Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **6**, Number **2**, May **2021**

Adaptive reversible data hiding based on multiple block scanning ways

Wenbin Zheng
School of Electronic, Electrical Engineering and Physics
Fujian University of Technology, Fuzhou 350118, China
zhwb190715@163.com

Ye Zhou
School of Computer Science and Mathematics
Fujian University of Technology, Fuzhou 350118, China
1962270759@qq.com

Tiancong Zhang
School of Electronic, Electrical Engineering and Physics
Fujian University of Technology, Fuzhou 350118, China
kushentian@163.com

Chunyu Zhang
College of information engineering
Xizang Minzu university, Xianyang 712082, China
zcy@xzmu.edu.cn

Fumin Zou
Fujian Key Lab for Automotive Electronics and Electric Drive
Fujian University of Technology, 350118, China
22652032@qq.com

Yunqing Shi
Department of Electrical and Computer Engineering
New Jersey Institute of Technology, Newark, NJ 07103 USA
shi@njit.edu

Corresponding author: Tiancong Zhang (kushentian@163.com)

Received March 2021; revised April 2021

Abstract. *The advantage of the improved pixel value ordering (IPVO) lies in introducing the positions of two pixels to calculate the difference value between these two pixels, so that each prediction error valued 0 or 1 can be embedded with one bit of data. However, in IPVO, each prediction error at -1 provides no capacity but merely degrade the visual quality. To increase the number of prediction errors valued 1 while decreasing the number of prediction errors valued -1, 16 different block scanning ways are designed to obtain the optimal scanning way generating the large inverse number for each block. Besides, the single-layer embedding strategy of Weng et al.'s method is employed in the proposed method to adaptively select pixels of a block for carrying data. By combining 16 different block scanning ways with the single-layer embedding strategy, the proposed method not only improves the rate-distortion performance but also reduces the computational complexity. Extensive experiments have proved that the performance of our scheme is superior to those of several related schemes.*
Keywords: Reversible data hiding, Multiple block scanning ways, Adaptive block modification

1. **Introduction.** The development of communication technology brings great convenience to people's life while causing great security risks. The information transmitted through the network may suffer from some malicious attacks, easily leading to privacy leaks, copyright disputes and other issues [1, 42]. Data hiding is one of the most effective methods used for copyright protection and content authentication [2–4], which hides imperceptibly secret data in the carrier and transmits this carrier in an unobtrusive way [43]. However, the most data hiding schemes cannot recover the stego carrier to the original state because of the permanent distortion introduced by embedding data. These data hiding schemes cannot be applied in some special applications such as military, medical processing [6, 7], in which the permanent distortion is unacceptable. To this end, a special branch of data hiding, namely reversible data hiding (RDH), was proposed to recover losslessly the original image after the secret data is extracted correctly [44].

In recent years, RDH has been studied extensively, and some valuable RDH schemes have been proposed. According to the adopted data embedding methods, all the RDH schemes can be divided into four classes: difference expansion (DE) [8–10], histogram shifting (HS) [12–14], lossless compression [16–18], and integer-to-integer transform [20, 21, 45]. The prediction error expansion (PEE), as an extension of DE, is widely used and studied because of its good performance [28–30]. Different from DE that embeds 1-bit data into the difference between two adjacent pixels, PEE embeds 1-bit data into the prediction error of each pixel. Afterwards, PEE has been improved from different perspectives such as compressed bitmaps [31, 32] and optimized predictors [33–36].

In 2013, Li *et al.* proposed an RDH scheme based on pixel value sorting (PVO) [37]. In their method, the cover image is divided into non-overlapping equal-sized pixel blocks. For a pixel block, all the pixels are sorted in ascending order to obtain the maximum and minimum pixels in the block. Taking the maximum pixel for example, it is predicted by the second largest pixel to generate one prediction error larger than or equal to 0. Only the prediction errors larger than or equal to 1 are modified, where the prediction error valued 1 are embedded with 1-bit data and the others are shifted by 1. This is similar for the minimum pixel. However, the PVO excludes the prediction error valued 0 from data embedment.

Later, Peng *et al.* [38] proposed an improved PVO (IPVO) that introduces the position information of pixels into PVO, enabling the prediction error valued 0 that are excluded from data embedment in PVO to embed data. Compared with PVO, Peng *et al.*'s method can obtain higher embedding capacity and better image quality because the number of the prediction errors valued 0 is larger than that number of the prediction errors valued -1.

Afterwards, many scholars have developed IPVO from different aspects [39, 40]. Among them, Weng *et al.* [40] proposed a two-layer embedding scheme that is capable of adaptively embedding data according to the smoothness of pixel blocks. They divide all the blocks into four levels according to the local complexity: high, medium, low and complex. The more the to-be-embedded data are, the higher the smoothness of a block is.

He *et al.* find that in Peng *et al.*'s method, each prediction error valued 1 is advantageous for the embedding capacity while the prediction errors valued -1 cannot provide any embedding capacity. To this end, He *et al.* proposed a flexible spatial position strategy based on regional characteristics [39]. In their scheme, eight spatial position modes of blocks are designed, and the optimal position mode is determined for blocks such that the number of prediction errors valued 1 is further larger than that of prediction errors

values -1. The experimental results in [39] show that different spatial position are of great help to the performance improvement of PVO.

Although Weng *et al.*'s scheme achieves good the rate-distortion performance, their method still does not fully exploit the position information of all pixels in a block. Inspired by He *et al.*'s scheme, we propose an improved IPVO scheme based on multiple block scanning ways. For a pixel block, before sorting, adopting different pixel scanning ways leads to different pixel sequences, directly affecting the original positions of all the pixels in the block. For the IPVO, the prediction error is strongly related with the original positions of all the pixels in a block. Therefore, in the proposed method, 16 pixel scanning ways are adopted such that an optimal scanning way causing that the number of prediction errors valued 1 is further larger than that of the prediction errors valued -1 is selected for scanning blocks. In a word, incorporating multiple block scanning ways into Weng *et al.*'s scheme can further improve the rate-distortion performance.

The rest of this paper is organized as follows. Section 2 introduces briefly two related works. Afterwards, Section 3 presents the proposed method, followed by the experimental results. Finally, Section 5 gives the conclusion.

2. **Related works.**

In this section, the principle of IPVO [38] is firstly described. Then, Weng *et al.*'s method [40] and He *et al.*'s method [39] are simplify described, respectively.

2.1. **IPVO.**

An image is divided into non-overlapping pixel blocks of size n. For one block, the pixels are scanned in the raster scan order to yield an original pixel list denoted by $\{z_1,\ z_2,\ \cdots,\ z_n\}$, and then are arranged in ascending order to generate a sorted pixel list $\left\{z_{\sigma(1)},\ z_{\sigma(2)},\ \cdots,\ z_{\sigma(n)}\right\}$ with $z_{\sigma(1)} \leq z_{\sigma(2)} \leq \cdots \leq z_{\sigma(n)}$. $z_{\sigma(2)}$ and $z_{\sigma(n-1)}$ as the reference pixels remain unchanged in the data embedment process, while $z_{\sigma(1)}$ and $z_{\sigma(n)}$ are modified to embed data or be shifted by 1.

