A Survey on Reversible Data Hiding for Uncompressed Images

CHENG ZHANG and BO OU, Hunan University, Changsha, China
FEI PENG, Guangzhou University, Guangzhou, China
YAO ZHAO, Beijing Jiaotong University, Beijing, China
KEQIN LI, State University of New York, New Paltz, USA

Reversible data hiding (RDH) has developed various theories and algorithms since the early 1990s. The existing works involve a large amount of specialized knowledge, making it difficult for researchers, especially primary learners, to have a good grounding in the basic ideas. In this survey, we will review the mainstream RDH algorithms in uncompressed images and analyze their unique features to provide readers with an introduction to basic topics in RDH. We analyze the most effective RDH frameworks and their common extensions. The classic techniques, including lossless compression-based RDH, difference expansion, integer transform, histogram shifting, prediction-error expansion (PEE), and their extensions, will be reviewed first. Then, three currently popular investigated schemes, i.e., multiple histograms modification, pairwise PEE, and pixel-value-ordering, are presented in detail. Four aspects of these mainstream techniques are reviewed and analyzed, including the evolution of embedding frameworks, detailed technological features, extensions, and the current state of the art. Furthermore, we look forward the possible future research based on early-age motivations.

CCS Concepts: • **Security and privacy** → **Digital rights management**; **Authentication;**

Additional Key Words and Phrases: Reversible data hiding, histogram shifting, prediction-error expansion, multiple histograms modification, pairwise prediction-error expansion, pixel-value-ordering

ACM Reference Format:
Cheng Zhang, Bo Ou, Fei Peng, Yao Zhao, and Keqin Li. 2024. A Survey on Reversible Data Hiding for Uncompressed Images. *ACM Comput. Surv.* 56, 7, Article 180 (April 2024), 33 pages. https://doi.org/10.1145/3645105

1 INTRODUCTION

Reversible data hiding (RDH) [91] is considered a valid method for protecting the security of multimedia data. Figure 1 describes the features of RDH. With RDH, one can achieve covert communication using the common channel. As the embedding does not result in obvious quality degradation, it becomes difficult for attackers to detect the marked carrier within the vast amount

This work was supported by the National Science Foundation of China (Grants. No. 61972031, No. U1936212, No. U22A2030, No. 62372128 and No. 92067104) and Natural Science Foundation of Guangdong Province under (Grant. No. 2023A1515011575).

Authors' addresses: C. Zhang and B. Ou (Corresponding author), Hunan University, Chang-sha, China, 410082; e-mails: zcheng@hnu.edu.cn, oubo@hnu.edu.cn; F. Peng, Guangzhou University, Guangzhou, China, 510000; e-mail: eepengf@gmail.com; Y. Zhao, Beijing Jiaotong University, Beijing, China, 100044; e-mail: yzhao@bjtu.edu.cn; K. Li, State University of New York, 1 Hawk Drive, New Paltz, NY, USA, 12561; e-mail: lik@newpaltz.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 0360-0300/2024/04-ART180
https://doi.org/10.1145/3645105

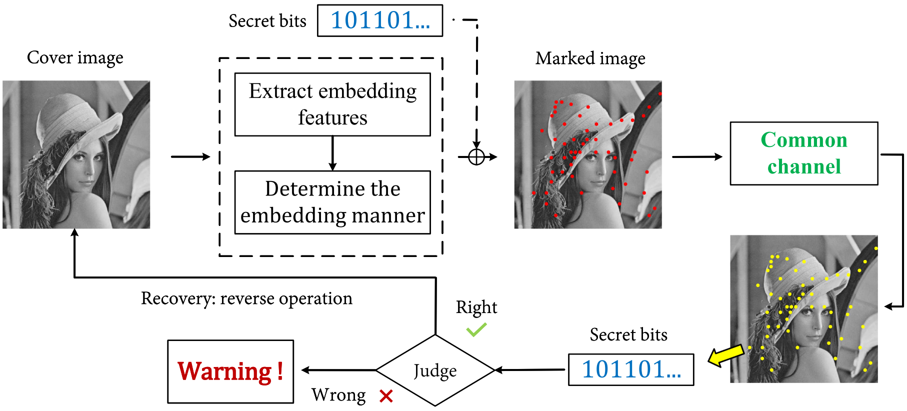


Fig. 1. The presentation for the function and characteristics of RDH.

of data. On the receiver side, the RDH performs two tasks: reading the embedded information and restoring the primary content of the carrier without any loss. The remarkable characteristic of RDH is its ability to recover the carrier perfectly. It means that permanent distortion is avoided. This characteristic makes RDH a promising application in special scenarios that are highly sensitive to data distortion, such as commercial files, judicial evidence, engineering images, and so on.

In existing RDH methods, the digital image is the commonly investigated carrier, due to its convenience in transmission and simplicity in operation. Since there are various storage formats, the related methods could be further classified into several categories, e.g., RDH in uncompressed images [1, 2, 9, 14, 16, 30, 38, 42, 45, 47, 48, 51, 57, 62, 72, 86, 90, 94, 95, 101, 135, 138], RDH in encrypted images [6, 19, 37, 59, 76, 79, 81, 83, 87, 111, 117, 125, 127–129, 136, 137], and RDH in compressed images [13, 20, 25, 31, 32, 80, 85, 107, 113]. Among them, the methods for RDH in uncompressed images have been the hottest topic in the academic community. This is because the information expression of uncompressed images contains a large amount of redundant space, making it easier to design an effective reversible embedding framework. Many techniques usually get validated first in uncompressed images and then are applied to other carriers. The statistical results for the research hotspots are shown in Figure 2 and Table 1. We have reviewed 454 published journal articles in the RDH community to complete the classification. In Figure 2, the number of existing works for four research directions is presented, including RDH in uncompressed images, RDH in encrypted images, RDH in JPEG images, and RDH in videos. Clearly, the RDH in uncompressed images has always been the most popular theme. Table 1 provides a more detailed description, including the citation data, the proportion, and simple explanations. The citation data are obtained from the Google Scholar search. Notice that the denominator used in Table 1 for the proportions is set as 454. Besides, some research, such as the RDH for binary images and halftone images, and so on, are not listed in Table 1. So, the sum of proportions is 97.2%.

For the RDH in uncompressed images, the early algorithms [1, 2, 9, 14, 16, 30, 38, 42, 45, 47, 48, 51, 62, 72, 86, 90, 94, 95, 101, 135, 138] mainly focus on designing an effective way to reversibly modify the pixels. These initial algorithms laid a firm foundation for the current developments. In general, there are five classic embedding frameworks, i.e., the RDH using lossless compression [2, 16, 135, 138], the **difference expansion (DE)** [30, 38, 95], the **integer transform (IT)** [1, 9, 72, 86, 101], the **histogram shifting (HS)** [42, 45, 51, 62], and the **prediction-error expansion (PEE)** [14, 47, 48, 90, 94]. Although these five types of frameworks have different technological features, their core ideas are the same, i.e., exploiting image redundancy to achieve reversibility and data embedding. Later, researchers continued to explore and develop the embedding theory based on

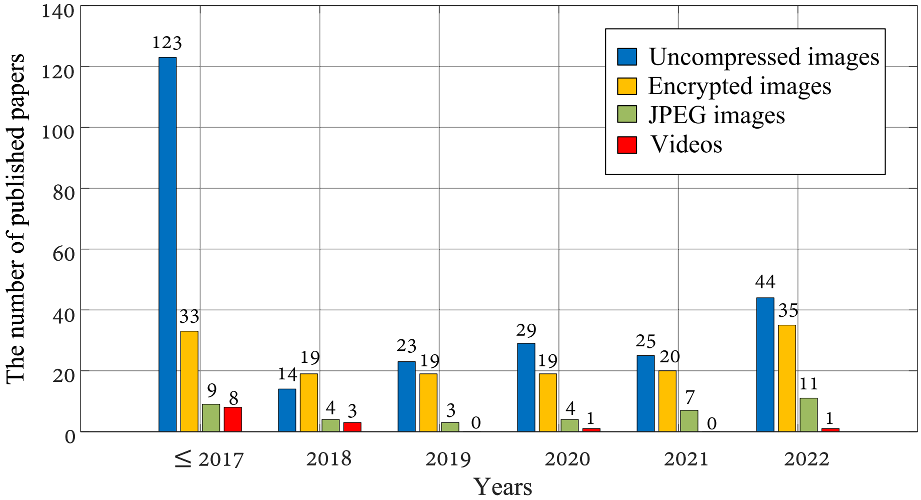


Fig. 2. The statistics for the published RDH works on the four common carriers in different years.

these classic RDH algorithms, aiming for better performance. Generally, the performance of an RDH scheme could be measured using two criteria, i.e., the **embedding capacity (EC)** and the **peak signal-to-noise ratio (PSNR)**. The first criterion is the carrying ability, usually measured by the **embedding rate (ER)**, which denote the average number of bits hidden in a pixel, also known as **bits per pixel (bpp)**. One would expect that the cover image could be embedded with more secret message bits to alleviate the storage burden of database. The second criterion counts the level of **embedding distortion (ED)**. The PSNR is used to measure the similarity between the marked images and the original content based on the difference in pixels. A higher PSNR indicates better quality preservation, meaning that the hidden message is less perceptible. One always wants to design the RDH algorithm that can either achieve a higher PSNR for a given capacity or embed more secret bits under the specific distortion level. To achieve a better tradeoff between the two criteria, data embedding is required to be adaptive according to the image content. For now, various adaptive RDH algorithms have been proposed, including **multiple histograms modification (MHM)** [49], pairwise PEE [68], and **pixel-value-ordering (PVO)** [46]. Each of them is developed with different motivations, to solve the adaptive embedding problem and enrich the theory of embedding.

In practice, the techniques mentioned above usually need to be adjusted according to the specific application and requirements. This adjustment allows the algorithm to have better generalization ability and practicality. Many effective algorithms have been reported, including the RDH in the encrypted domain [6, 19, 37, 59, 76, 79, 81, 83, 87, 111, 117, 125, 127–129, 136, 137], the RDH for JPEG images [13, 20, 25, 31, 32, 80, 85, 107, 113, 126], the RDH for two-dimensional (2D) vector graphics [71, 92], the RDH for 3D mesh models [35, 56, 74], the RDH with contrast enhancement [39, 109, 110, 121, 134], the RDH for color images [3, 4, 26, 33], robust RDH [43, 50, 100, 116], visible reversible watermarking [77, 82, 123], the RDH for videos [54, 93, 118, 119, 124], and the RDH considering the **human visual system (HVS)** [24, 36, 122, 132]. The introductions for these schemes are presented in Table 1. To address the challenges encountered in practice, these works have examined the relevant theories of RDH and its extensions.

In this survey, the RDH algorithms in uncompressed images are reviewed. We will introduce classic techniques and their popular extensions. The rest of the article contains five sections. In Section 2, five classic RDH techniques, i.e., the compression-based method, DE, IT, HS, and PEE,

Table 1. Overview of the Hot Spots in the Field of RDH

Research	Proportion	Maximal citations	Description
Lossless compression-based RDH	1.5%	1,463	The early RDH method for uncompressed images.
DE-based RDH	2.0%	3,637	The fundamental technique for the RDH in the unc-ompressed images.
IT-based RDH	2.0%	3,369	The extension of DE.
HS-based RDH	4.4%	1,703	An effective RDH scheme developed for uncompress-ed images. It also has broad applicability in other c-arriers, e.g., JPEG images and videos.
PEE-based RDH	13.7%	1,594	The popular technique derived from DE and HS.
MHM-based RDH	3.1%	342	The typical adaptive embedding strategy.
Pairwise PEE	3.1%	494	A more effective method of pixel modification deriv-ed from PEE.
PVO-based RDH	8.3%	485	• An effective extension of PEE. • One of the hottest RDH technique.
RDH in encrypted images	30.4%	1,066	• A practical framework considering the environme-nt of cloud storage and data sharing. • Hide the secret message in the encrypted images, to ensure the safety of plaintext.
RDH in JPEG images	8.4%	413	• Conduct the embedding in the transformed coefficients. • A new objective: the file size preservation.
RDH in 2D vector graphics	1.8%	68	Consider the different features of 2D vector graphics, e.g., the floating-point numbers.
RDH in 3D mesh models	1.8%	99	Consider the various structures of 3D mesh models, e.g., the vertex and mesh topology.
RDH with contrast enhancement	4.4%	179	Enhance the contrast of the carrier image after the embedding.
RDH in color images	3.1%	97	• Multiple channels embedding. • Other objective: grayscale-Invariance.
Robust RDH	3.5%	280	Make sure to perform the right extraction after the carrier is attacked, e.g., the JPEG compression, cr-opping, rotation, etc.
Visible reversible watermarking	1.1%	149	To achieve the specific objectives, the concealed in-formation is no longer imperceptive, i.e., it can now be observed from the marked carrier.
RDH in videos	3.3%	94	• Embed the message into the frames of the video. • Realize other tasks, such as the intra-frame error concealment.
RDH considering the HVS	1.3%	124	Reduce the quality distortion captured by human ey-es, rather than just controlling the PSNR.

are introduced. Then, three major extensions and their developments, including MHM, pairwise PEE, and PVO, will be described in Section 3. The embedding frameworks are compared, and the state-of-the-art RDH algorithms are analyzed in Section 4. The potential future topics of RDH are discussed in Section 5. Finally, the main conclusions are drawn in Section 6.

Table 2. Overview of the Classic RDH Techniques in Uncompressed Images

Technique	Description	Expectable ER	ED	Portability	Complexity	Reference
Lossless compression	Compress the image to obtain room and then embed the bits.	1.5 bpp	High	✓	Low	[2, 16, 135, 138]
DE	Use difference expan-sion for data embed-ding.	0.9 bpp	Medium	✓	Low	[30, 38, 95]
IT	Apply the integer tra-nsform for data emb-edding.	2.17 bpp	Medium	✓	Low	[1, 9, 72, 101]
HS	Embed the bits accor-ding to the histogram.	1 bpp	Low	✓	Low	[17, 42, 62]
PEE	Modify the prediction-error histogram for da-ta embedding.	1.8 bpp	Low	✓	Low	[34, 47, 90, 94]

2 CLASSIC RDH TECHNIQUES IN UNCOMPRESSED IMAGES

In general, there are two stages in the development of RDH for uncompressed images. In the early stage, researchers focused on designing different RDH frameworks and formed five classic schemes, namely, lossless compression-based RDH, DE-based RDH, IT-based RDH, HS-based RDH, and PEE-based RDH. The second stage involves working on the theory and exploring more effective algorithms. It aims to address the challenge of redundant utilization to improve performance. The effective algorithms in this stage, including MHM, pairwise PEE, and PVO, are mainly developed within the classical PEE framework. In this section, we will focus on the classic RDH frameworks developed in the first stage and briefly review their extensions. A simple illustration is presented in Table 2.

2.1 Compression-based RDH Scheme

At the beginning, RDH methods [2, 16, 135, 138] were mainly designed based on lossless compression. The partial image content is losslessly compressed into a compact representation with smaller size, so that the saved space can be used for data embedding. The image can be recovered, because compression is a lossless process. Accordingly, the embedding performance is closely related to the compression method. One task is to improve the tradeoff between compression efficiency and performance. Celik et al. [2] proposed the conventional **least significant bit–(LSB)** based RDH method. In this method, the lowest level of the cover pixels is determined through quantization and then used for embedding. In Reference [135], the decompression and compression processes of an entropy coder are used recursively to embed the information. Zhang [138] proposed to divide the cover pixels into several subsets and conducted the modification based on the optimal value transfer rule. In this methods, reversible embedding is successfully conducted using various compression algorithms. The compression-based RDH scheme has been fruitful. However, one drawback is that it is difficult to keep the image semantic when attempting to embed additional secret bits into it. Obvious visual distortion is usually found at high capacities.

