



Improved PVO-based reversible data hiding: A new implementation based on multiple histograms modification[☆]



Bo Ou^{a,*}, Xiaolong Li^b, Jinwei Wang^c

^a College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

^b Institute of Computer Science and Technology, Peking University, Beijing 100871, China

^c School of Computer Science and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

ARTICLE INFO

Article history:

Received 26 September 2015

Revised 6 March 2016

Accepted 7 March 2016

Available online 10 March 2016

Keywords:

Reversible data hiding

Pixel-value-ordering

Multiple histograms modification

ABSTRACT

Pixel-value-ordering (PVO) technique is based on ranking the pixels within a block and often employed in reversible data hiding (RDH) to preserve the image fidelity. In this paper, an improved PVO-based algorithm is proposed to achieve a high-performance RDH. We formulate the PVO embedding as a problem of multiple histograms modification (MHM) and make two contributions. Firstly, the existing PVO-based algorithms are generalized in an MHM form and the systematic analysis of performance is offered by considering the parameter determination. Secondly, based on the new framework, an improved implementation of PVO embedding is proposed to adaptively determine the MHM manner by solving the distortion minimization of multiple histograms, which can handle different characteristics in image areas. Experimental results show the superiority of the new algorithm over the state-of-the-art methods.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Due to the ability to reconstruct the original medium after data extraction, reversible data hiding (RDH) [1,2] has been identified as an effective way for copyright and integrity protection in sensitive scenarios, such as remote sensing, medical diagnose and judicial image processing. By exploiting the rich correlations, RDH can hide the secret message in an image and ensure the embedded data being invisible to human perception. In this sense, RDH can also be used to transmit the secret message by using the widely applied media on the Internet as carriers. So far, there is a rapid increase of applications using RDH including image authentication [3], medical image processing [4], video transmission [5], recovery of vector map in two-dimensional CAD engineering graphics [6], etc. Recently, with the development of high-resolution display devices, the visual perception of marked image is becoming more and more important. The less modifications on an image, the better visual perception and the less possibly the marked image is suspected. So, a method of high-fidelity RDH with a very low embedding distortion is greatly needed in most existing applications.

Existing RDH methods are mainly based on lossless compression [7–11], difference expansion (DE) [12–15], histogram shifting

(HS) [16–23], prediction-error expansion (PEE) [24–37] and integer-to-integer transform [38–40]. Among them, one classical approach of RDH is the use of prediction-error for data embedding, which is also well known as PEE [24]. Generally, in PEE-based methods, there are two basic steps, i.e., histogram generation and histogram modification. In the first step, the prediction-error of a pixel is calculated as the difference between the pixel and its estimated value. The prediction-error histogram (PEH) is then generated by counting the frequency of prediction-errors. Since there exists rich redundancy in a nature image, the derived prediction-errors are closed to zero and the PEH behaves like a Laplacian-like distribution. The statistical property of prediction-errors has obvious advantages in reducing the embedding distortion for a given capacity than the direct use of pixels. Herein, histogram generation can be viewed as a procedure to derive a compact feature representation of image. To yield a sharp PEH, the advanced prediction techniques [29,30,37] are often applied. The underlying idea is simple: the more accurate the prediction, the sharper the PEH distributes and the less distortion for a capacity. However, these methods are based on the one-dimensional (1D) PEH as the pixels are modified one by one, and the complex correlations within an image cannot be fully exploited in such a low dimensional space. To remedy this, the high-dimensional PEH is utilized as complement [15,33]. In pairwise PEE [33], the two-dimensional (2D) PEH is employed for data embedding by modifying every two adjacent prediction-errors together, and an obvious improvement is usually obtained than the 1D methods.

[☆] This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author.

E-mail addresses: oubo@hnu.edu.cn (B. Ou), lixiaolong@pku.edu.cn (X. Li), wjwei@nuist.edu.cn (J. Wang).

In histogram modification, the performance of PEE is improved by discriminately modifying the histogram bins corresponding to the pixels in the smooth and rough areas. The average embedding distortion on a prediction-error is determined by its magnitude. The smaller the magnitude of prediction-error, the less distortion caused by expansion. So, for the low capacity case, the most common way of PEH modification is to use the histogram bins 0 and -1 to carry bits and shift the other bins for the sake of reversibility. The main advantage lies in that it is easy to understand and simple to implement, yet lacking of the adaptivity to modify the pixels according to the image content. Recent works [19,41–43] show that, instead of fixing the modification manner, the optimal expansion bins could be adaptively chosen according to the image content. In some cases, the larger-magnitude bins can be used to yield a superior embedding performance than the bins 0 and -1 . However, the determination for expansion bins is a complex optimization problem to solve, and the price paid for an improved bin selection is the increase in computation complexity. Up to now, there is no general rule yet for finding the optimal expansion bins, and the better bins are selected by hand or iterative search based on a predefined manner. Wang and Ni [41] proposed to use the generic algorithm to determine the optimal bins for different capacities. Li et al. [43] proposed a simplified parameter determination mechanism for the bin selection of multiple histograms.

Notice that, the histogram generation is the basis for PEE as the embedding performance is intimately related to the derived PEH. It discovers a low dimensional representation of image that is favorable for RDH. To improve the histogram generation, employing an advanced predictor is by far the most used means, but it is by no means the only one. In the recent years, a new prediction technique based on pixel-value-ordering (PVO) continues to appear in literatures [44–47]. PVO prediction is based on ranking the pixels within a pixel block, and can achieve a more accurate prediction than the previous methods especially for a large-sized block. It is verified to be effective in generating a very sharper PEH and preserving a high-fidelity of marked image for the low capacity case. Li et al. [44] first proposed a PVO-based algorithm, in which the image is divided into blocks, and then the maximum/minimum pixel of a block is predicted by the second largest/smallest one. Recently, Qu and Kim [47] extended the PVO algorithm in a pixel-wise manner, where a pixel is predicted by the maximum or the minimum in the context. Since the smooth pixels are sufficiently used for data embedding, the capacity can be further increased while the high-fidelity property is still preserved.

Although the existing PVO-based methods demonstrate an impressive embedding performance for low capacity case, they still have some limitations. Firstly, in the histogram modification process, PVO embedding modifies the prediction-error in a fixed manner, i.e., the histogram bin 0 or 1 is always used to embed data and the others are shifted. As mentioned before, such a fixed histogram modification manner is not the advisable choice for all the cases. Secondly, by computing the pixel's local complexity, the PEH generated in PVO embedding can be further split into multiple sub-histograms with different noise levels. In the conventional PVO methods, these PEHs are modified uniformly according to the same manner. But, the uniform modification ignores the diversity of different PEHs, and may not guarantee the optimal performance for all cases. From this point of view, we believe that the full potential of PVO embedding has not been discovered yet, and aim to develop an adaptive bin-selection mechanism for the advanced PVO embedding.

In this paper, we propose a new PVO-based RDH method based on multiple histograms modification (MHM). The key observations behind our approach are that the prediction-errors with different

local complexities have discriminate statistical properties, and could be modified flexibly. Thus, our objective is to design a mechanism that can adaptively determine the expansion bins for multiple histograms of PVO. To achieve this target, we first generalize the PVO embedding as a problem of expansion-bin-selection to encapsulate a number of PVO algorithms, and then simplify the multiple-parameter determination problem of MHM into a two-parameter one. In the generalized framework, one can see that the conventional PVO embedding tends to modify multiple PEHs homogeneously as it lacks a guidance to automatically find the optimal parameters for MHM. Then, a two-stage mechanism is proposed to optimize the bin selection, which takes into account of both heuristic and automatic determinations. In the first stage, 54 heuristic bin-selections are given as candidates, which includes the conventional PVO manner, and an iterative search is then employed to find the optimal parameters. Next, in the second stage, the parameters are further locally tuned by considering the embedding efficiency. Unlike the single PEH modification, MHM is region-based, leading to a more efficient data embedding process which can further reduce the embedding distortion. All experimental results show that the proposed method consistently enhances the performance of PVO embedding, and yields a superior performance than some state-of-the-art methods as well.

Interest in the area of high-fidelity RDH has been renewed due to the emerging algorithms of adaptive embedding. In these algorithms, the adaptive embedding is extended to the selection of expansion bin, the design of high-dimensional mapping, and the discriminate modifications for multiple histograms, which are not only challenging in optimization but also computationally demanded. In the future, one way for the advanced data embedding is to finely modify the different image regions according to their content, which requires the determination of multiple parameters related to different histograms. However, few works have dedicated to exploiting the embedding manner for multiple histograms and its optimization. So, in this work, the study of MHM and the corresponding simple yet effective mechanism for optimization would be valuable.

The rest of paper is organized as follows. Section 2 presents a brief review of the related work. The generalization of PVO embedding in terms of MHM is proposed as well. Section 3 introduces the proposed MHM scheme for PVO embedding in details, where a simple mechanism for adaptively determining the expansion bins is proposed. Experimental results are provided in Section 4. Finally, Section 5 concludes this paper.

2. Related works

PVO-based techniques have been well studied in recent works [44–47] and a remarkable progress has been made. As a branch of PEE, the PVO embedding also consists of two steps, i.e., histogram generation and histogram modification. Despite of the histogram generation, the existing PVO-based methods share a common manner for histogram modification. That is, based on the derived PEH, the histogram bin 0 or 1 is used to carry data, and the others are shifted to vacate space. For simplicity, we only review the latest work proposed by Qu and Kim [47] to illustrate the histogram modification manner of PVO.

