited to embed secret data. The embedding rules for PE_{max} and PE_{min} are given by

$$PE'_{max} = \begin{cases} PE_{max}, & \text{if } PE_{max} = 0\\ PE_{max} + b, & \text{if } PE_{max} = 1\\ PE_{max} + 1, & \text{if } PE_{max} > 1 \end{cases}$$
(3)

$$PE'_{min} = \begin{cases} PE_{min}, & \text{if } PE_{min} = 0\\ PE_{min} - b, & \text{if } PE_{min} = -1\\ PE_{min} - 1, & \text{if } PE_{min} < -1 \end{cases}$$
(4)

where $b \in \{0, 1\}$ is the binary secret bit to be embedded. Then, the maximum pixel value $x_{\sigma(n)}$ is modified to

$$x'_{\sigma(n)} = x_{\sigma(n-1)} + PE'_{max} = \begin{cases} x_{\sigma(n)}, & \text{if } PE_{max} = 0\\ x_{\sigma(n)} + b, & \text{if } PE_{max} = 1\\ x_{\sigma(n)} + 1, & \text{if } PE_{max} > 1 \end{cases}$$
(5)

while the minimum pixel value $x_{\sigma(1)}$ is modified to

$$x'_{\sigma(1)} = x_{\sigma(2)} + PE'_{min} = \begin{cases} x_{\sigma(1)}, & \text{if } PE_{max} = 0\\ x_{\sigma(1)} - b, & \text{if } PE_{max} = -1\\ x_{\sigma(1)} - 1, & \text{if } PE_{max} < -1 \end{cases}$$
(6)

The rest pixels in the block remain unchanged. To make the embedding rules easier to understand, the prediction errors can be translated into a location-based representation, as introduced in [25] by

$$\begin{cases} e_{max}^{1} = x_{u} - x_{\sigma(n-2)} \\ e_{max}^{2} = x_{v} - x_{\sigma(n-2)} \end{cases}$$
(7)

where

$$\begin{pmatrix} u = \min(\sigma(n), \sigma(n-1)) \\ v = \max(\sigma(n), \sigma(n-1)) \end{pmatrix} .$$
(8)

Based on the location-based prediction error pair (e_{max}^1, e_{max}^2) , the embedding rules of the PVO scheme can be illustrated by the state transition map, as given in Figure 1a. For each pixel block, its corresponding prediction error pair can be mapped to a 'state' on the map. The solid purple lines indicate the state transition rules for embedding the embeddable states, while the pink dashed lines indicate the expansions of non-embeddable states. The states without any guiding line are non-embeddable states with no expansion required. During the secret data embedding phase, the maximum value (minimum value) is either increasing (decreasing) or unchanged. The order of the pixel values is preserved, which is crucial for reversibility. In the secret data extraction phase, the prediction errors PE'_{max} and PE'_{min} can be calculated in the same way. Then, the secret data extraction and the pixel value restoration processes can be designed according to their corresponding embedding rules.



Figure 1. The state transition map for PVO-based RDH schemes. (a) PVO [21]; (b) IPVO [22]; (c) Pairwise-PVO [24]; (d) PVO-k [26]; (e) k-pass PVO [27]; (f) Improved k-pass PVO [28].

2.2. IPVO

In the PVO scheme, bin 1 of the PE_{max} histogram and bin -1 of the PE_{min} histogram are exploited to embed the secret data. Bin 0 is kept unchanged. That is, the states on the main diagonal line of the state transition map (Figure 1a) are left unexploited. The IPVO scheme [22] defines a new prediction error calculation method, which takes the position information into account. The prediction error PE_{max} is defined by

$$PE_{max} = x_u - x_v, \tag{9}$$

where u and v are defined the same as Equation (8). The embedding rules for IPVO are given by

$$PE'_{max} = \begin{cases} PE_{max} - 1, & \text{if } PE_{max} < 0\\ PE_{max} - b, & \text{if } PE_{max} = 0\\ PE_{max} + b, & \text{if } PE_{max} = 1\\ PE_{max} + 1, & \text{if } PE_{max} > 1 \end{cases}$$
(10)

The corresponding pixel value modification rules are given by

$$x'_{\sigma(n)} = \begin{cases} x_{\sigma(n)} + 1, & \text{if } PE_{max} < 0\\ x_{\sigma(n)} + b, & \text{if } PE_{max} = 0\\ x_{\sigma(n)} + b, & \text{if } PE_{max} = 1\\ x_{\sigma(n)} + 1, & \text{if } PE_{max} > 1 \end{cases}$$
(11)

Again, the modification of prediction errors and pixel values can be translated into a state transition map, as shown in Figure 1b. Prediction error states that satisfy $e_{max}^2 = e_{max}^1$ are exploited instead of $e_{max}^2 = e_{max}^1 + 1$. According to the experiments, the IPVO scheme outperforms the original PVO scheme in embedding capacity. That is, the accumulated

value of the histogram bins exploited by IPVO is greater than that of PVO for the conventional test images.

2.3. Pairwise-PVO

To improve the embedding capacity, the pairwise-PVO scheme [24] allows two maximum-valued pixels to be increased by one at the same time. The prediction error pair (e_{max}^1, e_{max}^2) is defined the same way as in Equations (7) and (8). The corresponding state transition map of the pairwise-PVO is given in Figure 1c. The diagonal guiding lines indicate the simultaneous increase of two maximum-valued pixels for embedding or expansion. There are more embeddable states near the origin. Additionally, the state of $(e_{max}^1, e_{max}^2) = (1, 1)$ is exploited to embed a ternary digit.

Although the embedding capacity is effectively increased, the simultaneous increase of pixel values severely degrades the visual quality of the stego images. A complexity measure is applied to classify the image blocks into different categories. Only low complexity blocks are embedded with the pairwise-PVO scheme. Median complexity blocks are embedded with the IPVO scheme, while high complexity blocks are left unembedded.