2.2 DE-based RDH Scheme

The DE-based RDH utilizes the Haar transformation to modify every two adjacent pixels for data embedding. The similarity between the two pixels are utilized to realize the reversible data embedding and extraction. For a pixel pair (p_1, p_2), the marked pixels $(\widetilde{p}_1, \widetilde{p}_2)$ are determined as $\widetilde{p}_1 = 2p_1 - \lceil \frac{1}{2}(p_1 + p_2) \rceil$ and $\widetilde{p}_2 = 2p_2 - \lceil \frac{1}{2}(p_1 + p_2) \rceil + m$ respectively, where m

represents the binary message bit. That is, one bit is embedded by expanding the difference between the two pixels to $2(p_2 - p_1) + m$. For the receiver, the embedded bit can be determined as the LSB of the expanded difference, and the cover pixels are recovered using the inverse transformation.

In DE, the mean of two pixels before and after embedding are unchanged, and the ER is up to 0.5 bpp in one layer embedding. As a fundamental technique of RDH, many extensions are derived from it. However, in the initial stage, the DE method seldom considers the discriminate processing for the pixel pairs during embedding. So, when pursuing a higher ER, the multi-layer embedding can easily lead to excessive distortion. In addition, DE-based methods typically need to employ the location map to record the unavailable pixels and then losslessly compress it to help reduce the consumption of embedding capacity.

2.3 IT-based RDH Scheme

The core idea of IT-based RDH is to embed multiple bits by simultaneously modifying a sequence of pixels. During the embedding, a pixel sequence $(p_1, p_2, \ldots, p_n)$ can be modified as a unit to embed $n - 1$ bits. In this sense, we can view DE as a special case of IT with $n = 2$. In Reference [9], the marked pixels $(\widetilde{p}_1, \widetilde{p}_2, \ldots, \widetilde{p}_n)$ are determined as

$$\widetilde{p}_i = \begin{cases} 2p_i - \left\lfloor 2\overline{p} - \frac{1}{n}\sum_{j=1}^{n-1} m_j \right\rfloor + \lfloor \overline{p} \rfloor, & \text{if } i = 1 \\ \\ 2p_i - \left\lfloor 2\overline{p} - \frac{1}{n}\sum_{j=1}^{n-1} m_j \right\rfloor + \lfloor \overline{p} \rfloor + m_{i-1}, & \text{if } i \geq 2 \end{cases}, \tag{1}$$

where $\overline{p} = \frac{1}{n}\sum_{t=1}^{n} p_t$ and $m_j \in \{0, 1\}$ represents the message bit, $j \in \{1, 2, \ldots, n-1\}$, $i \in \{1, 2, \ldots, n\}$. Clearly, a IT-based RDH method could achieve a higher capacity compared with DE-based embedding when $n > 2$. The maximum ER is nearly 1 bpp for a large n. In the later works [72, 86, 101], the performance in terms of ER and PSNR is further improved. Besides, considering multiple pixels together makes the integer average more stable, which allows it to withstand perturbations.

2.4 HS-based RDH Scheme

The HS-based RDH scheme was first proposed by Ni et al. [62]. This scheme modifies the pixels based on the gray-scale histogram. In HS, the pixels are categorized based on their gray values, and the highest bins in the histogram are chosen as the expansion bins to embed the secret bits. The pixels from different categories will be modified in a reversible manner without causing any ambiguity. Generally, pixels can be classified into three categories: unchanged pixels, expansion pixels, and shifting pixels. During the embedding process, they perform different tasks. In general, only the expansion pixels and the shifting pixels are modified. When embedding, the expansion pixels will either be kept unchanged or have "±1" added to them based on the message bits. Meanwhile, the shifting pixels will either have "1" or "−1" added to them to guarantee the reversibility. More specifically, for the cover pixel sequence $(p_1, p_2, \ldots, p_N)$, the marked pixels $(\widetilde{p}_1, \widetilde{p}_2, \ldots, \widetilde{p}_N)$ are determined as

$$\widetilde{p}_i = \begin{cases} p_i + 1, & \text{if } p_i > b \\ p_i + m, & \text{if } p_i = b \\ p_i, & \text{if } a < p_i < b \\ p_i - m, & \text{if } p_i = a \\ p_i - 1, & \text{if } p_i < a \end{cases}, \tag{2}$$

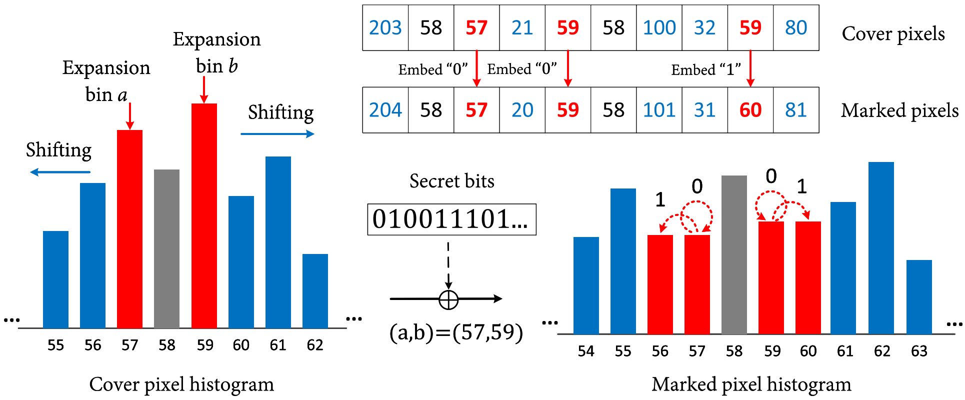


Fig. 3. An example of the pixels modification and histogram shifting in HS-based RDH.

where $0 < (a, b) < 255$ are the two expansion bins, m represents the message bit, and N counts the number of pixels.

To realize the reversible embedding, the parameters (a, b) must be stored as side information. In general, the gray values corresponding to the largest categories are selected as the expansion bins, i.e., the highest peaks of the pixel histogram. The pixels larger than b or smaller than a are considered as shifting pixels, while the pixels with values of a and b are referred to expansion pixels. Based on this rule, the distribution of the pixel value histogram will change in a predictable manner after the embedding. For pixels with values larger than b, their histogram bins will be shifted rightwards. The histogram bins for the pixels with values of b will be expanded to embed one additional bit. For the pixels with values equal to or smaller than a, the histogram bins are modified toward the left. An example of HS-based modification is shown in Figure 3. Here, the expansion bins are selected as $(a, b) = (57, 59)$. For the three expansion pixels $(57, 59, 59)$, they are modified according to the secret bits $(0, 0, 1)$, and the marked pixels are $(57, 59, 60)$ due to the HS modification rule. The pixels marked in blue are the shifting pixels. They are added by 1 or -1 without embedding any bits. The predictable modification method of HS allows the receiver to identify the hidden bits and complete the restoration in an inverse manner, i.e., determine the cover pixels as

$$p_i = \begin{cases} \widetilde{p_i} - 1, & \text{if } \widetilde{p_i} > b \\ \widetilde{p_i}, & \text{if } a \le \widetilde{p_i} \le b \\ \widetilde{p_i} + 1, & \text{if } \widetilde{p_i} < a \end{cases}. \tag{3}$$

The embedded bits are extracted based on the category of the marked pixel, i.e., $m = 1$ if $\widetilde{p_i} \in \{a - 1, b + 1\}$, or $m = 0$ if $\widetilde{p_i} \in \{a, b\}$.

Compared to DE-based and IT-based methods, the embedding method of HS is simple and intuitive. Its modification processes do not involve complex mathematical formulas, and the performance can be directly estimated based on the simulation of HS before the embedding. This technique also has strong portability. Generally speaking, the embedding design of HS is also effective for the statistical features generated from other coefficients, e.g., the histogram of the quantified discrete cosine transform coefficients [32, 113] and the histogram of the deep convolutional network parameters [17], and so on.

2.5 PEE-based RDH Scheme

In PEE, the prediction is introduced for RDH and the embedding performance is further enhanced due to the better ability of correlation exploitation. Based on the cover pixels and its

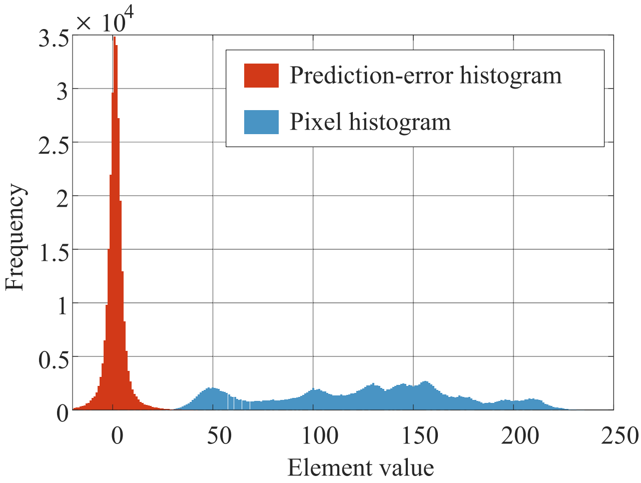


Fig. 4. The comparison of PEH and the pixel histogram for the test image Lena. Here the prediction is calculated by using the rhombus predictor.

neighboring pixels, the **prediction-errors (PE)** are calculated sequentially, and the **prediction-error histogram (PEH)** can be generated. The PEH is defined as $h(e) = \#\{1 \leq i \leq N : e_i = e\}$, where the function $\#\{\cdot\}$ counts the number of pixels collected in the given sequence. Here $e_i = p_i - p_i^*$ is the PE of pixel and its prediction p_i^*.

In PEE, the pixels will be classified by the PEs instead of the gray values. The generated PEH approximates the Laplacian distribution. Compared to the pixel histogram, the distribution of PEH is sharper and more concentrated. This means that the expression of image information becomes more compact, allowing for more space for the data embedding. Figure 4 compares the histogram generated from PEs and pixels. It can be seen that PEH is concentrated at zero, and its bins have much higher values. For the embedding, the PEE-based scheme applies the similar manner with HS. One difference is that PEE determines the expansion pixels according to the values of PEs. That is, the marked pixels $(\widetilde{p}_1, \widetilde{p}_2, \ldots, \widetilde{p}_N)$ are obtained as

$$\widetilde{p}_i = \begin{cases} p_i + 1, & \text{if } e_i > b \\ p_i + m, & \text{if } e_i = b \\ p_i, & \text{if } a < e_i < b \\ p_i - m, & \text{if } e_i = a \\ p_i - 1, & \text{if } e_i < a \end{cases}, \tag{4}$$

where $-255 < a < 0 \leq b < 255$ are the two expansion bins that are selected adaptively according to the distribution of PEH and the given capacity. Notice that the expansion bins in PEE are redefined as the values of PEs rather than the pixel values. Similarly to HS, the histogram bins of PEH will be modified in a predictable manner during the embedding. The framework of PEE is shown in Figure 5. Compared with that shown in Figure 3, the notable distinction lies in that the expansion pixels, shifting pixels, and unchanged pixels are classified according to the PEs.

In PEE, the prediction for a pixel should be kept unchanged before and after embedding, for ensuring the reversibility. The receiver should obtain the same predictions as those of sender. A common solution is to make sure that the same context used for prediction can be retrieved after the embedding. In the popular predictors [8, 14, 90, 94], the local context could be the same for the receiver by scanning and processing the marked pixels in the inverse order. In this way, the PEs

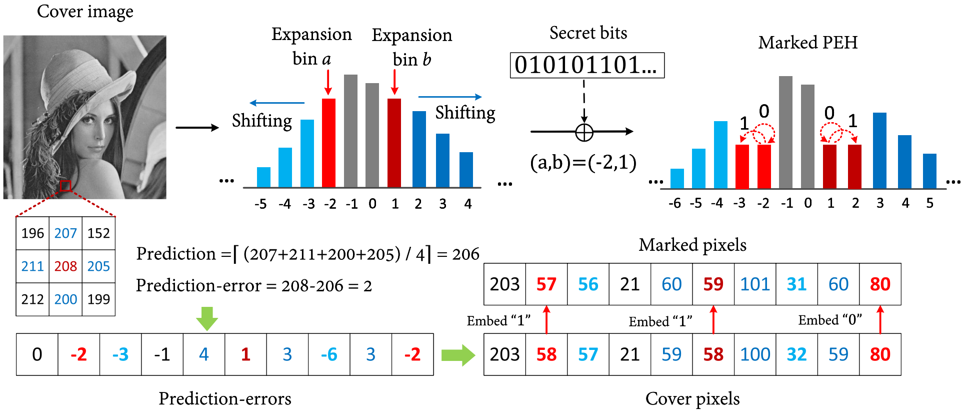


Fig. 5. An example of the pixels modification and histogram shifting in PEE-based RDH.

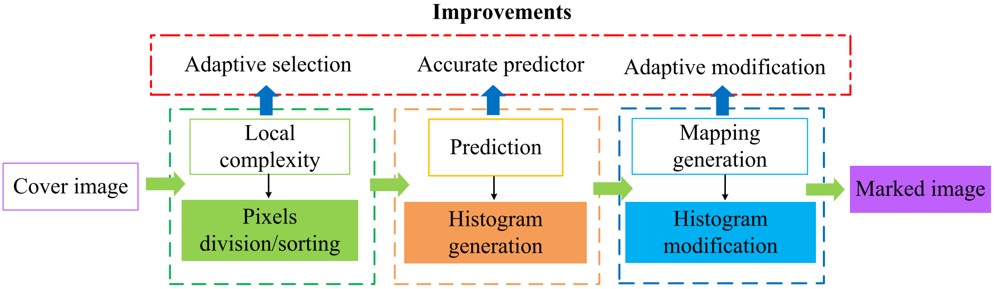


Fig. 6. The embedding modules of PEE and the corresponding improvements used in the extensions.

calculated from the marked image could be used to precisely classify the marked pixels. There are following three cases, i.e., for the marked pixel $\widetilde{p}_i$, its PE $\widetilde{e}_i = \widetilde{p}_i - p_i^*$, and the parameters (a, b) obtained from the side information,

- The cover pixels is recovered as $\widetilde{p}_i - 1$, when its marked PE is larger than b.
- The cover pixels is recovered as $\widetilde{p}_i$ itself, when its marked PE is not larger than b and not smaller than a.
- The cover pixels is recovered as $\widetilde{p}_i + 1$, when its marked PE is smaller than a.

Correspondingly, for the embedded bits, they will be extracted as "1" if $\widetilde{e}_i \in \{a - 1, b + 1\}$ and extracted as "0" if $\widetilde{e}_i \in \{a, b\}$. PEE inherits both the advantages of DE and HS. It has an acceptable computational complexity and higher embedding efficiency. Its series of embedding modules provides researchers with ample opportunities for investigation and improvement. The flow of PEE and the corresponding improvements are illustrated in Figure 6.

2.6 Extensions of PEE

Among the PEE-based methods, there are mainly two kinds of improvements, i.e., the accurate prediction [8, 10, 11, 14, 28, 29, 55, 58, 66, 90, 102, 139, 140] and the adaptive embedding strategy [34, 47, 48, 89, 90].

