Neurocomputing 226 (2017) 23–34

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

High-fidelity reversible data hiding based on geodesic path and pairwise prediction-error expansion

Bo Ou[a,*], Xiaolong Li[b], Jinwei Wang[c], Fei Peng[a]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[b] Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
[c] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

ARTICLE INFO

Communicated by Dr Li Bing

Keywords:
Reversible data hiding
Pairwise prediction-error expansion
Histogram generation
Geodesic path

ABSTRACT

It is often important for reversible data hiding (RDH) to reduce the amount of image modifications for a given capacity. To this end, recently, a pairwise prediction-error expansion (pairwise PEE) is proposed to better exploit the image redundancy in the two-dimensional (2D) space. However, in conventional pairwise PEE, a drawback is that the pixel pair is generated by a fixed combination manner, which may limit the further improvement of embedding performance. Based on this consideration, we propose a new histogram generation strategy for the 2D RDH by using geodesic path. In contrast to the fixed manner, the pixels are adaptively combined into pairs with respect to the local similarity in terms of both spatial distance and intensity. As a result, the prediction-errors in a pair are more correlated to each other and the derived 2D PEH is more advisable for pairwise PEE. Experimental results show that, the proposed method can reduce the embedding distortion of conventional pairwise PEE, and yields a superior performance than some state-of-the-art methods.

1. Introduction

In the information era, the advancement of multimedia technology has greatly facilitated the digital media applications on video, image, and audio. Such an advantage also poses a challenging task of protecting the intellectual property of digital content. To verify the content integrity and prevent forgery, the common ways are encryption, fingerprinting, data hiding and so on. However, these techniques may alter the image content permanently, which can cause serious problems in sensitive fields, such as military, judical and medical imagery. This limitation promotes the need for another way of intellectual protection that can recover the original content after data extraction. In doing so, reversible data hiding (RDH) [1] is designed to provide the perfect recovery of both the hidden message and the original content.

Generally, in RDH, the secret message should be embedded in an invisible way without changing the visual meaning conveyed by the image. This is because that the intrusive changes may easily arouse the attackers' attention and malicious attacks, leading to a serious consequence in medical and military operations, e.g., when the forged data is fed into the medical system. So, the issue of reducing the amount of modifications on an image is still of current interest. Up to now, existing techniques developed for RDH can be roughly divided into five categories: lossless compression [2–7], difference expansion (DE) [8–13], histogram shifting (HS) [14–26], prediction-error expansion (PEE) [27–43] and integer-to-integer transform [44–48].

Early lossless compression based methods are basically heuristic procedures, in which a selected feature in an image is directly compressed to spare the space for embedding the secret message. Fridrich et al. [2] first proposed a general compression-embedding framework for RDH, in which the least significant bit (LSB) of a pixel is selected as the to-be-compressed feature, and the data hiding is implemented by overwriting the LSBs of pixels with the payload bits. To recover the original pixels, the payload bits should include both the secret message and the compressed LSBs. Later, Celik et al. [3] integrated the conditional entropy coding into the compression-embedding framework, in order to increase the compression rate on LSB. However, since the correlations among the LSBs are weak, the compression of LSBs usually results in an unacceptable fidelity loss. To remedy this drawback, several methods began to explore a compact feature for better compression, e.g., prediction-error. The idea behind this is to first decorrelate the image and then obtain a more compact feature for compression. Zhang et al. [5] proposed a recursive code reconstruction to optimize the modifications of prediction-error histogram. By assuming that the gray-scale host signal is independent identical distributed, this work can achieve the optimal embedding

[*] Corresponding author.
E-mail addresses: oubo@hnu.edu.cn (B. Ou), lixiaolong@pku.edu.cn (X. Li), wjwei_2004@163.com (J. Wang), pengfei@hnu.edu.cn (F. Peng).

http://dx.doi.org/10.1016/j.neucom.2016.11.017
Received 26 January 2016; Received in revised form 17 May 2016; Accepted 1 November 2016
Available online 17 November 2016
0925-2312/

performance with respect to the rate-distortion model. Chang et. al [49] proposed an improved side match vector quantization (SMVQ) method based on the search order coding, which can offer a higher embedding rate and retain the reversibility for SMVQ codes. Zhang [6] first proposed to use the transfer matrix to model RDH, where the optimal embedding is determined by an optimal value transfer rule. Under this framework, RDH can be efficiently optimized to achieve a good payload-distortion trade-off. Qian et al. [7] proposed a successful RDH scheme for encrypted JPEG bitstream, in order to meet the potential demand in reality, i.e., provide the privacy protection for the the content owner from the malicious data-hider.

The second class of techniques is based on DE, which is first proposed by Tian [8]. As its name implies, the expansion procedure consists of first calculating the difference of two adjacent pixels and then multiplying the difference by two, such that the LSB of the expanded difference can be left to hide a bit. The embedding distortion is determined by the similarity of two adjacent pixels, i.e., the more similar the two pixels, the small magnitude of the derived difference and the lower the distortion caused by expansion. As a pair of pixels can be embedded a bit at most, the capacity bound is nearly 0.5 bit per pixel. Alattar et al. [9] extended the DE notation to the vector of pixels, where multiple pixel differences are expanded together for data embedding. By using the vector expansion, a higher capacity can be obtained than Tian's method. Kamstra et al. [10] proposed a sorting technique to improve the embedding efficiency by modifying the smooth pixels preferentially. An indication of whether a pixel is smooth enough is obtained by computing the variance of its neighborhood. For the low capacity case, sorting technique has its own advantage, i.e., the embedding distortion can be controlled at a low level as more rough pixels are left unused. Li et al. [13] proposed a two-dimensional DE by modifying the pixel differences jointly, and can obtain a high quality marked image for the low capacity case.

Another simple yet effective way for RDH is based on HS, in which the statistical property of image, e.g., the histogram, is slightly modified to embed data. As a pixel is modified at most 1, HS-based methods can guarantee a high-fidelity marked image. Ni et al. [14] first proposed a HS based method by modifying the intensity histogram, in which the peak bin of histogram is utilized to carry data and the other bins are shifted to vacate the space. Here, the peak bin corresponds to the gray-level with the maximum population. Once the gray intensity of the peak bin is known, the hidden bits can be extracted by identifying the pixel value. Lee et al. [50] applied HS to the pixel difference histogram which has a more concentrated distribution, and thus obtained the higher capacity and image fidelity. Luo et al. [16] utilized an interpolation technique to optimize the pixel prediction, and then used the derived prediction-error histogram (PEH) for data embedding. Tsai et al. [21] proposed to construct the difference histogram by calculating the differences between a pixel and its left and upper neighbors. Hong et al. [23] proposed a prediction-error histogram shifting method using Delaunay triangulation, which can not only provide a better image quality, but is also robust to the detection of steganalysis. Very recently, Wang et al. [38] proposed to utilize the rate-distortion model to solve the optimal single-layer and multiple-layer embedding, where the optimal peak value pair can be determined based on this model. The research shows promising results of HS especially for high capacity case.

PEE techniques that extends the expansion on pixel differences to prediction-errors are evolved by integrating DE with HS. PEE was first proposed by Thodi and Rodriguez [27], in which the prediction-error between the pixel and its estimated value is expanded for data embedding. Usually, the PEE technique is consisted of two successive steps as shown in Fig. 4, i.e., histogram generation and histogram modification. Histogram generation can be viewed as a transformation that seeks to produce a low-dimensional representation for an image, which has a particular statistical property and is helpful for distortion reduction of RDH. For example, in PEE, the PEH is a Laplacian-like distribution with the peak near the zero value, and introduces much less distortion than the intensity histogram in data embedding. Histogram generation is the basis for numerous histogram-modification-based techniques. Histogram modification is a procedure to determine the modification manner for pixels according to their statistical property. The effectiveness of histogram modification is highly relied on the shape of the derived histogram. So, an efficient histogram generation has obvious advantages for a low-distortion histogram modification design over the normal one which simply counts the frequencies of pixel values. Currently, the existing PEE-based researches are roughly categorized into following four topics: one-dimensional (1D) histogram generation, 1D histogram modification, two-dimensional (2D) histogram generation, and 2D histogram modification. For the 1D histogram generation, a common way for the PEE enhancement is to employ an advance predictor for accurate predictions [28,51,52,34]. So, in these works, PEE is often investigated as a subproblem of image prediction, owing to the intuition that the more accurately you predict the pixel, the more small-magnitude prediction-errors you can obtain, and then the less distortion is caused by data embedding. Another way for improving PEE is based on the 1D and 2D histogram modification, which aims to adaptively embed data according to the image content. The basic idea behind this is simple: regions characterized by small values in smoothness introduce a small embedding distortion, and should be used for embedding at first. In [29], Sachnev et al. determined the modification manner of 1D PEH by sorting the prediction-errors, and can obtain an excellent performance. Li et al. [31] modified the 1D PEH by adaptively embedding 1 or 2 bits into a pixel based on its local smoothness. Here, the property based on the absolute pixel differences in a local region is used to quantify the smoothness of a pixel. Coatrieux et al. [35] proposed to adaptively modify the pixel according to its local specificity, and determine the most suitable carrier-class in the 1D PEH for data embedding. In our previous work [37], several new 2D mappings are proposed for the modifications of 2D PEH.

Integer-to-integer transform based RDH methods are designed to modify several adjacent pixels together by the mathematical transform, and can achieve a very high embedding rate. Coltuc [44] proposed a very fast transform by modifying the pixel pair to hide a bit. Wang et al. [46] proposed to embed data into the pixel block by extending Coltuc's scheme [44]. Coltuc [47] proposed a low-distortion transform by using the prediction-errors for embedding. In the transform, both the current prediction-error and its context are modified to hide a bit, and the performance is optimized by minimizing the square errors of pixels. Peng et al. [48] proposed an adaptive transform scheme which embeds a different amount of data in blocks according to the block type.