2.1. Qu et al.'s pixel-wise PVO embedding algorithm

In [47], a pixel p_i is predicted by its context U_i , which contains the right and bottom neighbors. For simplicity, the minimum and maximum values in U_i are denoted as a_i and b_i , respectively. For the case of $a_i < b_i$, by comparing the pixel p_i with a_i and b_i , respectively, the prediction \hat{p}_i is obtained as

$$\hat{p}_i = \begin{cases} b_i, & \text{if } p_i \geq b_i \\ a_i, & \text{if } p_i \leq a_i \\ \emptyset, & \text{if } a_i < p < b_i \end{cases} \quad (1)$$

where \emptyset indicates that the pixel cannot be predicted and will be skipped during data embedding. For the case of $a_i = b_i$, the prediction \hat{p}_i is calculated as

$$\hat{p}_i = \begin{cases} a_i, & \text{if } p_i \geq a_i \\ a_i - 1, & \text{if } p_i \leq a_i - 1 \end{cases} \quad (2)$$

The prediction-error e_i is then computed as $e_i = p_i - \hat{p}_i$ if there exists a prediction for the pixel. By counting the occurrences of prediction-errors, the corresponding PEH is generated as

$$h(k) = \#\{1 \leq i \leq N : e_i = k\} \quad (3)$$

where N denotes the number of pixels, and $k \in [-255, 255]$ for an 8-bit gray-scale image. Evidently, as some rough pixels cannot be predicted by the context in (1), the total occurrences of prediction-errors $\sum_{k=-255}^{255} h(k)$ is less than N . This can also explain why the PVO embedding can achieve a high-fidelity performance, as the data embedding mainly occurs in the smooth areas. Based on the PEH, the marked prediction-error e'_i is obtained as

$$e'_i = \begin{cases} e_i + m, & \text{if } p_i = b_i \\ e_i - m, & \text{if } p_i = a_i \\ e_i + 1, & \text{if } p_i > b_i \\ e_i - 1, & \text{if } p_i < a_i \end{cases} \quad (4)$$

where m is a binary bit. To embed a data bit, the prediction-error 0 is added/subtracted by 1 or kept unchanged, respectively. The reversibility is guaranteed by the fact that the prediction of a pixel is invariant before and after embedding. Thus, at decoder, one can recover the original value and extract the data bit by comparing the marked pixel with its prediction.

Considering that $p'_i = \hat{p}_i + e'_i$, the histogram modification can also be interpreted in terms of pixel value. The marked pixel value p'_i is obtained as

$$p'_i = \begin{cases} p_i + m, & \text{if } p_i = b_i \\ p_i - m, & \text{if } p_i = a_i \\ p_i + 1, & \text{if } p_i > b_i \\ p_i - 1, & \text{if } p_i < a_i \\ p_i, & \text{if } a_i < p_i < b_i \end{cases} \quad (5)$$

Accordingly, the illustration of histogram modification based on (5) is given in Fig. 1, where x-axis denotes the pixel value. In the figure, it is clearly that the pixels in the range $[a_i + 1, b_i - 1]$ are not used for data embedding.

2.2. MHM framework for PVO embedding

Once a specific histogram is given, the embedding performance is determined by the selection of expansion bins. So, the optimization of PVO algorithms can be simplified as a function of

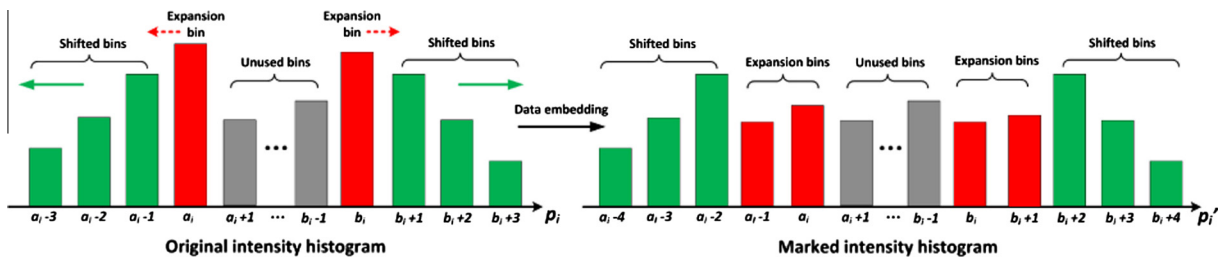


Fig. 1. Histogram modification used in PVO embedding, where x-axis denotes the values of pixel p .

expansion-bin-selection parameterized by the two offsets r and t . Taking the minimum and maximum values a_i and b_i as benchmark, two new histograms can be defined as

$$g_1(r) = \#\{1 \leq i \leq N : p_i - a_i = r\} \quad (6)$$

and

$$g_2(s) = \#\{1 \leq i \leq N : p_i - b_i = s\}, \quad (7)$$

respectively. For the sake of reversibility, the parameters r, s should satisfy that $b_i + s > a_i + r$ for all $i \in [1, N]$. Suppose that the bins r and s are selected for expansion, the capacity $EC(r, s)$ and the embedding distortion $ED(s, t)$ can be generalized as

$$EC(r, s) = g_1(s) + g_2(t), \quad (8)$$

and

$$ED(r, s) = \frac{1}{2} (g_1(r) + g_2(s)) + \sum_{k_1=r-1}^{-255} g_1(k_1) + \sum_{k_2=s+1}^{255} g_2(k_2), \quad (9)$$

respectively. Evidently, without considering the case of $a_i = b_i$, the modification manner shown in Fig. 1 is a special case of the new framework, where (r, s) is chosen as $(0, 0)$. One can obtain that the capacity $EC(0, 0) = g_1(0) + g_2(0)$, indicating that the to-be-expanded pixels are chosen as the ones equaled to the maximum or minimum in the context.

In addition, the new framework can be extended for MHM. To modify the smooth pixels at first, a pixel-selection technique is usually used to distinguish the smooth ones by measuring the noise level. For the PVO algorithm, the noise level NL_i of a pixel can be calculated as the difference between the maximum and the minimum in the context, i.e., $NL_i = b_i - a_i \geq 0$. Here, by means of noise level, the histograms defined in (6) and (7) can be further classified into multiple sub-histograms as

$$g_1(r_t, t) = \#\{1 \leq i \leq N : p_i - a_i = r_t, b_i - a_i = t\} \quad (10)$$

and

$$g_2(s_t, t) = \#\{1 \leq i \leq N : p_i - b_i = s_t, b_i - a_i = t\}, \quad (11)$$

respectively. Since the parameter $t \in [0, 255]$, let the parameter vectors for the multiple histograms be $\mathbf{r} = (r_0, r_1, \dots, r_{255})$ and $\mathbf{s} = (s_0, s_1, \dots, s_{255})$, respectively, where the element in the vector specifies the expansion bin for a histogram. For example, $r_0 = -1$ and $s_0 = 0$ represent that, for the histogram with $t = 0$, the two expansion bins are chosen as -1 and 0 , respectively. Considering the vectors of MHM, the capacity $EC(\mathbf{r}, \mathbf{s})$ and the embedding distortion $ED(\mathbf{r}, \mathbf{s})$ are obtained as

$$EC(\mathbf{r}, \mathbf{s}) = \sum_{t=0}^{255} g_1(r_t, t) + \sum_{t=0}^{255} g_2(s_t, t), \quad (12)$$

and

$$ED(\mathbf{r}, \mathbf{s}) = \frac{1}{2} \left(\sum_{t=0}^{255} g_1(r_t, t) + \sum_{t=0}^{255} g_2(s_t, t) \right) + \sum_{t=0}^{255} \sum_{k_1=r_t-1}^{-255} g_1(k_1, t) + \sum_{t=0}^{255} \sum_{k_2=s_t+1}^{255} g_2(k_2, t) \quad (13)$$

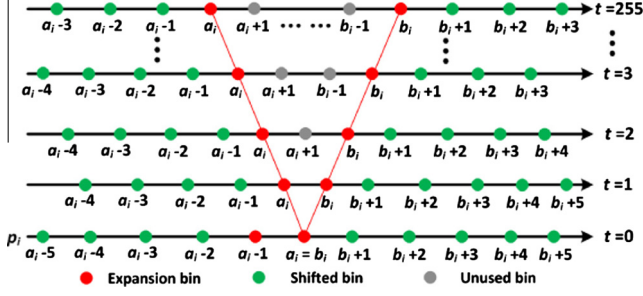


Fig. 2. MHM for Qu et. al's method, where the pixels equal to either the maximum or the minimum are utilized for data embedding. It is clearly that, for multiple histograms, the conventional modification manner is independent of histograms, and cannot guarantee the optimal performance for all the cases.

Accordingly, Qu et al.'s modification manner can be plotted in terms of MHM as shown in Fig. 2. Note that, in Qu et al.'s work, for the case of $t = 0$ (i.e., $a_i = b_i$), both of bins a_i and $a_i - 1$ (i.e., $r_0 = -1$ and $s_0 = 0$) are utilized for expansion.

To better illustrate the MHM framework, an example on how the pixels with different noise levels are discriminately modified is given in Fig. 3. In the example, the context only includes three neighbors for simplicity, and the data embedding for two pixels with $NL_i = 1$ and $NL_i = 2$ is given. In Fig. 3(a), since the noise level is computed as 1, the encoder refers to the offsets $r_1 = -1$ and $s_1 = 1$ in the predefined MHM manner for prediction. As the pixel $p_i = 156$ is less than $a_i + s_1$, the prediction is computed as $\hat{p}_i = a_i + r_1 = 157$, and the marked pixel is obtained as $p'_i = p_i - 1$. Similarly, in Fig. 3(b), by referring to the offset vector, one can determine the offsets $r_2 = 0$ and $s_2 = 1$ for the current data embedding. Consequently, the pixel is predicted as $\hat{p}_i = b_i + s_2 = 160$ and the marked pixel is obtained as $p'_i = p_i + m = 161$ for a data bit $m = 1$. It is found that the modification manner on a type of pixels with the noise level $NL_i = t$ is determined by the predefined offset vectors \mathbf{r} and \mathbf{s} . By setting different offset values, the pixels in the smooth and rough areas are allowed to be modified differently, leading to a potential for the adaptive embedding on the image.

From the point of view of MHM, one can see that there is a drawback for the conventional PVO embedding shown in Fig. 2. That is, no matter how a histogram distributes, the histograms with different noise levels are uniformly modified and the pixels

that are equal to the maximum or the minimum are always chosen for expansion. In this case, the parameters $(r_0, s_0, r_1, s_1, \dots, r_{255}, s_{255})$ for the 255 histograms are all specified as 0, except that $r_0 = -1$. The modification manner is independent of the histogram. It can be inferred that the conventional PVO embedding cannot guarantee the optimal performance for all the cases. Consequently, it is desired to adaptively modify the multiple histograms based on their characteristics. The key issue here is to give a guidance for bin selection and then mitigate the computation problem of determining the optimal expansion bins for multiple histograms.