The first kind of improvement is to enhance the accuracy of the predictor, so that the generated PEH can be more sharply distributed. For the PEE-based framework, a more accurate prediction means that the obtained statistical features contain a higher number of embeddable pixels. It

Table 3. The Comparison for Parts of the Existing Predictors

Predictor	Description	Mean	Variance	MSE
MED [94]	Median-edge-detector (MED) is a commonly used model. It completes the prediction based on the correlation between the target pixel and its three neighboring pixels.	6.34	125.13	182.86
DP [95]	The prediction derived from the DE-based method. Here, the target pixel is predicted by its adjacent pixel.	4.38	152.68	178.07
GAP [8, 14]	One of the popular predictors, i.e., gradient-adjusted-predictor (GAP). Compared with MED, it utilizes the more complex factors for prediction. The number of the used neighboring pixels are increased to seven. The local gradients are calculated to adaptively adjust the prediction.	8.39	262.76	355.56
RP [90]	The commonly used predictor, namely rhombus predictor (RP). It offers an accurate prediction based on the average of the four neighbors.	5.37	79.10	119.17
CNNP-1 [28]	The CNN-based predictor proposed by Hu and Xiang. It builds the training sets by marking the image content as two nonoverlapping sets. The pixels from one set are all inputted into the model to predict the pixels from another set.	4.16	53.52	81.05
CNNP-2 [29]	The improved predictor of CNNP-1. It divides the target image into four parts. The prediction is conducted according to the global context.	2.80	27.72	41.30

contributes to cutting down the degradation caused by unnecessary shifting and improves the efficiency of embedding. In terms of information entropy, an accurate predictor can generate a more concentrated histogram. This means that the original content is compressed into an expression with lower entropy. It could offset the entropy increase caused by the embedding. Collecting the appropriate neighboring pixels [10, 140] and developing a well-defined formulation [8, 11, 14, 28, 58, 66, 115] are effective approaches to improve prediction accuracy. Ou et al. [66] utilized the partial differential equation and the pixel gradient to learn more suitable weights for the neighboring pixels. Dragoi et al. [10, 11] provided the target image with multiple prediction models and employed the least squares method to obtain a fitted solution. In Reference [102], ridge regression is introduced to ensure that the trained weights have greater generalization ability. As the price, it takes more time to search for the appropriate coefficients. Recently, the **convolutional neural network (CNN)** has become a powerful tool for establishing predictors. Luo et al. [55] first pointed out that the non-linear mapping relationship between the target pixel and its neighborhood could be captured by using CNN. They trained a model for predicting stereo images and demonstrated its outstanding performance. Soon, Hu and Xiang [28, 29] developed a CNN-based predictor for grayscale images. They discussed the division of pixels, the training method, and other details in depth. Zhang et al. [139] and Yang et al. [120] proposed improved CNN-based predictors, respectively, using different network designs. Table 3 lists the comparison results for parts of the existing predictors. Here, the average, the variance, and the **mean square error (MSE)** are presented. The first two results are calculated based on the absolute values of the obtained PEs, and MSE is obtained according to the target pixels and the prediction values. The experimental data, which come from Reference [29], are calculated on the 500 test images.

In addition, the asymmetric PEH modification [7, 40, 41] is also a noteworthy extension. Different from previous PEE-based schemes, this technique advocates for generating a skewed histogram by using several extreme predictions. Figure 7 provides an illustration of the generated

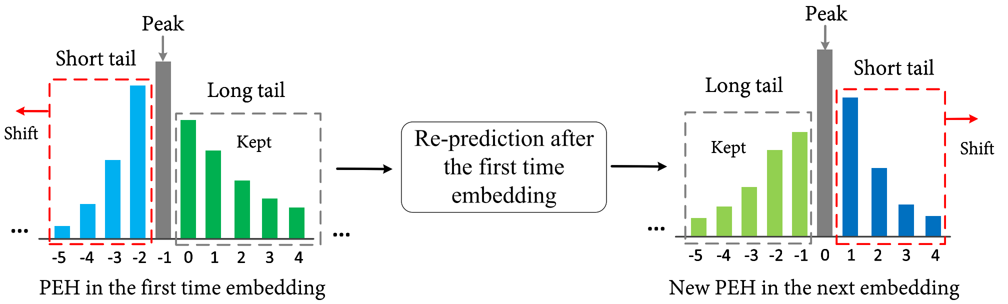


Fig. 7. The illustration of the histogram modification for the asymmetric PEH.

skewed PEH and its modification framework. The so-called skewed histogram is asymmetrically distributed. That is, it consists of a peak and two asymmetric sides. The sharper side is called the short tail, and the other side is called the long tail. In this method, the embedding is accordingly completed twice. In each layer of embedding, only the shorter side is modified for embedding. One advantage of this technique is that the modification in the first time may be recovered in the next step. This means that the ED can be dynamically estimated and adjusted during the embedding.

The second kind of improvement for PEE is to adaptively select the appropriate pixels for modification. Notice that this kind of improvement is not restricted to PEE-based extensions. Other schemes also have similar improvements. Here, we will only use PEE as an example, as it is currently the most popular RDH method. Sachnev et al. [90] proposed to sort the pixels based on the magnitudes of their local variances. This approach aims to use the smooth pixels for embedding as much as possible. Considering reversibility, it employs the local variances instead of the values of PEs, to perform element selection. The pixels with smaller local variances are preferentially used for embedding, which significantly reduces the embedding distortion. It has been proven that this sorting technique is effective for the distortion reduction, especially for the low-capacity embedding. Furthermore, a hotspot research direction [47–49] is to adaptively determine the expansion bins or modification manner based on the given image content. Li et al. [47] proposed to divide image pixels into two categories according to their smooth degrees, i.e., the pixels from smooth regions and pixels from texture regions. Then, the expansion pixels of smooth regions will be embedded with two bits, since their PEs are usually smaller. For the texture pixels, they will still be embedded with one bit. In this method, by its nature, the pixels are classified in a finer way, resulting in a more efficient use of redundancy. Table 4 summarizes the existing adaptive embedding strategies.

In the next section, we will focus on several mainstream extensions including MHM [27, 49, 60, 69, 78, 96–98, 103, 105, 108], pairwise PEE [5, 12, 15, 65, 68, 84, 114, 131], and PVO [21–23, 44, 46, 61, 63, 64, 67, 70, 73, 88, 99, 104, 112, 130, 133]. Among them, MHM and pairwise PEE can be seen as representative techniques of adaptive embedding, which achieve multiple-level and deep-level embedding, respectively. While PVO contributes to formulating a more natural predictor that locally sorts the pixel values for prediction.

3 CURRENT TYPICAL RDH TECHNIQUES FOR ADAPTIVE EMBEDDING

In this section, three of the most popularly utilized RDH schemes and their extensions are presented, including MHM [27, 49, 60, 69, 78, 96–98, 103, 105, 108], pairwise PEE [5, 12, 15, 65, 68, 84, 114, 131], and PVO [21–23, 44, 46, 61, 63, 64, 67, 70, 73, 88, 99, 104, 112, 130, 133]. Those schemes are improvements on the classic methods mentioned above and have their own technological characteristics. In addition, the discussion of these representative techniques will also be presented last.

Table 4. Overview of Adaptive Embedding Strategies

Strategy	Year	Country	Citations	Description
Pixel sorting [90]	2009	South Korea	967	The smooth pixels are modified preferentially by the sorting strategy.
Pixel selection [47]	2011	China	711	Smooth pixels are embedded with more secret bits.
Multi-level embedding[18]	2014	China	94	The pixels are partitioned based on the levels of smoothness. Each level is allocated a different embedding capacity.
MHM [49]	2015	China	342	Generate multiple PEHs and set suitable parameters for each sub-PEH. Pixels are treated differently based on the local context.
Pairwise PEE [68]	2013	China	494	The modification of PEH is updated to the 2D space by utilizing the second-order correlations of pixels.
PVO [46]	2013	China	485	Select both ends of the ordered pixel sequence from each block for embedding.
AGM [89]	2016	Malaysia	40	The PE with a high frequency is embedded with more than one bit and the low-frequency PEs are discarded for embedding.
Reduce shifting[34]	2019	China	135	The surplus shifting pixels are eliminated adaptively according to the local context.

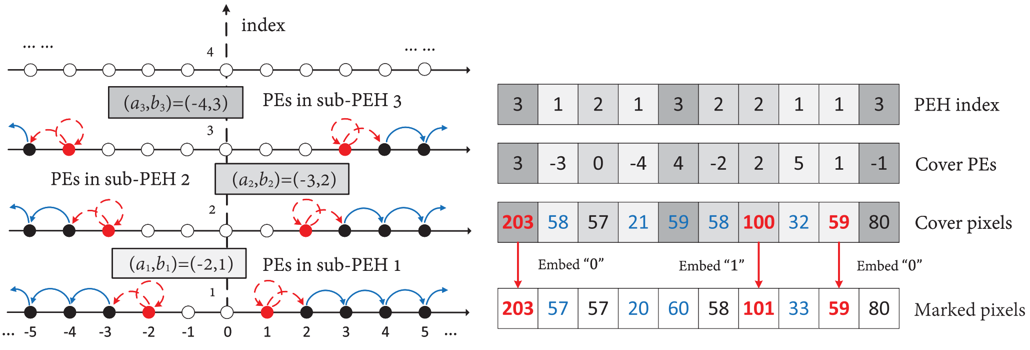
3.1 Multiple Histograms Modification

MHM is a generalized scheme of PEE. In MHM, the cover pixels are classified based on the combination of PEs and texture levels. In the PEE-based methods, the PEs of smooth pixels are usually with the values near zero. This is because the smooth regions usually have the high correlations. So, it is easier to obtain accurate predictions for the smooth pixels. The PEH generated from those regions has a more concentrated distribution. This means that embedding message into the smooth pixels will result in the less embedding distortion. A more reasonable modification approach is to assign different expansion bins to pixels with varying texture degrees.

In Reference [49], the N cover pixels $(p_1, p_2, \ldots, p_N)$ are divided into M categories. Each category is processed separately using its own set of parameters. First, the pixels with the same complexity level are collected into a subset. The complexity level is calculated based on the local context. Second, each subset will derive a PEH with two expansion bins. In other words, the original PEH will be divided into M sub-PEHs satisfying $h(e) = \sum_{c=1}^{M} h_c(e)$. Here, h_c represents the sub-PEH that tallies the PEs with complexities of c_i, $c \in \{1, 2, \ldots, M\}$. As a result, the cover pixels are classified according to the pair (e_i, c_i). Third, by using exhaustive search, the optimal expansion bin set $\{(a_c, b_c)\}_{c=1}^{M}$ are found adaptively based on the content of each subset. In this way, the pixels in each subset will have their respective mappings. This forms the multiple levels modification model.

An illustration of the MHM-based embedding mechanism [49] is shown in Figure 8. The cover pixels are marked with varying gray levels according to their histogram indexes. For each sub-histogram, based on the expansion bins $\{(a_c, b_c)\}_{c=1}^{M}$, the modifications for the cover pixels belonging to the sub-PEH h_c could be summarized as the following five cases:

- The cover pixel is added “1" to move the space, when $e_i > b_c$.
- The cover pixel is expanded as $p_i + m$ to embed one secret bit, when $e_i = b_c$.
- The cover pixel is kept unchanged, when $a_c < e_i < b_c$.
- The cover pixel is expanded as $p_i - m$ to embed one secret bit, when $e_i = a_c$.



(a) Mapping of the PEs (b) Pixels modification

Fig. 8. The data embedding example of MHM.

- The cover pixel is added "-1" to move the space, when $e_i < a_c$.

The pixels from different levels are modified according to their respective embedding parameters. Correspondingly, it is necessary to allocate a small amount of storage space to record the selected parameters. During the extracting, the cover pixels are recovered in a reverse manner.

In general, MHM is flexible by changing the expansion bin set. The meticulous modification can be realized by selecting suitable expansion bins for the pixels from different texture areas. Later, Ou et al. [69] and Qi et al. [78] successively improved [49] by expanding its framework to a more general form. For the cth histogram, K_c pairs of expansion bins $\{(a_{c,k}, b_{c,k})\}_{k=1}^{K_c}$ satisfying $a_{c,k} < \ldots < a_{c,1} \leq 0 \leq b_{c,1} < \ldots < b_{c,k}$ will be selected. During the embedding, the modifications for the cover pixels in each sub-PEH are still classified into five cases. The difference is that the size of shifting step for each pixel is adjusted. That is, the marked pixels are determined as follows:

- Case 1: If $b_{c,k} < e_i < b_{c,k+1}$, then the cover pixel should be shifted rightwards by k steps, i.e., $\widetilde{p_i} = p_i + k$.
- Case 2: If $e_i = b_{c,k}$, then the pixel is expanded to obtain the marked one as $p_i + (k-1) + m$.
- Case 3: If $a_{c,1} < e_i < b_{c,1}$, then the cover pixel is unchanged.
- Case 4: If $e_i = a_{c,k}$, then the marked pixel is $p_i - (k-1) - m$.
- Case 5: If $a_{c,k+1} < e_i < a_{c,k}$, then the cover pixel should be shifted leftwards by k steps, i.e., $\widetilde{p_i} = p_i - k$.

In References [69, 78], each histogram could be embedded with more secret messages. On average, the largest ER achieved in Reference [69] is up to 0.612 bpp, which is significantly higher than the ER of the method proposed by Li et al. [49] (0.176 bpp).

Considering the outstanding performance of MHM, various extensions have been proposed. Table 5 summarizes the existing MHM-based schemes. The improvements for MHM can be mainly divided into two types, i.e., the pixel division [27, 97, 103, 105] and parameter optimization [60, 96, 98, 108]. First, a more effective method of division could ensure that the frequent elements are grouped together as much as possible. It is convenient to select suitable expansion bins. Second, a well-designed expansion bin selection strategy could find a more effective modification method. Furthermore, reducing computational complexity is also a motivation for optimizing the parameter search algorithm. Wang et al. [97] mentioned that the heuristic pixel division method used in [49] lacks adaptability and suggested generating the PEH using fuzzy c-means clustering. It successfully builds multiple sharper distributed PEHs. Later, Wang et al. [96] pointed out that the bin selection in MHM is actually the process of determining the appropriate payload for each

Table 5. Overview of the MHM-based Methods

Work	PEH generation	Bins selection	Complexity	Maximum ER	PSNR
MHM [49]	Calculate the back complexity based on 12 neighbors.	Exhaustive search with limits in a specific range.	$O(2^{18})$	0.25 bpp	61.02 dB
Wang et al. [97]	Use fuzzy c-means clustering.	Same as in Reference [49].	$O(2^{18})$	0.18 bpp	61.15 dB
Ou et al. [69]	Same as in Reference [49].	Find multiple pairs of bins by searching large solutions.	$O(2^{20})$	0.8 bpp	61.01 dB
Qi et al. [78]	Same as in Reference [49].	Get series of bins for each PEH using dynamic programming.	$O(2^{27})$	0.64 bpp	61.04 dB
Wang et al. [96]	Divide the obtained complexity levels in a same proportion.	Allocate the payload for each sub-PEH using evolutionary algorithm.	—	0.7 bpp	60.92 dB
Hou et al. [27]	Use a deep neural network to learn a suitable clustering model.	Enlarge the searching region and set memos to speed the search.	$O(2^{26})$	0.24 bpp	61.46 dB
Weng et al. [105]	Use the k-means clustering.	Use the improved criss-cross optimization algorithm.	—	0.15 bpp	61.51 dB
Wu et al. [108]	Same as in Reference [49].	Derive the mapping based on the reversible embedding function.	—	0.61 bpp	61.00 dB

Here, the strategies of pixel division and the selection of expansion bins in each work are simply introduced. The PSNR is calculated based on the result when embedding 10,000 bits into the test image Lena.

sub-PEH. They established the ER allocation model and introduced the evolutionary algorithm to solve it. With this method, the EC of MHM is increased and a higher PSNR can be achieved for a given payload. Weng et al. [105] introduced the k-means clustering algorithm to classify the PEs and improved the clustering precision and speed. In addition, they found a similarity between the selection of expansion bins and the typical multi-choice knapsack problem. The searching method is replaced with the improved crisscross optimization algorithm. Wu et al. [108] investigated the generalization form of histogram modification-based RDH and introduced the so-called reversible embedding function to provide a mathematical description for it. Based on the results obtained from the function, the expansion bins can be determined. Compared to the conventional optimization strategy, this approach is more flexible, because it considers a greater number of possible solutions. Ma et al. [60] developed a fast expansion bins determination method. The original optimization objective is modeled as the real value differentiable function and advanced analysis tools such as Lagrange multipliers are used to solve it. This method reduces the average runtime from 41.7 s to 0.05 s.