Among the above techniques, the high-fidelity RDH methods have been continuously receiving much attention in recent years. For the high-fidelity RDH, several methods [13,23,37,53–57] have been proposed in literature. Among them, the pairwise PEE [37] is proposed to exploit the high-order correlations among prediction-errors. The primary difference from the traditional methods is that every two prediction-errors are combined as an unit, and modified together according to a given mapping in the two-dimensional space. This makes it suitable to fully exploit the complex correlations within an image. Obviously, as illustrated in Fig. 1, the work [37] mainly focuses on the 2D histogram modification, and the other question regarding histogram generation has not been discussed yet. That is, the conditions regarding how a pair of pixels is combined properly is unclear. From this point of view, the conventional pairwise PEE could be further improved. In this paper, rather than simply employing an advanced predictor, we turn to improve the combination of pixel pairs such that a lower-entropy 2D histogram can be obtained. Our work is an extension of pairwise PEE [37] by considering the optimization of 2D histogram generation.

In this paper, we propose a high-fidelity RDH method using pairwise PEE and geodesic path to implement an improved 2D

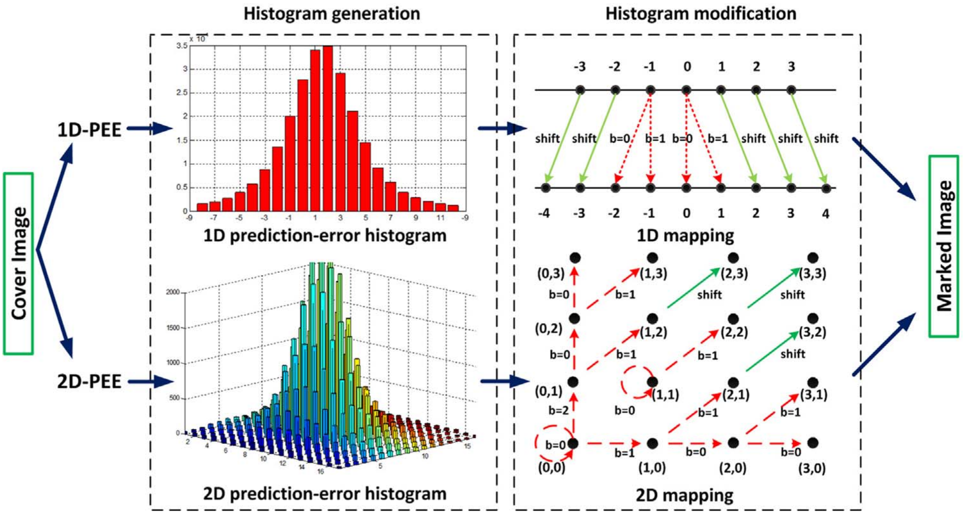


Fig. 1. Working flow of PEE: PEE first builds a PEH by histogram generation. Then, a given mapping is performed according to the derived histogram. Finally, the marked image is obtained.

histogram generation. The objective of the proposed method is to further reduce the embedding distortion by easily and effectively determining pixels into pairs for pairwise PEE, such that the combined pixels are more correlated to each other. Specifically, the scan order for pixels is first permuted in a distributed way according to a permutation algorithm. Then, for each pixel, we compute its geodesic paths to all neighbors within a pre-defined window, and choose the one to which the distance along the path is shortest as its partner. In contrast with the previous pairwise PEE method [37], both spatial and intensity distances are taken into account for measuring the similarity of two pixels. The underlying algorithm is based on the simple premise that nearby pixels in spatial-intensity space should also have similar prediction-errors. The pixel pairing technique proposed in this paper explores a new strategy, which jointly utilizes both the spatial and intensity correlations to derive a more advanced 2D histogram generation method. Thus our contribution is a simple yet effective 2D histogram generation technique to enhance 2D RDH methods. Experimental results demonstrate that the proposed method can yield a superior method than the previous pairwise PEE method [37], but also some other state-of-the-art methods [29,13,23,40].

The rest of paper is organized as follows. Section 2 presents a review of pairwise PEE, and establishes a link between the embedding performance and the similarity of paired pixels. In Section 3, we first introduce the fundamental notations of geodesic path, and then propose a design of adaptive pixel combination for pairwise PEE accounting for the local content. After that, a detailed extension for pairwise PEE using double-layer embedding and sorting technique is described in this section as well. Experimental results and some discussions are given in Section 4. Finally, Section 5 concludes this paper.

2. Related works

There are two basic ways to improve the data embedding of pairwise PEE. The first way is to design a content-based 2D mapping, and the price for this enhancement is an increase of computation complexity for automatically searching a 2D mapping suitable for a given image. The second way is to improve the 2D PEH, i.e., construct a more compact pair sequence where the pixels in a pair is more correlated to each other. In this paper, we focus on the second way for improving the pairwise PEE.

From a data compression viewpoint, a compact pair sequence indicates that the average number of bits required to represent the original status are less, thus leaving more spaces for data embedding. Suppose that a pixel can be represented by one bit, if the pixels in a pair are the same, we normally need only one bit to represent them, and can use the saved bit for data hiding; if they are totally different, two bits are required and no secret bit can be embedded. This is the fundamental motivation of the proposed method for pixel combination. We shall show how to construct a pair of prediction-errors in a reasonable way. Before our presentation, we simply review the mechanism of pairwise PEE and show how a pair of pixels is constructed in the conventional pairwise PEE.

2.1. Conventional pairwise PEE [37]

Pairwise pixel combination, 2D PEH generation and 2D mapping design constitute a complete pairwise PEE embedding procedure. Thus, the basic algorithm for data embedding proceeds as follows.

- **Step 1: Pairwise pixel combination** The pixels in a cover image I is categorized into two types, namely, shadow and blank pixels, based on the check-board like partition shown in Fig. 2. For the sake of reversibility, the data embedding on the two types of pixels are processed separately. Here, we take the shadow pixels for illustration. As shown in the figure, to construct a pair sequence, let W be a 3×3 sized window that scans I in the left-to-right, top-to-bottom order. In the window, the central pixel p_i and its bottom-right neighbor $p_{\gamma(i)}$, with the coordinates (s,t) and $(s+1,\ t+1)$, respectively, are combined as a pair. So, the Euclidian distance between them is $\sqrt{2}$. Note that, to prevent the overlaps between different pairs, the window is moved with an interval of one pixel in the

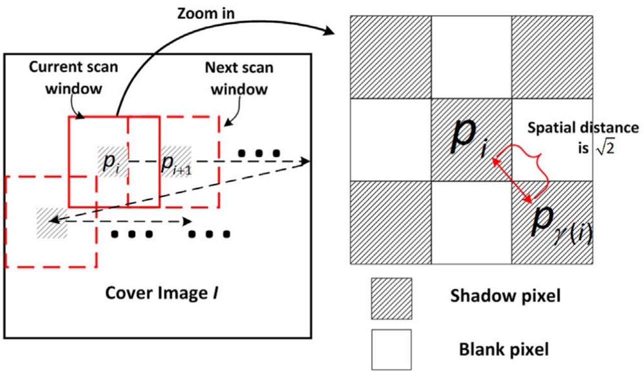
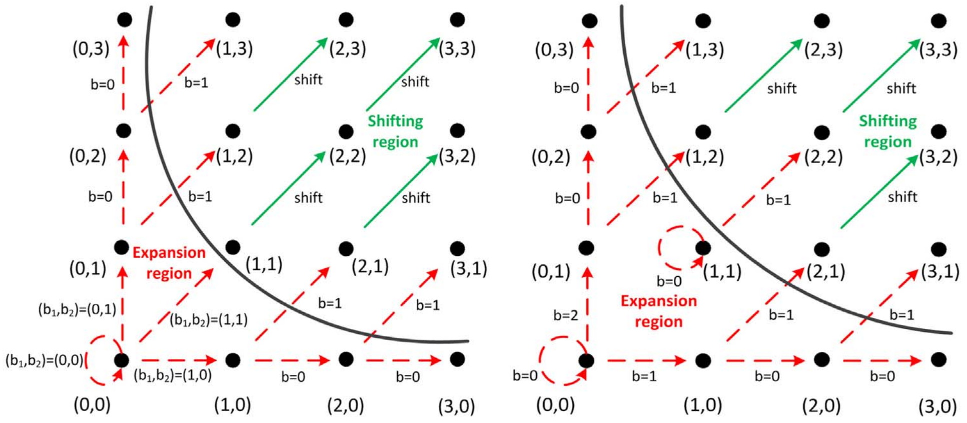


Fig. 2. The pixel combination mode in the conventional pairwise PEE [37], where the central pixel is always combined with its bottom-right neighbor as a pair.



(a) (b)

Fig. 3. Transformation of prediction-errors of PEE in terms of 2D mapping by limiting the maximum modification on a pixel as 1. (a) Conventional PEE. (b) Conventional pairwise PEE [37].

horizontal or vertical direction as illustrated in the figure.

- **Step 2: 2D PEH generation** After the pixel pair sequence is determined, use the blank pixels to predict the shadow ones. Specifically, for a pair of pixels $(p_i, p_{\gamma(i)})$, the corresponding predictions $(\widehat{p}_i, \widehat{p}_{\gamma(i)})$ are computed as the average of their four nearest blank pixels, respectively. For simplicity, suppose that the prediction is integer-valued. The prediction-error pair $\mathbf{e}_i = (x_i, y_i)$ is obtained as $x_i = p_i - \widehat{p}_i$ and $y_i = p_{\gamma(i)} - \widehat{p}_{\gamma(i)}$, respectively. Given a 2D histogram function h: $\mathbb{Z}^2 \rightarrow \mathbb{Z}$, the associated population of prediction-error pair $h(k_1, k_2)$ is

$$h(k_1, k_2) = \mathbf{card}\{1 \leq i \leq N/2\colon x_i = k_1,\ y_i = k_2\} \tag{1}$$

where the function **card** returns the cardinal number of a set and N is the number of shadow pixels.