3. Proposed method

In this section, we develop a two-stage bin-selection mechanism and use it to optimize the embedding performance of MHM in PVO embedding. The benefit of this two-stage mechanism is twofold: firstly, the computational complexity can be controlled at a low level and the processing speed is fast; secondly, the parameters of MHM can be finely tuned based on the criteria of embedding efficiency, and thus different PEHs can be modified flexibly. Before the presentation, we discuss the ways to improve the bin-selection at first. The motivation for determining the better expansion bins of multiple histograms are based on the following observations.

- Better expansion bins can be selected as the ones which can not only increase the capacity but also reduce the embedding distortion. In this case, the embedding performance is definitely improved.
- Better expansion bins can be selected as the ones that can improve the embedding efficiency, which is computed as the average distortion for embedding a bit. In this case, although either the capacity or the distortion may be decreased or increased, respectively, the total embedding performance can be still improved in general as the embedding efficiency is increased.

For better illustration, we take the determination of the left expansion bin for example as shown in Fig. 4. In the figure, for two cases, the arrow points out a better selection for expansion based on the above guidance.

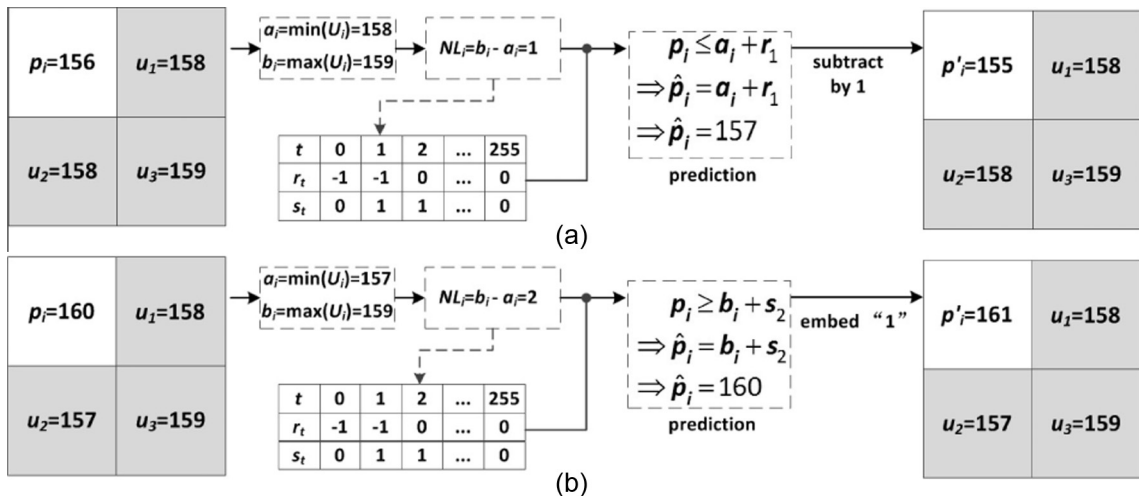


Fig. 3. Examples for the data embedding of MHM, where the modification manner of a pixel is related to its noise level NL_i . (a) Data embedding for $NL_i = 1$. (b) Data embedding for $NL_i = 2$.

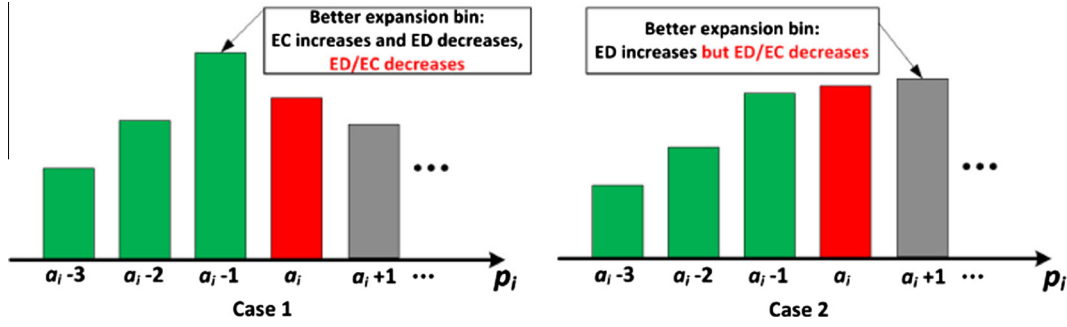


Fig. 4. Observations for an improved bin selection: the expansion bin used in the conventional PVO embedding can be replaced by the ones that can improve the embedding efficiency.

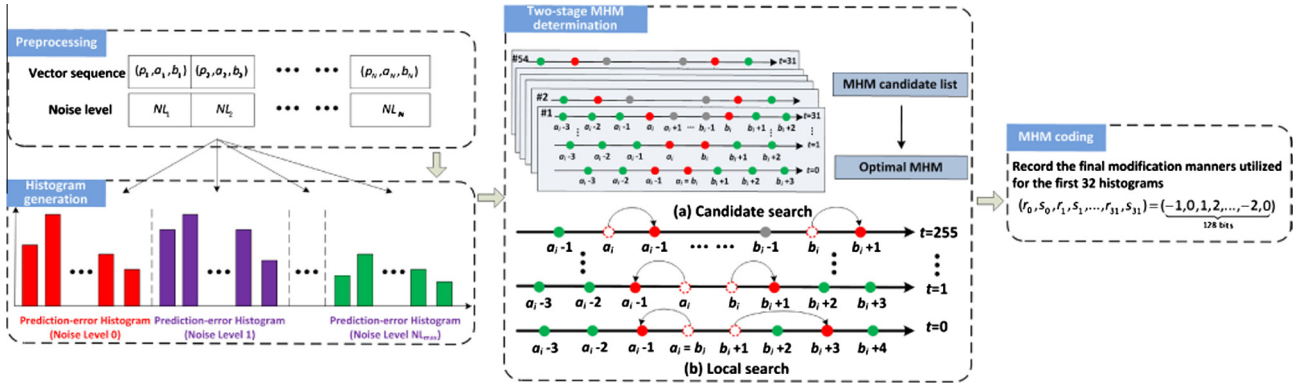


Fig. 5. The overview of embedding process in the proposed method.

3.1. Two-stage bin-selection mechanism for MHM

Our methodology is mainly based on selecting a modification manner from a predefined MHM manner list. Once the MHM manner is determined, it will be further tuned by locally searching a better choice according to the criterion of embedding efficiency. The whole framework of the proposed bin-selection mechanism is given in Fig. 5, including preprocessing, histogram generation, two-stage MHM determination and MHM coding.

In the preprocessing step, for each pixel, we can obtain a triple-element vector (p_i, a_i, b_i) corresponding to the pixel value, the minimum and maximum in the context, respectively. The context $U_i = (u_1, \dots, u_n)$ is defined in Fig. 6, where the number of context pixels $n \in [1, 15]$. The noise level of p_i is calculated as $NL_i = b_i - a_i$. The multiple histograms with different noise level t are generated using (6) and (7), where the expansion bins (i.e., $a_i + r_t$ and $b_i + s_t$) are parameterized by two offsets r_t and s_t ,

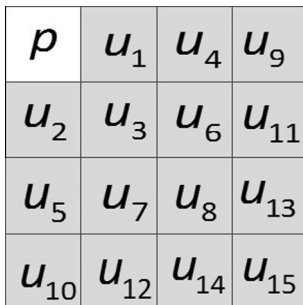


Fig. 6. Context U for a pixel p .

respectively. For the pixel p_i with the noise level t , the prediction \hat{p}_i is obtained as

$$\hat{p}_i = \begin{cases} b_i + s_t, & \text{if } p_i \geq b_i + s_t \text{ and } NL_i = t \\ a_i + r_t, & \text{if } p_i \leq a_i + r_t \text{ and } NL_i = t \\ \emptyset, & \text{if } a_i + r_t < p_i < b_i + s_t \text{ and } NL_i = t \end{cases} \quad (14)$$

Note that the offset parameters r_t and s_t should satisfy the constraint that $a_i + r_t < b_i + s_t$ with $NL_i = t$ for all $i \in [1, N]$. Consequently, the marked pixel p'_i is modified as

$$p'_i = \begin{cases} p_i + m, & \text{if } p_i = b_i + s_t \text{ and } NL_i = t \\ p_i - m, & \text{if } p_i = a_i + r_t \text{ and } NL_i = t \\ p_i + 1, & \text{if } p_i > b_i + s_t \text{ and } NL_i = t \\ p_i - 1, & \text{if } p_i < a_i + r_t \text{ and } NL_i = t \\ p_i, & \text{if } a_i + r_t < p_i < b_i + s_t \text{ and } NL_i = t \end{cases} \quad (15)$$

For data recovery, the original pixel is recovered as

$$p_i = \begin{cases} p'_i - 1, & \text{if } p'_i \geq b_i + s_t + 1 \text{ and } NL_i = t \\ p'_i + 1, & \text{if } p'_i \leq a_i + r_t - 1 \text{ and } NL_i = t \\ p'_i, & \text{if } a_i + r_t \leq p'_i \leq b_i + s_t \text{ and } NL_i = t \end{cases} \quad (16)$$

The data bit m is extracted as

$$m = \begin{cases} 0, & \text{if } p'_i \in \{b_i + s_t, a_i + r_t\} \text{ and } NL_i = t \\ 1, & \text{if } p'_i \in \{b_i + s_t + 1, a_i + r_t - 1\} \text{ and } NL_i = t \end{cases} \quad (17)$$

In the two-stage parameter determination, the parameters $(r_0, s_0, r_1, s_1, \dots, r_{255}, s_{255})$ are first initialized as the conventional PVO embedding manner (see Fig. 2). Since the majority of smooth pixels are with a low noise level, we only consider the optimization of the first 32 histograms (i.e., $t \leq 31$). In the first stage, we give 54

heuristic MHM manners and take them as the candidate list. Here, the candidate MHM manner is generated by using Algorithm 1, whose design is based on the junction level ϕ . The junction is a start point on the multiple histogram grid from which the offsets of expansion bins are chosen as the fixed values. It should be noted that two manners are generated for an odd-valued ϕ , but only one manner for an even value.