3.2 Pairwise PEE

In the conventional PEE [90, 94], the modification of pixels is designed based on the low-dimensional statistical results of the image data. The mapping generated based on the 1D PEH is not capable of fully utilizing the complex correlation between the adjacent PEs. Pairwise PEE [68] offers an effective solution, i.e., considering the modification in a higher-dimensional space. This method first takes every two nearby PEs as a unit to classify the pixels and generate a 2D PEH. Then, the expansion pixels, shifting pixels, and unchanged pixels are determined by designing modification rules for the 2D PEH bins, i.e., establishing a 2D mapping. Next, we will simply introduce the specific framework of method [68] and analyze its characteristic.

Table 6. The Nine Possibilities for the 2D Mappings of PE Pairs in the Conventional PEE

Case	Mapping directions	Condition
(a)	$(x,y) \longrightarrow \{(x,y),(x+1,y),(x,y+1),(x+1,y+1)\}$	$x=b, y=b$
(b)	$(x,y) \longrightarrow \{(x+1,y),(x+1,y+1)\}$	$x>b, y=b$
(c)	$(x,y) \longrightarrow \{(x,y),(x,y+1)\}$	$x<b, y=b$
(d)	$(x,y) \longrightarrow \{(x,y),(x+1,y)\}$	$x=b, y<b$
(e)	$(x,y) \longrightarrow \{(x,y+1),(x+1,y+1)\}$	$x=b, y>b$
(f)	$(x,y) \longrightarrow (x+1,y)$	$x>b, y<b$
(g)	$(x,y) \longrightarrow (x,y+1)$	$x<b, y>b$
(h)	$(x,y) \longrightarrow (x+1,y+1)$	$x>b, y>b$
(i)	$(x,y) \longrightarrow (x,y)$	$x<b, y<b$

Here, "$\longrightarrow$" denotes the mapping direction of the PE pair, and b is the selected expansion bin.

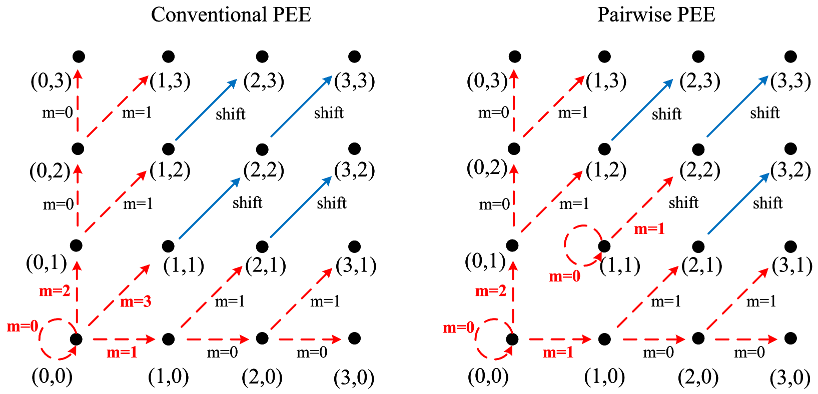
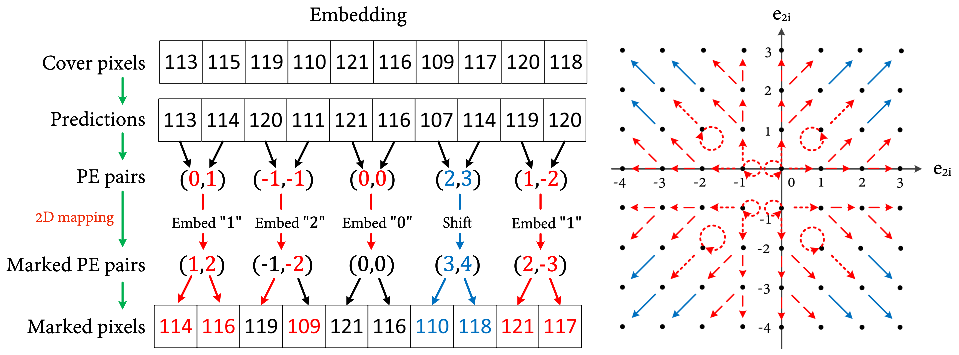


Fig. 9. Two-dimensional mapping comparison, where the expansion bin is $b = 0$ for the conventional PEE.

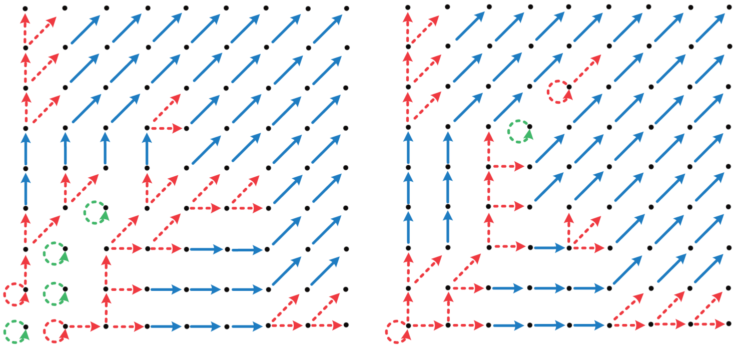
In pairwise PEE, the adjacent two PEs are combined to form the sequence $\{(e_1, e_2), (e_3, e_4), \ldots, (e_{N-1}, e_N)\}$. Similarly, the PEH is generated, and the mapping is established based on its distribution. The difference is that the PEH is generated in the 2D coordinate system, i.e., $h(x,y) = \#\{1 \leq i \leq \frac{N}{2} : e_{2i-1} = x, e_{2i} = y\}$, where h counts the frequencies for the PE pairs of (x,y). Then, the expansion bins are adaptively selected to generate a 2D mapping. During the embedding, the adjacent pixels will be modified jointly to either hide one bit or shift it. This approach allows for more possibilities in the shifting direction of each PEH bin.

As an extension of PEE, the modification manner of pairwise PEE is compatible with the 1D mapping-based methods. The modification of the conventional PEE, as defined in Equation (4), can also be explained in the 2D space. For the conventional PEE, the mapping of PE pairs (x,y) has nine possibilities. They are listed in Table 6. Note that only the mapping for the PEs with positive values is illustrated for brevity. In Reference [68], the authors developed a different modification model that was based on an analysis of embedding efficiency. The obvious improvement is case (a) mentioned in Table 6. For the pair (b,b), the pairwise PEE transforms its mapping outputs to $\{(b,b),(b+1,b),(b,b+1)\}$. Meanwhile, the pair $(b+1,b+1)$ is selected as the expansion bin, and the self-mapping is added to its outputs. Figure 9 displays two different 2D mappings. The 2D mapping used in pairwise PEE [68] introduces a new modification framework. In Reference



(a) (b)

Fig. 10. The pixels modification using pairwise PEE. (a) The embedding steps in the pairwise PEE. (b) The whole mapping manner of the pairwise PEE.



(a) Lena (b) Airplane

Fig. 11. The generated 2D mapping using the adaptive strategy, for the test image Lean and Airplane, when EC=10,000 bits.

[68], the performance of the newly designed 2D mapping has been demonstrated. Based on the determined 2D mapping, it is possible that the secret message is no longer embedded in binary form. The data embedding of pairwise PEE is shown in Figure 10.

The pairwise PEE provides a new representation of data embedding for RDH design. The potential of the high-dimensional-based modification becomes attracted with more attentions, and many 2D RDH methods are proposed. In the successive research, it has been proven that flexibly adjusting the expansion bins according to the given histogram distribution is helpful in improving performance. Considering this, the strategy for generating adaptive 2D mapping [5, 15, 65, 84, 114, 131]is proposed. Ou et al. [65] proposed two solutions to determine the 2D mapping for a given image. The first solution involves traversing a large number of possible solutions. By comparing the distortion-capacity ratio of each candidate, the optimum mapping can be determined. The second solution is to use the optimal probability matrix. Based on the estimation of mapping probability, the final modification model is confirmed. Its modification manner is image dependent, so that better performance could be achieved. Chang et al. [5] and Zhang et al. [131] successively designed an adaptive 2D mapping method. In Reference [5], the optimal 2D mapping is initialized as a random solution and the optimal one is found by the iterative updating process. Figure 11 presents

Table 7. Overview of the Optimization Strategies for the Pairwise PEE

Work	Year	Complexity	PSNR	Description
Pairwise PEE [68]	2013	$O(2^{10})$	59.75 dB	• The first 2D mapping-based scheme. • The 2D mapping is fixed for different images.
M2D-PEHs [84]	2019	$O(2^{18})$	60.42 dB	Design eight alternative 2D mappings and select t-he optimal combination of mappings for the MHM.
Hybrid PEE [65]	2019	$O(2^{10})$	60.77 dB	• The PEs are adaptively paired. • The optimal mapping is searched in a certain re-gion.
OPPEE [114]	2019	$O(2^{25})$	61.00 dB	Turn the 2D mapping optimization to the optimal routine determination issue and solve it using dyn-amic programming.
APPEE [5]	2021	$O(2^{10})$	61.35 dB	Utilize the random iteration strategy to update the 2D mapping in the 8 × 8 sized region.
SMPPEE [15]	2021	$O(2^{18})$	61.58 dB	Design the skewed 2D histogram modification, and apply it to the multiple skewed PEHs.
AMPPEE [131]	2022	$O(2^{18})$	60.93 dB	• The mapping outputs of the 2D PEH bins are ad-justed in a pre-set order. • The MHM framework is employed.

Here, the PSNR is calculated based on the result when embedding 10,000 bits into the test image Lena.

two examples. The 2D mapping can change in accordance with the image content. Zhang and Ou [131] proposed an alternative approach to optimize the mapping. In their method, the collected PE pairs are sorted according to their frequencies and positions, and the mapping inputs of these pairs are adjusted one by one. Fan et al. [15] pointed out that the pairwise PEE could be combined with skewed histogram to improve performance. Furthermore, they applied the skewed PEH-based 2D mapping to the MHM framework and can conduct more effective modifications. Recently, the authors of References [3, 4] have been developed to extend the pairwise modification technique to color images and design a corresponding 3D mapping optimization strategy. In Reference [3], a mapping generation model is trained using the reinforcement learning algorithm, i.e., double deep Q-network. In Table 7, various adaptive optimization strategies for the pairwise PEE are summarized.

In addition, the adaptive pairing strategy [12, 65] is also an effective technique to enhance performance. In the initial pairwise PEE [68], the PE pairs are generated by considering the adjacent two pixels jointly. The combination is conducted only based on the positions, i.e., the diagonal or the anti-diagonal directions. References [12, 65] suggest that this approach may lead to imperfect matching, particularly for pixels in special regions like the edges. This is because the spatial distance cannot always perfectly reflect the pixel similarity. Dragoi et al. [12] suggested that the PEs should be paired based on the gray values rather than the spatial location. By using this approach, the frequencies of PEH bins that are concentrated near zero are increased. Ou et al. [65] devised a new embedding framework for the pairwise PEE, namely hybrid-dimensional histogram modification. In this scheme, the pixels from the smooth region and those from the texture region are processed using different methods. Only the smooth pixels are combined into pairs and used for embedding. This is because smooth pixels usually generate a more sharply distributed histogram. The texture pixels are collected to generate the 1D PEH. They will be shifted in the manner of 1D mapping, without embedding any bit. This pairing manner brings a more precise classification for the pixels. The generated 2D PEH has lower entropy and the embedding distortion is further reduced.

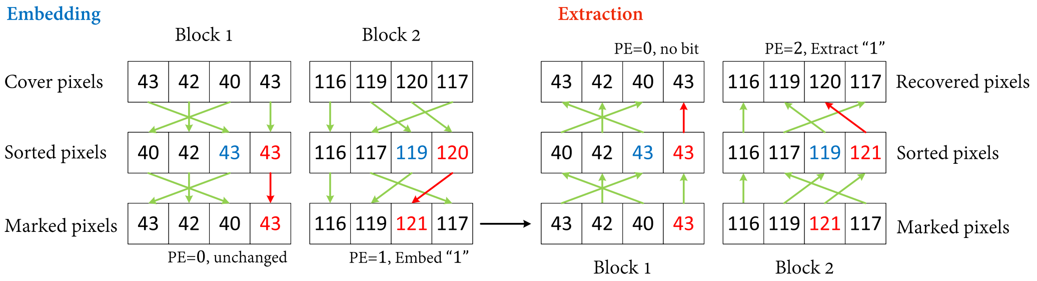


Fig. 12. Illustration of the embedding and the data extraction in PVO-based PEE.

3.3 Pixel-value-ordering

PVO is a more natural prediction by utilizing the pixel similarity within a block. Different from calculating the prediction for each pixel, Li et al. [46] suggested utilizing only the maximal pixel and the minimal one in a block for data embedding. The maximal and minimal pixels are predicted by the second largest and smallest pixels, respectively. Both the pixel positions and the gray values of pixels are considered during the prediction, making it more accurate. Besides, only two pixels are modified in a block, and the embedding distortion is less compared with the conventional PEE methods. Since the embedding processes are similar, we will just explain the modification of the maximum for simplicity.

For the pixels block $(p_1, p_2, \ldots, p_n)$, the n pixels will be first organized based on gray values to regenerate the sequence as $p_{\sigma(1)} \leq p_{\sigma(2)} \leq \ldots \leq p_{\sigma(n)}$. Here, the mapping σ establishes a one-to-one relationship between the new index and the original one. For two pixels with the same value, the new order is determined based on their original indexes, i.e., $\sigma(i) < \sigma(j)$ when $i < j$. In the ordered sequence, only the largest pixel is predicted and modified. First, it directly uses the nearest element as the prediction, and the PE is calculated as $e_{max} = p_{\sigma(n)} - p_{\sigma(n-1)}$. Clearly, the PEs in PVO are always greater than or equal to 0. Second, the PE with the value of 1 is used for expansion. As a result, the largest pixels are modified in a reversible manner to embed the secret bits. Specifically, the pixels with PEs "0" are unchanged and will be skipped during the embedding. For the pixels with PEs "1," they are expanded as $p_{\sigma(n)} + m$ to conceal the secret bit. If the corresponding PE is larger than 1, then the pixel is shifted and incremented by 1.

Since the maximum pixel will be either added by 1 or unchanged, the maximum value remains the largest one in the marked block. The pixel value order in the marked block remains unchanged. It guarantees reversibility. For the marked pixel, the restoration and extraction are completed in the opposite manner. The example of the data embedding and extraction for the PVO method [46] is shown in Figure 12.

In PVO, the most similar pixels in a block can be found by sorting the pixel values. It can obtain accurate predictions and generate the PEH with lower entropy, making the embedding more efficient. For a low capacity embedding, PVO typically exhibits better performance in terms of PSNR. So far, several extensions of PVO have been proposed, including the **improved PVO (IPVO)** [61, 73], PVO-k [67], k-pass PVO [23], **pixel-based PVO (PPVO)** [88, 106, 130], pairwise PVO [21, 63, 112, 133], MHM-based PVO [22, 44, 64], adaptive block complexity calculation [70, 112], and dynamical block size determination [99, 104].