- **Step 3: 2D mapping design** Because every two prediction-errors are modified together in pairwise PEE, the transformation between the original and marked prediction-errors can be equivalently represented as a 2D mapping. In the method [37], by slightly modifying the conventional mapping shown in Fig. 3 (a), a better performed mapping is designed as shown in Fig. 3 (b), where only the first quarter is plotted for simplicity. In Fig. 3 (b), the prediction-error pairs are classified into four types as shown in Fig. 4 and modified as follows. Note that here only the mapping for the 2D bins of the first quarter are described, and the modification for the other three quarters are omitted for simplicity.
  - Type A: If $(x_i, y_i) = (0, 0)$, it has three candidate mapping directions for data embedding and can be embedded with log_{23} bits. More specifically, it will be mapped into $(0, 0)$, $(0, 1)$ and $(1, 0)$, when the to-be-embedded ternary bit is 0, 1 and 2, respectively.
  - Type B: If $(x_i, y_i) = (1, 1)$, it can be embedded with 1 bit. Specifically, the marked pair (x'_i, y'_i) is obtained as $(1, 1)$ and $(2, 2)$ for a binary bit 0 and 1, respectively.
  - Type C: If $x_i = 0$, $y_i \geq 1$ or $x_i \geq 1$, $y_i = 0$, the prediction-error pair is carried with 1 bit. The marked pair is obtained as $(x'_i, y'_i) = (x_i + b, y_i + 1)$ or $(x'_i, y'_i) = (x_i + 1, y_i + b)$, respectively, where b is a binary bit.
  - Type D: If $(x_i, y_i) \notin$ Type (A-C), it will be shifted as $(x'_i, y'_i) = (x_i + 1, y_i + 1)$ to create vacancy.

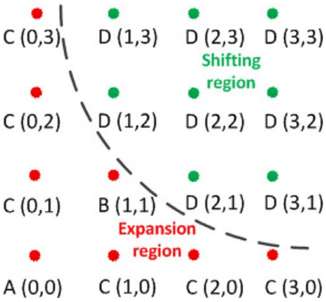


Fig. 4. Classification of 2-D bins for the first quarter of the 2D PEH.

At decoder, one needs to know the 2D mapping of data embedding, and then can make the perfect recovery of image by the inverse manner. Compared with the conventional PEE, the fundamental difference is that the prediction-error pair $(x_i, y_i) = (0, 0)$ is designed to carry log_{23} bits on average, thus making the pair $(1, 1)$ available for carrying 1 bit. The capacity EC and embedding distortion ED are

$$EC = \log_2 3 \sum_{\mathbf{e}\in A} h(\mathbf{e}) + \sum_{\mathbf{e}\in B} h(\mathbf{e}) + \sum_{\mathbf{e}\in C} h(\mathbf{e}) \tag{2}$$

and

$$ED = \frac{2}{3} \sum_{\mathbf{e}\in A} h(\mathbf{e}) + \sum_{\mathbf{e}\in B} h(\mathbf{e}) + \frac{3}{2} \sum_{\mathbf{e}\in C} h(\mathbf{e}) + 2 \sum_{\mathbf{e}\in D} h(\mathbf{e}), \tag{3}$$

respectively. Comapred with the conventional PEE (see Fig. 3 (a)), the distortion reduction Δ_{ED} and the increasement in capacity Δ_{EC} can be estimated as

$$\Delta_{ED} = \frac{1}{3} \sum_{\mathbf{e}\in A} h(\mathbf{e}) + \sum_{\mathbf{e}\in B} h(\mathbf{e}) > 0 \tag{4}$$

and

$$\Delta_{EC} \approx 0.415 \sum_{\mathbf{e}\in A} h(\mathbf{e}) - \sum_{\mathbf{e}\in B} h(\mathbf{e}), \tag{5}$$

respectively. In [37], the new 2D mapping is verified to be useful in reducing the embedding distortion while preserving the capacity for most of images.

In the above procedure, the pixel combination for pairs is based on the assumption that the similarity of pixels is dependent of spatial coordinate. Although this assumption is valid, it has limitations in dealing with the complex regions of an image, such as edges, where the relationship of neighbor pixels cannot be simply measured by the spatial distance. In this case, a precise pixel combination can bring a performance improvement. Hence, there is a huge potential for improving the way of pixel combination in pairwise PEE.

3. Proposed method

In this section, we present a new pixel combination way and the associated algorithm to achieve a better embedding performance of pairwise PEE. Fig. 5 gives the overview of the proposed embedding

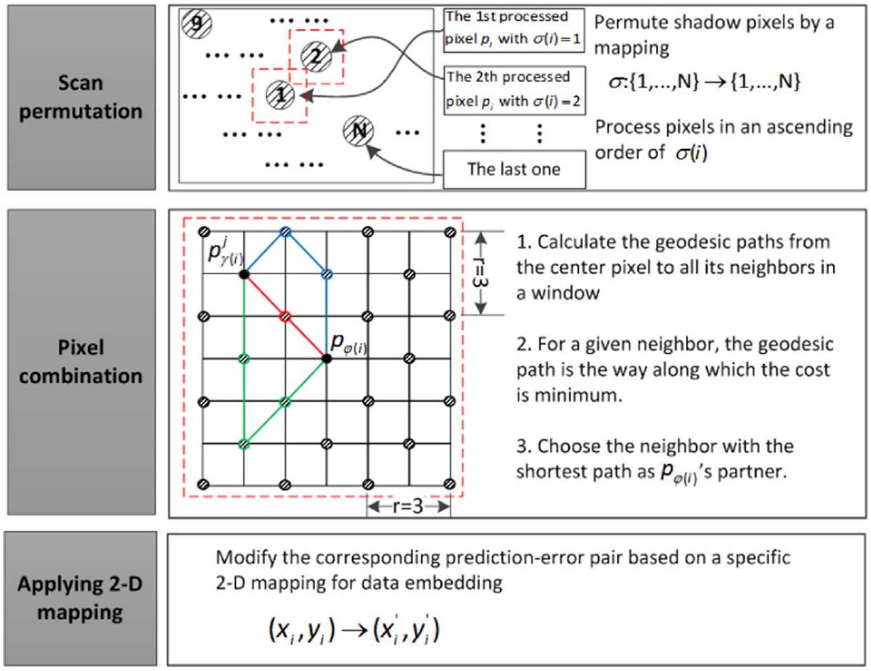

Scan permutation
9
2
1
N
The 1st processed pixel p_i with $\sigma(i)=1$
The 2th processed pixel p_i with $\sigma(i)=2$
The last one
Permute shadow pixels by a mapping
$\sigma:\{1,\ldots,N\}\rightarrow\{1,\ldots,N\}$
Process pixels in an ascending order of $\sigma(i)$
Pixel combination
$p^j_{\gamma(i)}$
$p_{\varphi(i)}$
r=3
r=3
1. Calculate the geodesic paths from the center pixel to all its neighbors in a window
2. For a given neighbor, the geodesic path is the way along which the cost is minimum.
3. Choose the neighbor with the shortest path as $p_{\varphi(i)}$'s partner.
Applying 2-D mapping
Modify the corresponding prediction-error pair based on a specific 2-D mapping for data embedding
$(x_i, y_i)\rightarrow(x_i', y_i')$

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

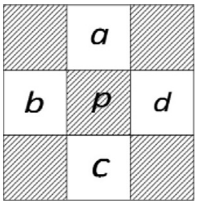


Fig. 6. The classification of shadow and blank pixels, where the context of a pixel p includes the upper, left, bottom and right neighbors $\{a, b, c, d\}$.

process involving scan permutation, pixel combination and 2D modification on prediction-error pair sequence. The prior work has shown that designing the 1D-PEE algorithm based on a concentrated PEH can yield a better performance. The similar reasoning applies to the proposed method in 2D space. Thus, the proposed pixel combination aims to determine the most similar neighbor to a pixel by measuring the geodesic path between them. As shown in Fig. 5, to compute the geodesic path between two pixels, one needs to first find all feasible paths and then choose the optimal way with the smallest geodesic distance. After the pixels are combined into pairs, a specific 2D mapping is applied for data embedding. Since our method also adopts the double-layer embedding (see the layer classification in Fig. 6), in the following, we only take the data embedding on shadow pixels for illustration.

In the proposed scheme, the scan order of pixels are first permuted before the pixel combination. Because a center pixel located in a window shares the neighborhood with the ones in adjacent windows, a conflict may occur when two pixels choose the same neighbor for combination, and one of them cannot choose the most similar neighbor as it has been paired previously. This would cause a decrease in performance when a certain number of smooth pixels have to combine with the rough ones. To reduce the conflicts, the permutation is used to rearrange the scan order, so that the pixels are processed in a distributed way. By permutation, it is expected that the preferentially processed pixels can have high priorities in selection of partner because most of its neighbors are available for the current combination. Since the smooth pixels are utilized first in RDH, this operation is thus helpful for the reduction of embedding distortion.

Let $(p_1, p_2, \ldots, p_N)$ be the shadow pixels. By a given one-to-one unique mapping σ: $\{1,\ldots,N\}\rightarrow\{1,\ldots,N\}$, the pixels are permuted as $(p_{\sigma(1)}, p_{\sigma(2)}, \ldots, p_{\sigma(N)})$ with the ascending order of σ. It is noted that the permutation key should be recorded as side information, and transmitted to the receiver for recovering the order. Let the final pair sequence be $(p_{\varphi(1)}, p_{\gamma(1)}), \ldots, (p_{\varphi(N/2)}, p_{\gamma(N/2)})$. For a pixel $p_{\varphi(i)}$, we search its partner in the window with the size of $(2r + 1) \times (2r + 1)$, where r is the window size parameter. As we use double-layer embedding, only a half of pixels in the window are taken as the candidates. The maximum number of candidates for the combination of $p_{\varphi(i)}$ is 2r^{2+2r}, and the candidate set is denoted as $\{p^1_{\gamma(i)}, \ldots, p^{2r^2+2r}_{\gamma(i)}\}$. Here, once a neighboring pixel has been paired, it is regarded unavailable for the later combination.

For two points p_i and p_j, the geodesic distance $d_g(p_i, p_j)$ utilized by Yatziv and Sapiro [58] is defined as

$$d_g(p_i, p_j) = \min_{\Gamma} \int_0^1 |\nabla\Omega\cdot\dot{\Gamma}(m)|dm, \tag{6}$$

where Γ denotes a path from p_i to p_j, $\dot{\Gamma}(m)$ denotes the tangent of the path Γ at the pixel m, and the integral counts the accumulative directional derivative at all pixels along the path, respectively. The geodesic path between two pixels reflects the shortest distance among all possible paths in terms of gradient. The smoother the path and the shorter in spatial distance, the smaller the geodesic value $d_g(p_i, p_j)$. It can be viewed as a powerful similarity measure for two pixels by considering both the intensity and spatial correlations. However, the brute-force application of geodesic path is not suitable for RDH due to the reversibility requirement. Because the distance along the same path changes as the corresponding pixels are modified after data embedding.