For the largest two pixels $z_{\sigma(n-1)}$ and $z_{\sigma(n)}$, the prediction error $e_{\max}$ is calculated in Eq. (1).

$$e_{\max} = z_u - z_v, \tag{1}$$

where,

$$\begin{aligned} u &= \min(\sigma(n), \sigma(n-1)), \\ v &= \max(\sigma(n), \sigma(n-1)). \end{aligned} \tag{2}$$

When $e_{\max} \in \{0, 1\}$, $z_{\sigma(n)}$ is modified to $z_{\sigma(n)} + w$ via Eq. (3) to embed 1-bit data w ($w \in \{0, 1\}$). In contrast, $z_{\sigma(n)}$ with $e_{\max} > 1$ is shifted outwards to vacate embedding space for $z_{\sigma(n)}$ with $e_{\max} = 1$. This is similar for $e_{\max} < 0$.

$$z'_{\sigma(n)} = z_{\sigma(n-1)} + |e'_{\max}| = \begin{cases} z_{\sigma(n)} + w, & \text{if } e_{\max} = 1, \\ z_{\sigma(n)} + 1, & \text{if } e_{\max} > 1, \\ z_{\sigma(n)} + w, & \text{if } e_{\max} = 0, \\ z_{\sigma(n)} + 1, & \text{if } e_{\max} < 0, \end{cases} \tag{3}$$

where w is 1-bit data.

$$e_{\max}^{'} = \begin{cases} e_{\max} + w, & \text{if } e_{\max} = 1, \\ e_{\max} + 1, & \text{if } e_{\max} > 1, \\ e_{\max} - w, & \text{if } e_{\max} = 0, \\ e_{\max} - 1, & \text{if } e_{\max} < 1. \end{cases} \tag{4}$$

At the receiving end, according to Eq. (5) and Eq. (6), 1-bit data is extracted and $z_{\sigma(n)}^{'}$ is restored to $z_{\sigma(n)}$.

$$z_{\sigma(n)} = \begin{cases} z_{\sigma(n)}^{'} - e_{\max}^{'} + 1, & \text{if } e_{\max}^{'} \in \{1, 2\}, \\ z_{\sigma(n)}^{'} - 1, & \text{if } e_{\max}^{'} > 2, \\ z_{\sigma(n)}^{'} + e_{\max}^{'}, & \text{if } e_{\max}^{'} \in \{0, -1\}, \\ z_{\sigma(n)}^{'} - 1, & \text{if } e_{\max}^{'} < -1. \end{cases} \tag{5}$$

$$w = \begin{cases} e_{\max}^{'} - 1, & \text{if } e_{\max}^{'} \in \{1, 2\}, \\ -e_{\max}^{'}, & \text{if } e_{\max}^{'} \in \{0, -1\}. \end{cases} \tag{6}$$

For the smallest two pixels, the prediction error is calculated in the following manner:

$$e_{\min} = z_s - z_t, \tag{7}$$

where,

$$\begin{aligned} s &= \min(\sigma(1), \sigma(2)), \\ t &= \max(\sigma(1), \sigma(2)). \end{aligned} \tag{8}$$

After obtaining $e_{\min}$, $z_{\sigma(1)}$ is embedded with 1-bit of data to generate the corresponding stego pixel $z_{\sigma(1)}^{'}$ via Eq. (9).

$$z_{\sigma(1)}^{'} = \begin{cases} z_{\sigma(1)} - w, & \text{if } e_{\min} = 1, \\ z_{\sigma(1)} - 1, & \text{if } e_{\min} > 1, \\ z_{\sigma(1)} - w, & \text{if } e_{\min} = 0, \\ z_{\sigma(1)} - 1, & \text{if } e_{\min} < 0. \end{cases} \tag{9}$$

At the receiving end, one bit of data is extracted from $z_{\sigma(1)}^{'}$ according to Eq. (10).

$$w = \begin{cases} e_{\min}^{'} - 1, & \text{if } e_{\min}^{'} \in \{1, 2\}, \\ -e_{\min}^{'}, & \text{if } e_{\min}^{'} \in \{0, -1\}. \end{cases} \tag{10}$$

Subsequently, $z_{\sigma(1)}^{'}$ is restored to its original state $z_{\sigma(1)}$ using the rule:

$$z_{\sigma(1)} = \begin{cases} z_{\sigma(1)}^{'} + e_{\min}^{'} - 1, & \text{if } e_{\min}^{'} \in \{1, 2\}, \\ z_{\sigma(1)}^{'} + 1, & \text{if } e_{\min}^{'} > 2, \\ z_{\sigma(1)}^{'} - e_{\min}^{'}, & \text{if } e_{\min}^{'} \in \{0, -1\}, \\ z_{\sigma(1)}^{'} + 1, & \text{if } e_{\min}^{'} < -1. \end{cases} \tag{11}$$

From Eq. (2), we know that $e_{\max}$ is determined by u and v. That is, different u and v can produce two numbers, i.e., $-|e_{\max}|$ or $|e_{\max}|$, where $|e_{\max}|$ denotes of the absolute

value of $e_{\max}$. Similarly, it can be known from Eq. (8) that different s and t will produce $-|e_{\min}|$ or $|e_{\min}|$.

2.2. **Weng *et al.*'s method.** Weng *et al.* realize that IPVO only selects two pixels, i.e., the maximum and minimum pixels, for data embedment, and thus, the embedding capacity is limited. Taking a block with $z_{\sigma(n)} = z_{\sigma(n-1)} = z_{\sigma(n-2)}$ or $z_{\sigma(1)} = z_{\sigma(2)} = z_{\sigma(3)}$ for example, although its three largest (or smallest) pixels are strongly related with each other, it cannot provide more than two bits using IPVO. To this end, Weng *et al.* divide all the blocks into four categories based on the local complexity: high, moderate, low and complex. Note that all complex blocks are excluded from embedding data. The smoother a block is, the higher the embedding capacity is.

For a highly-smooth block, the largest three pixels and the smallest three pixels are used to carry 6-bit data at most. Specifically $z_{\sigma(n-3)}$ is used to predict $z_{\sigma(n)},\ z_{\sigma(n-1)},\ z_{\sigma(n-2)}$, respectively, to obtain three prediction errors. Since each prediction-error is capable of 1-bit data at most, three bits at most can be embedded in three largest pixels. Similarly, $z_{\sigma(4)}$ can be used to predict $z_{\sigma(1)},\ z_{\sigma(2)},\ z_{\sigma(3)}$ to achieve a capacity of 3 bits at most.

For a moderately-smooth block, the largest two pixels and the smallest two pixels are used to carry 4 bits at most. That is, $z_{\sigma(n-2)}$ is used to predict $z_{\sigma(n)}$ and $z_{\sigma(n-1)}$, respectively. Depending on this way, a moderately-smooth block can be embedded with 4 bits at most.

Similarly, a lowly-smooth block can carry 2 bits at most by only modifying the maximum and minimum pixels. In addition, Weng *et al.* design a two-layer embedding mode to achieve high embedding capacity.