2.4. Other Typical PVO-Based Schemes

To further improve the embedding capacity, studies [26–28] were conducted to embed secret data by modifying the k maximum-valued pixels. The corresponding state transition maps for PVO-k [26], k-pass PVO [27], and improved k-pass PVO [28] schemes are given in Figure 1d–f.

Although their motivation is to modify k maximum-valued pixels, the concluded feasible value of k is still 2. Further extension of the k value leads to an unbearable degradation of image quality. The only difference between PVO-k and the original PVO, referring to Figure 1a,d, is the usage of $(e_{max}^1, e_{max}^2) = (1, 1)$ with the required additional expansions. In k-pass PVO, the state $(e_{max}^1, e_{max}^2) = (1, 1)$ is exploited to embed two bits at a time, as shown in Figure 1e. The improved k-pass PVO further exploits the states of $|e_{max}^1 - e_{max}^2| = 3$ to embed secret data, as shown in Figure 1f. As a result, the pixel value may be increased by two.

The image quality degradation caused by modifying two pixel-values at a time is still not solved. The improved PVO-based data hiding schemes [26–28] are all hybrid methods. Only smooth blocks are processed with the newly proposed schemes. The remaining complex blocks are embedded with IPVO or left unembedded.

3. Proposed Scheme

As discussed in the previous section, the embeddable states are concentrated in the smooth blocks of a cover image. To enlarge the embedding capacity, the improved PVO versions vacate more states for embedding by expanding the surrounding states outward. In most cases, this results in the simultaneous modification of two pixel-values and thus degrades the image quality.

In this section, a new PVO reversible data hiding scheme based on a prediction-error triplet is proposed. By extending the state space from two dimensions to three dimensions (3D), the space for vacating embeddable states can be effectively expanded. By modifying, at most, one of the three maximum-valued pixels in a block, the proposed scheme can embed secret data with the least sacrifice in image quality.

The flowcharts of the proposed data embedding and extraction processes are given in Figure 2. In the following subsections, the framework and pre-process are introduced first. Then, the new proposed 3D state transition map is presented. The complete data embedding process of two example blocks is given in the third subsection, and then their corresponding data extraction and pixel value recovery are demonstrated. Finally, an additional complex block exclusion process is provided to skip non-embeddable blocks.



Figure 2. Flowcharts of the proposed scheme. (a) Embedding phase; (b) Extraction phase.

3.1. Framework and Preprocess

As shown in Figure 2, the data embedding phase consists of a pre-process and a blockwise process. The cover image, the secret data stream, and the block size and complexity threshold parameters are provided as the input information. The framework of block processing is introduced first. Then, the required pre-process to ensure perfect recovery of both the cover image and secret data is presented.

Like all the PVO-based schemes, the RDH scheme proposed in this paper first divides the cover image into non-overlapping and equally sized blocks. Then, the pixels in each block is rearranged into a sequence of *n* pixels $(x_1, x_2, ..., x_n)$ in the raster scan order. The secret data embedding process includes two steps: the maximum-prediction-error expansion (max-PEE) and the minimum-prediction-error expansion (mini-PEE). In the first step, the *n* pixel values are sorted to obtain the sequence $x_{\sigma(1)} \le x_{\sigma(2)} \le ... \le x_{\sigma(n)}$, where $\sigma(i)$ represents the original location index of the *i*-th term after sorting. When repeated pixel value occurs, the spatial order of pixels with the same value is preserved as given in Equation (1). The 4-th pixel $x_{\sigma(n-3)}$ is used to predict the three maximum-valued pixels $(x_{\sigma(n)}, x_{\sigma(n-1)}, x_{\sigma(n-2)})$ and three prediction errors are obtained. The specific operation is as follows: first sort the spatial indices $(\sigma(n), \sigma(n-1), \sigma(n-2))$ to obtain $a_1 < a_2 < a_3$. That is, the three pixels $(x_{\sigma(n)}, x_{\sigma(n-1)}, x_{\sigma(n-2)})$ are resorted in their original raster-scanned order into $(x_{a_1}, x_{a_2}, x_{a_3})$. Then, calculate the prediction errors $(e_{max}^1, e_{max}^2, e_{max}^3)$ by

$$e_{max}^{1} = x_{a_{1}} - x_{\sigma(n-3)}$$

$$e_{max}^{2} = x_{a_{2}} - x_{\sigma(n-3)}$$

$$e_{max}^{3} = x_{a_{3}} - x_{\sigma(n-3)}$$
(12)

The prediction error triplet $(e_{max}^1, e_{max}^2, e_{max}^3)$ is a process in a similar manner as the PVO-based schemes introduced in the related works to embed secret data except for the state transition map is expanded to 3D space. The proposed 3D state-space transition map is provided in the next subsection. Suppose the prediction error triplet $(e_{max}^1, e_{max}^2, e_{max}^3)$ is modified into $(e_{max}^{\prime 1}, e_{max}^{\prime 2}, e_{max}^{\prime 3})$ after data embedding or expansion, the three maximum-valued pixels $(x_{a_1}, x_{a_2}, x_{a_3})$ are modified accordingly by

$$x'_{a_i} = x_{\sigma(n-3)} + e'^i_{max}, \ (i = 1, 2, 3).$$
(13)

In our design, the modification of the maximum-valued pixels is always incremented or unchanged, while the remaining pixels are kept unchanged. The maximum-valued group can be preserved. Further, the three pixels are resorted in their spatial order. This order cannot be altered even if their order in value is changed during the embedding process. Therefore, the embedding scheme's reversibility can be ensured.