Suppose Γ be a path from $p_{\varphi(i)}$ to $p^j_{\gamma(i)}$, with the length of l, on the discrete shadow pixels. To facilitate the geodesic path for RDH, for the pixel $p_{\varphi(i)}$ and its candidate $p^j_{\gamma(i)}$, we define the geodesic distance $d_g(p_{\varphi(i)}, p^j_{\gamma(i)})$ along the path Γ as

$$d_g(p_{\varphi(i)}, p^j_{\gamma(i)}) = \min_{\Gamma} \sum_{k=1}^{l-1} d_1(m_k, m_{k+1}) + d_2(m_k, m_{k+1}), \tag{7}$$

where $j \in [1, 2r^2 + 2r]$, m_k denotes the pixel on the path, and the functions d_1, d_2 compute the spatial and intensity distances, respectively. More specifically, $m_1 = p_{\varphi(i)}$ and $m_l = p^j_{\gamma(i)}$. The spatial distance

$d_1(m_k, m_{k+1})$ is computed as the Euclidian distance of coordinates, i.e.,

$$d_1(m_k, m_{k+1}) = \|(s, t) - (u, v)\|_2, \tag{8}$$

where (*s*,*t*) and (*u*,*v*) denote the coordinates of the pixels m_k and m_{k+1}, respectively. The intensity distance $d_2(p_{\varphi(i)}, p^j_{\gamma(i)})$ is calculated as the summed absolute differences between the 4-neighbors of $p_{\varphi(i)}$ and $p^j_{\gamma(i)}$, i.e.,

$$d_2(m_k, m_{k+1}) = \|\phi(m_k) - \phi(m_{k+1})\|_1, \tag{9}$$

where the neighborhood set $\phi(\cdot)$ for the coordinate (*s*,*t*) contains its four neighboring pixels in the vertical and horizontal directions, whose coordinates are given by $(s + 1, t)$, $(s, t + 1)$, $(s - 1, t)$, $(s, t - 1)$. Specifically, for a shadow pixel, its 4-connected neighbors are blank ones as shown in Fig. 6. This guarantees that the computations of d_1 and d_2 are both invariant before and after embedding. After the geodesic distances between the center pixel $p_{\varphi(i)}$ and all its candidates $p^j_{\gamma(i)}, j \in [1, 2r^2 + 2r]$ are obtained, the optimal pair is chosen with the minimum distance as

$$(p_{\varphi(i)}, p_{\gamma(i)}) = \underset{j}{\operatorname{argmin}} d_g(p_{\varphi(i)}, p^j_{\gamma(i)}). \tag{10}$$

It should be noted that if the candidate set is empty, the pixel $p_{\varphi(i)}$ would be skipped and not utilized in data embedding.

The minimization of 10 can be reduced to the search of the shortest path in a weighted graph, where the pixels within a window are taken as the vertices of graph. To find the shortest path from a pixel to its faraway neighbor, we are asked to view the edges as distances, and the cost of a path is the sum of weights of the edges on that path. For example, when $d_g(p_{\varphi(i)}, p^j_{\gamma(i)})=+\infty$, it means that there is no path at all from $p_{\varphi(i)}$ to $p^j_{\gamma(i)}$. There are lots of ways to solve the shortest path problem. We use the Dijkstra's algorithm proposed by Dutch computer scientist Edsger Dijkstra [59] to find the costs of a path from a pixel to another one.

Fig. 7 gives an example of the determination of geodesic paths from the center pixel to its neighbors by using Dijkstra's algorithm, where the radius of window is *r*=1. The candidate set for the center pixel $p_{\varphi(i)}$ is obtained as $\{p^1_{\gamma(i)}, \ldots, p^4_{\gamma(i)}\}$. From the figure, one can see that, in the initial stage (see Fig. 7 (a)), the edges between adjacent shadow pixels are assigned as the sum of the corresponding d_1 and d_2, respectively. Then, by Dijkstra's algorithm, the minimum geodesic distance is updated step by step. For example, the initial geodesic distance $d_g(p_{\varphi(i)}, p^1_{\gamma(i)})$ from $p_{\varphi(i)}$ to $p^1_{\gamma(i)}$ is 15.4 (see Fig. 7 (a)), because the spatial distance d_1 and the difference of neighborhood d_2 are computed as $d_1(p_{\varphi(i)}, p^1_{\gamma(i)}) = \sqrt{2}$ and $d_2(p_{\varphi(i)}, p^1_{\gamma(i)}) = \|\phi(p_{\varphi(i)}) - \phi(p^1_{\gamma(i)})\|_1 = 14$, respectively, where the neighbor sets $\phi(p^1_{\gamma(i)}) = \{160, 160, 158, 155\}$ and $\phi(p_{\varphi(i)}) = \{155, 158, 160, 160\}$. In Fig. 7 (b), the final distance $d_g(p_{\varphi(i)}, p^1_{\gamma(i)})$ is updated as 12.4, since the path of $p_{\varphi(i)} \to p^3_{\gamma(i)} \to p^1_{\gamma(i)}$ costs less than that of $p_{\varphi(i)} \to p^1_{\gamma(i)}$.

Because the complexity of Dijkstra's algorithm is determined by the number of vertices, the search complexity is dependent on the size of window. Evidently, a bigger window provides some extra freedom in selection of two similar pixels, and has the potential to yield a more accurate pixel combination. This is because that the pixel can search its partner in a larger neighborhood and makes better use of possible self-prediction that the image can provide. However, the complexity would increase dramatically. In practice, the choice of *r* is a trade-off. In our method, the window parameter *r* ranges from 1 to 3, and the best one is determined to achieve the highest PSNR.

To better describe the details of pixel combination after permutation, Fig. 8 gives an example of the pair sequence construction for a 4×4 pixel block. It can be seen that the first window is located at the position *i*=6 as the permutation value $\sigma(6) = 1$. The partner of p_6 is then determined within the window. Suppose that p_2 is the partner of p_6, the corresponding indices for the first pair are $\varphi(1) = 6$ and $\gamma(1) = 2$, respectively. Similarly, the second window is centered at the position with $\sigma(11) = 2$. Suppose that the second pair is determined as (p_{11}, p_{16}) with the indices of $\varphi(2) = 11$ and $\gamma(2) = 16$. It can be seen that the upper-left pixel in the second window is unavailable for combination, as it has been combined into a pair in the first window. Repeat the above process until the window is centered at $\sigma(12) = 16$, the pair sequence is finally obtained. For the whole image, the corresponding construction process is similar to that of the pixel block, and omitted here for simplicity.

To investigate the benefit of the proposed pixel combination, we use the entropy to evaluate the derived 2D histogram. The smaller the histogram entropy, the more concentrated the 2D histogram and the lower embedding distortion for a capacity. Table 1 lists the entropies for the 2D prediction-error histograms using the proposed adaptive pixel combination and the fixed manner [37], respectively. Only the shadow pixels are utilized for data embedding, and the prediction-error is computed as the average of four nearest neighbors. By the comparison, it is found that the proposed histogram generation can result in a lower entropy for the test images. It demonstrates that, by considering the geodesic distance as the similarity measurement between two pixels, a more compact pair sequence with a smaller entropy can be obtained than the previous method [37]. Besides, based on the two derived prediction-error histograms, the corresponding PSNRs for

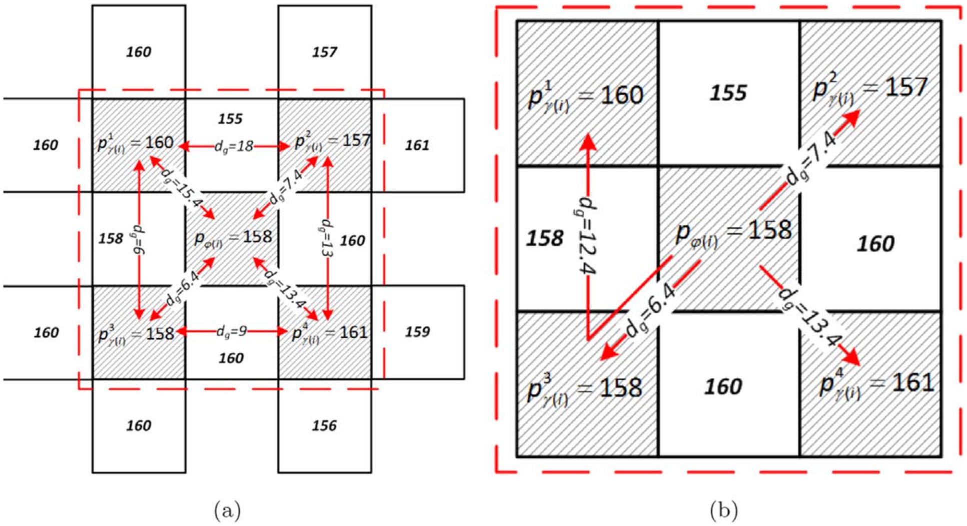


Fig. 7. An example to determine the shortest geodesic paths using Dijkstra's algorithm from the center pixel to all its neighbors, where the window radius *r* is 1. (a) The initialization of Dijkstra's algorithm by taking the distances defined in (8) and (9) as edges of graph. (b) The final geodesic paths from $p_{\varphi(i)}$ to its four neighbors.