2.3. **He *et al.*'s method.** In He *et al.*'s method, eight block spatial position modes are designed such that the optimal spatial position mode which can achieve the largest inverse number is determined for each block. For all the original positions of $r \times c$ pixels in a block, the inverse number is defined as the sum of $l_{1,2},\ l_{1,3},\ \cdots,\ l_{1,n},\ l_{2,3},\ l_{2,4},\ \cdots,\ l_{2,n},\ \cdots,\ l_{n-1,n}$, where $i \in \{1,\ 2,\ \cdots,\ r \times c\}$, $i < j$ and $j \in \{1,\ 2,\ \cdots,\ r \times c\}$, if $\sigma(i) > \sigma(j)$ and $i < j$, the corresponding $l_{i,j}$ is 1; otherwise $l_{i,j} = 0$. The larger the inverse number is, the greater the number of the prediction error valued 1 is, and the more data to be embedded are.

For one block with a size of $w \times h$, it is scanned according to one of eight block spatial position modes to generate one pixel list $\{z_1,\ z_2,\ \cdots,\ z_n\}$, and then it is sorted in ascending order to generate $\{z_{\sigma(1)},\ z_{\sigma(2)},\ \cdots,\ z_{\sigma(n)}\}$. To ensure that the inverse number remains unchanged before and after data embedment , $z_{\sigma(n)}$ that is to be modified during data embedment is modified to $z_{\sigma(n-1)}$, and similarly, $z_{\sigma(1)}$ is modified to $z_{\sigma(2)}$. Next, this block and two columns and two rows surrounding it are combined to form an expanded block of size $(w+2) \times (h+2)$. Then, the spatial position of this expanded block is calculated to get the inverse number. Finally, the way with the largest inverse number is determined as the scanning way of the block.

The larger the inverse number is, the larger the provided embedding space is, and the smaller the embedding distortion is. After the spatial position mode is determined, the data is embedded and extracted according to IPVO.

3. **The proposed scheme.** The main aim of the proposed method is to largely increase the number of prediction errors valued 1 by designing different scanning ways while decreasing the number of prediction errors valued -1.

From Eq. (2), it can be observed that prediction errors are strongly-related with $\sigma(n)$ and $\sigma(n-1)$ (i.e., the original scanning positions of the largest two pixels. Different scanning ways lead to different u and v. Based on the above considerations, we design 16

block scanning ways such that the optimal block scanning way of each block obtaining the largest inverse number is determined. Afterwards, Weng *et al.*'s method is used to adaptively embed data into.

Firstly, an image O of size $W \times H$ is divided into non-overlapping blocks $\{O_k\}_{k=1}^{N}$ of size $r \times c$, where N is the total number of blocks. For one block, suppose $\{z_1,\ z_2,\ \cdots,\ z_n\}$ is the original scanning pixel list obtained according to one of 16 block scanning ways, and let $\left\{z_{\sigma(1)},\ z_{\sigma(2)},\ \cdots,\ z_{\sigma(n)}\right\}$ be the sorted pixel list by sorting the original pixel list $\{z_1,\ z_2,\ \cdots,\ z_n\}$ in ascending order, where $z_{\sigma(1)} < z_{\sigma(2)} < \cdots < z_{\sigma(n)}$.

3.1. **Four levels of the local smoothness.** To exploit as many pixels as possible in a smooth block, we divide the smoothness of an image block O_k into four levels, and design adaptive embedding strategy for each level. It is well known that adjacent pixels are strongly related to each other. Therefore, the local complexity of a block is evaluated using the variance of the pixels surrounding this block.

For the block O_k of size $r \times c$ marked in white in FIGURE. 1 (a), it has $(r + c + 1)$ adjacent pixels marked in green in FIGURE. 1(a) that forms a set denoted by H. To evaluate the local complexity more accurately, we incorporate the pixels of O_k that are not used during data embedment (See the pixels marked in green in FIGURE. 1 (b)) into the set H to form a new set G. The variance of the set G is used to evaluate the smoothness of the block O_k, which is defined as:

$$\Delta = \sqrt{\frac{\sum_{k\in\{1,2,\cdots,r+1\}}(z_{k,c+1}-\mu)^2+\sum_{k\in\{1,2,\cdots,c\}}(z_{r+1,k}-\mu)^2+\sum_{k\in\{4,5,\cdots,n-3\}}(z_{\sigma(k)}-\mu)^2}{r+c+n-5}}, \quad (12)$$

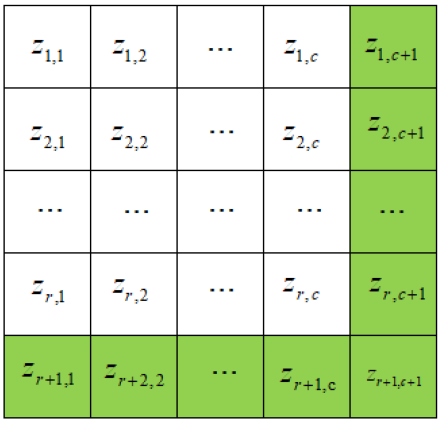
where μ is the average value of all the pixels of G. Based on the local complexity, all the blocks are divided into four levels, namely G_0 ($\Delta > vT$), G_1 ($\frac{vT}{2} < \Delta \leq vT$), G_2($\frac{vT}{4} < \Delta \leq \frac{vT}{2}$) and G_3($\Delta \leq \frac{vT}{4}$), where vT is a predefined threshold value. The smoother the local complexity of a block, the more the embedding capacity is and vice versa. To this end, four different embedding strategies are designed so that each block adaptively selects one of four strategies.

Strategy 1: Specifically, $\Delta > vT$ means that the correlations between pixels in the block are weak, and therefore, the block is excluded from data embedment . And meanwhile, the block is classified into the set G_0.

Strategy 2: A block with $\frac{vT}{2} < \Delta \leq vT$ is classified into G_1, and it can carry 2 bits at most by only changing the maximum and the minimum pixels. Taking the maximum pixel $z_{\sigma(n)}$ for example to explain its specific embedding process, the prediction error $e_{\max}$ is obtained by using the second largest pixel $z_{\sigma(n-1)}$ to predict $z_{\sigma(n)}$. Depending on $e_{\max}$, $z_{\sigma(n)}$ is modified via Eq. (3) to yield $z'_{\sigma(n)}$. Similarly $z'_{\sigma(1)}$ is obtained by changing $z_{\sigma(1)}$ via Eq. (9) after $e_{\min}$ is generated by using $z_{\sigma(2)}$ to predicting $z_{\sigma(1)}$.

Strategy 3: If $\frac{vT}{4} < \Delta \leq \frac{vT}{2}$, then the block belongs to the set G_2 and it can be embedded with 4 bits at most by modifying the largest two pixels and the smallest two pixels. Specifically, the third largest pixel $z_{\sigma(n-2)}$ is used to predict the largest two pixels $z_{\sigma(n-1)}$ and $z_{\sigma(n)}$, respectively, to generate two prediction errors $e_{1\max}$ and$e_{2\max}$. Afterwards, $z'_{\sigma(n-1)}$ (or $z'_{\sigma(n)}$) is embedded with 1 bit of data according to Strategy 2. Similarly, the third smallest pixel $z_{\sigma(3)}$ is used to predict the smallest two pixels $z_{\sigma(1)}$ and $z_{\sigma(2)}$, respectively, to generate two prediction errors $e_{1\min}$ and $e_{2\min}$. Subsequently, $z'_{\sigma(1)}$ (or $z'_{\sigma(2)}$) is embedded with 1 bit of data according to Strategy 2.