In the second step, the modified pixel sequence obtained in the previous step $(x'_1, x'_2, \ldots, x'_n)$ is sorted again to obtain $x'_{\sigma(1)} \leq x'_{\sigma(2)} \leq \ldots \leq x'_{\sigma(n)}$. The indices $(\sigma'(1), \sigma'(2), \sigma'(3))$ are then resorted according to their spatial order to obtain $c_1 < c_2 < c_3$. The prediction errors of the three minimum-valued pixels $(e^1_{min}, e^2_{min}, e^3_{min})$ are calculated by Equation (14). The second round of data embedding is executed by modifying $(e^1_{min}, e^2_{min}, e^3_{min})$ according to the same 3D state transition map into $(e'^{11}_{min}, e'^{21}_{min}, e'^{31}_{min})$. The corresponding modification of pixel values is given by Equation (15). Again, the three minimum-valued pixels are decremented or unchanged to ensure reversibility. The remaining pixels are kept unchanged, and the final stego pixel sequence $(x''_1, x''_2, \ldots, x''_n)$ of the block can be obtained.

$$\begin{cases}
e_{min}^{1} = x'_{\sigma'(4)} - x'_{c_{1}} \\
e_{min}^{2} = x'_{\sigma'(4)} - x'_{c_{2}} \\
e_{min}^{3} = x'_{\sigma'(4)} - x'_{c_{3}}
\end{cases}$$
(14)

$$x_{c_i}'' = x_{\sigma'(4)}' - e_{min}'^i, \ (i = 1, 2, 3).$$
(15)

The pre-process is illustrated in Figure 2a. There are some auxiliary messages, including the length and width of a block (4 bits), the threshold of block complexity (8 bits), and the secret data size (16 bits). To avoid overflow and underflow, a block that contains any 0-valued or 255-valued pixel is classified as a non-embeddable block and is marked by 1 in the location map. Then, the compressed location map and the other auxiliary messages are hidden in the LSB of the pixels in the first two rows of the cover image, and these two rows are skipped in the embedding process. The original LSB bits of these two lines are appended to the front of the secret stream and embedded into the cover image. At the receiving end, these auxiliary messages are retrieved first to help extract the secret data and restore the cover image.

3.2. 3D State Transition Map

To design a proper 3D state transition map for a prediction-error triplet $(e_{max}^1, e_{max}^2, e_{max}^3)$, the 3D histogram of the triplet is investigated first. Figure 3 shows the 3D histogram of prediction-error triplet for cover image 'Lena' slice-wise. As shown in the figure, the maximum value appears at (1, 1, 1), which is a common phenomenon for ordinary cover images. A 3D view of the histogram is given in Figure 4, where a cross-sectional view at the maximum value is also plotted. The values of the prediction error triplets of an entire cover image are highly concentrated near the state (1, 1, 1), and the population gradually decreases outward.



Figure 3. Slice representation of 3D prediction error histogram. (a) Lena; (b) $e_{max}^3 = 0$; (c) $e_{max}^3 = 1$; (d) $e_{max}^3 = 2$; (e) $e_{max}^3 = 3$; (f) $e_{max}^3 = 4$; (g) $e_{max}^3 = 5$; (h) $e_{max}^3 = 6$.



Figure 4. 3D prediction error histogram with a view at the maximum value. (a) 3D histogram; (b) Cross section view.

A rule of thumb for transition map design is to select the maximum bins in the histogram as the embedding targets while reducing the required expansion of non-embeddable bins. The 3D prediction-error space is divided into layered structures according to the equation

$$L_{p} = \left\{ \left(e_{max}^{1}, e_{max}^{2}, e_{max}^{3} \right) \middle| \min\left(e_{max}^{1}, e_{max}^{2}, e_{max}^{3} \right) = p. \right\},$$
(16)

where L_p is the set of all states included in the *p*-th layer. The layered structure is illustrated in Figure 5. In fact, the 0-th layer is the boundary surface of an octant in the 3D space. More specifically, the *p*-th layer contains all of the states on three quarter plane

$$\begin{cases}
e_{max}^{1} = p, e_{max}^{2} \ge p, e_{max}^{3} \ge p \\
e_{max}^{2} = p, e_{max}^{1} \ge p, e_{max}^{3} \ge p \\
e_{max}^{3} = p, e_{max}^{1} \ge p, e_{max}^{2} \ge p
\end{cases}$$
(17)



Figure 5. Layered structure of the 3D transition map.

Since the maximum value of the 3D prediction-error histogram appears at (1, 1, 1), which is located on the 1-st layer, the 3D state transition map for the 0-th layer and the remaining layers are designed differently, as shown in Figures 6 and 7, respectively. In each figure, the state transition maps for the three quarter-planes are given in (a–c), while their projection versions are given in (d–f) for convenience. The major difference between the 0-th layer and the remaining layers is that the state (0, 0, 0) is left unembedded, while the states (p, p, p), $p \neq 0$ are embedded with two bits each. Thus, the highest bin of (1, 1, 1) is efficiently exploited. On each quarter plane, the embeddable states are aligned on two diagonal lines, and the secret data is embedded by expanding the states along the horizontal and the vertical axes. This design can reduce the embedding distortion to a minimum since only a single value of the prediction-error triplet is modified. Based on the 3D state transition map, the proposed PVO scheme can efficiently embed the secret data with the least sacrifice of image quality. The corresponding rules for data embedding or pixel value expansion of the proposed scheme are summarized as follows.

For the 0-th layer (L_0) :

1.
$$e_{max}^3 = 0$$
:

$$\begin{pmatrix}
e_{max}^{\prime 1}, e_{max}^{\prime 2}, e_{max}^{\prime 3}
\end{pmatrix} = \begin{cases}
\begin{pmatrix}
e_{max}^{1}, e_{max}^{2}, e_{max}^{3}, & e_{max}^{1} = e_{max}^{2} = 0 \\
(e_{max}^{1} + 1, e_{max}^{2}, e_{max}^{3}), & e_{max}^{1} - 1 > e_{max}^{2} \\
(e_{max}^{1} + b, e_{max}^{2}, e_{max}^{3}), & e_{max}^{1} - 1 = e_{max}^{2} \\
(e_{max}^{1}, e_{max}^{2} + b, e_{max}^{3}), & e_{max}^{1} = e_{max}^{2} \neq 0 \\
(e_{max}^{1}, e_{max}^{2} + 1, e_{max}^{3}), & 0 \neq e_{max}^{1} < e_{max}^{2}
\end{pmatrix}$$
(18)