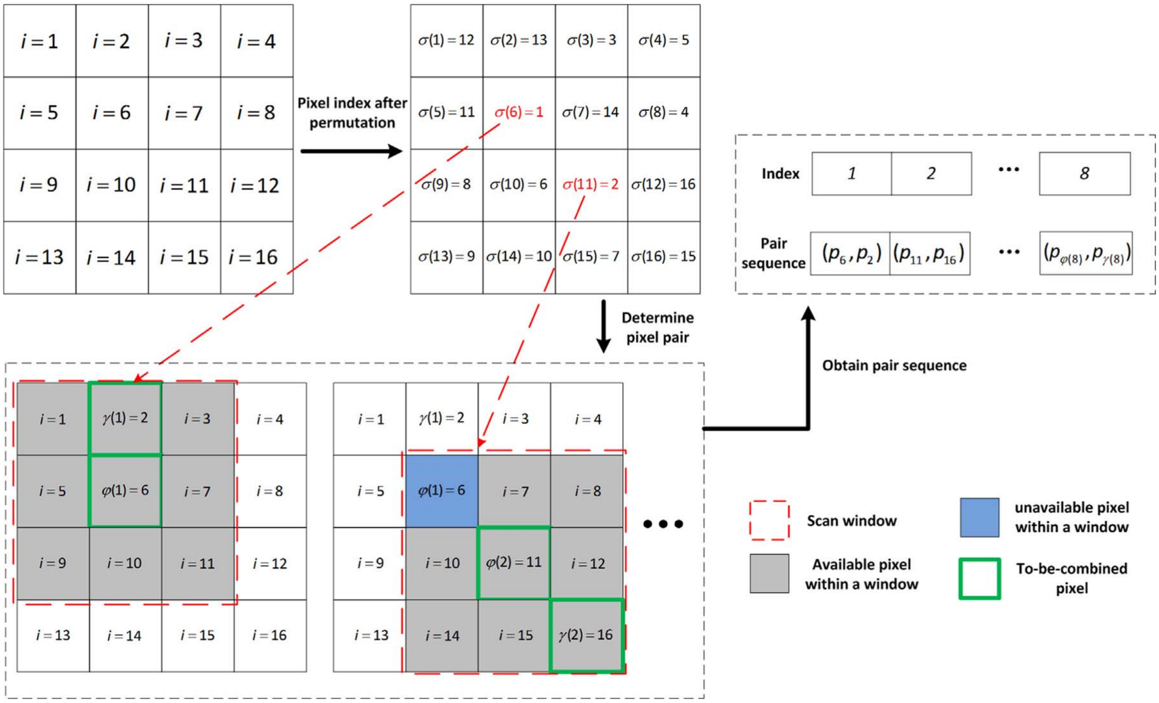

i=1 i=2 i=3 i=4
i=5 i=6 i=7 i=8
i=9 i=10 i=11 i=12
i=13 i=14 i=15 i=16
Pixel index after permutation
σ(1)=12 σ(2)=13 σ(3)=3 σ(4)=5
σ(5)=11 σ(6)=1 σ(7)=14 σ(8)=4
σ(9)=8 σ(10)=6 σ(11)=2 σ(12)=16
σ(13)=9 σ(14)=10 σ(15)=7 σ(16)=15
Determine pixel pair
Index
1
2
8
Pair sequence
(p6, p2) (p11, p16) (pφ(8), pγ(8))
Obtain pair sequence
γ(1)=2
φ(1)=6
φ(2)=11
γ(2)=16
Scan window
Available pixel within a window
unavailable pixel within a window
To-be-combined pixel

Fig. 8. Construction of pair sequence for a 4×4 pixel block, where the radius of window is set as r=1. The scan window is moved in an ascending order of the permutation value σ.

Table 1
Entropies of 2D prediction-error histograms derived by the conventional pairwise PEE [37] and the proposed method. Note that here only the shadow pixels are used for comparison.

Images	Lena	Baboon	Airplane	Barbara	Elaine	Lake
Conventional	5.592	8.059	5.188	6.679	6.729	6.771
Proposed	5.567	8.037	5.152	6.572	6.704	6.753

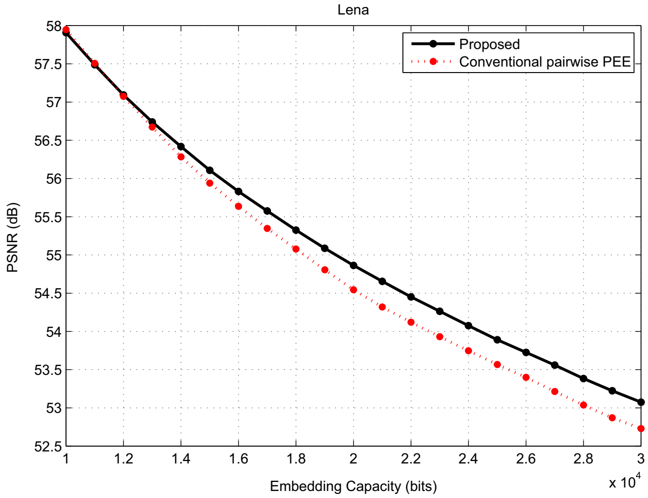


Fig. 9. Performance comparison on Lena image by using the conventional pairwise PEE [37] and the proposed method. Here, only the shadow pixels are utilized for data embedding.

Lena image are given in Fig. 9. It is noted that the same 2D mapping is employed for the two histograms. From the figure, one can see that the proposed pixel combination yields a higher PSNR than the conventional pairwise PEE method in most cases.

In addition, similar to the previous PEE-based methods, we also employ a sorting technique to modify the smooth pixel pairs first for data embedding. Here, the noise level of a pixel $p_{\varphi(i)}$ is computed as the summed absolute differences of the context, i.e.,

$$nl\,(p_{\varphi(i)}) = |a-b| + |b-c| + |c-d| + |d-a|, \tag{11}$$

where a, b, c and d are the upper, left, bottom and right neighbors of a pixel as shown in Fig. 6. Then, the noise level NL_i for a pair $(p_{\varphi(i)}, p_{\gamma(i)})$ is defined as

$$NL_i = nl\,(p_{\varphi(i)}) + nl\,(p_{\gamma(i)}). \tag{12}$$

By setting a threshold t, only the pairs with $NL_i \le t$ are processed during embedding. For a given window size, the optimal threshold is determined as the smallest integer to meet the required payload. So, there are actually two parameters, r and t in the proposed method. The parameters are adjusted according to the following rule

$$\begin{cases} \underset{1\le r\le 3, 0\le t\le NL_{max}}{\text{minimize}} \dfrac{ED\,(r,t)}{EC\,(r,t)} \\ \text{subject to } EC\,(r,t) \ge PS \end{cases} \tag{13}$$

where NL_{max} is the maximum of noise level for a pair and PS is the payload size.

Embedding and extracting procedures.

The side information required in the proposed method is used to help the perfect recovery of the scan order and determine the to-be-processed pairs during data extraction. It is embedded into the LSBs of a predefined image region by LSB replacement. For a 512×512 sized image, it consists of four parts, including the permutation key (12 bits), two parameters $\{r, t\}$ (12 bits), the end position (18 bits) and a location map. Here, the location map is used to avoid the overflow and underflow problems, i.e., ensure that the pixels fall in the range $[0, 255]$ after data embedding. Specifically, for a pair $(p_{\varphi(i)}, q_{\gamma(i)})$, it is marked with “1” in the location map if the pair has a pixel with a value of 0 or 255; else it will be marked with “0”. Based on the location map, a pair would be skipped if it is marked with “1” during data embedding/extraction.

For the double-layer embedding, the data embedding is processed by two steps, i.e., first manipulate the shadow pixels and then the blank ones. Thus, the payload is split into two parts and equally embedded into each layer. The diagram of embedding procedure is given in Fig. 10. It is noted that the parameters $\{r, t\}$ are first determined to

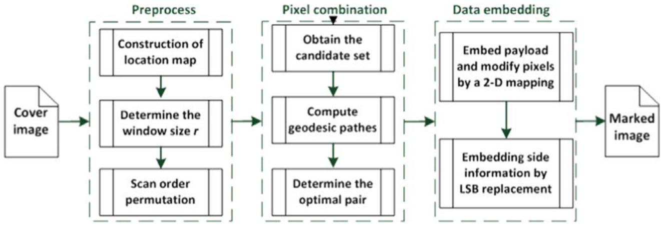

Preprocess
Pixel combination
Data embedding
Cover image
Construction of location map
Determine the window size r
Scan order permutation
Obtain the candidate set
Compute geodesic pathes
Determine the optimal pair
Embed payload and modify pixels by a 2-D mapping
Embedding side information by LSB replacement
Marked image

Fig. 10. The details of embedding procedure.

satisfy the minimization of 13, where the embedding capacity and distortion are estimated by using (2) and (3), respectively.

For better illustration, only the detailed data embedding for a given window size r is described in the following. First, permute the scan order of pixels by their locations based on a one-to-one unique mapping σ, and process the pixel in an ascending order of σ. For a pixel $p_{\varphi(i)}$, obtain the candidate neighbor set in the $(2r + 1) \times (2r + 1)$ window as $\{p^1_{\gamma(i)}, \dots, p^n_{\gamma(i)}\}$, where the maximum number of neighbors is $2r^{2+2r}$. Here, the candidate set only includes the available neighbors which have not been combined previously. If the number of candidate is larger than 1 ($n \geq 1$), compute the geodesic distances from $p_{\varphi(i)}$ to its