Strategy 4: A block having $\Delta \leq \frac{vT}{4}$ implies that it has strong correlation and each block is embedded with 6 bits at most. Note that the detailed embedding process is described in Strategy 3.



(a) The pixels of a block are marked in white while the pixels surrounding the block are marked in green.

(b) The pixels of a block marked in green represent unmodified pixels of the block.

Fig. 1. A block and its neighborhood.

3.2. **16 block scanning ways.** According to Eq. (1), the prediction error $e_{\max}$ between $z_{\sigma(n)}$ and $z_{\sigma(n-1)}$ is determined by the original spatial positions $\sigma(n)$ and $\sigma(n-1)$ of two pixels $z_{\sigma(n)}$ and $z_{\sigma(n-1)}$ in a block. To obtain more prediction errors valued 1, we design 16 kinds of different block scanning ways, and obtain the optimal way capable of achieving the largest inverse number for each block. FIGURE. 2 shows the 16 block scanning ways for 4×4-sized-sized block.

For a block of size 4×4 as shown in FIGURE. 3, 16 pixels are arranged in ascending order to get the sorted list {190, 191, 192, 192, 192, 192, 192, 193, 193, 193, 193, 194, 194, 194, 195, 196}. If the 1^{st} block scanning way is adopted to scan this block, the original positions of 16 sorted pixels are {14, 15, 2, 7, 11, 13, 16, 1, 4, 10, 12, 3, 5, 9, 6, 8}, where $\sigma(16) = 8$ and $\sigma(15) = 6$. According to Eq. (1), $u = 6$, $v = 8$ obtained via Eq. (2) means $e_{\max} = -1$. Since $e_{\max} = -1$, $z_{\sigma(16)}$ cannot be used for embedding data. Similarly, $e_{\min} = z_s - z_t = z_{14} - z_{15} = 190 - 191 = -1$ due to $\sigma(1) = 14$, $\sigma(2) = 15$, so $z_{\sigma(1)}$ is also excluded from embedding 1-bit data. However, if we adopt the 3rd block scanning way to scan this block, the sorted pixel list is still {190, 191, 192, 192, 192, 192, 192, 193, 193, 193, 193, 194, 194, 194, 195, 196}, but the original positions of 16 sorted pixels are changed to be {3, 2, 1, 4, 6, 10, 15, 5, 7, 13, 16, 8, 12, 14, 11, 9}. In such case, $\sigma(1) = 3, ; \sigma(2) = 2, ; \sigma(16) = 9, ; \sigma(15) = 11$, and therefore, $e_{\min} = z_s - z_t = z_2 - z_3 = 191 - 190 = 1$ and $e_{\max} = z_u - z_v = z_9 - z_{11} = 196 - 195 = 1$. By adopting different block scanning way, both $z_{\sigma(1)}$ and $z_{\sigma(16)}$ can carry 1-bit data.

3.3. **Optimal scanning way.** To further increase the embedding performance, the inverse number in He *et al.*'s method is introduced into Weng *et al.*'s method such that the optimal scanning way corresponding to the largest inverse number for each block is determined.

To ensure reversibility, the inverse number must remain unchanged before and after data embedment . Therefore, we need to utilize some unchanged pixels of a block during

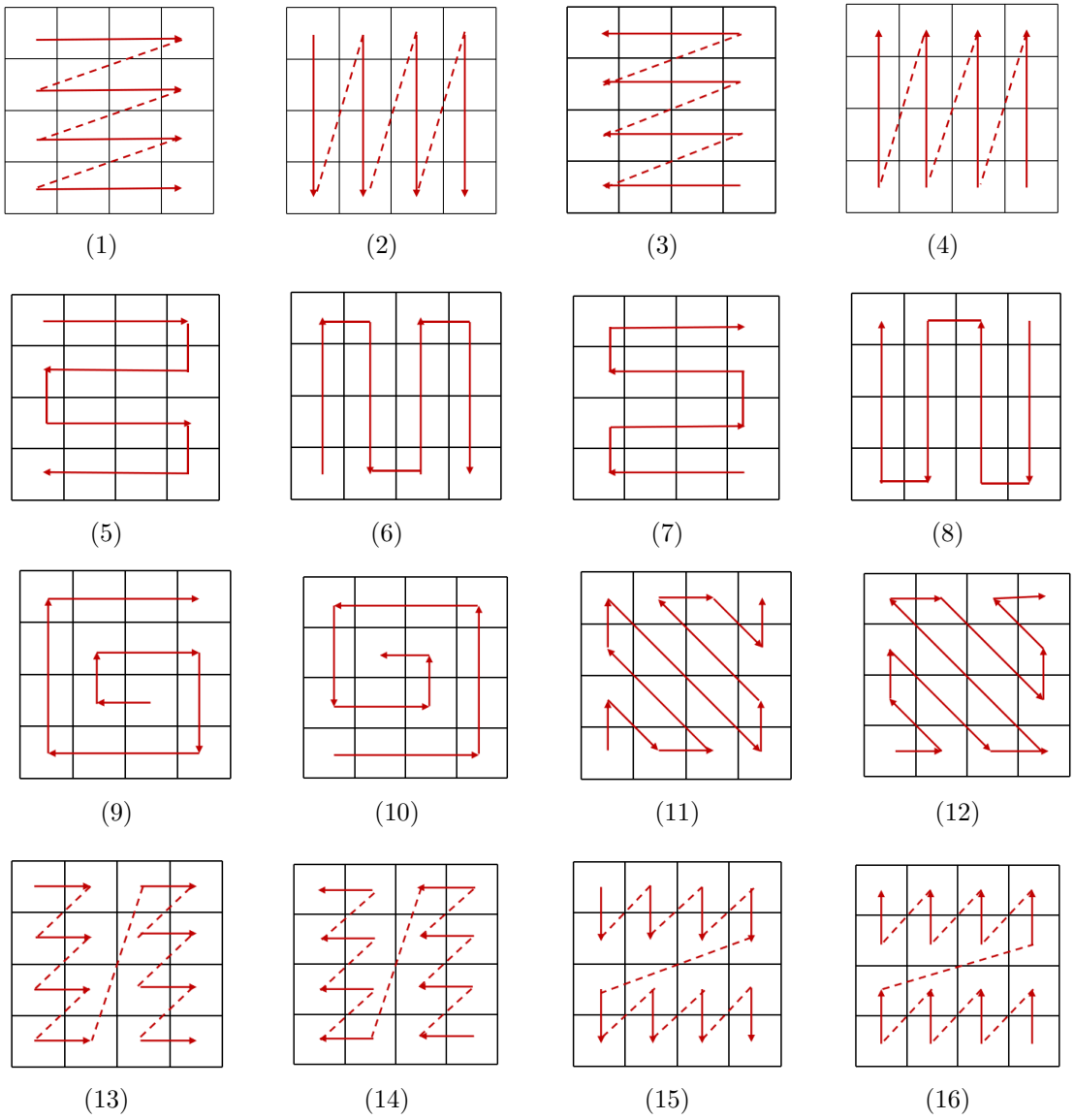


Fig. 2. 16 block scanning ways.