2.
$$e_{max}^2 = 0$$
:

$$\left(e_{max}^{\prime 1}, e_{max}^{\prime 2}, e_{max}^{\prime 3}\right) = \begin{cases} \left(e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1\right), & e_{max}^{1} < e_{max}^{3} - 1\\ \left(e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + b\right), & e_{max}^{1} = e_{max}^{3} - 1\\ \left(e_{max}^{1} + b, e_{max}^{2}, e_{max}^{3}\right), & e_{max}^{1} = e_{max}^{3} \neq 0\\ \left(e_{max}^{1} + 1, e_{max}^{2}, e_{max}^{3}\right), & e_{max}^{1} > e_{max}^{3} \neq 0 \end{cases}$$
(19)

3.
$$e_{max}^1 = 0$$
:

$$\left(e_{max}^{\prime 1}, e_{max}^{\prime 2}, e_{max}^{\prime 3}\right) = \begin{cases} \left(e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1\right), & e_{max}^{2} < e_{max}^{3} - 1\\ \left(e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + b\right), & e_{max}^{2} = e_{max}^{3} - 1\\ \left(e_{max}^{1}, e_{max}^{2} + b, e_{max}^{3}\right), & e_{max}^{2} = e_{max}^{3} \neq 0\\ \left(e_{max}^{1}, e_{max}^{2} + 1, e_{max}^{3}\right), & e_{max}^{2} > e_{max}^{3} \neq 0 \end{cases}$$
(20)

 e_{max}^2 5 4

3 2 b=1 1 b=0

For the *p*-th (except for 0) layer ($L_{p\neq 0}$):

2.

3.

1.
$$e_{max}^{1} = e_{max}^{2} = e_{max}^{3} = p$$
:
 $\left(e_{max}^{\prime 1}, e_{max}^{\prime 2}, e_{max}^{\prime 3}\right) = \begin{cases} \left(e_{max}^{1}, e_{max}^{2}, e_{max}^{3}\right), & b_{1}b_{0} = 00\\ \left(e_{max}^{1}, +1, e_{max}^{2}, e_{max}^{3}\right), & b_{1}b_{0} = 01\\ \left(e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1, e_{max}^{3}\right), & b_{1}b_{0} = 10\\ \left(e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1\right), & b_{1}b_{0} = 11 \end{cases}$
(21)

$$e_{max}^{3} = p:$$

$$\begin{pmatrix} e_{max}^{\prime 1}, e_{max}^{\prime 2}, e_{max}^{\prime 3} \end{pmatrix} = \begin{cases} \begin{pmatrix} e_{max}^{1} + 1, e_{max}^{2}, e_{max}^{3} \end{pmatrix}, & e_{max}^{1} - 1 > e_{max}^{2} \\ (e_{max}^{1} + 1, e_{max}^{2}, e_{max}^{3}), & e_{max}^{1} - 1 = e_{max}^{2} = p \\ (e_{max}^{1} + b, e_{max}^{2}, e_{max}^{3}), & e_{max}^{1} - 1 = e_{max}^{2} \neq p \\ (e_{max}^{1}, e_{max}^{2}, e_{max}^{3}), & e_{max}^{1} - 1 = e_{max}^{2} \neq p \\ (e_{max}^{1}, e_{max}^{2} + b, e_{max}^{3}), & e_{max}^{1} = e_{max}^{2} \neq p \\ (e_{max}^{1}, e_{max}^{2} + 1, e_{max}^{3}), & p \neq e_{max}^{1} < e_{max}^{2} \end{cases}$$
(22)

$$e_{max}^{2} = p:$$

$$\begin{pmatrix} e_{max}^{\prime 1}, e_{max}^{\prime 2}, e_{max}^{\prime 3} \end{pmatrix} = \begin{cases} (e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1), & e_{max}^{1} < e_{max}^{3} - 1 \\ (e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1), & e_{max}^{1} = e_{max}^{3} - 1 = p \\ (e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + b), & e_{max}^{1} = e_{max}^{3} - 1 \neq p \\ (e_{max}^{1}, e_{max}^{2}, e_{max}^{3}, e_{max}^{3}), & e_{max}^{1} = e_{max}^{3} \neq p \\ (e_{max}^{1} + 1, e_{max}^{2}, e_{max}^{3}), & e_{max}^{1} > e_{max}^{3} \neq p \\ (e_{max}^{1} + 1, e_{max}^{2}, e_{max}^{3}), & e_{max}^{1} > e_{max}^{3} \neq p \end{cases}$$

$$(23)$$

4.
$$e_{max}^{1} = p$$
:

$$\begin{pmatrix} e_{max}^{1}, e_{max}^{2}, e_{max}^{3} \end{pmatrix} = \begin{cases} \begin{pmatrix} e_{max}^{1}, e_{max}^{2}, e_{max}^{3}, e_{max}^{2}, e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{2}, e_{max}^{3} - 1 = p \\ e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} - 1 = p \\ e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} - 1 = p \\ e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} - 1 = p \\ e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} - 1 = p \\ e_{max}^{1}, e_{max}^{2}, e_{max}^{2} + 1, e_{max}^{3} \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} + p \end{pmatrix}.$$
 (24)

$$\begin{pmatrix} e_{max}^{1}, e_{max}^{2}, e_{max}^{2} + 1, e_{max}^{3} \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} + p \\ e_{max}^{1}, e_{max}^{2}, e_{max}^{2} + 1, e_{max}^{3} \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} + p \end{pmatrix}$$

$$\begin{pmatrix} e_{max}^{1}, e_{max}^{2}, e_{max}^{2} + 1, e_{max}^{3} \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} + p \\ e_{max}^{1}, e_{max}^{2}, e_{max}^{2} + 1, e_{max}^{3} \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} + p \end{pmatrix}$$

$$\begin{pmatrix} e_{max}^{1}, e_{max}^{2}, e_{max}^{2} + 1, e_{max}^{3} \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} + p \\ e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{2} = e_{max}^{3} + p \end{pmatrix}$$