(1) IPVO [73]

In IPVO, the formulation for calculating the PEs is redefined by taking the original positions of target pixel and its prediction into consideration. For the sorted block pixels $(p_{\sigma(1)}, p_{\sigma(2)}, \ldots, p_{\sigma(n)})$, the PEs are calculated as follows: ① If the original index of maximum is smaller than that of the

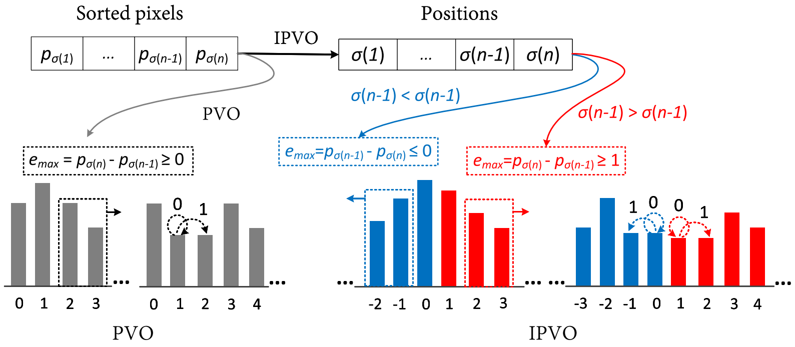

Sorted pixels
$p_{\sigma(1)}$ … $p_{\sigma(n-1)}$ $p_{\sigma(n)}$
IPVO
Positions
$\sigma(1)$ … $\sigma(n-1)$ $\sigma(n)$
PVO
$\sigma(n-1) < \sigma(n-1)$
$\sigma(n-1) > \sigma(n-1)$
$e_{max} = p_{\sigma(n)} - p_{\sigma(n-1)} \geq 0$
$e_{max}=p_{\sigma(n-1)} - p_{\sigma(n)} \leq 0$
$e_{max}=p_{\sigma(n)} - p_{\sigma(n-1)} \geq 1$
0 1
0 1 2 3
0 1 2 3 4
PVO
1 0 0 1
-2 -1 0 1 2 3
-3 -2 -1 0 1 2 3 4
IPVO

Fig. 13. Comparison of the PEH modification for PVO and IPVO during the embedding.

prediction, then the difference is the same as the PVO, i.e., $e_{max} = p_{\sigma(n)} - p_{\sigma(n-1)}$. ② If the original index of maximum is larger than that of the prediction, then the value of PE will be negative, i.e., $e_{max} = p_{\sigma(n-1)} - p_{\sigma(n)}$. With this definition, the generated PEH is more like that in conventional PEE, i.e., a Laplace-like distribution centered at zero. The comparison of the histogram modification for PVO [46] and IPVO [73] is shown in Figure 13. In IPVO, since the histogram is symmetrical distributed, the shifting of PEs could have two directions. The expandable PEs are selected as the two peak points, i.e., the PE “0” and the PE “1.”

During embedding, the pixels with PEs “0” or “1” are all selected for carrying secret bits. These pixels are kept to embed bit “0" or added by 1 to embed bit “1.” For PEs larger than 1 or smaller than 0, the corresponding pixels are shifted rightwards, i.e., added by 1, to ensure reversibility. Notice that, in IPVO, the shifting direction of PEs is different from that of pixels. For the cover pixels, they are either added by 1 or remain unchanged after the embedding. Compared to conventional PVO, different PEs are utilized in IPVO to carry the secret bits. Its classification method is more refined, allowing for a higher payload and better visual quality.

(2) PVO-*k* [67]

PVO-*k* is a generalized extension of PVO. It takes more situations into consideration. In conventional PVO, only the last pixel in the sequence $(p_{\sigma(1)}, p_{\sigma(2)}, \ldots, p_{\sigma(n)})$ is used for embedding. However, there is one special case that one block has several largest pixels, such as $p_{\sigma(n-2)} = p_{\sigma(n-1)} = p_{\sigma(n)}$. The conventional PVO approach results in skipping many blocks during the embedding. To address this issue, in PVO-*k*, the simultaneous modification of the top *k* largest pixels to embed one bit is enabled.

For the block pixels satisfying $p_{\sigma(1)} \leq \ldots \leq p_{\sigma(n-k)} < p_{\sigma(n-k+1)} = \ldots = p_{\sigma(n)}$, there are *k* largest pixels. Taking the second largest pixel as the prediction, these *k* pixels can share the same PE, i.e., $e_{max} = p_{\sigma(n)} - p_{\sigma(n-k)}$. By selecting PE “1" as the expansion bin, the embedding and extraction processes are essentially identical to those in PVO. Notice that, the *k* largest pixels should be modified as a whole to ensure reversibility, i.e., each block is still hidden one bit. The authors in [67] has proven that better embedding performance can be achieved by flexibly using PVO-2 and PVO-1.

(3) *k*-pass PVO [23]

The aim of *k*-pass PVO is similar to PVO-*k*, i.e., using multiple pixels in each block for embedding. The difference is that the selected multiple pixels in *k*-pass PVO may not be equivalent. In *k*-pass PVO, for the pixel sequence satisfying $p_{\sigma(1)} \leq \ldots \leq p_{\sigma(n-k)} \leq p_{\sigma(n-k+1)} \leq \ldots \leq p_{\sigma(n)}$, each of the top *k* largest pixels are allocated with a PE. The prediction starts with the maximum value and continues for the rest. First, the $n - k$ indexed pixel in the ordered sequence is selected as

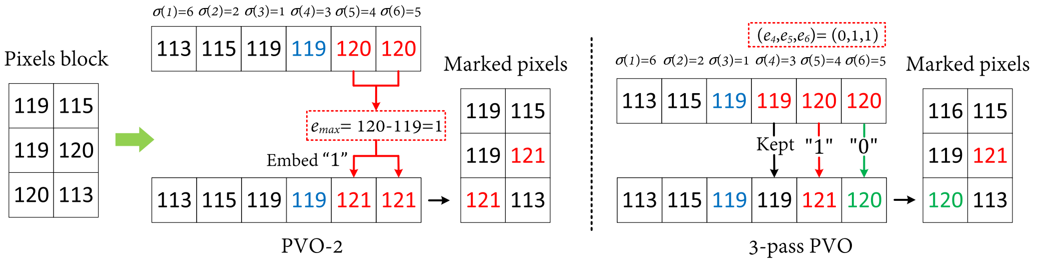


Fig. 14. Embedding comparison of PVO-k and k-pass PVO.

the prediction for the maximum, and the PE is obtained by using $e_n = p_{\sigma(n)} - p_{\sigma(n-k)}$. Next, the PEs of the other top $k-1$ largest pixels are determined based on e_n. For the pixel $p_{\sigma(i)}$, where $i \in \{n-k+1, n-k, \ldots, n-1\}$, there are two possibilities for its PE, i.e.,

- If the PE of the maximum is larger than 1, then the PE is determined as the difference between it and the $n-k$ indexed pixel, i.e., $e_i = p_{\sigma(i)} - p_{\sigma(n-k)}$.
- Otherwise, the maximum becomes the prediction, i.e., the PE is calculated as $e_i = p_{\sigma(i)} - (p_{\sigma(n)} - 2)$.

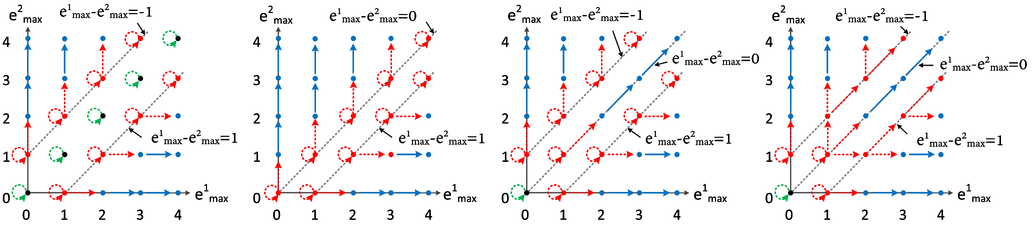
By using this prediction method, each to-be-modified pixel will have a PE and the relative order can be restored after the embedding. When embedding, the pixels with PEs "1" are determined as the expansion pixels, the pixels with PEs larger than 1 are classified as the shifting pixels, and the rests are the unchanged pixels.

In k-pass PVO method, one bit is embedded into each available pixel instead of processing them as a whole. Each block can carry k bits at most. The comparison of PVO-k [67] and k-pass PVO [23] is presented in Figure 14. In PVO-k, the top two largest pixels (120, 120) are expanded to (121, 121) simultaneously to embed the secret bit "1." While in the k-pass PVO, the top two largest pixels (120, 120) are embedded with the bits "1" and "0" respectively. In the k-pass PVO, it can be observed that the relative order of the marked pixels differs from that of the cover pixels. That is, the pixel $120_{\sigma(5)}$ is expanded to 121, making it the largest one in the marked sequence. However, this will not affect the extraction result, because the recovered bits can be further adjusted based on their positions prior to ordering.

(4) PPVO [88]

In PPVO, the prediction model of PVO is combined with the conventional predictors utilized in PEE. That is, each pixel of the block will be assigned a PE. During the prediction, the target pixel and its neighbors are collected together to form a pixel block. If the target pixel is the largest element in the block, then it will be predicted by its largest neighbor. If it is the smallest element, then its prediction becomes the minimum of its neighbors. Otherwise, this pixel will be marked as an unchanged pixel during the embedding. One special case is when all the neighbors are the same and the target pixel becomes the smallest one. In this case, the target pixel is predicted by "maximum - 1."

During the embedding process, the modification method can be determined by considering the relationship between the target pixel and its neighboring pixels. When the current pixel is equal to the largest element or the minimum of the neighbors, i.e., PE = 0, it is expanded to carry the information. When the target pixel is larger than the maximum value of its neighbors, its value will be incremented by 1 to shift. Similarly, if it is smaller than the minimal element, then the current pixel will be added by -1. With PPVO, the number of embeddable pixels is increased, allowing for the hiding of more additional bits.



(a) PVO (b) IPVO (c) PVO-2 (d) 2-pass PVO

Fig. 15. Two-dimensional mapping comparison for (a) PVO, (b) IPVO, (c) PVO-k when $k = 2$, and (d) k-pass PVO when $k = 2$.

(5) Pairwise PVO [63]

In the schemes mentioned above, one can find that the expansion bins are always selected as either PE "1" or "0." It lacks adaptive variations. As an effective adaptive strategy, 2D mapping could provide the modification of PVO with more options, so that the performance can be further enhanced. Ou et al. [63] extended the pairwise PEE to PVO by considering the top two largest elements in each block as a pair.

For the ordered pixel sequence $p_{\sigma(1)} \leq p_{\sigma(2)} \leq \ldots \leq p_{\sigma(n)}$, the pixels $p_{\sigma(n)}$ and $p_{\sigma(n-1)}$ are first predicted by the third largest one. Here, for the sake of reversibility, the pairing order should be constrained. That is, the PE pair (e^1_{max}, e^2_{max}) is generated by considering both the positions and the gray values. Specifically, the first PE of the pair is the difference of the lower indexed one, i.e., $e^1_{max} = p_u - p_{\sigma(n-2)}$, where $u = \min(\sigma(n), \sigma(n-1))$. Accordingly, the second PE of the pair is the difference of the higher indexed one, i.e., $e^2_{max} = p_v - p_{\sigma(n-2)}$, where $v = \max(\sigma(n), \sigma(n-1))$.

In this way, the block pixels could be modified based on a created 2D mapping. Actually, the four methods mentioned above, namely PVO [46], IPVO [73], PVO-k [67], and k-pass PVO [23] could also be explained in the 2D space. The comparison of their 2D mappings is shown in Figure 15. Similarly to the improvements mentioned in Section 3.2, several studies have been conducted to enhance the pairwise PVO through 2D mapping optimization [21, 130] and adaptive pairing [112, 133].

(6) MHM-based PVO [64]

It has been proved that MHM is one of the most effective adaptive embedding strategies. Therefore, by extending the MHM framework to PVO, a more satisfied performance can be obtained. For the N cover pixel blocks $\{B_1, B_2, \ldots, B_N\}$, the 1D histogram can be generated by collecting the maximum value from each block into a set, where $B_i = (p_{\sigma(1)}, p_{\sigma(2)}, \ldots, p_{\sigma(n)})$. In conventional method, the pixel $p_{\sigma(n)}$ is typically classified based on its PE, i.e., e_{max}, and then the optimal embedding parameters are confirmed adaptively by considering the histogram distribution.

In MHM-based PVO, the generated 1D histogram is further divided into multiple sub-histograms based on the complexity measurements. In Reference [64], the block complexity c_i is identified as the second largest difference in the Nth block. i.e., $c_i = p_{\sigma(n-1)} - p_{\sigma(1)}$. Based on the pair (e_{max}, c_i), the maximum values are allocated to different sub-histograms. Then, by considering the distributions of the obtained sub-histograms, the appropriate expansion bin set is adaptively determined.

3.4 Discussion

In this section, the three current typical RDH techniques mentioned above, including MHM, pairwise PEE, and PVO, will be further analyzed to illustrate their advantages compared to the conventional methods.

Table 8. Comparison of the Prediction Precision for the Common Used Predictors and PVO on the 512 × 512 Sized Test Images Airplane and Lena

Image	Predictor	Mean	Variance	MSE	PEH entropy
Airplane	DP [95]	5.31	93.95	120.67	4.59
	MED [94]	5.51	105.71	134.52	4.67
	GAP [14]	3.53	31.87	43.81	4.12
	RP [90]	3.81	41.84	56.24	4.19
	PVO [46]	2.84	30.46	8.50	1.27
Lena	DP [95]	6.48	87.05	127.57	5.05
	MED [94]	5.89	72.71	106.16	4.91
	GAP [14]	3.96	25.38	40.58	4.38
	RP [90]	4.31	32.86	50.86	4.49
	PVO [46]	2.99	23.78	7.22	1.39

The MHM-based methods can be seen as the natural evolution of the adaptive embedding strategy. In the previous methods, such as pixel sorting and adaptive expansion-bins-selection, some of the shifting pixels will be labeled as the unchanged class, effectively reducing unnecessary shifting. These approaches can be seen as refined modifications. However, these methods are typically conducted based on global statistical features. It conflicts with the finer modification. One price is that a large number of embeddable pixels are also labeled as the unchanged class. In MHM, to address this issue, clustered local statistical features are utilized to divide the pixels, transforming the original histogram into multiple sub-histograms. In this way, the embedding becomes a modification with multiple levels. It is possible to determine a more suitable modification manner for the local feature.

The pairwise PEE is another method of multi-level modification. For the 1D mapping, the modifications should adhere to certain rules to ensure reversibility. This imposes certain restrictions on the finer modifications to some extent. While in pairwise PEE, the 2D mapping provides the embedding with more flexibility in terms of modification directions. This makes the 2D mapping could modify the pixels in a finer way, i.e., the multiple levels modification. This is why the pairwise PEE could achieve better performance.

As for PVO, it leverages the inherent characteristics of the local image content to achieve more accurate predictions instead of relying on artificial formulations. For a given pixel block, the two largest pixels are naturally closer. The reason is that the neighboring pixels used for prediction in the traditional predictors [14, 90, 94, 95] are usually chosen from the local block. While in the same block, the neighboring pixels for the maximum value are all no more than the second largest one. It makes the prediction of PVO more consistent with the local context, i.e., the estimation is more reasonable. The experimental results listed in Tables 8 and 9 support this theory. The block size in PVO is set as 3 × 3 here. According to Table 8, it can be seen that the mean and variance for the absolute PEs obtained by PVO are much lower than those of the other four predictors. It illustrates that PVO is capable of generating PEs with lower values, i.e., the predictions are more accurate. Correspondingly, the MSE between the pixels and the predictions calculated using PVO is also the smallest. In terms of the histogram distribution, the PEH generated by PVO has a relatively lower entropy. This means that its histogram is sharper. Table 9 compares the performance of PVO and traditional predictors [14, 90, 94, 95] for predicting the maximum of each block. For the eight test images, PVO can achieve better predictions in a large proportion (higher than 83%).