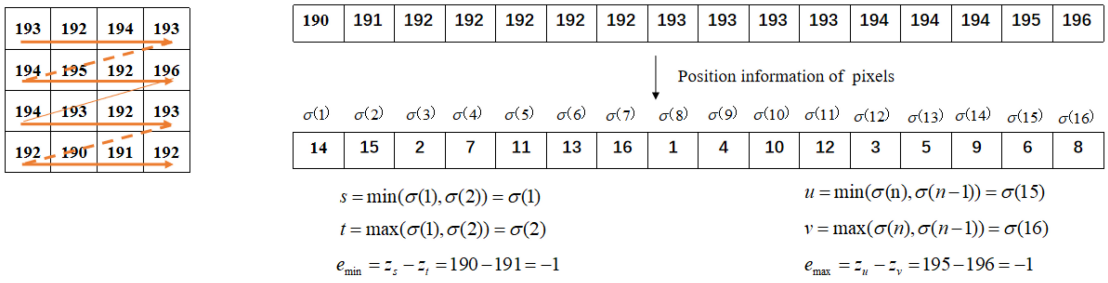
data embedment and some pixels surrounding the block for calculating the inverse number of this block. For example, for a block in G_2 since the largest two pixels and the smallest two pixels are used for data embedment, the remaining $r \times c - 4$ pixels remains unchanged, and therefore, they can be included into the set S_I that is used for calculating the inverse number. In addition, the pixels surrounding the blocks, i.e., the pixels in the right two columns and bottom two rows of the block, can also be included into the set S_I. Finally, each block scanning way corresponds to one inverse number of the set S_I that is calculated according to Section 2.3. The optimal scanning way corresponding to the largest inverse number is applied for this block.

3.4. **Data embedding process.** The following sub-section is used to describe the detailed process of data embedment step by step.

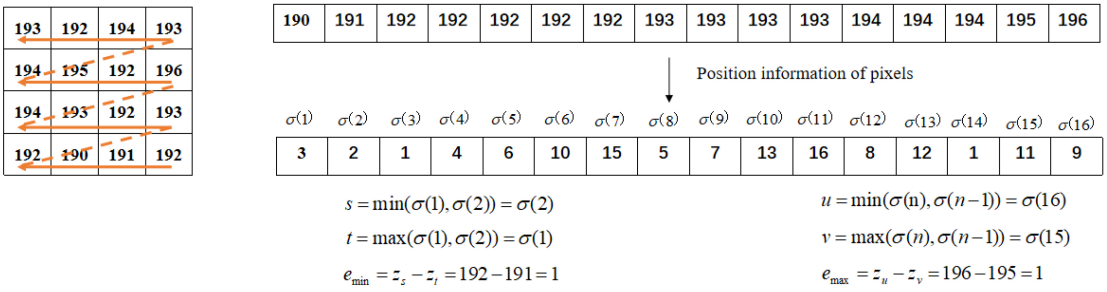
Input: A cover image O of size $W \times H$, the payload to-be-embedded.

Output: A stego $O^{'}$ image of size $W \times H$.

Step 1: Divide the cover image O into non-overlapping blocks of size $r \times c$.



(a) The number of prediction errors valued 1 by the 1^{st} scanning way.



(b) The number of prediction errors valued 1 by the 3^{nd} scanning way.

Fig. 3. Comparison of the number of pixels prediction errors at 1 produced by different scanning ways.

Step 2: According to Eq. (12), the local complexity, i.e., the variance of each block Δ is calculated, and then the blocks are divided into four categories based on the variance Δ: $G_0(\Delta > vT)$, $G_1(\frac{vT}{2} < \Delta \leq vT)$, $G_2(\frac{vT}{4} < \Delta \leq \frac{vT}{2})$, $G_3(\Delta \leq \frac{vT}{4})$.

Step 3: For a block O_k in $G_1 \cup G_2 \cup G_3$, it is scanned using one of 16 scanning ways in Section 3.2, and the corresponding inverse number is calculated according to Section 3.3. 16 scanning ways generate 16 inverse numbers, and the optimal way corresponding to the largest inverse number is applied to scan the block O_k for generating a pixel list $\{z_1, z_2, \cdots, z_n\}$. Then, it is sorted in ascending order to generate the sorted pixel list $\{z_{\sigma(1)}, z_{\sigma(2)}, \cdots, z_{\sigma(n)}\}$. For one block in G_1, its maximum and minimum pixels, namely $z_{\sigma(1)}$ and $z_{\sigma(n)}$ are modified using IPVO to carry 2 bits of data at most. One block of G_2 can be embedded with 4 bits by modifying the largest two and the smallest two pixels. In contrast, for a block of G_3, the largest three and the smallest three pixels are modified according to IPVO to carry 6 bits at most.

Step 4: Overflow or underflow occurs when the pixels valued 0 or 255 may exceed the range of [0,255] after data embedment . Therefore, a location map is created to record the locations of pixels valued 0 or 255 during data embedment . Afterwards, the location map is compressed losslessly to generate its compressed version LM. To avoid overflows or underflows, the values of pixels valued 0 are modified to 1 while the values of pixels valued 255 is modified to 254.

In addition to the to-be-embedded data, the auxiliary data denoted by A are also needed to be embedded into the cover image, where the auxiliary data A are composed of the block width r and the block height c, the threshold vT and the compressed location map LM.

After a small part of the payload is embedded into the first $\lfloor H/c \rfloor$ blocks in the raster scan order according to Step 4 to a temporary image I_t, the first $|A|$ pixels of I_t in the

raster scan order are used for carrying the A, where $|A|$ is the length of A. Specifically, the LSBs (least significant bits) of the first $|A|$ pixels are collected and replaced by A, and the collected LSBs form a bit stream B. B along with the rest of the payload are embedded into the remaining blocks using Weng *et al.*'s method. After all the blocks are modified, the stego image is generated.

3.5. **Data extraction and image restoration.** Data extraction and image restoration are processed using six steps.

Step 1: Extraction of auxiliary information.

The LSBs of the $|A|$ pixels are extracted according to the raster scan order to obtain the block width r and the block height c, the threshold vT and the compressed map LM.

Step 2: Divide the stego image into non-overlapping blocks of size $r \times c$.

Step 3: Smoothness classification of blocks.

The blocks are extracted according to the reverse order of the embedding process. For each block $O_k^{'}$, the variance Δ_k is calculated via Eq. (12), and then, it is classified into one of the following four classes G_0, G_1, G_2, or G_3 by comparing Δ_k with the threshold vT. If the block belongs to G_0, it remains unchanged.

Step 4: Scanning way determination.

Referring to the description in Section 3.3, we calculate 16 inverse numbers of one block. The scanning way corresponding to the largest inverse number is the optimal one.

Step 5: Data extraction.