Algorithm 1. Algorithm for generating the MHM manner based on the junction level ϕ . For the PEHs with $t \geq \phi + 1$, the offsets r_t and s_t are set as constant.

Data Input the junction position ϕ . Initialize the offsets r_t and s_t as the conventional PVO embedding manner.

for $t \leftarrow \phi + 1$ **to** 255 **do**

if $\phi \geq 0$ and $t \geq 0$ **then**

 The expansion positions are shifted inwards.

if $\text{mod}(\phi, 2) = 0$ **then**

 Generate one MHM manner.

$r_t = \lfloor \phi/2 \rfloor, s_t = -(t - r_t)$.

end

if $\text{mod}(\phi, 2) = 1$ **then**

 Generate two MHM manners.

 Manner 1: $r_t = \lfloor \phi/2 \rfloor + 1, s_t = -\lfloor \phi/2 \rfloor$.

Manner 2: $r_t = \lfloor \phi/2 \rfloor, s_t = -\lfloor \phi/2 \rfloor - 1$.

end

if $\phi < 0$ and $t \geq 0$ **then**

 The expansion positions are shifted outwards.

if $\text{mod}(\phi, 2) = 0$

 Generate one MHM manner.

$r_t = \lfloor \phi/2 \rfloor, s_t = -\lfloor \phi/2 \rfloor$.

end

if $\text{mod}(\phi, 2) = 1$ **then**

 Generate two MHM manners.

 Manner 1: $r_t = \lfloor \phi/2 \rfloor + 1, s_t = -\lfloor \phi/2 \rfloor$.

 Manner 2: $r_t = -\lfloor \phi/2 \rfloor, s_t = \lfloor \phi/2 \rfloor + 1$.

end

end

For better illustration, six modification manners in the proposed MHM candidate list are plotted in Fig. 7, where the junction level ϕ is set as 1, 2, -1 and -2, respectively. For instance, for $\phi = 1$, one can yield two MHM manners as shown in Fig. 7(a) and (b), respectively. In the two sub-figures, since the junction position $\phi = 1$, r_t and s_t for $t \geq 2$ are set as the constant values respectively, and the expansion bins on these histograms are therefore symmetrically distributed. Similarly, for $\phi = 2$, r_t and s_t with $t \geq 3$ are all

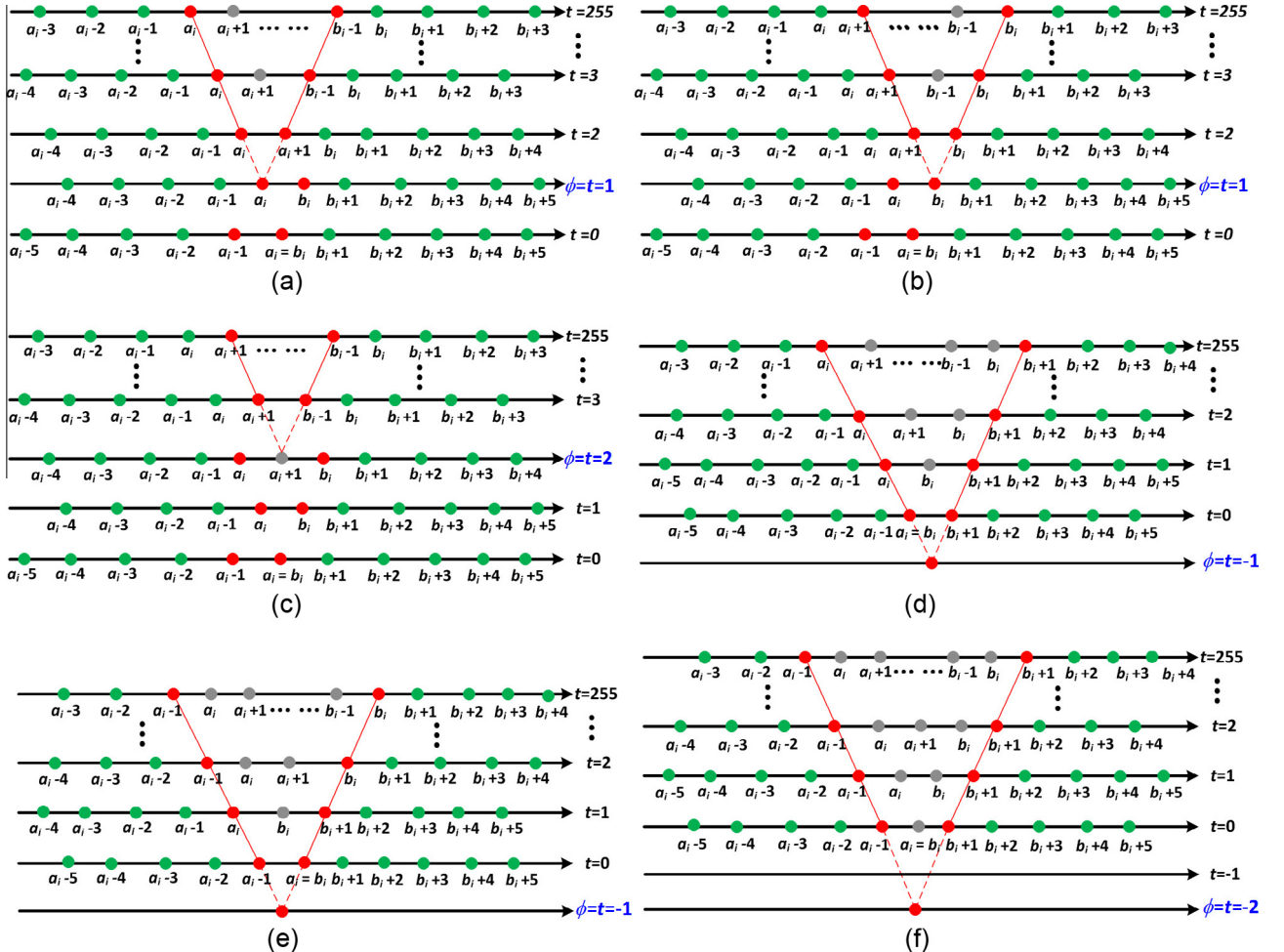


Fig. 7. Six MHM manners in the proposed candidate list by using Algorithm 1, where the expansion bins are designed to symmetrically distribute from a junction position ϕ . (a) $\phi = 1$, (b) $\phi = 1$, (c) $\phi = 2$, (d) $\phi = -1$, (e) $\phi = -1$, and (f) $\phi = -2$. For a better view, the noise level with the junction is marked blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
The offset parameters r_t and s_t for the MHM manners shown in Fig. 7.

	Fig. 7(a)		Fig. 7(b)		Fig. 7(c)		Fig. 7(d)		Fig. 7(e)		Fig. 7(f)	
t	r_t	s_t	r_t	s_t	r_t	s_t	r_t	s_t	r_t	s_t	r_t	s_t
0	-1	0	-1	0	-1	0	0	1	-1	0	-1	1
1	0	0	0	0	0	0	0	1	-1	0	-1	1
2	0	-1	1	0	0	0	0	1	-1	0	-1	1
3	0	-1	1	0	1	-1	0	1	-1	0	-1	1
4	0	-1	1	0	1	-1	0	1	-1	0	-1	1
...
30	0	-1	1	0	1	-1	0	1	-1	0	-1	1
31	0	-1	1	0	1	-1	0	1	-1	0	-1	1

set as 1 and -1, respectively, and only one MHM manner is obtained. For a negative value ϕ , a virtual junction point is given to illustrate the design of MHM manner. It can be found that, compared with the conventional PVO embedding, the expansion positions are shifted inwards for $\phi > 0$ and outwards for $\phi < 0$ by Algorithm 1. When the junction level ϕ is set as 0, the MHM manner of the conventional PVO algorithm is obtained. The corresponding offset parameters for the six MHM manners in Fig. 7 are given in Table 1. Note that here, only the parameters for the first 32 histograms need to be recorded, since the other histograms are modified in the conventional way.

In our experiments, we set $\phi \in [-4, 31]$ and therefore can obtain 54 different MHM manners by using Algorithm 1. For a given capacity, the optimal MHM manner in the list is determined as

$$\begin{cases} (\tilde{\mathbf{r}}, \tilde{\mathbf{s}}) = \arg \min_{\mathbf{r}, \mathbf{s} \in \mathbf{S}^{256}} \frac{ED(\mathbf{r}, \mathbf{s})}{EC(\mathbf{r}, \mathbf{s})} \\ \text{subject to } EC(\mathbf{r}, \mathbf{s}) \geq PS \end{cases} \quad (18)$$

where \mathbf{S}^{256} denotes the vector space with the dimension 256 for the 54 candidate MHM manners, and PS is the payload size. For the exact recovery, the index of the optimal MHM manner (6 bits) will be recorded as the side information.

In the second stage, based on the derived vectors $\tilde{\mathbf{r}}$ and $\tilde{\mathbf{s}}$, the offsets for the histograms with $t \leq 31$ are further optimized. Specifically, the better expansion bins are locally searched near the current positions $a_i + \tilde{r}_t$ and $b_i + \tilde{s}_t$ as shown in Fig. 8. The search range is limited in a window with size of 4. Thus, the issue here is to choose the optimal expansion bin from the four candidates in the window. For example, for the left expansion bin, the four-candidate set V_t^a includes $\{a_i + \tilde{r}_t - 2, \dots, a_i + \tilde{r}_t + 1\}$.