Table 9. Experimental Results for the Proportion of the Better Predictions Obtained by PVO on the Eight Standard Test Images Downloaded from the CVG-UGR Databset

Predictor	Airplane	Baboon	Barbara	Boat	Elaine	Lake	Lena	Peppers
PVO [46] vs. DP [95]	91.1%	88.3%	93.8%	93.3%	91.9%	91.4%	91.9%	92.5%
PVO [46] vs. MED [94]	87.9%	82.5%	90.0%	89.9%	85.6%	85.9%	86.1%	86.2%
PVO [46] vs. GAP [14]	83.0%	84.2%	83.8%	85.9%	89.0%	86.1%	82.9%	88.2%
PVO [46] vs. RP [90]	82.5%	87.5%	85.7%	86.0%	90.7%	86.1%	83.8%	89.0%

Table 10. Overview of the Uncompressed Image Databases

Database	Description	Image size	Link
USC-SIPI	It contains 28 grayscale images and 16 color images. All of them are in the TIFF format.	256 × 256, 512 × 512, 1024 × 1024	http://sipi.usc.edu/database/
Kodak	The Kodak set owns 24 true color images stored in the PNG format.	768 × 512, 512 × 768	http://r0k.us/graphics/kodak/
RAISE	This dataset possesses 8,156 high resolution nature color images stored in the RAW format.	4288 × 2848, 4928 × 3264	http://loki.disi.unitn.it/RAISE/
BOSSbase1.0.1	A large database with 10,000 gray images in the PGM format.	512 × 512	http://agents.fel.cvut.cz/boss
BOWS-2	The images offered by the 2nd BOWS contest. It consists of 10,000 gray images in the PGM format.	512 × 512	http://bows2.ec-lille.fr/
CVG-UGR	A free online dataset. It contains several sets of uncompressed images collected from open databases.	512 × 512	https://ccia.ugr.es/cvg/dbimagenes/g512.php
MedPix	A freely accessible medical images set. It has 59,000 CT scan images.	512 × 512	https://medpix.nlm.nih.gov/home

4 THE STATES OF THE ART

In this section, the performances of the representative works of PEE [29, 40, 41, 90], MHM [27, 49, 78, 105], pairwise PEE [5, 15, 65, 68, 131], and PVO [22, 23, 46, 63, 67, 73, 112, 133] are compared and discussed.

Before conducting the comparison, we will provide a brief overview of the commonly used databases (Table 10) and the evaluation criteria (Table 11) utilized in the RDH community. Table 10 presents the descriptions, image sizes, and online links for various uncompressed image databases. They are mainly nature grayscale or color images captured by the camera. Parts of them are the medical images, which have sharp contrast black and white content. Generally, the test images used in RDH come from the databases of USC-SIPI, Kodak, Bossbase, and BOWS-2. In Table 10, the measurements for RDH and their corresponding application scenarios are summarized. There are four main fundamental criteria, i.e., the embedding capacity, the imperceptibility, the robustness, and the reversibility. The imperceptibility of RDH is usually evaluated in terms of visual quality. The most commonly used measurement is PSNR. The reversibility is evaluated based on the quality of the restored image. Usually, to ensure perfect recovery, RDH requires the transmission channel to be lossless. So, conventional RDH methods seldom consider robustness.

Table 11. Overview of the Assessment Criteria in RDH

Measurement	Description	Application	Reference
Capacity (in bits, bpp)	It evaluates the number of the hidden bits. A larger capacity means a better transfer efficiency.	The basic performance of RDH.	[49, 90]
PSNR (in dB)	This score measures the changes in pixel values caused by embedding. A higher PSNR indicates that the modified image has lower difference compared to the original one.	A commonly used measurement in the RDH.	[13, 33]
SSIM	The structural similarity (SSIM) aims to assess the visual differences in the terms of construction, brightness, and contrast, etc. It is more consistent with human visual perception.	The RDH considering HVS.	[36, 132]
RCE	The relative contrast error (RCE) is a measurement of contrast variation.	The RDH pursing the quality enhancement in the terms of the contrast.	[109, 134]
Extracted-bit error rate	It assesses the accuracy of extracting the secret message.	• Robust RDH. • RDH in encrypted images.	[75, 100]
NCC	The normalized cross correlation (NCC) compares the similarity between the primary secret message and the message read from the attacked carrier. It can be used to measure the robustness.	• Robust RDH. • RDH in encrypted images.	[50, 116]

Next, a performance comparison will be conducted based on the PSNR results for a fixed capacity.

The compared methods include the sorting-based PEE [90], the CNN-based RDH method [29], the pixel-residual PEH modification [115], the skewed histogram shifting [40], the dynamic asymmetry-based PEE [41], the conventional MHM [49], Qi et al.'s enhanced MHM [78], the **deep neural networks– (DNN)** based MHM [27], the clustering-based MHM [105], the conventional pairwise PEE [68], adaptive 2D mapping [5, 15, 131], adaptive pairing [65], the conventional PVO [46], IPVO [73], PVO-*k* [67], *k*-pass PVO [22, 23], and pairwise PVO [63, 112, 133]. Here, the test image Lena is embedded with 10,000 bits and 20,000 bits, respectively. The PSNRs and technological characteristics for the compared methods are listed in Table 12. The experimental data all come from these published works.

The results of Table 12 show that the MHM-based methods [22, 27, 49, 78, 105] and 2D mapping-based methods [5, 15, 29, 63, 65, 112, 131, 133] can generally achieve better performance. When embedding 10,000 bits, there are five outstanding methods, i.e., the CNN-based RDH method [29], the clustering-based MHM method [105], the skewed MHM-based pairwise PEE method [15], the location-based PVO [133], and the adaptive complexity-based scheme [112]. The PSNRs of all of them are larger than 61.50 dB. Among them, Reference [29] achieves the highest PSNR of 61.70 dB, which is one of the latest advancements in the RDH community. Its CNN-based predictor yields a more concentrated PEH. Besides, adaptive 2D mapping is also used to provide more effective modification. The scheme designed by Weng et al. [105] is one of the latest works for MHM, which obviously improves the performance of conventional MHM. In this method, the *k*-means clustering algorithm is used to generate multiple histograms. Compared with the conventional MHM [49], its PSNR gain is 0.49 dB. Fang et al. [15] introduced the asymmetric predictor to MHM and generated

Table 12. Comparison for the Existing Representative RDH Methods

Method	Techique	10,000 bits	20,000 bits
Sachnev et al. [90]	PEE + sorting	58.21	55.04
Hu et al. [29]	CNN predictor + 2D mapping	61.70	57.96
Xiao et al. [115]	Improved predictor and mapping model	61.40	57.77
Kim et al. [40]	Asymmetric PEE	59.92	56.67
Abolfazl et al. [41]	Asymmetric PEE	60.86	57.26
Li et al. [49]	MHM	61.02	57.55
Qi et al. [78]	MHM + Multi-parameter	61.04	57.64
Hou et al. [27]	MHM + DNN	61.46	57.65
Weng et al. [105]	MHM + Clustering	61.51	57.65
Ou et al. [68]	Pairwise PEE (2D mapping)	59.75	56.21
Chang et al. [5]	Adaptive 2D mapping	61.35	57.83
Zhang et al. [131]	Adaptive 2D mapping	60.93	57.38
Fang et al. [15]	2D mapping + Skewed MHM	61.58	58.02
Ou et al. [65]	2D mapping + Adaptive pairing	60.77	57.25
Li et al. [46]	PVO	60.34	56.21
Peng et al. [73]	IPVO	60.49	56.57
Ou et al. [67]	PVO-k	60.59	56.58
He et al. [23]	k-pass PVO	60.64	56.77
He et al. [22]	k-pass PVO + MHM	61.39	57.75
Ou et al. [63]	PVO + 2D mapping	60.91	56.69
Zhang et al. [133]	PVO + 2D mapping	61.53	57.74
Xiang et al. [112]	PVO + 2D mapping	61.57	57.69

The PSNR (in dB) is obtained when embedding 10,000 bits and 20,000 bits into the test image Lena, respectively.

20 candidate 2D mappings for adaptive embedding. As the combination of pairwise PEE and MHM, it outperforms a series of pairwise PEE-based methods [5, 65, 68, 131] and MHM-based methods [27, 49, 78, 105]. The methods in References [133] and [112] are representative works in the field of PVO. The PSNR gains of the two methods are 1.19 and 1.23 dB, respectively, compared to the conventional PVO [46]. Zhang et al. [133] proposed to generate the pixel pairs in a different manner to achieve a more uniform distribution. To establish the image-content-dependent 2D mapping, they applied an exhaustive search to find a suitable solution from the candidates obtained based on the statistical results. Xiang et al. [112] offered an enhancement in terms of block selection. The pixel blocks are allocated with the adaptive local context to calculate the texture levels.

In addition to the five algorithms mentioned above, the methods developed by Chang et al. [5], He et al. [22], and Xiao et al. [115] have also demonstrated satisfactory performance (with a PSNR greater than 61.30 dB for 10,000 bits). Chang et al. [5] fully explored the potential of pairwise PEE. More options are introduced for the 2D mapping. In Reference [22], the MHM framework is combined with the k-pass PVO method to pursue improvement. He et al. enhanced the prediction by considering the absolute positional relationship between two pixels. The multiple PEHs are dynamically generated based on the adaptive collection strategy. Xiao et al. [115] formulated a novel prediction method to mitigate the precision loss caused by round-off errors. Besides, the mapping model of MHM and pairwise PEE were adjusted by introducing a larger shifting step size.

When encountering a larger payload (20,000 bits), it becomes evident that the majority of the eight aforementioned methods still outperform others. The six methods, namely the CNN-based RDH method [29], the pixel-residual PEH modification [115], the adaptive 2D mapping [5], the

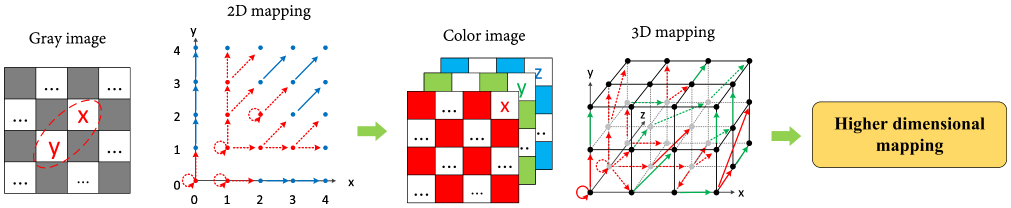


Fig. 16. Future work: High dimensional mapping.

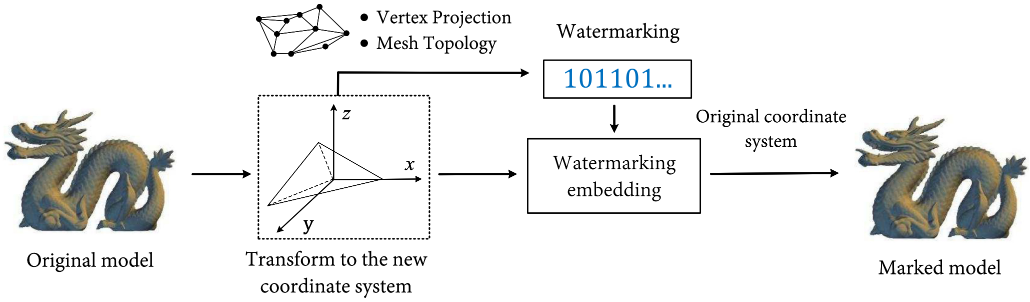


Fig. 17. Future work: RDH for the 3D mesh model.

skewed MHM-based pairwise PEE method [15], the MHM-based k-pass PVO [22], and the location-based PVO [133], can achieve a PSNR greater than 57.73 dB.

5 FUTURE WORK

For now, the RDH in uncompressed images has been thoroughly investigated. The existing framework has extensive experience in utilizing image redundancy. In our opinion, the future RDH works may mainly focus on two directions: perfecting the schematic design and exploring applied theory. For the existing frameworks, the design of the embedding steps and the optimization of parameters are still open problems. The existing strategies, such as adaptive mapping, histogram generation, and complexity calculation, could be further investigated to explore their potential. Based on the combination of various strategies, there are also numerous effective variations to be explored. Furthermore, by implementing new techniques such as deep learning algorithms, it has the potential to greatly enhance RDH and even lead to significant breakthroughs in schematic design. Next, we introduce four possible directions for future work, as shown in Figures 16–19, respectively.

(1) High dimensional mapping. This direction for future work organizes the modification mapping in higher dimensions, allowing for a more meticulous embedding. In high-dimensional space, the mapping will exhibit more variations, and the modification model can be customized for a specific carrier. The potential of adaptive embedding could be fully realized. Correspondingly, there are numerous works to be investigated for designing an effective optimization algorithm. To achieve a better balance between capacity, distortion, and time cost, it may be preferable to use powerful computing algorithms such as CNN and reinforcement learning. An example of this is the 3D mapping method proposed by Chang et al. [3].

(2) RDH for 2D vector graphics and 3D mesh models. This direction for future work is designed for commonly used multimedia data, which has new requirements for the embedding method and performance. Nowadays, 2D vector graphics and 3D mesh models are frequently used data formats in engineering design and game modeling, among other applications. They all have high precision

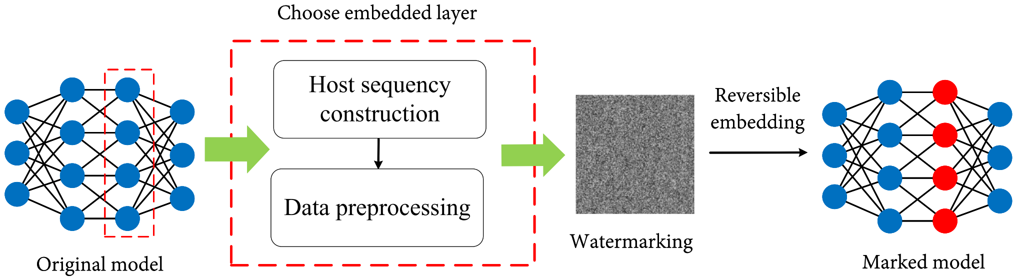


Fig. 18. Future work: RDH for DNN.

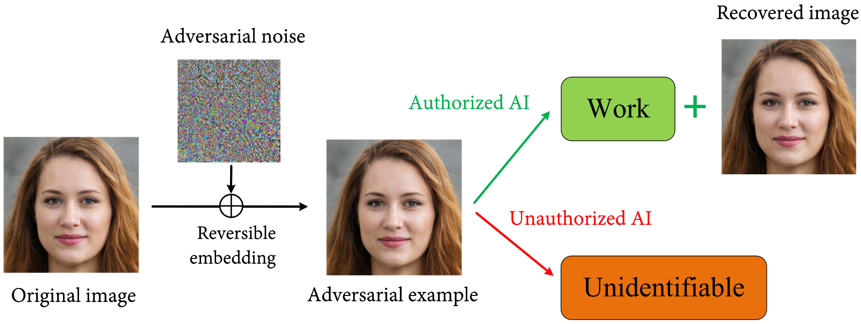


Fig. 19. Future work: Reversible adversarial example.

requirements. The RDH in 2D vector graphics and 3D mesh models should be redesigned due to their different structures compared to common raster images. For the image content, the RDH should consider the unique structure of vertices and mesh topology. The data format becomes more troublesome with floating-point numbers rather than integers. Furthermore, the 2D vector graphics and 3D mesh models used in practice are typically encrypted. So, it is necessary to investigate the RDH in the encryption domain for these purposes. Lin et al [52] designed an embedding framework for 2D engineering graphics. In their method, the region nesting algorithm is utilized to generate multiple sub-regions of the watermarking. The reversible embedding is achieved by establishing a correlation between the original vertices and the generated subspaces. Later, Peng et al. [75] further applied this algorithm to 3D mesh models and designed a semi-fragile RDH method. It is possible to learn the history of manipulation and its type based on the extracted watermarking.

(3) RDH for DNN. This direction for future work is aimed at addressing the security risks in the era of AI. Today, it is difficult to determine the owner of an AI model due to the abundance of mature models available on the Internet and the low consequences for illegal tampering. For significant outcomes, the owners may want to ensure copyright protection while also sharing their networks to enable more researchers to access and build upon their work. With RDH, the watermarking could be reversibly embedded into the DNN for integrity authentication. The marked DNN could still complete its task and maintain its original performance. At the appropriate time, the network could be reconstructed without any loss to safeguard the integrity of the network. The concept of model watermarking has been depicted by Guan et al. [17].

(4) Reversible adversarial example. Recently, Liu et al. [53] introduced the concept of reversible adversarial examples. This direction for future work aims to generate adversarial examples by reversibly embedding the corresponding perturbance into the target image. For the authorized AI model, the watermarking is extracted first to conduct authentication, and then the original

image can be restored from the received adversarial example. Finally, the authorized AI model can complete its tasks using the restored image. While other AIs may not function properly due to the adversarial effect. The reversible adversarial example can deceive unauthorized AI, thereby protecting the user's privacy. In this scene, the target image is viewed as the data that require protection, and the optimization objective shifts toward the adversarial effect.