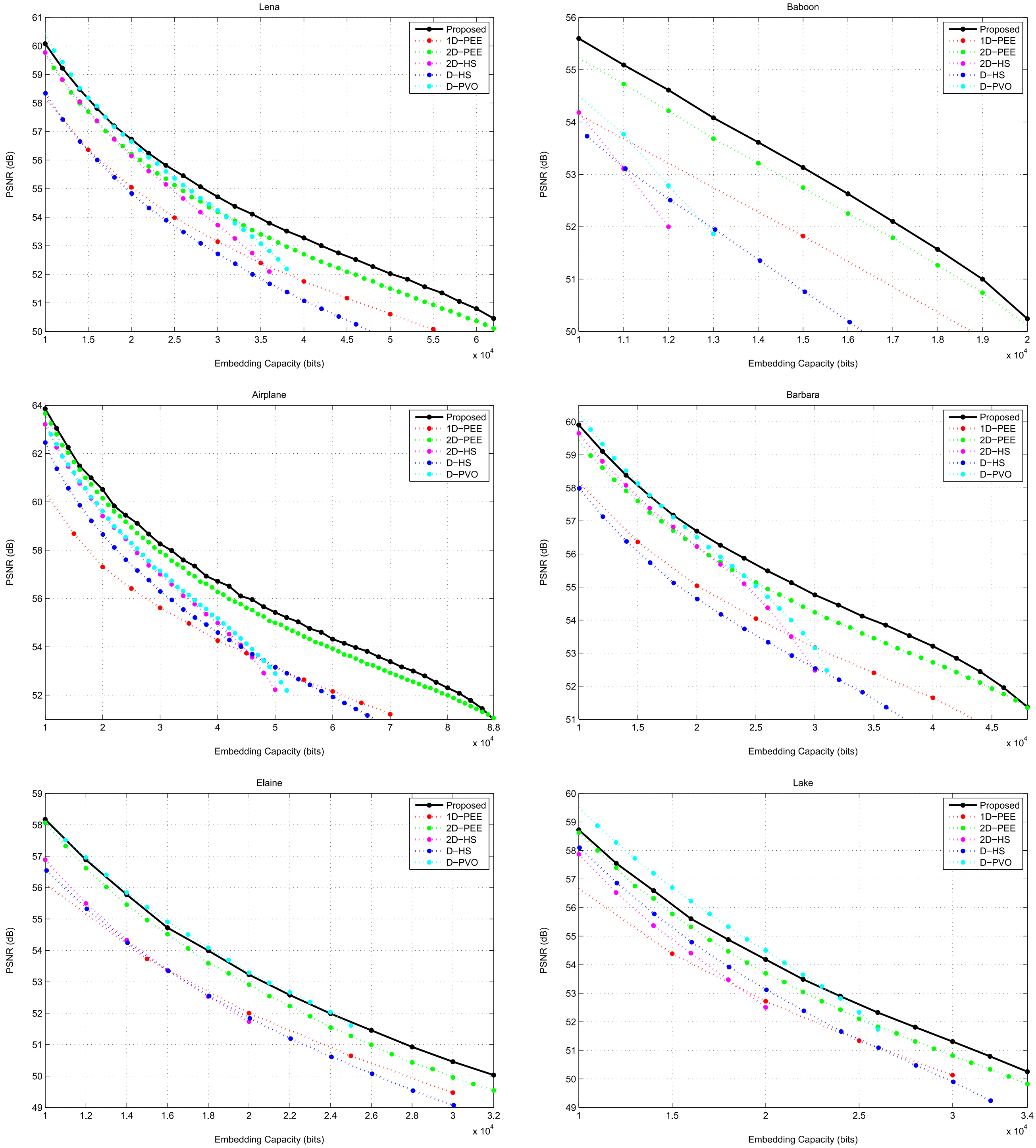


Fig. 11. Performance evaluation in terms of capacity-distortion curve by comparing the proposed method with four methods including 1D-PEE [29], 2D-PEE [37], 2D-HS [13], D-HS [23], D-PVO [40].

Table 2
The corresponding PSNR values (dB) of the proposed method and 2D-PEE [37] for the capacities of 10,000, 20,000, 30,000 and 40,000 bits on the six standard images. "–" indicates that the image cannot be embedded with such a capacity.

Capacity (bits)		10,000	20,000	30,000	40,000
Lena	2D-PEE	59.70	56.21	54.19	52.70
	Proposed	**60.08**	**56.73**	**54.71**	**53.27**
Baboon	2D-PEE	55.22	50.11	–	–
	Proposed	**55.60**	**50.24**	–	–
Airplane	2D-PEE	63.66	60.15	57.93	56.27
	Proposed	**63.85**	**60.51**	**58.26**	**56.71**
Barbara	2D-PEE	59.43	56.23	54.24	52.72
	Proposed	**59.90**	**56.69**	**54.76**	**53.21**
Elaine	2D-PEE	58.06	52.91	49.95	–
	Proposed	**58.17**	**53.23**	**50.46**	–
Lake	2D-PEE	58.62	53.70	50.82	–
	Proposed	**58.72**	**54.18**	**51.31**	–

Table 3
Performance comparison on Kodak image database for a capacity of 10,000 bits. The best result for an image is shown in boldface.

Image	1D-PEE	2D-PEE	2D-HS	D-HS	Proposed
kodim01	61.86	63.94	61.41	59.11	**63.95**
kodim02	61.83	**64.21**	63.67	62.92	64.18
kodim03	63.53	65.01	64.68	63.83	**65.15**
kodim04	61.88	**63.84**	63.03	61.82	63.82
kodim05	61.74	63.44	61.47	60.54	**63.64**
kodim06	65.33	66.86	65.70	64.52	**66.92**
kodim07	63.04	64.42	63.99	63.46	**64.65**
kodim08	**58.44**	57.52	55.07	55.74	57.72
kodim09	60.53	61.76	61.69	60.74	**62.11**
kodim10	60.65	61.20	60.35	60.75	**62.44**
kodim11	62.86	64.84	64.40	63.60	**65.00**
kodim12	62.73	**64.54**	63.63	62.93	64.53
kodim13	57.79	58.94	52.47	51.12	**59.04**
kodim14	60.83	**62.93**	61.24	60.23	62.66
kodim15	62.31	62.61	62.29	61.06	**63.04**
kodim16	63.01	64.73	64.20	63.66	**64.89**
kodim17	61.58	62.95	62.44	61.74	**63.15**
kodim18	58.82	59.57	58.34	58.54	**60.01**
kodim19	60.75	62.08	61.79	60.14	**62.26**
kodim20	53.40	54.52	**60.22**	59.24	58.53
kodim21	60.70	61.91	61.24	59.61	**62.15**
kodim22	60.31	61.44	60.90	58.52	**61.94**
kodim23	62.27	63.79	62.94	61.77	**64.56**
kodim24	58.29	59.65	58.80	57.82	**60.13**
Average	61.02	62.36	61.49	60.53	**62.77**

Table 4
Performance comparison on Kodak image database for a capacity of 20,000 bits. The best result for an image is marked boldface. Here, the comparison on kodim13 is eliminated as 2D-HS cannot fulfill the capacity requirement.

Image	1D-PEE	2D-PEE	2D-HS	D-HS	Proposed
kodim01	58.10	60.02	54.15	57.43	**60.05**
kodim02	58.72	60.85	58.71	58.98	**60.86**
kodim03	60.09	61.83	60.16	60.68	**61.93**
kodim04	58.58	60.38	58.03	58.35	**60.41**
kodim05	58.40	60.05	55.23	57.82	**60.07**
kodim06	61.37	**63.37**	56.60	61.73	63.26
kodim07	59.47	61.17	60.05	59.98	**61.49**
kodim08	55.07	54.94	52.41	54.40	**55.59**
kodim09	57.44	58.51	58.51	57.23	**58.67**
kodim10	57.49	58.24	57.83	57.44	**59.03**
kodim11	59.37	61.49	61.15	59.94	**61.64**
kodim12	59.07	61.11	60.38	59.58	**61.13**
kodim14	57.42	**59.03**	56.94	53.12	58.93
kodim15	59.15	60.67	60.22	58.63	**61.41**
kodim16	59.48	61.39	60.91	60.29	**61.55**
kodim17	58.20	59.64	59.38	58.99	**59.82**
kodim18	55.70	56.57	55.52	57.03	**56.89**
kodim19	57.63	58.91	58.67	57.60	**59.06**
kodim20	51.87	54.29	**59.33**	56.75	58.92
kodim21	57.58	58.71	58.42	**62.33**	58.94
kodim22	57.24	**58.46**	58.11	56.31	58.79
kodim23	58.94	60.67	60.22	59.58	**61.20**
kodim24	56.64	58.33	56.78	57.55	**60.06**
Average	57.96	59.50	58.84	58.28	**59.94**

Table 5
Performance comparison on Kodak image database for a capacity of 40,000 bits. The best result for an image is marked boldface. Here, the results of 2D-HS2-HS are not listed as it cannot provide the capacity for most of images.

Image	1D-PEE	2D-PEE	D-HS	Proposed
kodim01	52.57	53.67	50.44	**53.85**
kodim02	55.66	57.37	55.00	**57.77**
kodim03	56.61	58.51	56.92	**59.12**
kodim04	55.45	56.81	54.45	**57.33**
kodim05	54.67	56.25	53.43	**56.60**
kodim06	57.04	58.45	56.63	**58.83**
kodim07	56.29	57.83	56.19	**58.51**
kodim08	50.14	49.88	48.91	**51.29**
kodim09	54.26	55.09	53.42	**55.72**
kodim10	54.34	55.04	53.55	**55.99**
kodim11	56.02	57.91	55.53	**58.34**
kodim12	55.86	57.49	55.54	**57.97**
kodim14	53.62	54.72	51.94	**55.10**
kodim15	56.30	57.31	57.34	**59.17**
kodim16	56.26	57.97	55.96	**58.47**
kodim17	55.01	56.27	54.15	**56.81**
kodim18	52.44	53.26	50.78	**53.87**
kodim19	54.44	55.73	53.38	**56.31**
kodim20	52.68	51.27	**54.07**	53.52
kodim21	54.37	55.39	53.30	**56.04**
kodim22	54.06	55.36	52.71	**55.94**
kodim23	55.88	57.28	55.74	**58.12**
kodim24	53.94	54.71	56.48	**56.94**
Average	54.69	55.81	54.17	**56.59**

available neighbors using 7; else skip the pixel and keep it unchanged. For the case of $n \geq 1$, determine the optimal pair $(p_{\varphi(i)}, p_{\gamma(i)})$ using (10), and mark them unavailable for the later combination. Next, repeat the pixel combination until the pixel pair sequence is determined. Calculate the smoothness of a pair using (12) and obtain the prediction-error pairs using the rhombus prediction, where the prediction of a pixel is computed as its four nearest neighbors. The threshold t is determined as the smallest value to meet the capacity. Based on a specific 2D mapping, modify prediction-error pairs whose noise level $NL_i \leq t$ and location map value is "0", and embed the data bits into the pairs which belong to the expansion set. The side information is embedded in the pre-defined region of image by LSB replacement. Finally, the marked image is obtained.

The corresponding extracting procedure is described as follows. First, extract the side information and obtain the same pixel sequence according to the permutation key. By using the geodesic path, obtain the marked pixel pair sequence and compute the corresponding prediction-errors. Based on the location map and the threshold t, process the marked pair with the noise level $NL_i \leq t$ and location map value "0". According to the inverse 2D mapping, extract the hidden bits and recover the original prediction-error pair until the payload bits are extracted completely. Finally, extract the LSBs from the payload bits and recover the pixels in the pre-defined region.