To alleviate the computation complexity, the histograms are sequentially processed and optimized one by one (i.e., from $t = 0$ to $t = 31$). More specifically, we first optimize the histogram with a low noise level t , and obtain the optimal offsets (r_t^*, s_t^*) as

$$\begin{cases} (r_t^*, s_t^*) = \arg \min_{r_t \in V_t^a, s_t \in V_t^b} \frac{ED(r_t, s_t)}{EC(r_t, s_t)} \\ \text{subject to } EC(\mathbf{r}_t, \mathbf{s}_t) \geq PS \end{cases} \quad (19)$$

where $\mathbf{r}_t = (r_0^*, \dots, r_{t-1}^*, r_t, \tilde{r}_{t+1}, \dots, \tilde{r}_{255})$ and $\mathbf{s}_t = (s_0^*, \dots, s_{t-1}^*, s_t, \tilde{s}_{t+1}, \dots, \tilde{s}_{255})$. After that, we proceed the next histogram with a

higher noise level $t + 1$ and repeat the same local optimization. Because not all the parameter combinations are examined, this solution provides a significant computational complexity reduction, and the complexity of optimization is linearly complex, w.r.t. the number of multiple histograms. After the optimization on a histogram is completed, the offset vectors are correspondingly updated. For the $(t + 1)$ -th iteration, the offset vectors are updated as

$$\mathbf{r}^{(t+1)} = (r_0^*, r_1^*, \dots, r_t^*, \tilde{r}_{t+1}, \dots, \tilde{r}_{255}), \quad (20)$$

and

$$\mathbf{s}^{(t+1)} = (s_0^*, s_1^*, \dots, s_t^*, \tilde{s}_{t+1}, \dots, \tilde{s}_{255}), \quad (21)$$

respectively, where the superscript (\cdot) denotes the number of iteration, and the initial vectors are $\mathbf{r}^{(0)} = \tilde{\mathbf{r}}$ and $\mathbf{s}^{(0)} = \tilde{\mathbf{s}}$. After the first 32 histograms are optimized, the final vectors are obtained as $\mathbf{r}^* = \mathbf{r}^{(32)}$ and $\mathbf{s}^* = \mathbf{s}^{(32)}$.

In the step of MHM coding, the first 32 offsets in the vectors are recorded as the side information. For each offset, two bits are required to represent its value since the window size is set as 4. As a result, there are totally $32 \times 4 = 128$ bits, which is acceptable for RDH.

3.2. Embedding and extracting procedures

Compared with the conventional PVO embedding, the two-stage mechanism above is employed to adaptively generate a better MHM manner based on the image content. It takes the payload size PS and the number of context pixels $n \in [2, 15]$ as input, and the output are the index of the optimal MHM manner in the candidate list and the offset vectors for the first 32 histograms. To determine the optimal context size, we exhaustively test all the 14 context sizes by using the two-stage mechanism, and select the one with the highest peak signal-to-noise ratio (PSNR) and the required capacity. Generally, the small context n provides a larger capacity, while the larger one usually yields a higher PSNR. The block diagram for the overall data embedding is given in Fig. 9.

As the maximum modification on a pixel is 1 at most, the location map that avoids the overflow/underflow problem is simple to implement. We mark the pixel with a value of 0 or 255 as "1", and the one with the value of 1 or 254 as "0". Then, the pixel with the value of 0 or 255 is modified as 1 or 254, respectively. For a 512×512 sized image, the total side information includes the following five parts:

- Offset vectors \mathbf{r}^* and \mathbf{s}^* (128 bits).
- Context size n^* (4 bits).
- Index of the optimal MHM manner in the candidate list (6 bits).
- End position S_{end} (18 bits).
- Location map LM (L bits).

Thus, the payload size is $PS = 156 + L + C$ bits, where C denotes the size of secret message. The side information is embedded in the rightmost column and the bottom row by the least-significant-bit

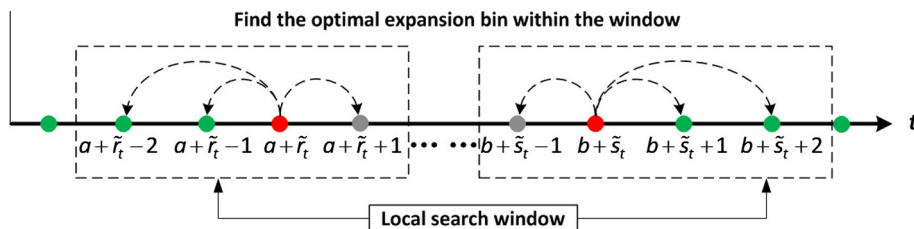


Fig. 8. The local optimization for determining the expansion bins with the noise level t .

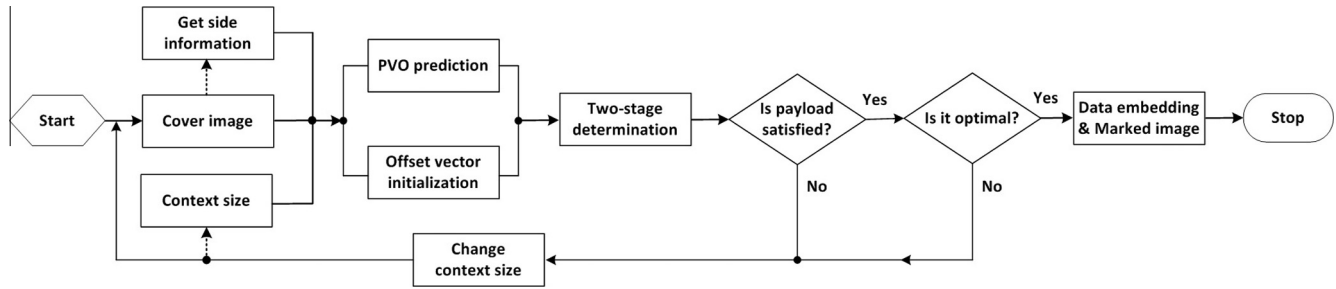


Fig. 9. The block diagram for the overall data embedding.

(LSB) replacement. The basic data embedding and recovery for a given context size proceed as follows.

Data embedding: Scan the whole image by a raster-scan order except the rightmost column and the bottom row, and obtain the pixel sequence (p_1, \dots, p_N) . For each pixel p_i , get the triple-element vector (p_i, a_i, b_i) , where a_i and b_i are the minimum and maximum in the given context $U_i = (u_1, \dots, u_n)$. Compute the pixel's noise level as $NL_i = b_i - a_i$, and generate the multiple histograms with different noise levels. Initialize the offset vectors \mathbf{r} and \mathbf{s} as the values utilized in the conventional PVO embedding. Use the two-stage bin-selection mechanism to determine the MHM manner, and obtain the optimal vectors \mathbf{r}^* and \mathbf{s}^* . For each pixel, based on (15), embed the secret data and obtain the marked pixel until the payload is satisfied. The marked pixel sequence is obtained as $(p'_1, \dots, p'_{S_{end}}, p_{S_{end}+1}, \dots, p_N)$, where the index S_{end} denotes the end position. Finally, embed the side information into the LSBs of the pixels in the rightmost column and the bottom row.

Data extraction and pixel recovery: Extract the side information from the predefined region. From the end position S_{end} , process the pixels in the inverse order of data embedding. For each pixel, the context is the same as that in data embedding, and one can get the same prediction and noise level. According to the noise level, the pixel is recovered by using (16), and checked whether there is a data bit or not using (17). Repeat the recovery until the beginning pixel is processed. Extract the LSBs from the payload to recover the pixels in the rightmost column and the bottom row. Finally, the original image and the secret message are restored.

4. Experimental results

For the performance evaluation, the proposed method is compared with three state-of-the-art methods [25,37,47] on six standard images and Kodak image database. Our method is mainly implemented using C++, and all the experiments are conducted on a PC with i7 3.6 GHz CPU and 12 GB RAM.

The performance comparison on six representative images is shown in Fig. 10. The test images includes Lena, Baboon, Airplane, Barbara, Lake and Peppers, with size of 512×512 . Results show that the proposed method performs superiorly or similarly against the state-of-the-art methods in the most of cases. Specifically, compared with the latest PVO-based method [47], the proposed method can uniformly get an improved performance, and is able to obtain a bigger gain in PSNR for the large capacity cases. The merit is due to the fact that, our method adjusts the modification manner for multiple histograms, and can better determine the expansion bins with respect to the embedding efficiency. As shown in Fig. 10, for the relatively textured images such as Baboon and Barbara, the PSNR gain is more significant and nearly 0.5 dB on average. For the smooth images Airplane and Lena, the derived MHM manner is the same as the conventional PVO embedding in

the most of cases, thus leading to a similar performance. Besides, the proposed method also increases the capacity than the method [47] as shown in the figure. This is because the adaptive MHM manner makes more rough pixels available for carrying data bits. For the six images, the capacity using the proposed method is increased 18% on average than Qu et al.'s.

Sachnev et al.'s method [25] is the typical PEE, which is based on the single PEH modification. Compared with their method, our method can yield a significant improvement for the relatively small capacity, because the adaptive MHM manner is employed to better exploit the correlations among different image areas. Ma et al.'s method [37] is also based on the single PEH modification, but utilizes the multiple predictors for prediction. It generally performs better than the conventional PEE method. Compared with Ma et al.'s method, the proposed method yields a higher PSNR for small capacities as well. For the Lena, Barbara, and Peppers images, the PSNR of the proposed method is 1 dB higher than theirs on average for low capacities. But for the very smooth and rough images such as Airplane and Baboon, our method is slightly better for low capacities. It is found that both the proposed method and Qu et al.'s method are worse in dealing with the high capacity cases than Sachnev et al.'s and Ma et al.'s, due to that the advantage of PVO disappears as the pixels in the rough areas have to be used to fulfill the capacity.

Similar to the other PVO-based methods, the superiority of the proposed method over the typical PEE method (e.g., Sachnev et al.'s) mainly lies in better exploiting image redundancy by using the large sized context, and thus leaving the rough pixels unchanged as much as possible. For low and moderate capacities, the smooth pixels are sufficient for PVO embedding. Therefore, one can only modify a minority of pixels and leave the others unchanged. But for large capacity case, almost all the pixels are utilized and the gain is diminishing as the embedding capacity goes near the upper bound. Especially, for a relative rough image such as Baboon, the neighboring pixels are less correlated, thus, the image does not contain sufficient smooth pixels.