6 SUMMARY

RDH is a special technique used for ensuring the security of messages. Its advantage in image recovery enables RDH to achieve several outcomes in sensitive applications. This survey article reviews the classic frameworks and mainstream adaptive techniques for the RDH in uncompressed images. These techniques include the compression-based method, DE, IT, HS, PEE, MHM, pairwise PEE, and PVO. The typical algorithms and methodologies of the eight subjects mentioned above are presented, analyzed, and discussed. Furthermore, in this article, we present the performances of the state-of-the-art RDH algorithms and analyze the advantages of the three typical techniques. Considering the new requirements and application scenarios, the reformulation of applied theory and embedding framework of RDH is an inevitable trend. Four possible directions for future research based on the early-age motivations are given and discussed.

REFERENCES

[1] Adnan M. Alattar. 2004. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. Image Process.* 13, 8 (2004), 1147–1156.

[2] Mehmet Utku Celik, Gaurav Sharma, Ahmet Murat Tekalp, and Eli Saber. 2005. Lossless generalized-LSB data embedding. *IEEE Trans. Image Process.* 14, 2 (2005), 253–266.

[3] Jie Chang, Guopu Zhu, Hongli Zhang, Yicong Zhou, Xiangyang Luo, and Ligang Wu. 2022. Reversible data hiding for color images based on adaptive 3D prediction-error expansion and double deep q-network. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 8 (2022), 5055–5067.

[4] Qi Chang, Xiaolong Li, and Yao Zhao. 2022. Reversible data hiding for color images based on adaptive three-dimensional histogram modification. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 9 (2022), 5725–5735.

[5] Qi Chang, Xiaolong Li, Yao Zhao, and Rongrong Ni. 2021. Adaptive pairwise prediction-error expansion and multiple histograms modification for reversible data hiding. *IEEE Trans. Circ. Syst. Vid. Technol.* 31, 12 (2021), 4850–4863.

[6] Kaimeng Chen and Chin-Chen Chang. 2019. High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement. *J. Vis. Commun. Image Represent.* 58 (2019), 334–344.

[7] Xianyi Chen, Xingming Sun, Huiyu Sun, Zhili Zhou, and Jianjun Zhang. 2013. Reversible watermarking method based on asymmetric-histogram shifting of prediction errors. *J. Syst. Soft.* 86, 10 (2013), 2620–2626.

[8] Dinu Coltuc. 2011. Improved embedding for prediction-based reversible watermarking. *IEEE Trans. Inf. Forens. Secur.* 6, 3 (2011), 873–882.

[9] Dinu Coltuc and Jean-Marc Chassery. 2007. Very fast watermarking by reversible contrast mapping. *IEEE Sign. Process. Lett.* 14, 4 (2007), 255–258.

[10] Ioan-Catalin Dragoi and Dinu Coltuc. 2014. Local-prediction-based difference expansion reversible watermarking. *IEEE Trans. Image Process.* 23, 4 (2014), 1779–1790.

[11] Ioan-Catalin Dragoi and Dinu Coltuc. 2015. On local prediction based reversible watermarking. *IEEE Trans. Image Process.* 24, 4 (2015), 1244–1246.

[12] Ioan-Catalin Dragoi and Dinu Coltuc. 2016. Adaptive pairing reversible watermarking. *IEEE Trans. Inf. Forens. Secur.* 25, 5 (2016), 2420–2422.

[13] Yang Du, Zhaoxia Yin, and Xinpeng Zhang. 2022. High capacity lossless data hiding in JPEG bitstream based on general VLC mapping. *IEEE Trans. Depend. Secure Comput.* 19, 2 (2022), 1420–1433.

[14] Mehdi Fallahpour. 2008. Reversible image data hiding based on gradient adjusted prediction. *EICE Electr. Exp.* 5 (2008), 870–876.

[15] Guojun Fang, Zhibin Pan, Zhou Quan, Gao Xinyi, and Zhang Xiaoran. 2021. Multiple histogram based adaptive pairwise prediction-error modification for efficient reversible image watermarking. *Inf. Sci.* 581 (2021), 515–535.

[16] Jiri Fridrich, Miroslav Goljan, and Rui Du. 2001. Invertible authentication. In *Proceedings of the SPIE: Security and Watermarking of Multimedia Contents III,* Vol. 4314 (SPIE, Bellingham, WA, 2001), 197–208.

[17] Xiquan Guan, Huamin Feng, Weiming Zhang, Hang Zhou, Jie Zhang, and Nenghai Yu. 2020. Reversible watermarking in deep convolutional neural networks for integrity authentication. In *Proceedings of the 28th ACM International Conference on Multimedia (MM'20).* Association for Computing Machinery, New York, NY, 2273–2280.

[18] Xinlu Gui, Xiaolong Li, and Bin Yang. 2014. A high capacity reversible data hiding scheme based on generalized prediction-error expansion and adaptive embedding. *Sign. Process.* 98 (2014), 370–380.

[19] Junhui He, Junxi Chen, Weiqi Luo, Shaohua Tang, and Jiwu Huang. 2019. A novel high-capacity reversible data hiding scheme for encrypted JPEG bitstreams. *IEEE Trans. Circ. Syst. Vid. Technol.* 29, 12 (2019), 3501–3515.

[20] Junhui He, Junxi Chen, and Shaohua Tang. 2020. Reversible data hiding in JPEG images based on negative influence models. *IEEE Trans. Inf. Forens. Secur.* 15 (2020), 2121–2133.

[21] Wenguang He and Zhanchuan Cai. 2021. Reversible data hiding based on dual pairwise prediction-error expansion. *IEEE Trans. Image Process.* 30 (2021), 5045–5055.

[22] Wenguang He, Gangqiang Xiong, and Yaomin Wang. 2021. Reversible data hiding based on adaptive multiple histograms modification. *IEEE Trans. Infor. Forens. Secur.* 16 (2021), 3000–3012.

[23] Wenguang He, Ke Zhou, Jie Cai, Long Wang, and Gangqiang Xiong. 2017. Reversible data hiding using multi-pass pixel value ordering and prediction-error expansion. *J. Vis. Commun. Image Represent.* 49 (2017), 351–360.

[24] Wien Hong, Gwoboa Horng, Chih-Wei Shiu, Tung-Shou Chen, and Yu-Chi Chen. 2015. Reversible steganographic method using complexity control and human visual system. *Comp. J.* 58 (2015), 2583–2594.

[25] Dongdong Hou, Haoqian Wang, Weiming Zhang, and Nenghai Yu. 2018. Reversible data hiding in JPEG image based on DCT frequency and block selection. *Sign. Process.* 148 (Jul. 2018), 41–47.

[26] Dongdong Hou, Weiming Zhang, Kejiang Chen, Sian-Jheng Lin, and Nenghai Yu. 2019. Reversible data hiding in color image with grayscale invariance. *IEEE Trans. Circ. Syst. Vid. Technol.* 29, 2 (2019), 363–374.

[27] Jiacheng Hou, Bo Ou, Huawei Tian, and Zheng Qin. 2021. Reversible data hiding based on multiple histograms modification and deep neural networks. *Sign. Process. Image Commun.* 92 (2021), 116118.

[28] Runwen Hu and Shijun Xiang. 2021. CNN prediction based reversible data hiding. *IEEE Sign. Process. Lett.* 28 (2021), 464–468.

[29] Runwen Hu and Shijun Xiang. 2022. Reversible data hiding by using CNN prediction and adaptive embedding. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 12 (2022), 10196–10208.

[30] Yongjian Hu, Heung-Kyu Lee, and Jianwei Li. 2009. DE-based reversible data hiding with improved overflow location map. *IEEE Trans. Circ. Syst. Video Technol.* 19, 2 (2009), 250–260.

[31] Yongjian Hu, Kan Wang, and Zhe-Ming Lu. 2013. An improved VLC-based lossless data hiding scheme for JPEG images. *J. Syst. Softw.* 86, 8 (2013), 2166–2173.

[32] Fangjun Huang, Xiaochao Qu, Hyoung Joong Kim, and Jiwu Huang. 2016. Reversible data hiding in JPEG images. *IEEE Trans. Circ. Syst. Vid. Technol.* 26, 9 (2016), 1610–1621.

[33] Chun-Liang Jhong and Hsin-Lung Wu. 2022. Grayscale-invariant reversible data hiding based on multiple histograms modification. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 9 (2022), 5888–5901.

[34] Yujie Jia, Zhaoxia Yin, Xinpeng Zhang, and Yonglong Luo. 2019. Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting. *Sign. Process.* 163 (2019), 238–246.

[35] Ruiqi Jiang, Hang Zhou, Weiming Zhang, and Nenghai Yu. 2018. Reversible data hiding in encrypted three-dimensional mesh models. *IEEE Trans. Multimedia* 20, 1 (2018), 55–67.

[36] Seung-Won Jung, Le Thanh Ha, and Sung-Jea Ko. 2011. A new histogram modification based reversible data hiding algorithm considering the human visual system. *IEEE Sign. Process. Lett.* 18 (2011), 95–98.

[37] Yan Ke, Minqing Zhang, Xinpeng Zhang, Jia Liu, Tingting Su, and Xiaoyuan Yang. 2022. A reversible data hiding scheme in encrypted domain for secret image sharing based on chinese remainder theorem. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 4 (2022), 2469–2481.

[38] Hyoung Joong Kim, Vasiliy Sachnev, Yun-Qing Shi, Jeho Nam, and Hyon-Gon Choo. 2008. A novel difference expansion transform for reversible data embedding. *IEEE Trans. Inf. Forens. Secur.* 3, 3 (2008), 456–465.

[39] Suah Kim, Rolf Lussi, Xiaochao Qu, Fangjun Huang, and Hyoung Joong Kim. 2019. Reversible data hiding with automatic brightness preserving contrast enhancement. *IEEE Trans. Circ. Syst. Vid. Technol.* 29, 8 (2019), 2271–2284.

[40] Suah Kim, Xiaochao Qu, Vasily Sachnev, and Hyoung Joong Kim. 2019. Skewed histogram shifting for reversible data hiding using a pair of extreme predictions. *IEEE Trans. Circ. Syst. Vid. Technol.* 29, 11 (2019), 3236–3246.

[41] Abolfazl Kouhi and Mohammad Hossein Sedaaghi. 2021. Prediction error distribution with dynamic asymmetry for reversible data hiding. *Expert Syst. Appl.* 184 (2021), 115475.

[42] Sang-kwang Lee, Young-ho Suh, and Yo-sung Ho. 2006. Reversible image authentication based on watermarking. In *Proceedings of the IEEE International Conference on Multimedia and Expo.* IEEE, 1321–1324.

[43] Jingxuan Li and Shijun Xiang. 2022. Audio-lossless robust watermarking against desynchronization attacks. *Sign. Process.* 198 (2022), 108561.

[44] Wenjie Li, Xiaolong Li, Rongrong Ni, and Yao Zhao. 2021. PVO-based reversible data hiding using adaptive multiple histogram generation and modification. *Sign. Process. Image Commun.* 99 (2021), 116405.

[45] Xiaolong Li, Bin Li, Bin Yang, and Tieyong Zeng. 2013. General framework to histogram-shifting-based reversible data hiding. *IEEE Trans. Image Process.* 22, 6 (2013), 2181–2191.

[46] Xiaolong Li, Jian Li, Bin Li, and Bin Yang. 2013. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Sign. Process.* 93, 1 (2013), 198–205.

[47] Xiaolong Li, Bin Yang, and Tieyong Zeng. 2011. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Trans. Image Process.* 20, 12 (2011), 3524–3533.

[48] Xiaolong Li, Weiming Zhang, Xinlu Gui, and Bin Yang. 2013. A novel reversible data hiding scheme based on two-dimensional difference-histogram modification. *IEEE Trans. Inf. Forens. Secur.* 8, 7 (2013), 1091–1100.

[49] Xiaolong Li, Weiming Zhang, Xinlu Gui, and Bin Yang. 2015. Efficient reversible data hiding based on multiple histograms modification. *IEEE Trans. Inf. Forens. Secur.* 10, 9 (Sep. 2015), 2016–2027.

[50] Xingyun Liang and Shijun Xiang. 2020. Robust reversible audio watermarking based on high-order difference statistics. *Sign. Process.* 173 (2020), 107584.

[51] Chia-Chen Lin, Wei-Liang Tai, and Chin-Chen Chang. 2008. Multilevel reversible data hiding based on histogram modification of difference images. *Pattern Recogn.* 41, 12 (2008), 3582–3591.

[52] Zi-Xing Lin, Fei Peng, and Min Long. 2018. A low-distortion reversible watermarking for 2D engineering graphics based on region nesting. *IEEE Trans. Inf. Forens. Secur.* 13, 9 (2018), 2372–2382.

[53] Jiayang Liu, Weiming Zhang, Kazuto Fukuchi, Youhei Akimoto, and Jun Sakuma. 2023. Unauthorized AI cannot recognize me: Reversible adversarial example. *Pattern Recogn.* 134 (2023), 109048.

[54] Min Long, Fei Peng, and Han-Yun Li. 2018. Separable reversible data hiding and encryption for HEVC video. *J. Real-Time Image Process.* 14 (2018), 171–182.

[55] Ting Luo, Gangyi Jiang, Mei Yu, Caiming Zhong, Haiyong Xu, and Zhiyong Pan. 2019. Convolutional neural networks-based stereo image reversible data hiding method. *J. Vis. Commun. Image Represent.* 61 (2019), 61–73.

[56] Wan-Li Lyu, Lulu Cheng, and Zhaoxia Yin. 2022. High-capacity reversible data hiding in encrypted 3D mesh models based on multi-MSB prediction. *Sign. Process.* 201 (2022), 108686.

[57] Bin Ma and Yun-Qing Shi. 2016. A reversible data hiding scheme based on code division multiplexing. *IEEE Trans. Inf. Forens. Secur.* 11, 9 (2016), 1914–1927.

[58] Bin Ma, Xiaoyu Wang, Qi Li, Bing Li, Jian Li, Chunpeng Wang, and Yun-Qing Shi. 2019. Adaptive error prediction method based on multiple linear regression for reversible data hiding. *J. Real-Time Image Process.* 16 (2019), 821–834.

[59] Kede Ma, Weiming Zhang, Xianfeng Zhao, Nenghai Yu, and Fenghua Li. 2013. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Trans. Inf. Forens. Secur.* 8, 3 (2013), 553–562.

[60] Shimei Ma, Xiaolong Li, Mengyao Xiao, Bin Ma, and Yao Zhao. 2022. Fast expansion-bins-determination for multiple histograms modification based reversible data hiding. *IEEE Sign. Process. Lett.* 29 (2022), 662–666.

[61] Ningxiong Mao, Fan Chen, Hongjie He, and Yaolin Yang. 2022. Reversible data hiding based on adaptive IPVO and two-segment pairwise PEE. *Sign. Process.* 198 (2022), 108577.

[62] Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, and Wei Su. 2006. Reversible data hiding. *IEEE Trans. Circ. Syst. Vid. Technol.* 16, 3 (2006), 354–362.

[63] Bo Ou, Xiaolong Li, and Jinwei Wang. 2016. High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion. *J. Vis. Commun. Image Represent.* 39 (2016), 12–23.

[64] Bo Ou, Xiaolong Li, and Jinwei Wang. 2016. Improved PVO-based reversible data hiding: A new implementation based on multiple histograms modification. *J. Vis. Commun. Image Represent.* 38 (2016), 328–339.

[65] Bo Ou, Xiaolong Li, Weiming Zhang, and Yao Zhao. 2019. Improving pairwise PEE via hybrid-dimensional histogram generation and adaptive mapping selection. *IEEE Trans. Circ. Syst. Vid. Technol.* 29, 7 (2019), 2176–2190.