4. Experimental results

In this section, serval experiments are conducted on the gray-scale images to evaluate the performance of the proposed method. The

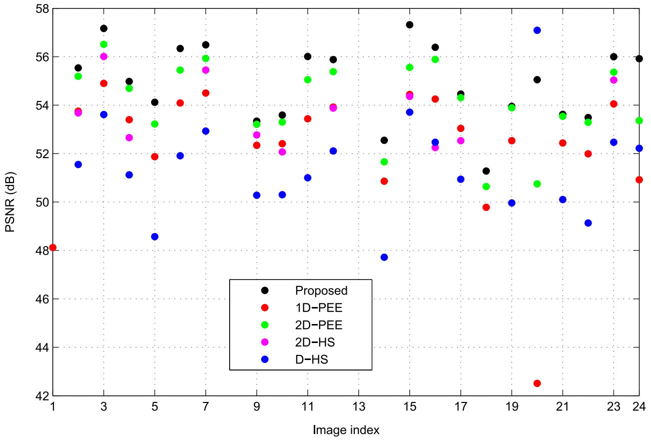


Fig. 12. Performance comparison on Kodak image database for the capacity of 60,000 bits. Note that here, the corresponding results are not plotted if the method cannot provide the capacity.

results are discussed and compared with four state-of-the-art methods [29,13,23,37,40]. For simple descriptions, we denote the compared methods as 2D-HS [13], D-HS [23], 1D-PEE [29], 2D-PEE (i.e., the conventional pairwise PEE) [37], and D-PVO [40], respectively. Here, except the method [29], the maximum modification on a pixel is 1 in the other methods. So the corresponding PSNR is guaranteed over 48.13 dB, but the embedding capacity for an image is limited.

The first experiment is conducted to evaluate the capacity-distortion performance on six standard images with the size of 512×512, including Lena, Baboon, Airplane, Barbara, Elaine and Lake. As shown in Fig. 11, the proposed method can uniformly outperform 1D-PEE, 2D-HS and D-HS. This is mainly due to that a more efficient 2D mapping (i.e., the mapping shown in Fig. 3 (b)) is employed in our method. Compared with 2D-PEE, although the same 2D mapping is adopted, our method can yield a sightly higher PSNR in most cases, and the improvement increases with the capacity. The average gains in PSNR for the capacities of 10,000, 20,000, 30,000 and 40,000 bits are 0.27, 0.38, 0.47 and 0.50 dB, respectively. Compared with the recent pixel-valuing-ordering (PVO) method [40], which is advantageous in dealing the low capacity embedding, our method yields a competitive performance at low capacities for most images. For a larger capacity, our method is superior than theirs because the advantage of PVO embedding is diminishing as the employed block goes larger. The detailed comparison between the proposed method and 2D-PEE is given in Table 2. The performance improvement varies for images because it depends on the derived pair sequence, whose distribution can be measured by the entropy. According to Tables 1 and 2, it is found that the performance gain is increased by the reduction of entropy. For example, for Barbara image, the entropies of pair sequences using ours and the conventional pixel combination are 6.572 and 6.679. The entropy reduction is equal to 0.107, and the average PSNR gain for different capacities is 0.49 dB. For Baboon and Airplane images, the entropy reductions are 0.022 and 0.036, and the average PSNR gains are decreased to 0.26 and 0.33 dB, respectively. It is clearly that the more reduction of entropy, the larger PSNR gain can be expected on average. This can be explained as that in a rough region, the similar pixels usually do not locate nearby and the direct combination in a fixed manner cannot fully utilize the self-similarity in the neighborhood. In this case, searching the candidate partner adaptively by considering the spatial and intensity correlations would be an advisable choice, and therefore can provide a better performance. But the improvement is highly dependent on the image content.

For a more comprehensive comparison, we evaluate the proposed method on Kodak image database, which consists of 24 color images with the size of 512×768 or 768×512. All the color images are transformed into the gray-scale version. Tables 3–5 list the PSNR values of the proposed and the four compared methods for the capacities of 10,000, 20,000 and 40,000 bits, respectively. Compared with 2D-PEE, the average gains using the proposed method are 0.31, 0.44 and 0.79 dB, respectively. It is observed that the performance of the proposed method is comparable to 2D-PEE for 10,000 and 20,000 bits, but yields a higher PSNR on average for 40,000 bits. This observation suggests the feasibility of obtaining a larger gain by using the proposed method for larger capacities. To illustrate the superiority of the proposed method, Fig. 12 gives the corresponding PSNR values of the proposed method, 1D-PEE and 2D-PEE for the case of 60,000 bits on Kodak image database. Compared with 2D-PEE, the average gain in PSNR is increased to 0.83 dB. It is noted that only 21 images can be embedded with a capacity of 60,000 bits by the proposed method and 2D-PEE. For some texture images, our method can yield a even higher PSNR than the others, especially when the image consists of repeated and similar structures.

To further validate the adaptivity of the proposed method for various 2D mappings, we apply another three mappings (as shown in Fig. 13) utilized in the method [37]. The related experimental results on Lena are given in Fig. 14. It is found that a higher PSNR value can also be obtained using the proposed method for the three mappings. It demonstrates that the proposed pixel combination and adaptive 2D mapping determination are helpful for the performance enhancement of different 2D mappings.

The main computation burden is in 10, where the shortest path problem is solved by using Dijkstra's algorithm for each pixel. Because Dijkstra's algorithm is performed in windows of $(2r+1)\times(2r+1)/2$ pixels. The time complexity of our method is roughly given by $O(N\cdot r^2)$, where N is the number of pixels and r is the radius of window size. Whereas, the time complexity of both 1D-PEE and 2D-PEE is about $O(N)$. The comparison of time complexity for once embedding is given

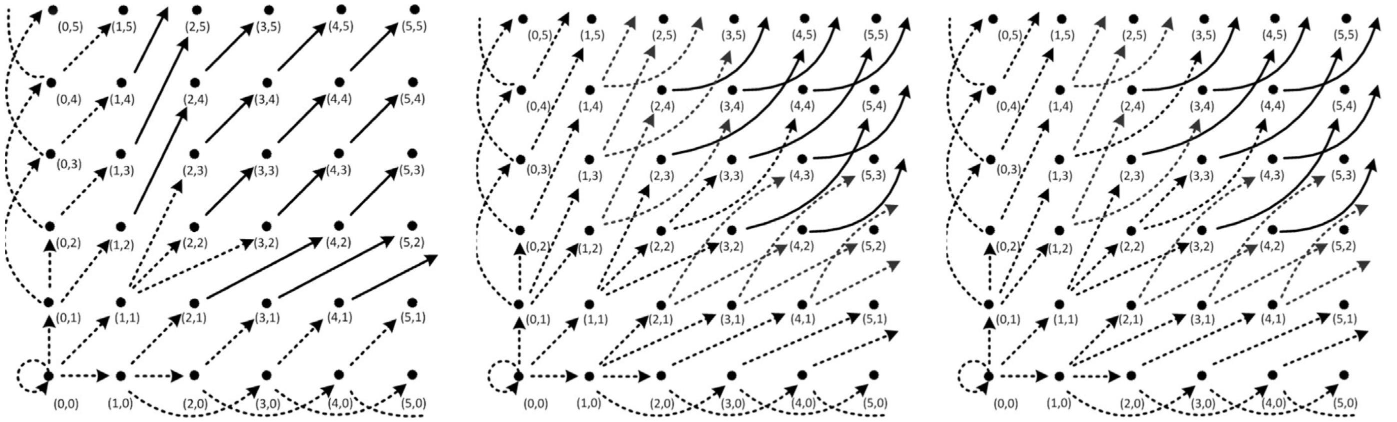


Fig. 13. Another three 2D mappings proposed in the method [37], denoted as M1, M2 and M3 from left to right.

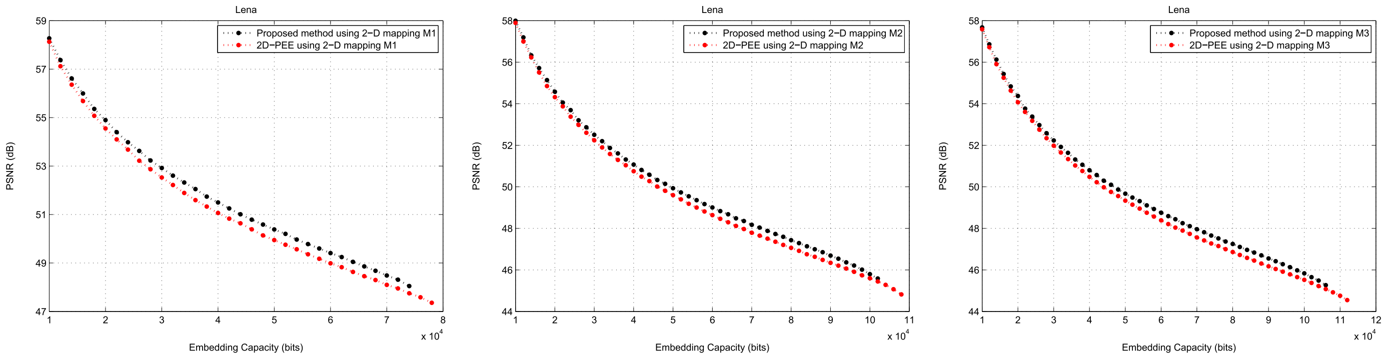


Fig. 14. Performance comparisons between the proposed method and 2D-PEE by using another three 2D mappings (from left to right: M1, M2 and M3) employed in the method [37].

Table 6
Comparison of average runtime for once embedding, where the unit is second and the test is conducted on on the six standard images.

Methods	1D-PEE	2D-PEE	D-HS	Proposed		
				r=1	*r*=2	*r*=3
Runtime	1.08	4.75	52.37	23.41	30.48	70.90

in Table 6, where the runtime is computed as the average on the six standard images, and the time consumed with varying window size is given for the proposed method. It is seen that our method consumes more time in general, and the time consumption is increased for a larger window size. This is because that as the increase of window size, the candidates for pixel combination is increased, leading to more similarity computation as well as the complexity of Dijkstra's algorithm. As we consider the adaptation of window size in the range from 1 to 3, for a standard image, the overall runtime equals to the sum of time spent by using three sizes, and is thus nearly 2 min on average. Note that all experiments are implemented by Matlab, and run on a personal PC. Based on our experiments, we also find that the runtime of ours for different images fluctuates around the average, and thus depends little on the image content. Besides, choosing a window larger than 7×7 does not obtain a significant gain. Hence, to offer a reasonable compromise between computational complexity and performance enhancement, it is suggested to restrict the window size as $1 \le r \le 3$.