The offset vectors using the proposed method for different capacities on Baboon and Barbara images are listed in Tables 2 and 3, respectively. Since the modification manner for the histograms with $t \geq 32$ are set as the same as the conventional PVO embedding, i.e., the offsets $(r_{32}, s_{32}, \dots, r_{255}, s_{255})$ are all set as 0, they are omitted in the tables for simplicity.

Furthermore, we have compared the proposed method with the three methods on Kodak image database, which consists of 24 color images with size of 768×512 or 512×768 . For comparison, the color images are all transformed into the gray-scale version. Tables 4 and 5 give the PSNRs of the proposed method and the three compared methods for a capacity of 10,000 and 20,000 bits, respectively. Compared with the methods [25,37,47], the average gains in PSNR by using the proposed method are 2.85, 1.28 and 0.8 dB for 10,000 bits, respectively, and 2.87, 1.26 and 0.4 dB for 20,000 bits, respectively. We can observe that the proposed

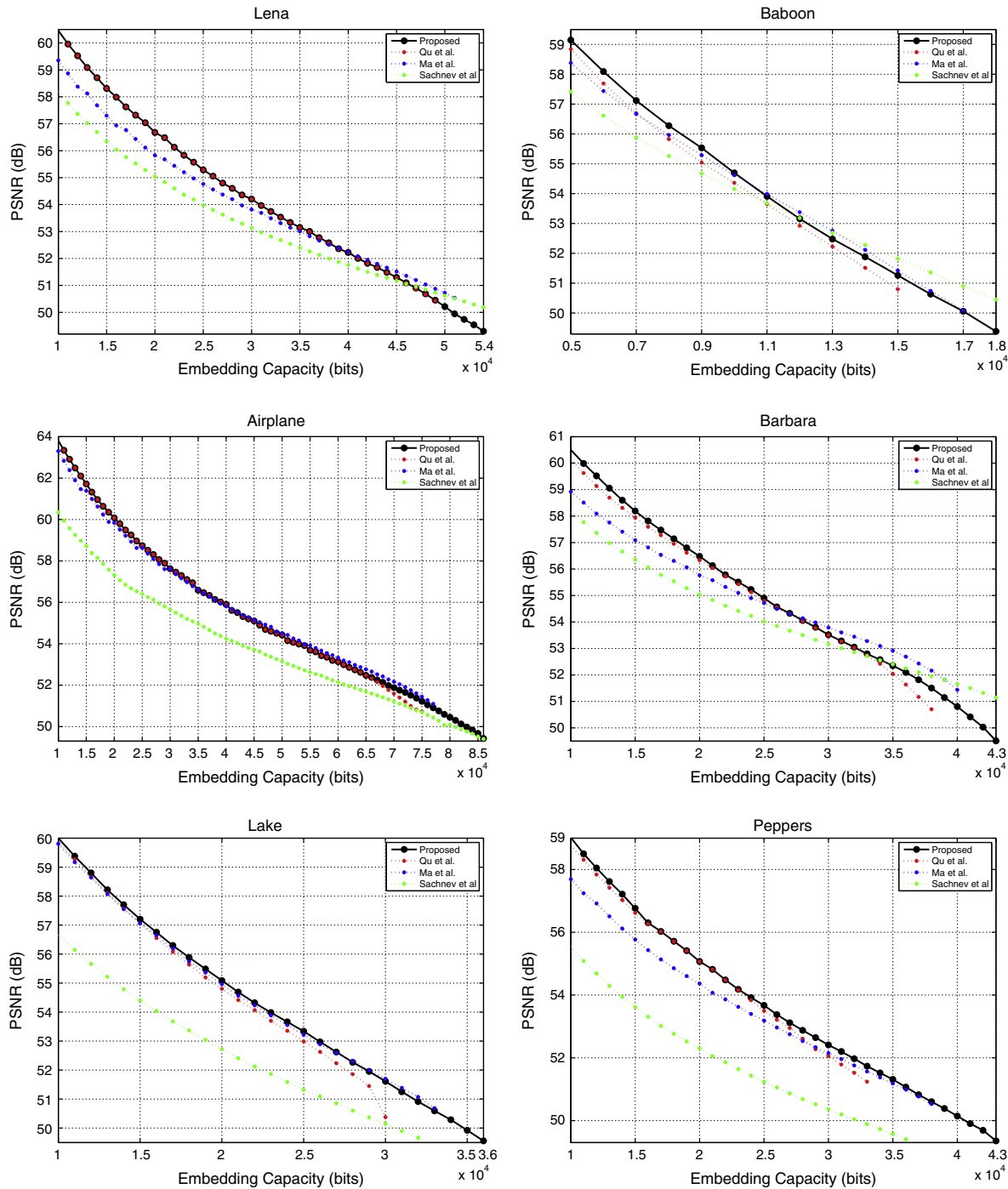


Fig. 10. Performance evaluation in terms of capacity-distortion curve by comparing the proposed method with three methods including Sachnev et al.'s [25], Ma et al.'s [37], and Qu and Kim's [47].

method tends to achieve a bigger gain in embedding performance for the large-sized image. This tendency is related to that, in the large-sized image, the rough areas become more important in improving the embedding efficiency and the highly flexible MHM manner contributes more in distortion reduction.

It is interesting to find that Ma et al.'s method can also be viewed as a variant of PVO embedding, which is based on prediction-value-ordering. So, the main difference is that, to predict a pixel, it sorts the multiple predictions $\{\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n\}$ derived from different predictors instead of ranking the context pixels $\{u_1, u_2, \dots, u_n\}$, and then choose either the maximum or minimum

prediction as the optimal estimate. By the multiple predictor technique, a more accurate prediction can be obtained than the classical predictor [25]. Compared with the PVO-based method, it usually yields a better performance for large capacities. However, Ma et al.'s method is designed for the single histogram modification, and has not considered the diversity of different histograms as well. In fact, the MHM framework can also be applied to Ma et al.'s method. Here, we use the proposed MHM determination for the enhancement of Ma et al.'s work. By replacing the PVO prediction with the multiple-predictor prediction, a very different PEH can be obtained and the corresponding performance is shown in

Table 2

Offsets of expansion bins determined by the proposed method for the first 32 histograms ($0 \leq t \leq 31$) on Baboon image.

Capacity (bits)	5000		10,000		15,000		18,000	
	r_t	s_t	r_t	s_t	r_t	s_t	r_t	s_t
t								
0	-2	2	-1	0	-1	0	-1	0
1	-2	2	0	0	0	0	0	0
2	-2	2	-1	0	-1	0	-1	0
3	-2	2	-1	0	-1	1	-1	1
4	-2	2	-1	0	-2	1	-2	1
5	-2	2	-1	0	-2	2	-2	2
6	-2	2	-1	0	-3	2	-3	2
7	-2	2	-1	0	-3	3	-3	3
8	-2	2	-1	0	-3	3	-4	3
9	-2	2	-1	0	-3	3	-4	4
10	-2	2	-1	0	-3	3	-5	4
11	-2	2	-1	0	-3	3	-5	5
12	-2	2	-1	0	-3	3	-6	5
13	-2	2	-1	0	-3	3	-6	6
14	-2	2	-1	0	-3	3	-7	6
15	-2	2	-1	0	-3	3	-7	7
16	-2	2	-1	0	-3	3	-8	7
17	-2	2	-1	0	-3	3	-8	8
18	-2	2	-1	0	-3	3	-9	8
19	-2	2	-1	0	-3	3	-9	8
20	-2	2	-1	0	-3	3	-9	8
21	-2	2	-1	0	-3	3	-9	8
22	-2	2	-1	0	-3	3	-9	8
23	-2	2	-1	0	-3	3	-9	8
24	-2	2	-1	0	-3	3	-9	8
25	-2	2	-1	0	-3	3	-9	8
26	-2	2	-1	0	-3	3	-9	8
27	-2	2	-1	0	-3	3	-9	8
28	-2	2	-1	0	-3	3	-9	8
29	-2	2	-1	0	-3	3	-9	8
30	-2	2	-1	0	-3	3	-9	8
31	-2	2	-1	0	-3	3	-9	8

Table 3

Offsets of expansion bins determined by the proposed method for the first 32 histograms ($0 \leq t \leq 31$) on Barbara image.

Capacity (bits)	10,000		20,000		30,000		40,000		43,000	
	r_t	s_t	r_t	s_t	r_t	s_t	r_t	s_t	r_t	s_t
t										
0	-1	1	-1	1	0	1	0	-1	0	-1
1	-1	1	-1	1	0	1	0	0	0	0
2	-1	1	-1	1	0	1	-1	0	-1	0
3	-1	1	-1	1	0	1	-1	1	-1	1
4	-1	1	-1	1	0	1	-2	1	-2	1
5	-1	1	-1	1	0	1	-2	2	-2	2
6	-1	1	-1	1	0	1	-2	3	-3	2
7	-1	1	-1	1	0	1	-2	3	-3	3
8	-1	1	-1	1	0	1	-2	3	-3	4
9	-1	1	-1	1	0	1	-2	3	-3	4
10	-1	1	-1	1	0	1	-2	3	-3	4
11	-1	1	-1	1	0	1	-2	3	-3	4
12	-1	1	-1	1	0	1	-2	3	-3	4
13	-1	1	-1	1	0	1	-2	3	-3	4
14	-1	1	-1	1	0	1	-2	3	-3	4
15	-1	1	-1	1	0	1	-2	3	-3	4
16	-1	1	-1	1	0	1	-2	3	-3	4
17	-1	1	-1	1	0	1	-2	3	-3	4
18	-1	1	-1	1	0	1	-2	3	-3	4
19	-1	1	-1	1	0	1	-2	3	-3	4
20	-1	1	-1	1	0	1	-2	3	-3	4
21	-1	1	-1	1	0	1	-2	3	-3	4
22	-1	1	-1	1	0	1	-2	3	-3	4
23	-1	1	-1	1	0	1	-2	3	-3	4
24	-1	1	-1	1	0	1	-2	3	-3	4
25	-1	1	-1	1	0	1	-2	3	-3	4
26	-1	1	-1	1	0	1	-2	3	-3	4
27	-1	1	-1	1	0	1	-2	3	-3	4
28	-1	1	-1	1	0	1	-2	3	-3	4
29	-1	1	-1	1	0	1	-2	3	-3	4
30	-1	1	-1	1	0	1	-2	3	-3	4
31	-1	1	-1	1	0	1	-2	3	-3	4

Table 4

Performance comparison in terms of PSNR (in dB) on Kodak image database for a capacity of 10,000 bits. The best result for an image is shown in boldface.