$O_k^{'} \in G_1$ implies that only the maximum and minimum pixels of a block are modified during data embedment . Therefore, taking the stego maximum pixel $z_{\sigma(n)}^{'}$ for example to illustrate the detailed extraction process, the prediction error $e_{\max}^{'}$ is calculated via Eq. (1) by using the second largest pixel $z_{\sigma(n-1)}$ to predict $z_{\sigma(n)}^{'}$. Based on $e_{\max}^{'}$, one bit is extracted from $e_{\max}^{'}$ according to Eq. (6), and $z_{\sigma(n)}$ is restored according to Eq. (5). Similarly, $z_{\sigma(1)}$ is retrieved by Eq. (11) and one bit are extracted via Eq. (10) after the prediction error $e_{\min}^{'}$ is obtained using the second smallest pixel to predict the minimum pixel.

If $O_k^{'} \in G_2$, then the block contains 4 modified pixels, i.e., the largest two pixels and the smallest two pixels. For the largest two pixels, the third largest pixel $z_{\sigma(n-2)}$ is used to predict the largest two pixels $z_{\sigma(n-1)}^{'}$ and $z_{\sigma(n)}^{'}$, so that two prediction errors $e_{1\max}^{'}$ and $e_{2\max}^{'}$ are generated. Depending on $e_{1\max}^{'}$ or $e_{2\max}^{'}$, one bit is extracted via Eq. (6), and meanwhile, $z_{\sigma(n-1)}$ or $z_{\sigma(n)}$ are retrieved according to Eq. (5). Similarly, $z_{\sigma(1)}$ and $z_{\sigma(2)}$ can be completely retrieved.

Similarly, if $O_k^{'}$ belongs to G_3, six pixels including the largest three pixels and the smallest three pixels are needed to be retrieved. The detailed data extraction and pixel restoration process can refer to that of G_2.

Step 6: Image recovery.

The extracted data bits are separated into two parts: the stream B and the payload. The LSBs of the $|A|$ pixels of the first $\lceil |A| /512 \rceil$ rows are recovered by B replacement. Then, both the cover image and secret data are recovered.

4. **Experimental results.** In this section, we use ten 512×512-sized test images including Barbara, Airplane(F-16), Lena, Sailboat, Baboon, Peppers, Milkdrop, Woman, Wine, Zelda to illustrate the performance of the proposed method.

FIGURE. 5 shows the PSNR comparison of the 10 images at different payload size. It is clearly observed that Baboon performs the worst among all images while Woman performs the best. This is because Baboon is a strongly textured image, and contrastingly,

(a) Barbara (b) Airplane (F-16) (c) Lena (d) Sailboat (e) Baboon

(f) Peppers (g) Milkdrop (h) Woman (i) Wine (j) Zelda

Fig. 4. Ten test images.

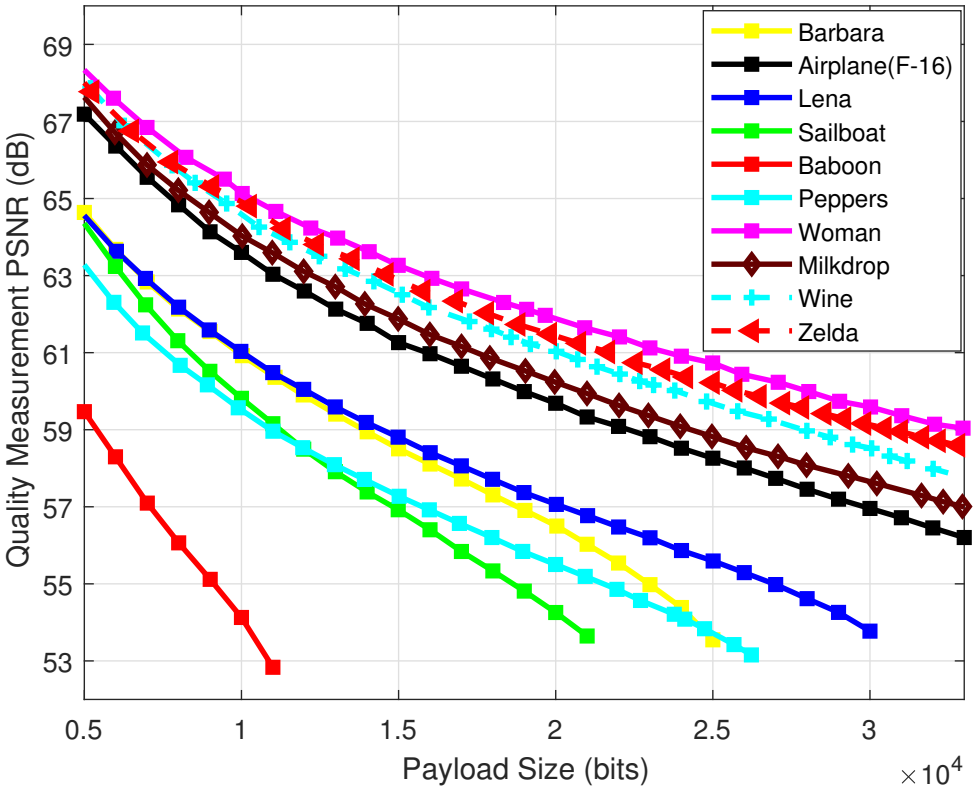
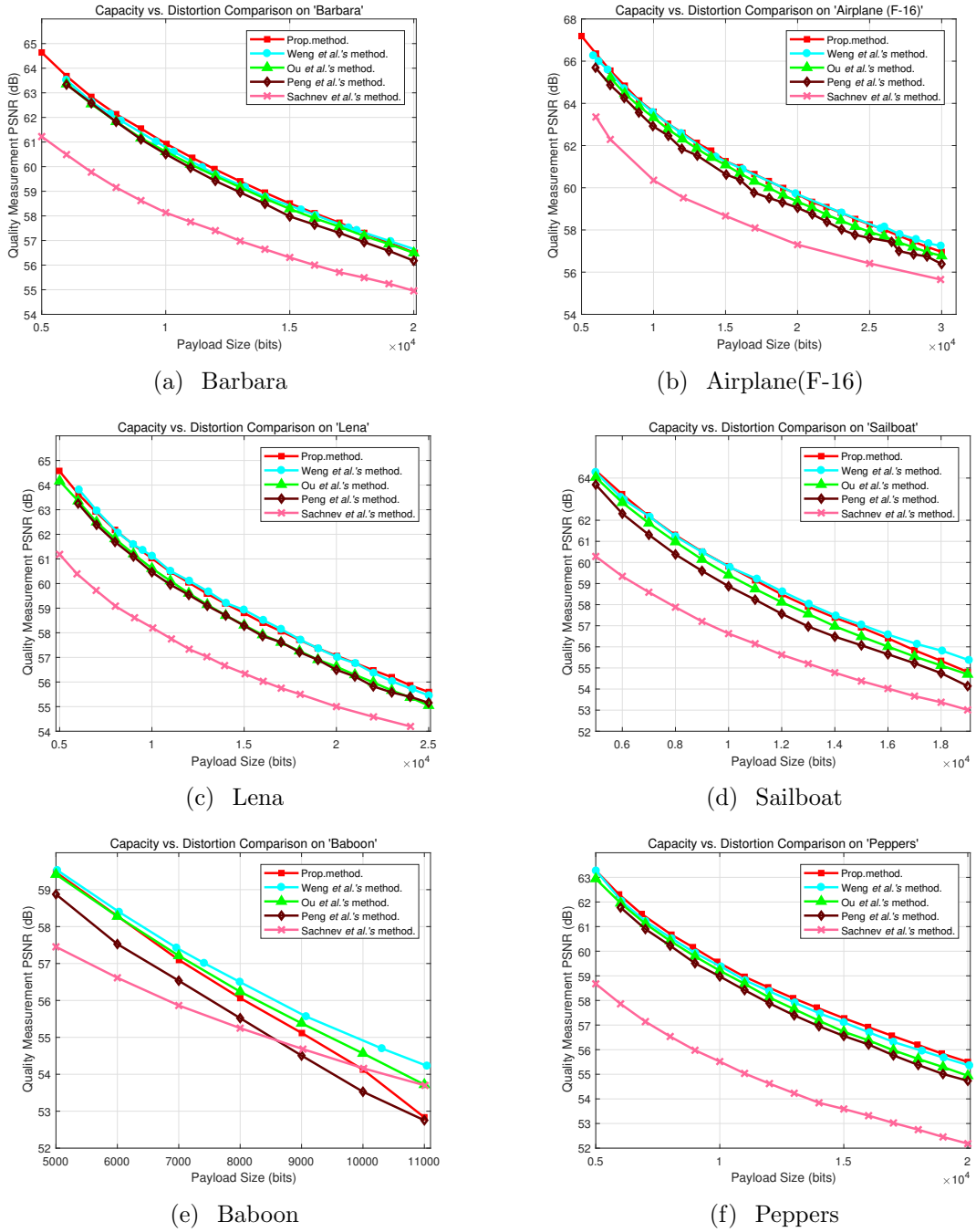