5. Conclusions

In this paper, we propose an improved pairwise PEE-based approach to incorporate the geodesic path for accurate pixel combination. It exploits both the spatial and intensity information within a local neighborhood to construct a more correlated pixel pair sequence, and addresses the question of how to determine the pixels into pairs properly for pairwise PEE. Since the proposed pixel pairing aims to adaptively combine the pixels into pairs, the distinct contribution here is the new two-dimensional (2D) histogram generation strategy that is benefited and general enough for the existing 2D RDH method. To the best of our knowledge, few existing work focuses on the topic regarding how the construction of the pair sequence influences the embedding performance for a 2D mapping. Experimental results show that the proposed method can outperform the state-of-the-art baselines [29,13,23,37]. In the future, designing a pixel combination manner suitable for a specific 2D mapping would be an interesting topic.

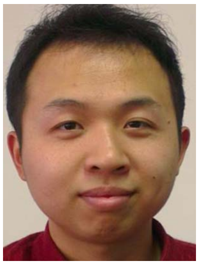
Acknowledgement

This work is supported by the National Science Foundation of China (Nos. 61502160, 61572052, 61272421, 61370225 and 61572182), the PAPD fund, and the CICAEET fund.

References

[1] Y.Q. Shi, Z. Ni, D. Zou, C. Liang, G. Xuan, Lossless data hiding: fundamentals, algorithms and applications, in: Proceedings IEEE ISCAS, vol. 2, 2004, pp. 33–36.
[2] J. Fridrich, M. Goljan, R. Du, Lossless data embedding – new paradigm in digital watermarking, EURASIP J. Appl. Signal Process. 2002 (2) (2002) 185–196.
[3] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, IEEE Trans. Image Process 14 (2) (2005) 253–266.
[4] 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 Tech. 16 (10) (2006) 1301–1308.
[5] 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.
[6] X. Zhang, Reversible data hiding with optimal value transfer, IEEE Trans. Multimed. 15 (2) (2013) 316–325.
[7] Z. Qian, X. Zhang, S. Wang, Reversible data hiding in encrypted jpeg bitstream, IEEE Trans. Multimed. 16 (5) (2014) 1486–1491.
[8] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 890–896.
[9] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process 13 (8) (2004) 1147–1156.
[10] L. Kamstra, H.J.A.M. Heijmans, Reversible data embedding into images using wavelet techniques and sorting, IEEE Trans. Image Process 14 (12) (2005) 2082–2090.
[11] W.L. Tai, C.M. Yeh, C.C. Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Trans. Circuits Syst. Video Technol. 19 (6) (2009) 906–910.
[12] Y. Hu, H.K. Lee, J. Li, DE-based reversible data hiding with improved overflow location map, IEEE Trans. Circuits Syst. Video Technol. 19 (2) (2009) 250–260.
[13] 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.
[14] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.
[15] 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.
[16] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, Inf. Forens. Secur. 5 (1) (2010) 187–193.
[17] Y.-C. Li, C.-M. Yeh, C.-C. Chang, Data hiding based on the similarity between neighboring pixels with reversibility, Digit. Signal Process. 20 (4) (2010) 1116–1128.
[18] H.-T. Wu, J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, Signal Process. 92 (12) (2012) 3000–3009.
[19] L. An, X. Gao, Y. Yuan, D. Tao, Robust lossless data hiding using clustering and statistical quantity histogram, Neurocomputing 77 (1) (2012) 1–11.
[20] L. An, X. Gao, Y. Yuan, D. Tao, C. Deng, F. Ji, Content-adaptive reliable robust lossless data embedding, Neurocomputing 79 (2012) 1–11.
[21] Y.-Y. Tsai, D.-S. Tsai, C.-L. Liu, Reversible data hiding scheme based on neighboring pixel differences, Digit. Signal Process. 23 (3) (2013) 919–927.
[22] 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.
[23] W. Hong, T.-S. Chen, J. Chen, Reversible data hiding using delaunay triangulation and selective embedment, Inf. Sci. 308 (0) (2015) 140–154.
[24] X. Hu, W. Zhang, X. Li, N. Yu, Minimum rate prediction and optimized histograms modification for reversible data hiding, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 653–664.
[25] H. Wu, J. Dugelay, Y. Shi, Reversible image data hiding with contrast enhancement, IEEE Signal Process. Lett. 22 (1) (2015) 81–85.
[26] 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.
[27] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process 16 (3) (2007) 721–730.
[28] W. Hong, T.S. Chen, C.W. Shiu, Reversible data hiding for high quality images using modification of prediction errors, J. Syst. Softw. 82 (11) (2009) 1833–1842.
[29] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking

algorithm using sorting and prediction, IEEE Trans. Circuits Syst. Video Technol. 19 (7) (2009) 989–999.
[30] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, IEEE Trans. Circuits Syst. Video Technol. 21 (8) (2011) 1061–1070.
[31] 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.
[32] G. Feng, Z. Qian, N. Dai, Reversible watermarking via extreme learning machine prediction, Neurocomputing 82 (2012) 62–68.
[33] B. Ou, Y. Zhao, R. Ni, Reversible watermarking using optional prediction error histogram modification, Neurocomputing 93 (2012) 67–76.
[34] B. Ou, X. Li, Y. Zhao, R. Ni, Reversible data hiding based on pde predictor, J. Syst. Softw. 86 (10) (2013) 2700–2709.
[35] 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.
[36] C. Qin, C.-C. Chang, Y.-H. Huang, L.-T. Liao, An inpainting-assisted reversible steganographic scheme using histogram shifting mechanism, IEEE Trans. Circuits Syst. Video Tech. 23 (7) (2013) 1109–1118.
[37] 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.
[38] 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.
[39] B. Ou, X. Li, Y. Zhao, R. Ni, Efficient color image reversible data hiding based on channel-dependent payload partition and adaptive embedding, Signal Process. 108 (0) (2015) 642–657.
[40] 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.
[41] I.-C. Dragoi, D. Coltuc, On local prediction based reversible watermarking, IEEE Trans. Image Process. 24 (4) (2015) 1244–1246.
[42] Y. Liu, L. Ju, M. Hu, X. Ma, H. Zhao, A robust reversible data hiding scheme for h.264 without distortion drift, Neurocomputing 151 (Part 3) (2015) 1053–1062.
[43] Y. Liu, L. Chen, M. Hu, Z. Jia, S. Jia, H. Zhao, A reversible data hiding method for h.264 with shamir's (t, n)-threshold secret sharing, Neurocomputing 188 (2016) 63–70.
[44] D. Coltuc, J.M. Chassery, Very fast watermarking by reversible contrast mapping, IEEE Signal Process. Lett. 14 (4) (2007) 255–258.
[45] S. Weng, Y. Zhao, J.S. Pan, R. Ni, Reversible watermarking based on invariability and adjustment on pixel pairs, IEEE Signal Process. Lett. 15 (2008) 721–724.
[46] X. Wang, X. Li, B. Yang, Z. Guo, Efficient generalized integer transform for reversible watermarking, IEEE Signal Process. Lett. 17 (6) (2010) 567–570.
[47] D. Coltuc, Low distortion transform for reversible watermarking, IEEE Trans. Image Process 21 (1) (2012) 412–417.
[48] F. Peng, X. Li, B. Yang, Adaptive reversible data hiding scheme based on integer transform, Signal Process. 92 (1) (2012) 54–62.
[49] C.-C. Chang, T.S. Nguyen, C.-C. Lin, A reversible compression code hiding using soc and smvq indices, Inf. Sci. 300 (0) (2015) 85–99.
[50] S.K. Lee, Y.H. Suh, Y.S. Ho, Reversible image authentication based on watermarking, in: Proceedings of the IEEE ICME, 2006, pp. 1321–1324.
[51] D. Coltuc, Improved embedding for prediction-based reversible watermarking, IEEE Trans. Inf. Forens. Secur. 6 (3) (2011) 873–882.
[52] W. Hong, Adaptive reversible data hiding method based on error energy control and histogram shifting, Opt. Commun. 285 (2) (2012) 101–108.
[53] 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.
[54] F. Peng, X. Li, B. Yang, Improved pvo-based reversible data hiding, Digit. Signal Process. 25 (0) (2014) 255–265.
[55] 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.
[56] 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.
[57] B. Ou, X. Li, J. Wang, Improved pvo-based reversible data hiding: a new implementation based on multiple histograms modification, J. Vis. Commun. Image Represent. 38 (2016) 328–339.
[58] L. Yatziv, G. Sapiro, Fast image and video colorization using chrominance blending, IEEE Trans. Image Process. 15 (5) (2006) 1120–1129.
[59] E. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271.



Bo Ou received the B.S. and Ph.D. degrees in Beijing Jiaotong University, Beijing, China, in 2008 and 2014, respectively. Since 2014, he has been on the faculty of College of Computer Science and Electronic Engineering, Hunan University, where he is currently an Associate Professor. His research interests include image processing and data hiding.

Xiaolong Li received the B.S. degree from Peking University, Beijing, China, the M.S. degree from Ecole Polytechnique, Palaiseau, France, and the Ph.D. degree in mathematics from ENS de Cachan, Cachan, France, in 1999, 2002, and 2006, respectively. Before joining Peking University as a researcher, he worked as a postdoctoral fellow at Peking University in 2007–2009. His research interests are image processing and information hiding.

Jinwei Wang was born in Inner Mongolia, China, in 1978. He received the B.A.Sc. in automatic control from Inner Mongolia University of Technology in 2000. He received the Ph.D. student in information security at Nanjing University of Science & Technology in 2007 and was a visiting scholar in Service Anticipation Multimedia Innovation (SAMI) Lab of France Telecom R & D Center (Beijing) in 2006. He worked as a senior engineer at the 28th research institute, CETC from 2007 to 2010. He worked as a visiting scholar at New Jersey Institute of Technology, NJ, USA from 2014 to 2015. Now he works as a professor at Nanjing University of Information Science & Technology. His research interests *Biography of the author(s) Click here to download Biography of the author(s): biboraphy.pdf include multimedia copyright protection, multimedia forensics, multimedia encryption and data authentication. He has published more than 40 papers, hosted and participated in more than 10 projects.

Fei Peng received the Ph.D. degree in Circuits and Systems from the South China University of Science and Technology, Guangzhou, China, in 2006. He was a visiting fellow of the Department of Computer Science, University of Warwick, U.K. in 2009–2010.

Currently, he is a professor in the School of Computer Science and Electronic Engineering, Hunan University, Changsha. He is also the Director of the Department of Cyber Security. His areas of interest include digital watermarking and digital forensics.