Image	Sachnev et al.	Ma et al.	Qu et al.	Proposed
kodim01	61.86	63.49	64.23	64.95
kodim02	61.83	63.55	64.49	64.91
kodim03	63.53	64.02	65.11	65.67
kodim04	61.88	63.25	64.17	64.58
kodim05	61.74	62.77	62.51	63.08
kodim06	65.33	66.16	66.05	66.88
kodim07	63.04	63.45	64.83	65.32
kodim08	58.44	57.45	57.83	58.97
kodim09	60.53	62.02	63.34	63.74
kodim10	60.65	62.02	62.68	63.31
kodim11	62.86	64.24	65.42	66.01
kodim12	62.73	64.38	64.75	65.28
kodim13	57.79	58.74	58.20	59.14
kodim14	60.83	62.19	62.62	63.05
kodim15	62.31	64.66	62.86	64.86
kodim16	63.01	64.21	65.06	65.67
kodim17	61.58	62.53	64.52	64.57
kodim18	58.82	59.38	61.02	61.34
kodim19	60.75	61.30	63.19	63.24
kodim20	53.40	60.72	57.76	60.74
kodim21	60.70	61.49	63.58	64.04
kodim22	60.31	61.06	63.00	63.43
kodim23	62.27	63.96	64.35	65.09
kodim24	58.29	65.06	62.22	65.06
Average	61.02	62.59	63.07	63.87

Table 5

Performance comparison in terms of PSNR (in dB) on Kodak image database for a capacity of 20,000 bits. The best result for an image is marked boldface.

Image	Sachnev et al.	Ma et al.	Qu et al.	Proposed
kodim01	58.10	59.81	60.47	60.87
kodim02	58.72	60.16	61.37	61.57
kodim03	60.09	61.17	62.42	62.67
kodim04	58.58	59.93	60.85	61.04
kodim05	58.40	59.13	58.94	59.31
kodim06	61.37	63.37	63.45	63.86
kodim07	59.47	60.56	62.03	62.27
kodim08	55.07	54.34	55.10	55.72
kodim09	57.44	58.49	60.50	60.69
kodim10	57.49	58.82	59.98	60.30
kodim11	59.37	61.38	62.63	62.91
kodim12	59.07	60.60	61.80	62.06
kodim13	53.26	53.41	53.76	54.12
kodim14	57.42	58.28	58.68	59.08
kodim15	59.15	61.53	61.99	62.51
kodim16	59.48	60.90	62.22	62.47
kodim17	58.20	59.10	61.17	61.19
kodim18	55.70	56.39	57.48	57.63
kodim19	57.63	58.25	60.14	60.15
kodim20	51.87	59.85	58.15	59.83
kodim21	57.58	58.61	60.87	61.12
kodim22	57.24	58.14	59.56	59.73
kodim23	58.94	60.86	61.61	61.92
kodim24	56.64	61.75	60.49	62.13
Average	57.76	59.37	60.23	60.63

Fig. 11. It can be seen that, by using the multiple predictor technique for prediction, the proposed method can uniformly outperform Ma et al.'s no matter what a capacity is. What interests us most here is that the results show that the proposed MHM determination leads to a definite improvement for whichever a prediction technique mentioned above (i.e., PVO prediction and multiple-predictor prediction). The superiority of the proposed MHM mechanism is further validated for efficient data hiding. It is expected that by combining a more efficient histogram generation, the proposed MHM mechanism can yield a better embedding performance.

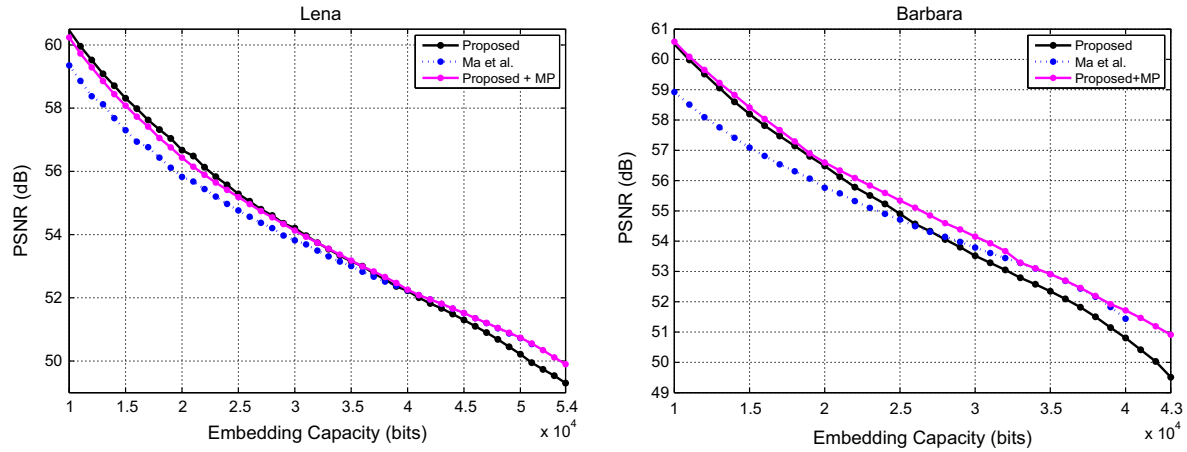


Fig. 11. Performance comparison between the proposed method, Ma et al.'s [37] and the proposed method integrating the multiple predictor technique, where MP denotes the multiple predictor technique.

Table 6

Comparison with respect to the number of shifted pixels N_s for the capacity of 10,000 bits. The best results are marked boldface.

Images	Sachnev et al.	Ma et al.	Qu et al.	Proposed
Lena	20,785	144,390	10,370	10,290
Baboon	60,312	53,715	57,549	52,008
Airplane	10,501	2565	2133	2109
Barbara	21,068	16,177	11,512	10,223
Lake	31,700	12,352	12,639	12,085
Peppers	42,296	23,239	17,741	16,369

Table 7

Comparison with respect to the number of shifted pixels N_s for the capacity of 20,000 bits. The best results are marked boldface.

Images	Sachnev et al.	Ma et al.	Qu et al.	Proposed
Lena	43,420	33,795	27,176	26,825
Airplane	21,566	7399	6951	6690
Barbara	43,420	34,371	29,644	28,420
Lake	81,202	44,434	46,390	42,787
Peppers	90,388	52,460	43,250	43,191

To further verify the effectiveness of our method in reducing distortion, we compare the number of the shifted pixels in different methods for a given capacity. It is known that, during the embedding procedure, the image pixels are modified sequentially, and the data embedding stops when the processed pixels just provides the capacity. Let N_e and N_s denote the numbers of expanded and shifted pixels, respectively, we have $N_e = EC$ for a given capacity EC . Because the average distortions for the expandable and shiftable pixels are 0.5 and 1, respectively, the introduced distortion can be calculated as $0.5N_e + N_s = 0.5EC + N_s$, where the capacity is a constant. So when the capacity is fixed, the number of expanded pixels are the same for different methods, and the embedding distortion is proportional to the number of shifted pixels. In this way, we can use N_s to measure the distortion on an image. Tables 6 and 7 list the numbers of shifted pixels N_s for capacities of 10,000 and 20,000 bits, respectively. Here, the less the shifted pixels, the better the embedding performance. As can be seen, the proposed method achieves the minimum number of N_s among the four methods. It is thus verified that our method is able to modify less pixels for a given capacity than the compared methods.

The comparison of runtime for the capacity of 10,000 bits is listed in Table 8. Based on our experiments, the most of computation time is spent in the preprocess, where there exist multiple loops for estimating the performances of all possible candidates.

Table 8

Comparison of runtime for the capacity of 10,000 bits, where the unit is second.

Image	C++ implementation			
	Sachnev et al.	Ma et al.	Qu et al.	Proposed
Lena	1.08	1.24	1.58	6.91
Baboon	1.11	1.48	1.42	7.14
Airplane	1.09	1.28	1.58	5.72
Barbara	1.07	1.23	1.48	6.14
Lake	1.07	1.25	1.53	5.70
Peppers	1.03	1.20	1.51	6.40
Average	1.08	1.28	1.52	6.34

Since it is difficult to process multiple loops on the Matlab platform, the acceleration of the proposed method is implemented by C++ to tackle with the computation of loops. As can be seen in Table 8, the average runtime of ours is nearly 6 s, which is still acceptable for practical uses. It is noted that, using C++ implementation, the runtime for the other three methods is less than 2 s on average. The comparison of time and space complexity is analyzed in Table 9. The computation complexity of our method is nearly $O(756N)$, and mainly depends on the image size N , and the number of MHM candidates. Here, the number of MHM candidates is calculated as 14×54 because there are 14 context sizes and 54 MHM manners for each size. For Sachnev et al.'s, Ma et al.'s and Qu et al.'s methods, the time complexities are approximated $O(N)$, $O(5N)$ and $O(14N)$. The space complexity here corresponds to the consumption for saving the possible candidates. So, to some extent, the complexity is proportional to the roughness of an image, i.e., the more textured an image, the more space is consumed. This also leads to an increase of runtime on the rough image such as Baboon. The increased complexity of our algorithm is due to the increased number of MHM manners for adaptive embedding, wherein the most time and memory are spent in the initialization of candidate pool. As the loops for estimating the possible performances are independent, for a further complexity, a parallel computing technique based on many-core processors [48] can be exploited to facilitate the proposed method.

Table 9

Comparison of time and space complexity, where N and NL_{max} denote the image size and the maximum of noise level for a block or a pixel.