Fig. 5. Performance comparison of the proposed method for ten test images.

Woman is a highly smooth image. FIGURE. 5 indicates that the higher the smoothness of an image, the better the embedding performance is.

In addition, we implement performance comparison among several methods including Weng *et al.*'s method [40], Ou *et al.*'s method [41], IPVO [38], Sachnev *et al.*'s method [34] and our proposed method. As shown in FIGURE. 6, for five images, namely Barbara,



(a) Barbara (b) Airplane(F-16)

(c) Lena (d) Sailboat

(e) Baboon (f) Peppers

Fig. 6. Performance comparisons between the proposed method and the four compared methods including Weng *et al.* [40], Ou *et al.* [41], Peng *et al.* [38], Sachnev *et al.* [34].

Airplane(F-16), Lena, Sailboat, Peppers our method is superior to Ou *et al.*'s method, IPVO, Sachnev *et al.*'s method. No matter what the local complexity is, IPVO modifies indiscriminately the maximum and minimum pixels such that each block is uniformly embedded with 2 bits at most. In contrast, our method divides all blocks into four categories according to the local complexity, and adaptively embeds data into blocks located in different smooth regions. For example, each block located in the smoothest regions can be embedded with 6 bits at most, while the block located in the least smooth region is only embedded with 2 bits at most. In addition, we choose the optimal block

scanning way from 16 different block scanning ways for each block so that the number of prediction errors valued 1 are largely increased. Therefore, our method can not only improve the embedding capacity, but also have satisfactory image quality. Sachnev *et al.*'s method introduces the sorting technique to select preferentially smooth prediction errors produced by the rhombus predictor for embedding data. However, since the rhombus predictor is weaker than IPVO, the performance of Sachnev *et al.*'s method is inferior to that of Weng *et al.*'s method, Ou *et al.*'s method, IPVO and the proposed method. In Ou *et al.*'s method, the smooth blocks satisfying $z_{\sigma(n-3)} > z_{\sigma(n-2)} = z_{\sigma(n-1)} = z_{\sigma(n)}$ may not be utilized for data embedment, while our method can embed up to 6 bits into these smooth blocks.

Weng *et al.*'s method needs to adopt a two-layer embedding strategy to obtain the given embedding capacity, leading to high computational complexity. To decrease the computational complexity, our method only adopts one single embedding strategy of Weng *et al.*'s method. By combining 16 block scanning way into the single embedding strategy, the proposed method achieves comparable or even better embedding performance than Weng *et al.*'s method for five test images containing Pepper, Barbara, Airplane(F-16), Lena and Sailboat(see FIGURE. 6 for more details). For smooth images like Pepper, the proposed method can obviously increase the number of prediction errors valued 1. For complex images like Baboon, due to weak correlation between two adjacent pixels, it is very difficult to increase the number of prediction errors. However, Weng *et al.*'s method can provide more smooth blocks for embedding data by means of two-layer embedding strategy. Therefore, for Baboon, Weng *et al.*'s method outperforms the proposed method.

5. **Conclusions.** In this paper, unlike that Weng *et al.* adopt the same scanning way for all blocks, our method can choose the optimal scanning way from 16 different scanning ways for each block. On the other hand, our method adopts adaptive embedding strategy for blocks with different smoothness, and thus, our scheme achieves higher embedding capacity and better image quality compared with several existing schemes.

Acknowledgment. This work was supported in part by the National NSF of China under Grant 61872095, Grant 61571139, Grant 61872128, in part International Scientific and Technological Cooperation of Guangdong Province under Grant 2019A050513012, in part by the Open Project Program of Shenzhen Key Laboratory of Media Security under Grant ML-2018-03, in part by Fujian Science Fund for Distinguished Young Scholars under Grant 2020J06043.

REFERENCES

[1] W. B. Zheng, C. C. Chang, S. W. Weng, A novel adjustable RDH method for AMBTC-compressed codes using one-to-many map, *IEEE Access* , vol.8, no. 1, pp.13105-13118, 2020.

[2] K. Muhammad, J. Ahmad, H. Farman, Z. Jan, M. Sajjad, S.W. Baik, A secure method for color image steganography using gray-level modification and multi-level encryption, *Ksii Transactions on Internet, Information Systems*, vol.9, no.5, pp.1938-1962, 2015.

[3] K. Muhammad, M. Sajjad, S.W. Baik, A novel magic lsb substitution method (m-lsb-sm) using multi-level encryption and achromatic component of an image, *Multimedia Tools , Applications,*, vol.75, no.22, pp.1-27, 2016.

[4] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho, S.W. Baik, Dual-level security based cyclic 18 steganographic method and its application for secure transmission of key frames during wireless capsule endoscopy, *Multimedia Tools , Applications*, vol.75, no.22, pp.14867-14893,2016. .

[5] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho, S.W. Baik, Image steganography using uncorrelated color space and its application for security of visual contents in online social networks, *Future Generation Comput. Syst*, vol.86, pp.951-960, 2018.