$$\begin{pmatrix} e_{max}^{1}, e_{max}^{2}, e_{max}^{2} + 1, e_{max}^{3} \end{pmatrix}, & e_{max}^{2} + e_{max}^{3} \end{pmatrix}, & e_{max}^{2} + e_{max}^{3} + p \end{pmatrix}$$

$$\begin{pmatrix} e_{max}^{1}, e_{max}^{2}, e_{max}^{2} + 1, e_{max}^{3} \end{pmatrix}, & e_{max}^{2} + e_{max}^{3} \end{pmatrix}, & e_{max}^{2} + e_{max}^{3} \end{pmatrix}$$

$$\begin{pmatrix} e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{2} + e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{2} + e_{max}^{3} \end{pmatrix}, & e_{max}^{3} + p \end{pmatrix}$$

$$\begin{pmatrix} e_{max}^{1}, e_{max}^{2}, e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{2} + e_{max}^{3} + 1 \end{pmatrix}, & e_{max}^{3} +$$

Figure 6. (**a**–**c**) 3D state transition map for the 0-th layer $L_{p=0}$; (**d**–**f**) the projection versions.



Figure 7. (**a**–**c**) 3D state transition map for the *p*-th layer $L_{p\neq 0}$; (**d**–**f**) the projection versions.

3.3. Examples of Secret Data Embedding

In this subsection, two example image blocks sized 2×2 are provided to demonstrate the data embedding process of the proposed scheme. The first example is shown in Figure 8a. The first step is to execute the max-PEE. The sorted sequence is (192, 192, 192, 193), the maximum-valued pixels in their spatial order are $(x_2, x_3, x_4) = (192, 192, 193)$, and the minimum-valued pixel is $x_1 = 192$. The prediction error triplet can be calculated by $(e_{max}^1, e_{max}^2, e_{max}^3) = (x_2 - x_1, x_3 - x_1, x_4 - x_1) = (0, 0, 1)$ and its corresponding layer $p = \min(e_{max}^1, e_{max}^2, e_{max}^3) = e_{max}^2 = 0$. According to the state transition map of the 0-th layer (see Figure 6e or Equation (19)), suppose the data to be embedded is b = 1, the triplet $(e_{max}^1, e_{max}^2, e_{max}^3)$ is modified from (0, 0, 1) to (0, 0, 2). Consequently, (x_2, x_3, x_4) is modified from (192, 192, 193) to $(x'_2, x'_3, x'_4) = (192, 192, 194)$. Then, the second step of the mini-PEE has proceeded. The sorted pixel sequence is (192, 192, 192, 194) and the minimum-valued pixels in their spatial order is (x'_1, x'_2, x'_3) . The prediction error triplet is given by $(e_{min}^1, e_{min}^2, e_{min}^3) = (x'_4 - x'_1, x'_4 - x'_2, x'_4 - x'_3) = (2, 2, 2)$ and its corresponding layer is $p = \min(e_{min}^1, e_{min}^2, e_{min}^3) = e_{min}^2 = 2$. According to the state transition map of the 2-nd layer (see Figure 7e or Equation (23)), suppose the two bits of data to be embedded are $b_1b_0 = 01, (e_{min}^1, e_{min}^2, e_{min}^3)$ is modified from (2, 2, 2) to (3, 2, 2). Consequently, (x'_1, x'_2, x'_3) is modified from (192, 192, 192) to $(x''_1, x''_2, x''_3) = (191, 192, 192)$. The second example block is given in Figure 8b. In the first embedding step, the maximum-valued pixels in their spatial order is $(x_1, x_2, x_4) = (194, 195, 193)$. The prediction error triplet is therefore $(e_{max}^1, e_{max}^2, e_{max}^3) = (x_1 - x_3, x_2 - x_3, x_4 - x_3) = (2, 3, 1)$. According to the state transition map of the 1-st layer (see Figure 7d or Equation (22)), the state is not embeddable and $(e_{max}^1, e_{max}^2, e_{max}^3)$ is expanded from (2, 3, 1) to (2, 4, 1). Consequently, (x_1, x_2, x_4) is modified from (194, 195, 193) to $(x'_1, x'_2, x'_4) = (194, 196, 193)$. In the second embedding step, the minimum-valued pixels in spatial order is $(x'_1, x'_3, x'_4) = (194, 192, 193)$. The triplet is $(e_{min}^1, e_{min}^2, e_{min}^3) = (x_2' - x_1', x_2' - x_3', x_2' - x_4') = (2, 4, 3)$. According to the state transition map of the 2-th layer (see Figure 7f or Equation (24)), suppose the data to be embedded



is b = 1, $(e_{min}^1, e_{min}^2, e_{min}^3)$ is modified from (2, 4, 3) to (2, 5, 3). Consequently, (x'_1, x'_3, x'_4) changes from (194, 192, 193) to $(x''_1, x''_3, x''_4) = (194, 191, 193)$.

(a) Embedding Example 1.



(b) Embedding Example 2.

Figure 8. Two examples of the secret data embedding process. (a) Embedding Example 1; (b) Embedding Example 2.

3.4. Data Extraction and Image Recovery

The data extraction is operated in the reverse embedding order. At the receiver end, the stego-image is first divided into non-overlapping blocks of the same size. For each block, the pixels are sorted to obtain the ordered sequence first. Then, the reverse operations of mini-PEE and max-PEE are executed sequentially to extract the secret data and recover the original pixel values.