Complexity type	Sachnev et al.	Ma et al.	Qu et al.	Proposed
Time	$O(N)$	$O(5N)$	$O(14N)$	$O(756N)$
Space	$O(NL_{max})$	$O(5 \cdot NL_{max})$	$O(14 \cdot NL_{max})$	$O(756 \cdot NL_{max})$

5. Conclusions

In this paper, a new pixel-value-ordering (PVO) algorithm based on the multiple histogram modification (MHM) is proposed. The new algorithm is designed in a unified framework of MHM, and parameterized by the selection of expansion bins. It allows the conventional PVO embedding to be implemented equivalently in the MHM form, and highlights the embedding efficiency by flexibly modifying different histograms. Based on the framework, a two-stage bin-selection mechanism is introduced to optimize the MHM manner, using both the heuristic and automatical determinations. Experimental results show that the proposed method can improve the performance of PVO-based methods and also yield a superior performance than some state-of-the-art methods. As the proposed MHM determination is a methodology of adaptive histogram modification, it is applicable for the existing techniques of histogram generation and naturally benefits the PVO-based methods and some other PEE-based methods. Two contributions here are (1) the general framework of MHM for PVO embedding (2) and a two-stage mechanism for the optimization of MHM.

As there are more options for data embedding than the conventional method, the complexity of time and space in our method are increased. To guarantee a fast processing speed, a more effective bin-selection strategy would be helpful to simplify the optimization of MHM manner while preserving the high performance. In the future, it is interesting to apply the MHM framework to the advanced histogram generation technique, or utilize more MHM manners under the proposed framework to further improve the embedding performance.

Acknowledgements

This work is supported by the National Science Foundation of China (Nos. 61502160, 61232016, U1405254, and 61272421) and the PAPD fund.

References

- [1] Y.Q. Shi, Z. Ni, D. Zou, C. Liang, G. Xuan, Lossless data hiding: fundamentals, algorithms and applications, *Proc. IEEE ISCAS*, vol. 2, 2004, pp. 33–36.
- [2] A. Khan, A. Siddiqui, S. Munib, S.A. Malik, A recent survey of reversible watermarking techniques, *Inf. Sci.* 279 (0) (2014) 251–272.
- [3] S.K. Lee, Y.H. Suh, Y.S. Ho, Reversible image authentication based on watermarking, in: *Proc. IEEE ICME*, 2006, pp. 1321–1324.
- [4] G. Coatrieux, C. Le Guillou, J.-M. Cauvin, C. Roux, Reversible watermarking for knowledge digest embedding and reliability control in medical images, *IEEE Trans. Inf. Technol. Biomed.* 13 (2) (2009) 158–165.
- [5] K.L. Chung, Y.H. Huang, P.C. Chang, H.Y.M. Liao, Reversible data hiding-based approach for intra-frame error concealment in H.264/AVC, *IEEE Trans. Circuits Syst. Video Technol.* 20 (11) (2010) 1643–1647.
- [6] F. Peng, Y. Liu, M. Long, Reversible watermarking for 2D CAD engineering graphics based on improved histogram shifting, *Comput. Aided Des.* 49 (2014) 42–50.
- [7] J. Fridrich, M. Goljan, R. Du, Lossless data embedding – new paradigm in digital watermarking, *EURASIP J. Appl. Signal Process.* 2002 (2) (2002) 185–196.
- [8] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, *IEEE Trans. Image Process.* 14 (2) (2005) 253–266.
- [9] C.-C. Chang, W.-L. Tai, C.-C. Lin, A reversible data hiding scheme based on side match vector quantization, *IEEE Trans. Circuits Syst. Video Technol.* 16 (10) (2006) 1301–1308.
- [10] W. Zhang, X. Hu, X. Li, N. Yu, Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression, *IEEE Trans. Image Process.* 22 (7) (2013) 2775–2785.
- [11] X. Zhang, Reversible data hiding with optimal value transfer, *IEEE Trans. Multimedia* 15 (2) (2013) 316–325.
- [12] J. Tian, Reversible data embedding using a difference expansion, *IEEE Trans. Circuits Syst. Video Technol.* 13 (8) (2003) 890–896.
- [13] W.L. Tai, C.M. Yeh, C.C. Chang, Reversible data hiding based on histogram modification of pixel differences, *IEEE Trans. Circ. Syst. Video Technol.* 19 (6) (2009) 906–910.
- [14] Y. Hu, H.K. Lee, J. Li, DE-based reversible data hiding with improved overflow location map, *IEEE Trans. Circ. Syst. Video Technol.* 19 (2) (2009) 250–260.
- [15] X. Li, W. Zhang, X. Gui, B. Yang, A novel reversible data hiding scheme based on two-dimensional difference-histogram modification, *IEEE Trans. Inf. Forensics Secur.* 8 (7) (2013) 1091–1100.
- [16] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, *IEEE Trans. Circ. Syst. Video Technol.* 16 (3) (2006) 354–362.
- [17] P. Tsai, Y.C. Hu, H.L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, *Signal Process.* 89 (6) (2009) 1129–1143.
- [18] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, *IEEE Trans. Inf. Forens. Secur.* 5 (1) (2010) 187–193.
- [19] H.-T. Wu, J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, *Signal Process.* 92 (12) (2012) 3000–3009.
- [20] Y.-Y. Tsai, D.-S. Tsai, C.-L. Liu, Reversible data hiding scheme based on neighboring pixel differences, *Digital Signal Process.* 23 (3) (2013) 919–927.
- [21] X. Li, B. Li, B. Yang, T. Zeng, General framework to histogram-shifting-based reversible data hiding, *IEEE Trans. Image Process.* 22 (6) (2013) 2181–2191.
- [22] W. Hong, T.-S. Chen, J. Chen, Reversible data hiding using Delaunay triangulation and selective embedding, *Inf. Sci.* 308 (0) (2015) 140–154.
- [23] H. Wu, J. Dugelay, Y. Shi, Reversible image data hiding with contrast enhancement, *IEEE Signal Process. Lett.* 22 (1) (2015) 81–85.
- [24] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, *IEEE Trans. Image Process.* 16 (3) (2007) 721–730.
- [25] 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.* 19 (7) (2009) 989–999.
- [26] W. Hong, T.-S. Chen, Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism, *J. Vis. Commun. Image Represent.* 22 (2) (2011) 131–140.
- [27] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, *IEEE Trans. Circ. Syst. Video Technol.* 21 (8) (2011) 1061–1070.
- [28] X. Li, B. Yang, T. Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, *IEEE Trans. Image Process.* 20 (12) (2011) 3524–3533.
- [29] W. Hong, Adaptive reversible data hiding method based on error energy control and histogram shifting, *Optics Commun.* 285 (2) (2012) 101–108.
- [30] B. Ou, X. Li, Y. Zhao, R. Ni, Reversible data hiding based on PDE predictor, *J. Syst. Softw.* 86 (10) (2013) 2700–2709.
- [31] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, C. Roux, Reversible watermarking based on invariant image classification and dynamic histogram shifting, *IEEE Trans. Inf. Forens. Secur.* 8 (1) (2013) 111–120.
- [32] C. Qin, C.-C. Chang, Y.-H. Huang, L.-T. Liao, An inpainting-assisted reversible steganographic scheme using histogram shifting mechanism, *IEEE Trans. Circ. Syst. Video Technol.* 23 (7) (2013) 1109–1118.
- [33] B. Ou, X. Li, Y. Zhao, R. Ni, Y.-Q. Shi, Pairwise prediction-error expansion for efficient reversible data hiding, *IEEE Trans. Image Process.* 22 (12) (2013) 5010–5021.
- [34] J. Wang, J. Ni, Y. Hu, An efficient reversible data hiding scheme using prediction and optimal side information selection, *J. Vis. Commun. Image Represent.* 25 (6) (2014) 1425–1431.
- [35] I.-C. Dragoi, D. Coltuc, Local-prediction-based difference expansion reversible watermarking, *IEEE Trans. Image Process.* 23 (4) (2014) 1779–1790.
- [36] X. Wang, J. Ding, Q. Pei, A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition, *Inf. Sci.* 310 (0) (2015) 16–35.
- [37] X. Ma, Z. Pan, S. Hu, L. Wang, High-fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism, *J. Vis. Commun. Image Represent.* 28 (2015) 71–82.
- [38] D. Coltuc, J.M. Chassery, Very fast watermarking by reversible contrast mapping, *IEEE Signal Process. Lett.* 14 (4) (2007) 255–258.
- [39] X. Wang, X. Li, B. Yang, Z. Guo, Efficient generalized integer transform for reversible watermarking, *IEEE Signal Process. Lett.* 17 (6) (2010) 567–570.
- [40] D. Coltuc, Low distortion transform for reversible watermarking, *IEEE Trans. Image Process.* 21 (1) (2012) 412–417.
- [41] J. Wang, J. Ni, A GA optimization approach to HS based multiple reversible data hiding, in: *Proc. IEEE WIFS*, 2013, pp. 203–208.
- [42] I. Caciula, D. Coltuc, Improved control for low bit-rate reversible watermarking, in: *Proc. IEEE ICASSP*, 2014.
- [43] X. Li, W. Zhang, X. Gui, B. Yang, Efficient reversible data hiding based on multiple histograms modification, *IEEE Trans. Inf. Forens. Secur.* 10 (9) (2015) 2016–2027.
- [44] X. Li, J. Li, B. Li, B. Yang, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, *Signal Process.* 93 (1) (2013) 198–205.
- [45] F. Peng, X. Li, B. Yang, Improved pvo-based reversible data hiding, *Digital Signal Process.* 25 (0) (2014) 255–265.
- [46] B. Ou, X. Li, Y. Zhao, R. Ni, Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion, *Signal Process.: Image Commun.* 29 (7) (2014) 760–772.
- [47] X. Qu, H.J. Kim, Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding, *Signal Process.* 111 (2015) 249–260.
- [48] C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, F. Wu, Efficient parallel framework for HEVC motion estimation on many-core processors, *IEEE Trans. Circuits Syst. Video Technol.* 24 (12) (2014) 2077–2089.