[6] A. Khan, A. Siddiqa, S. Munib, S.A. Malik, *A recent survey of reversible watermarking techniques*, Inf. Sci., vol.279, no.20, pp.251-272, 2014.
[7] T. C. Zhang, S. W. Weng, S.-C. Chu, Overview of reversible data hiding, *Journal of Computer Security and Data Forensics*, vol. 1, no. 1, pp. 60-76, 2021.
[8] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, *IEEE Trans. Image Process.*,vol.13, no.8, pp.1147-1156, 2004.
[9] Y. Hu, H.K. Lee, J. Li, DE-based reversible data hiding with improved overflow location map, *IEEE Trans. Circuits Syst. Video Technol.*,vol.19, no.2, pp.250-260, 2009.
[10] J. Tian, Reversible data embedding using a difference expansion, *IEEE Trans. Circuits Syst. Video Technol.*, vol.13, no.8, pp.890-896, 2003.
[11] S.K. Lee, Y.H. Suh, Y.S. Ho, Reversible image authentication based on watermarking, *in: Proc. IEEE ICME.*, pp.1321-1324, 2006.
[12] X. Li, B. Li, B. Yang, T. Zeng, General framework to histogram-shifting-based reversible data hiding, *IEEE Trans. Image Process.*, vol.22, no.6, pp.2181-2191, 2013.
[13] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, Reversible data hiding, *IEEE Trans. Circ. Syst. Video Technol.*, vol.16, no.3, pp.354-362, 2006.
[14] P. Tsai, Y.C. Hu, H.L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, *Signal Process.*,vol.89, no.6, pp.1129-1143, 2009.
[15] G.R. Xuan, C.Y. Yang, Y.Z. Zhen, Y.Q. Shi, Reversible data hiding using integer wavelet transform and companding technique, *in: Proceedings of IWDW.*, vol.5, pp.23-26, 2004.
[16] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, *in: IEEE Trans. Image Process.*, vol.14, no.2, pp.253-266, 2005.
[17] J. Fridrich, M. Goljan, R. Du, Lossless data embedding-new paradigm in digital watermarking, *in: EURASIP J. Appl. Signal Process.*, vol.2002, no.2, pp.185-196, 2002.
[18] L. Kamstra, H.J.A.M. Heijmans, Reversible data embedding into images using wavelet technique and sorting, *IEEE Trans. Image Process.*, vol.14, no.12, pp.2082-2090, 2005.
[19] D. Coltuc, J.M. Chassery, Very fast watermarking by reversible contrast mapping, *IEEE Signal Process. Lett.*, vol.14, no.4, pp.255-258, 2007.
[20] D. Coltuc, Low distortion transform for reversible watermarking, *IEEE Trans. Image Process.*, vol.21, no.1, pp.412-417, 2012.
[21] X. Wang, X. Li, B. Yang, Z. Guo, Efficient generalized integer transforms for reversible watermarking, *IEEE Signal Process.*, vol.17, no.6, pp.567-570, 2010.
[22] S. Weng, Y. Zhao, J.S. Pan, R. Ni, Reversible watermarking based on invariability and adjustment on pixel pairs, *IEEE Signal Process. Lett.*, vol.45, no.20, pp.1022-1023, 2008.
[23] D. Coltuc, Low distortion transform for reversible watermarking, *IEEE Trans. Image Process*, vol.21, no.1, pp.412-417, 2012.
[24] D. Coltuc, J.M. Chassery, Very fast watermarking by reversible contrast mapping, *IEEE Signal Process. Lett.*, vol.14, no.4, pp.255-258, 2007.
[25] S.W. Weng, Y. Zhao, R.R. Ni, J.S. Pan, Parity-invariability-based reversible watermarking, *IET Electronics Lett.*, vol. 1, no.2, pp.91-95, 2009.
[26] D. Coltuc, Improved embedding for prediction-based reversible watermarking, *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no.3, pp.873-882, 2011.
[27] I.-C. Dragoi, D. Coltuc, Local-prediction-based difference expansion reversible watermarking, *IEEE Trans. Image Process.*, vol. 23, no.4, pp.1779-1790, 2014.
[28] I.-C. Dragoi, D. Coltuc, Adaptive pairing reversible watermarking, *IEEE Trans. Image Process.*, vol. 25, no.5, pp.2420-2422, 2016.
[29] W.-G. He, J. Cai, K. Zhou, G.Q. Xiong, Efficient PVO-based reversible data hiding using multistage blocking and prediction accuracy matrix, *J. Vis. Commun. Image R.*, vol. 46, pp.58-69, 2017.
[30] W. Hong, Adaptive reversible data hiding method based on error energy control and histogram shifting, *Opt. Commun.*, vol.285, no.2, pp.101-108, 2012.
[31] L. Kamstra, H.J.A.M. Heijmans, Reversible data embedding into images using wavelet technique and sorting, *IEEE Trans. Image Process.*, vol. 14, no.12, pp.2082-2090, 2005. .
[32] H.J. Kim, V. Sachnev, Y.Q. Shi, J. Nam, H.G. Choo, A novel difference expansion transform for reversible data embedding, *IEEE Trans. Inf. Forensic Secur.*, 4 (3) (2008) 456–465.
[33] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, *IEEE Trans. Image Process.*, vol. 16, no.3, pp.721-730, 2007.
[34] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking algorithm using sorting and prediction, *IEEE Trans. Circ. Syst. Video Technol.*, vol. 19, no.7, pp.989-999, 2009.

[35] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no.1, pp.187-193, 2010.
[36] Q. Pei, X. Wang, Y. Li, H. Li, Adaptive reversible watermarking with improved embedding capacity, *IEEE J. Syst. Softw.*, vol.86, no.11, pp.2841-2848, 2013.
[37] X.L. Li, B. Yang, T.Y. Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, *IEEE Trans. Image Process.*, vol.20, no.12, pp.3524-3533, 2011.
[38] F. Peng, X.L. Li, B. Yang, Improved PVO-based reversible data hiding, *Digit. Signal Process.*, vol.25, pp.255-265, 2014.
[39] W. G. He, Z. C. Cai, Y.M. Wang, Flexible spatial location-based PVO predictor for high-fidelity reversible data hiding, *Information Sciences.*, vol.520, pp.431-444, 2020.
[40] S.W. Weng, J.S. Pan, L.D. Li, Reversible data hiding based on an adaptive pixel-embedding strategy and two-layer embedding, *Information Sciences.*, vol.396, pp.144-159, 2016.
[41] B. Ou, X.L. Li, Y. Zhao, R.R. Ni, Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion, *Signal Process. Image Commun.*, vol.29, no.7, pp.198–205 ,2014.
[42] X. Huang and M. P. Nia and Q. Ding, Research on image encryption based on hyperchaotic system, *Journal of Network Intelligence*, vol.5, no.1, pp.10–22 ,2020.
[43] W. B. Zheng and C. C. Chang and J. H. Horng and J. Lin and Y.Q. Shi, An adjustable RDH method for AMBTC-compressed images using unilateral pixel value modification strategy, *Journal of Network Intelligence*, vol.5, no.2, pp.77–92 ,2020.
[44] C. F. Lee and Y. C. Li and S. C. Chu and J. F. Roddick, Data hiding scheme based on a flower-shaped reference matrix, *Journal of Network Intelligence*, vol.3, no.2, pp.138–151 ,2018.
[45] S. W. Weng and J. S. Pan and J. H. Deng, Invariability of remainder based reversible watermarking, *Journal of Network Intelligence*, vol.1, no.1, pp.16–22 ,2016.