The stego-block in Figure 8a is applied as the first example shown in Figure 9a. The pixel sequence is sorted to obtain the minimum-valued pixels in spatial order $(x_1', x_2', x_3') =$ (191, 192, 192) and the triplet $(e'_{min}, e'_{min}, e'_{min}) = (194 - 191, 194 - 192, 194 - 192) = (3, 2, 2).$ According to the state transition map of the 2-nd layer (see Figure 7e or Equation (23)), (3,2,2) = (p+1, p, p+0) corresponds to the state (1,0) on the map and comes from (0,0) due to embedding of $b_1b_0 = 01$. Therefore, the original prediction error triplet is (2+0,2,2+0) and the pixel values are $(x'_1,x'_2,x'_3) = (192,192,192)$. Then, the sequence is sorted again to obtain the maximum-valued pixels $(x'_2, x'_3, x'_4) = (192, 192, 194)$ and the prediction error triplet $(et_{max}^1, et_{max}^2, et_{max}^3) = (192 - 192, 192 - 192, 194 - 192) = (0, 0, 2).$ According to the state transition map of the 0-th layer (see Figure 6e or Equation (19)), the triplet (0, 0, 2) corresponds to (0, 2) on the map and comes from (0, 1) due to embedding of b = 1. Therefore, the original triplet is (0, 0, 1) and the cover pixel values are $(x_2, x_3, x_4) =$ (192, 192, 193). The second stego-block is shown in Figure 9b. The pixel sequence is sorted to obtain the minimum-valued pixels in spatial order $(x_1'', x_3'', x_4'') = (194, 191, 193)$ and the prediction error triplet $(e'_{min}, e'_{min}, e'_{min}) = (196 - 194, 196 - 191, 196 - 193) =$ (2,5,3). According to the state transition map of the 2-nd layer (see Figure 7f or Equation (24)), (2,5,3) = (p, p+3, p+1) corresponds to the state (3,1) on the map and comes from (2,1) due to embedding of b = 1. Therefore, the original prediction error triplet is (2, 2+2, 2+1) and the pixel values are $(x'_1, x'_2, x'_3) = (194, 192, 193)$. Then, the sequence is sorted again to obtain the maximum-valued pixels $(x'_1, x'_2, x'_4) = (194, 196, 193)$ and the triplet $(e\ell_{max}^1, e\ell_{max}^2, e\ell_{max}^3) = (194 - 192, 196 - 192, 193 - 192) = (2, 4, 1)$. According to the state transition map of the 1-st layer (see Figure 7d or Equation (22)), the triplet (2, 4, 1)corresponds to (1,3) on the map and comes from (1,2) due to expansion. Therefore, the original triplet is (2,3,1), and the cover pixel values are $(x_1, x_2, x_4) = (194, 195, 193)$.



(b) Extract Example 2.

′p p+1

p+2 p+3 p+4 p+5

 e_{max}^1

Figure 9. Two examples of the secret data extraction and block recovery process. (a) Extract Example 1; (b) Extract Example 2.

3.5. Complex Block Exclusion

Since the embeddable states are aligned on two diagonal lines of each quarter plane in the layered structure, at least two pixel-values in a block should be equal or very close to each other to make it embeddable. Complex blocks are non-embeddable, and pixel value expansion is required without any contribution to the embedding capacity.

All PVO-based schemes introduced in the related works include a complexity measure of blocks and skip the complex blocks when embedding. In our scheme, the complexity measure of blocks proposed by Ou et al. in [26] was adopted. The noise level (*NL*) of a pixel block is evaluated through its context. When the size of the pixel block is $n_1 \times n_2$, referring to Figure 10, the two columns of pixels on the right and the two rows of pixels below the pixel block are regarded as the context of the current block. Calculating the sum of the absolute difference between every two consecutive pixels in the context can obtain the complexity *NL* of the current pixel block. The formula is given by

$$NL = \sum d_{ver} + \sum d_{hor},\tag{25}$$

where d_{ver} denotes an absolute difference of two consecutive pixels in the vertical direction, and d_{hor} denotes an absolute difference of two pixels in the horizontal direction.



Figure 10. A 2×2 sized block and its context (gray pixels), where NL is the noise level.

The pairwise-PVO [24], POV-k [26], k-pass PVO [27], and improved k-pass PVO [28] all classify the image blocks into three types and process them with hybrid schemes. In our work, the proposed single scheme was applied. The complexity measure was just used to exclude the non-embeddable blocks when *NL* exceeds a given threshold *T*.

4. Experimental Results and Comparisons

The proposed data hiding scheme is implemented using MATLAB language on the Windows PC environment. Eight standard grayscale test images sized 512×512 shown in Figure 11 are applied. The binary secret data *S* is produced by a random number generator. The performance is mainly evaluated through two indicators, the peak signal-to-noise ratio (PSNR) and the payload. Let *W* and *H* be the width and height of the cover image. The PSNR defined by Equation (26) evaluates the visual quality of a stego image.

$$PSNR = 10 \log_{10} \left(\frac{H \times W \times (255)^2}{\sum_{i=1}^{H} \sum_{j=1}^{W} (G_{ij} - G'_{ij})^2} \right),$$
 (26)

where G_{ij} and G'_{ij} are the pixel values in the *i*-th row and the *j*-th column of the cover image and the stego-image, respectively. In general, we think that if the PSNR of the hidden image is greater than 30, then the difference between it and the original image is already invisible. Of course, the higher the PSNR, the smaller the difference between the hidden image and the original image, and the less likely it is to discover the secret data on the hidden image. A stego image of a high PSNR value can be produced by minimizing the required modification of pixel values during data embedding. A major objective of data hiding is to resemble the original image as much as possible. Therefore, an excellent data hiding scheme should embed a satisfactory amount of secret data with a high PSNR value.



Figure 11. Eight gray level standard test images applied in our experiments. (**a**) Lena; (**b**) Baboon; (**c**) Barbara; (**d**) Elaine; (**e**) Airplane; (**f**) Peppers; (**g**) Lake; (**h**) Boat.

The payload is the valid part of the data we embed in the stego image, i.e., the secret data. In fact, the data we embed in the stego image consists of two parts: the payload and the auxiliary information. The bigger the payload, the more secret data we can embed.

4.1. Experimental Results

There are two adaptive parameters in our proposed scheme, which are the block size $n_1 \times n_2$ and the threshold of complexity *T*. In general, larger block size or a smaller threshold value can produce a stego image with a high PSNR value. However, both strategies lead to the reduction of payloads. A larger block size reduces the total number of blocks, while a smaller threshold value reduces the number of applicable smooth blocks.

In our experiments, the influence of the block size was investigated first. The block sizes of 1×4 , 2×2 , 4×1 , 2×3 , 3×2 , and 3×3 were tested, and the results are listed in Table 1. As shown in the table, the visual quality of the stego images improves with the expansion in block size. However, among different shapes of the same size, 2×2 and 3×2 outperform the others.

Table 1. PSNR values under different sizes (payload = 10,000 bits).

Image	1×4	2 imes 2	4 imes 1	2 imes 3	3 imes 2	3×3
Lena	59.51	60.26	60.20	61.05	61.18	61.42
Baboon	55.33	55.84	55.33	56.89	56.99	56.72
Barbara	59.65	60.01	59.68	60.97	60.98	61.28
Elaine	58.03	58.52	57.85	59.19	59.29	59.30
Airplane	62.63	63.07	62.97	63.73	63.80	63.63
Peppers	58.30	58.55	58.48	59.75	59.90	60.40
Lake	59.72	59.92	59.39	60.88	60.76	60.72
Boat	58.22	58.51	58.49	59.50	59.52	59.81

By taking both the block size and the threshold value into consideration, Table 2 lists the best block size and the complexity threshold to obtain the best PSNR for different test images under a payload of 10,000 bits. As shown in the table, 3×2 is the most frequent optimal solution and is the smallest size among all the sizes, which is advantageous when the application requires a high payload.

Image	PSNR	Block Size	Т
Airplane	63.80	3×2	20
Baboon	56.99	3×2	223
Barbara	61.29	3×3	65
Boat	59.82	2 imes 4	106
Elaine	59.29	3×2	97
Lake	60.88	2×3	68
Lena	61.57	3 imes 4	70
Peppers	60.55	3 imes 4	105

Table 2. PSNR values under different sizes (payload = 10,000 bits).

4.2. Comparisons

In this subsection, the performance of our proposed scheme is compared with the typical PVO-based schemes, namely, PVO [21], IPVO [22], pairwise PVO [24], and the improved k-pass PVO [28]. PSNR values under different payloads for different test images are plotted in Figure 12, where the block size of 3×2 is adopted. The payload starts with 5000 bits, and the step length is 5000 bits, except for the cover image 'Baboon'. Due to the image's special features, the initial and step sizes of the payload are set differently. The results show that, for all test images, the PSNR of the proposed scheme outperforms the other schemes under the same payload. As the amount of payload increases, the PSNR improvement of the proposed scheme is more obvious. Due to the high embedding capacity and the low pixel value distortion during embedding and expansion, the proposed data hiding scheme provides a significant improvement for test images of high complexity such as 'Baboon'. This further confirms the superiority of the proposed 3D state transition map.



Figure 12. Cont.









dB65

63

Figure 12. The PSNR values under different payloads. (a) Lena; (b) Baboon; (c) Barbara; (d) Elaine; (e) Airplane; (f) Peppers; (g) Lake; (h) Boat.

Tables 3 and 4 list the PSNR values of the stego images for related works under the payloads of 10,000 bits and 20,000 bits, where the block size is 3×2 . The embedding capacity of the test image 'Baboon' cannot achieve 20,000 bits for all schemes, so its corresponding row in Table 4 is empty.

Image	Proposed	[21]	[22]	[24]	[27]	[28]
Lena	61.18	60.28	60.47	60.90	60.64	60.68
Baboon	56.99	53.43	53.49	54.94	54.00	54.60
Barbara	60.98	60.39	60.48	60.14	60.37	60.72
Elaine	59.29	56.79	57.34	58.29	57.67	57.49
Airplane	63.80	61.95	62.90	63.63	63.45	63.28
Peppers	59.90	58.83	58.97	59.36	59.29	59.14
Lake	60.76	58.13	58.85	59.94	59.71	59.86
Boat	59.52	58.05	58.25	58.05	58.28	58.61

Table 3. Comparison of PSNR values when the pixel block is 3×2 (payload = 10,000 bits).

Table 4. Comparison of PSNR values when the pixel block is 3×2 (payload = 20,000 bits).

Image	Proposed	[21]	[22]	[24]	[27]	[28]
Lena	57.67	56.14	56.51	57.04	56.81	56.67
Baboon		_	—	—	—	_
Barbara	57.27	55.40	56.13	55.88	56.09	56.74
Elaine	54.93	52.31	52.58	53.50	53.08	53.14
Airplane	60.04	58.09	58.99	59.73	59.59	59.18
Peppers	56.42	54.61	54.71	55.34	55.10	54.97
Lake	56.31	53.32	53.54	54.84	54.53	54.66
Boat	55.57	53.34	53.81	53.72	54.07	54.27

5. Conclusions

A new 3D state transition map for the PVO-based RDH scheme was proposed. The newly defined prediction error triplet and a sophisticatedly devised 3D state transition map were applied to guide the hiding of secret data in an image. The peak state in the 3D prediction error histogram was efficiently exploited to embed two bits of secret data each. All states, either embeddable or non-embeddable, were processed so that only a single entry of the triplet was modified during the embedding of secret data or pure expansion.

The experimental results showed that the proposed scheme could efficiently embed secret data with very low distortion to the cover image. The PSNR of the proposed scheme significantly outperforms the existing PVO-based schemes. Our scheme is especially advantageous for images with complex textures. As the amount of payload increases, the improvement in the visual quality of stego images becomes more obvious. In the future, a PVO scheme based on a state space of prediction errors with even higher dimensions is promising.

Author Contributions: Conceptualization, H.-X.C. and J.-H.H.; methodology, J.-H.H.; validation, H.-X.C.; writing—original draft preparation, H.-X.C.; writing—review and editing, J.-H.H. and C.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology of Taiwan, grant number MOST 110-2221-E-507-003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

- 1. Chang, C.-C. Adversarial Learning for Invertible Steganography. IEEE Access 2020, 8, 198425–198435. [CrossRef]
- Chang, C.-C.; Li, C.-T.; Shi, Y.-Q. Privacy-Aware Reversible Watermarking in Cloud Computing Environments. *IEEE Access* 2018, 6, 70720–70733. [CrossRef]

- Chi, H.; Chang, C.-C.; Liu, Y. An SMVQ Compressed Data Hiding Scheme Based on Multiple Linear Regression Prediction. Connect. Sci. 2020, 1–20. [CrossRef]
- 4. Chen, K.-M. High Capacity Reversible Data Hiding Based on the Compression of Pixel Differences. *Mathematics* 2020, *8*, 1435. [CrossRef]
- 5. Yu, C.; Ye, C.; Zhang, X.; Tang, Z.; Zhan, S. Separable Reversible Data Hiding in Encrypted Image Based on Two-Dimensional Permutation and Exploiting Modification Direction. *Mathematics* **2019**, *7*, 976. [CrossRef]
- 6. Sun, Y.; Lu, Y.; Chen, J.; Zhang, W.; Yan, X. Meaningful Secret Image Sharing Scheme with High Visual Quality Based on Natural Steganography. *Mathematics* 2020, *8*, 1452. [CrossRef]
- Chang, C.C.; Liu, Y.; Nguyen, T.S. A Novel Turtle Shell Based Scheme for Data Hiding. In Proceedings of the Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan, 27–29 August 2014; pp. 89–93. [CrossRef]
- 8. Qin, J.; Wang, J.; Tan, Y.; Huang, H.; Xiang, X.; He, Z. Coverless Image Steganography Based on Generative Adversarial Network. *Mathematics* **2020**, *8*, 1394. [CrossRef]
- 9. Alattar, A.M. Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156. [CrossRef]
- 10. Hu, Y.; Lee, H.-K.; Li, J. DE-Based Reversible Data Hiding With Improved Overflow Location Map. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 250–260. [CrossRef]
- Tian, J. Reversible Data Embedding Using a Difference Expansion. *IEEE Trans. Circuits Syst. Video Technol.* 2003, 13, 890–896. [CrossRef]
- 12. Li, X.; Li, B.; Yang, B.; Zeng, T. General Framework to Histogram-Shifting-Based Reversible Data Hiding. *IEEE Trans. Image Process.* **2013**, *22*, 2181–2191. [CrossRef]
- 13. Ni, Z.; Shi, Y.-Q.; Ansari, N.; Su, W. Reversible Data Hiding. IEEE Trans. Circuits Syst. Video Technol. 2006, 16, 354–362. [CrossRef]
- 14. Tsai, P.; Hu, Y.-C.; Yeh, H.-L. Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting. *Signal Process.* **2009**, *89*, 1129–1143. [CrossRef]
- 15. Dragoi, I.-C.; Coltuc, D. Local-Prediction-Based Difference Expansion Reversible Watermarking. *IEEE Trans. Image Process.* 2014, 23, 1779–1790. [CrossRef]
- 16. Dragoi, I.-C.; Coltuc, D. Adaptive Pairing Reversible Watermarking. IEEE Trans. Image Process. 2016, 25, 2420–2422. [CrossRef]
- 17. He, W.; Xiong, G.; Weng, S.; Cai, Z.; Wang, Y. Reversible Data Hiding Using Multi-Pass Pixel-Value-Ordering and Pairwise Prediction-Error Expansion. *Inf. Sci.* 2018, *467*, 784–799. [CrossRef]
- Sachnev, V.; Kim, H.J.; Nam, J.; Suresh, S.; Shi, Y.Q. Reversible Watermarking Algorithm Using Sorting and Prediction. *IEEE Trans. Circuits Syst. Video Technol.* 2009, 19, 989–999. [CrossRef]
- Luo, L.; Chen, Z.; Chen, M.; Zeng, X.; Xiong, Z. Reversible Image Watermarking Using Interpolation Technique. *IEEE Trans. Inf. Forensics Secur.* 2010, 5, 187–193. [CrossRef]
- Ou, B.; Li, X.; Zhao, Y.; Ni, R.; Shi, Y.-Q. Pairwise Prediction-Error Expansion for Efficient Reversible Data Hiding. *IEEE Trans. Image Process.* 2013, 22, 5010–5021. [CrossRef] [PubMed]
- Li, X.; Li, J.; Li, B.; Yang, B. High-Fidelity Reversible Data Hiding Scheme Based on Pixel-Value-Ordering and Prediction-Error Expansion. Signal Process. 2013, 93, 198–205. [CrossRef]
- 22. Peng, F.; Li, X.; Yang, B. Improved PVO-Based Reversible Data Hiding. Digit. Signal Process. 2014, 25, 255–265. [CrossRef]
- 23. He, W.; Cai, Z.; Wang, Y. Flexible Spatial Location-Based PVO Predictor for High-Fidelity Reversible Data Hiding. *Inf. Sci.* 2020, 520, 431–444. [CrossRef]
- Ou, B.; Li, X.; Wang, J. High-Fidelity Reversible Data Hiding Based on Pixel-Value-Ordering and Pairwise Prediction-Error Expansion. J. Vis. Commun. Image Represent. 2016, 39, 12–23. [CrossRef]
- Zhang, T.; Li, X.; Qi, W.; Guo, Z. Location-Based PVO and Adaptive Pairwise Modification for Efficient Reversible Data Hiding. IEEE Trans. Inf. Forensics Secur. 2020, 15, 2306–2319. [CrossRef]
- Ou, B.; Li, X.; Zhao, Y.; Ni, R. Reversible Data Hiding Using Invariant Pixel-Value-Ordering and Prediction-Error Expansion. Signal Process. Image Commun. 2014, 29, 760–772. [CrossRef]
- He, W.; Zhou, K.; Cai, J.; Wang, L.; Xiong, G. Reversible Data Hiding Using Multi-Pass Pixel Value Ordering and Prediction-Error Expansion. J. Vis. Commun. Image Represent. 2017, 49, 351–360. [CrossRef]
- Weng, S.; Chen, Y.; Ou, B.; Chang, C.-C.; Zhang, C. Improved K-Pass Pixel Value Ordering Based Data Hiding. *IEEE Access* 2019, 7, 34570–34582. [CrossRef]