[66] Bo Ou, Xiaolong Li, Yao Zhao, and Rongrong Ni. 2013. Reversible data hiding based on PDE predictor. *J. Syst. Softw.* 86, 10 (2013), 2700–2709.

[67] Bo Ou, Xiaolong Li, Yao Zhao, and Rongrong Ni. 2014. Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. *Sign. Process. Image Commun.* 29, 7 (2014), 760–772.

[68] Bo Ou, Xiaolong Li, Yao Zhao, Rongrong Ni, and Yun-Qing Shi. 2013. Pairwise prediction-error expansion for efficient reversible data hiding. *IEEE Trans. Image Process.* 22 (Dec. 2013), 5010–5021.

[69] Bo Ou and Yao Zhao. 2020. High capacity reversible data hiding based on multiple histograms modification. *IEEE Trans. Circ. Syst. Video Technol.* 30, 8 (2020), 2329–2342.

[70] Zhibin Pan, Xinyi Gao, Erdun Gao, and Guojun Fan. 2020. Adaptive complexity for pixel-value-ordering based reversible data hiding. *IEEE Sign. Process. Lett.* 27 (2020), 915–919.

[71] Fei Peng, Wen-Yan Jiang, Ying Qi, Zi-Xing Lin, and Min Long. 2020. Separable robust reversible watermarking in encrypted 2D vector graphics. *IEEE Trans. Circ. Syst. Vid. Technol.* 30, 8 (2020), 2391–2405.

[72] Fei Peng, Xiaolong Li, and Bin Yang. 2012. Adaptive reversible data hiding scheme based on integer transform. *Sign. Process.* 92, 1 (2012), 54–62.

[73] Fei Peng, Xiaolong Li, and Bin Yang. 2014. Improved PVO-based reversible data hiding. *Digit. Sign. Process.* 25 (2014), 255–265.

[74] Fei Peng, Tongxin Liao, and Min Long. 2022. A semi-fragile reversible watermarking for authenticating 3D models in dual domains based on variable direction double modulation. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 12 (2022), 8394–8408.

[75] Fei Peng, Bo Long, and Min Long. 2021. A general region nesting-based semi-fragile reversible watermarking for authenticating 3D mesh models. *IEEE Trans. Circ. Syst. Vid. Technol.* 31, 11 (2021), 4538–4553.

[76] Pauline Puteaux and William Puech. 2021. A recursive reversible data hiding in encrypted images method with a very high payload. *IEEE Trans. Multimedia* 23 (2021), 636–650.

[77] Wenfa Qi, Sirui Guo, and Wei Hu. 2022. Generic reversible visible watermarking via regularized graph fourier transform coding. *IEEE Trans. Image Process.* 31 (2022), 691–705.

[78] Wenfa Qi, Xiaolong Li, Tong Zhang, and Zongming Guo. 2020. Optimal reversible data hiding scheme based on multiple histograms modification. *IEEE Trans. Circ. Syst. Vid. Technol.* 30, 8 (2020), 2300–2312.

[79] Zhenxing Qian, Haisheng Xu, Xiangyang Luo, and Xinpeng Zhang. 2019. New framework of reversible data hiding in encrypted JPEG bitstreams. *IEEE Trans. Circ. Syst. Vid. Technol.* 29, 2 (2019), 351–362.

[80] Zhenxing Qian and Xinpeng Zhang. 2012. Lossless data hiding in JPEG bitstream. *J. Syst. Softw.* 85, 2 (2012), 309–313.

[81] Zhenxing Qian, Xinpeng Zhang, and Shuozhong Wang. 2014. Reversible data hiding in encrypted JPEG bitstream. *IEEE Trans. Multimedia* 16, 5 (2014), 1486–1491.

[82] Chuan Qin, Zhihong He, Heng Yao, Fang Cao, and Liping Gao. 2018. Visible watermark removal scheme based on reversible data hiding and image inpainting. *Sign. Process. Image Commun.* 60 (2018), 160–172.

[83] Chuan Qin, Xiaokan Qian, and Xinpeng Zhang. 2019. An efficient coding scheme for reversible data hiding in encrypted image with redundancy transfer. *Inf. Sci.* 487 (2019), 176–192.

[84] Jianqiang Qin and Fangjun Huang. 2019. Reversible data hiding based on multiple two-dimensional histograms modification. *IEEE Sign. Process. Lett.* 26, 6 (2019), 843–847.

[85] Yingqiang Qiu, Zhenxing Qian, Han He, Hui Tian, and Xinpeng Zhang. 2021. Optimized lossless data hiding in JPEG bitstream and relay transfer-based extension. *IEEE Trans. Circ. Syst. Vid. Technol.* 31, 4 (2021), 1380–1394.

[86] Yingqiang Qiu, Zhenxing Qian, and Lun Yu. 2016. Adaptive reversible data hiding by extending the generalized integer transformation. *IEEE Sign. Process. Lett.* 23, 1 (2016), 130–134.

[87] Yingqiang Qiu, Qichao Ying, Yuyan Yang, Huanqiang Zeng, Sheng Li, and Zhenxing Qian. 2022. High-capacity framework for reversible data hiding in encrypted image using pixel prediction and entropy encoding. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 9 (2022), 5874–5887.

[88] Xiaochao Qu and Hyoung Joong Kim. 2015. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Sign. Process.* 111 (2015), 249–260.

[89] Reza Moradi Rad, KokSheik Wong, and Jing-Ming Guo. 2016. Reversible data hiding by adaptive group modification on histogram of prediction errors. *Sign. Process.* 125 (2016), 315–328.

[90] Vasiliy Sachnev, Hyoung Joong Kim, Jeho Nam, Sundaram Suresh, and Yun-Qing Shi. 2009. Reversible watermarking algorithm using sorting and prediction. *IEEE Trans. Circ. Syst. Vid. Technol.* 19, 7 (2009), 989–999.

[91] Yun-Qing Shi, Xiaolong Li, Xinpeng Zhang, Hao-Tian Wu, and Bin Ma. 2016. Reversible data hiding: Advances in the past two decades. *IEEE Access* 4 (2016), 3210–3237.

[92] Zhiyong Su, Ying Ye, Qi Zhang, Weiqing Li, and Yuewei Dai. 2018. Robust 2D engineering CAD graphics hashing for joint topology and geometry authentication via covariance-based descriptors. *IEEE Trans. Inf. Forens. Secur.* 13, 4 (2018), 1018–1030.

[93] Xiaoxu Tang, Hongxia Wang, and Yi Chen. 2020. Reversible data hiding based on a modified difference expansion for H.264/AVC video streams. *Multimedia Tools Appl.* 79 (2020), 28661–28674.

[94] Diljith M. Thodi and Jeffrey J. Rodrguez. 2007. Expansion embedding techniques for reversible watermarking. *IEEE Trans. Image Process.* 16, 3 (2007), 721–730.

[95] Jun Tian. 2003. Reversible data embedding using a difference expansion. *IEEE Trans. Circ. Syst. Vid. Technol.* 13, 8 (2003), 890–896.

[96] Junxiang Wang, Xin Chen, Jiangqun Ni, Ningxiong Mao, and Yun-Qing Shi. 2020. Multiple histograms-based reversible data hiding: Framework and realization. *IEEE Trans. Circ. Syst. Vid. Technol.* 30, 8 (2020), 2313–2328.

[97] Junxiang Wang, Ningxiong Mao, Xin Chen, Jiangqun Ni, Chuntao Wang, and Yun-Qing Shi. 2019. Multiple histograms based reversible data hiding by using FCM clustering. *Sign. Process.* 159 (2019), 193–203.

[98] Junxiang Wang, Jiangqun Ni, Xing Zhang, and Yun-Qing Shi. 2017. Rate and distortion optimization for reversible data hiding using multiple histogram shifting. *IEEE Trans. Cyber.* 47, 2 (2017), 315–326.

[99] Xiang Wang, Jing Ding, and Qingqi Pei. 2015. A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition. *Inf. Sci.* 310 (2015), 16–35.

[100] Xiang Wang, Xiaolong Li, and Qingqi Pei. 2020. Independent embedding domain based two-stage robust reversible watermarking. *IEEE Trans. Circ. Syst. Vid. Technol.* 30, 8 (2020), 2406–2417.

[101] Xiang Wang, Xiaolong Li, Bin Yang, and Zongming Guo. 2010. Efficient generalized integer transform for reversible watermarking. *IEEE Sign. Process. Lett.* 17, 6 (2010), 567–570.

[102] Xiaoyu Wang, Xingyuan Wang, Bin Ma, Qi Li, and Yun-Qing Shi. 2021. High precision error prediction algorithm based on ridge regression predictor for reversible data hiding. *IEEE Sign. Process. Lett.* 28 (2021), 1125–1129.

[103] Shaowei Weng, Tanshuai Hou, Tiancong Zhang, and Jeng-Shyang Pan. 2023. Adaptive smoothness evaluation and multiple asymmetric histogram modification for reversible data hiding. *J. Vis. Commun. Image Represent.* 90 (2023), 103732.

[104] Shaowei Weng, Yun-Qing Shi, Wien Hong, and Ye Yao. 2019. Dynamic improved pixel value ordering reversible data hiding. *Inf. Sci.* 489 (2019), 136–154.

[105] Shaowei Weng, Wenlong Tan, Bo Ou, and Jeng-Shyang Pan. 2021. Reversible data hiding method for multi-histogram point selection based on improved crisscross optimization algorithm. *Inf. Sci.* 549 (Mar. 2021), 13–33.

[106] Shaowei Weng, Guohao Zhang, Jeng-Shyang Pan, and Zhili Zhou. 2017. Optimal PPVO-based reversible data hiding. *J. Vis. Commun. Image Represent.* 48 (2017), 317–328.

[107] Shaowei Weng and Tiancong Zhang. 2022. Adaptive reversible data hiding for JPEG images with multiple two-dimensional histograms. *J. Vis. Commun. Image Represent.* 85 (2022), 103487.

[108] Haorui Wu, Xiaolong Li, Xiangyang Luo, Xinpeng Zhang, and Yao Zhao. 2022. General expansion-shifting model for reversible data hiding: Theoretical investigation and practical algorithm design. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 9 (2022), 5989–6001.

[109] Hao-Tian Wu, Xin Cao, Ruoyan Jia, and Yiu-ming Cheung. 2022. Reversible data hiding with brightness preserving contrast enhancement by two-dimensional histogram modification. *IEEE Trans. Circ. Syst. Video Technol.* 32, 11 (2022), 7605–7617.

[110] Hao-Tian Wu, Jean-Luc Dugelay, and Yun-Qing Shi. 2015. Reversible image data hiding with contrast enhancement. *IEEE Sign. Process. Lett.* 22, 1 (2015), 81–85.

[111] Youqing Wu, Youzhi Xiang, Yutang Guo, Jin Tang, and Zhaoxia Yin. 2020. An improved reversible data hiding in encrypted images using parametric binary tree labeling. *IEEE Trans. Multimedia* 22, 8 (2020), 1929–1938.

[112] Shijun Xiang and Guanqi Ruan. 2022. Efficient PVO-based reversible data hiding by selecting blocks with full-enclosing context. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 5 (2022), 2868–2880.

[113] Mengyao Xiao, Xiaolong Li, Bin Ma, Xinpeng Zhang, and Yao Zhao. 2021. Efficient reversible data hiding for JPEG images with multiple histograms modification. *IEEE Trans. Circ. Syst. Vid. Technol.* 31, 7 (2021), 2535–2546.

[114] Mengyao Xiao, Xiaolong Li, Yangyang Wang, Yao Zhao, and Rongrong Ni. 2019. Reversible data hiding based on pairwise embedding and optimal expansion path. *Sign. Process.* 158 (2019), 210–218.

[115] Mengyao Xiao, Xiaolong Li, Yao Zhao, Bin Ma, and Guodong Guo. 2023. A novel reversible data hiding scheme based on pixel-residual histogram. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 1s (2023), 1–19.

[116] Lizhi Xiong, Xiao Han, Ching-Nung Yang, and Yun-Qing Shi. 2022. Robust reversible watermarking in encrypted image with secure multi-party based on lightweight cryptography. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 1 (2022), 75–91.

[117] Lizhi Xiong, Xiao Han, Ching-Nung Yang, and Zhihua Xia. 2023. RDH-DES: Reversible data hiding over distributed encrypted-image servers based on secret sharing. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 1 (2023), 1–19.

[118] Dawen Xu and Yong Liu. 2022. Reversible data hiding in H.264/AVC videos based on hybrid-dimensional histogram modification. *Multimedia Tools Appl.* 81 (2022), 29305–29319.

[119] Dawen Xu and Rangding Wang. 2016. Two-dimensional reversible data hiding-based approach for intra-frame error concealment in H.264/AVC. *Sign. Process. Image Commun.* 47 (2016), 369–379.

[120] Xie Yang and Fangjun Huang. 2022. New CNN-based predictor for reversible data hiding. *IEEE Sign. Process. Lett.* 29 (2022), 2627–2631.

[121] Yang Yang, Weiming Zhang, Dong Liang, and Nenghai Yu. 2016. Reversible data hiding in medical images with enhanced contrast in texture area. *Digital Sign. Process.* 52 (2016), 13–24.

[122] Yang Yang, Tianrui Zou, Genyan Huang, and Weiming Zhang. 2022. A high visual quality color image reversible data hiding scheme based on B-R-G embedding principle and CIEDE2000 assessment metric. *IEEE Trans. Circ. Syst. Video Technol.* 32, 4 (2022), 1860–1874.

[123] Yuanzhi Yao, Weiming Zhang, Hui Wang, Hang Zhou, and Nenghai Yu. 2019. Content-adaptive reversible visible watermarking in encrypted images. *Sign. Process.* 164 (2019), 386–401.

[124] Yuanzhi Yao, Weiming Zhang, and Nenghai Yu. 2016. Inter-frame distortion drift analysis for reversible data hiding in encrypted H.264/AVC video bitstreams. *Sign. Process.* 128 (2016), 531–545.

[125] Shuang Yi and Yicong Zhou. 2019. Separable and reversible data hiding in encrypted images using parametric binary tree labeling. *IEEE Trans. Multimedia* 21, 1 (2019), 51–64.

[126] Zhaoxia Yin, Yuan Ji, and Bin Luo. 2020. Reversible data hiding in JPEG images with multi-objective optimization. *IEEE Trans. Circ. Syst. Vid. Technol.* 30, 8 (2020), 2343–2352.

[127] Zhaoxia Yin, Yinyin Peng, and Youzhi Xiang. 2022. Reversible data hiding in encrypted images based on pixel prediction and bit-plane compression. *IEEE Trans. Depend. Secure Comput.* 19, 2 (2022), 992–1002.

[128] Zhaoxia Yin, Youzhi Xiang, and Xinpeng Zhang. 2020. Reversible data hiding in encrypted images based on multi-MSB prediction and huffman coding. *IEEE Trans. Multimedia* 22, 4 (2020), 874–884.

[129] Chunqiang Yu, Xianquan Zhang, Guoxiang Li, Shanhua Zhan, and Zhenjun Tang. 2022. Reversible data hiding with adaptive difference recovery for encrypted images. *Inf. Sci.* 584 (2022), 89–110.

[130] Chunqiang Yu, Xianquan Zhang, Dewang Wang, and Zhenjun Tang. 2022. Reversible data hiding with pairwise PEE and 2D-PEH decomposition. *Sign. Process.* 196 (2022), 108527.

[131] Cheng Zhang and Bo Ou. 2022. Reversible data hiding based on multiple adaptive two-dimensional prediction-error histograms modification. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 7 (2022), 4174–4187.

[132] Cheng Zhang, Bo Ou, Xiaolong Li, and Jianqin Xiong. 2023. Human visual system guided reversible data hiding based on multiple histograms modification. *Comp. J.* 66, 4 (2023), 888–906.

[133] Tong Zhang, Xiaolong Li, Wenfa Qi, and Zongming Guo. 2020. Location-based PVO and adaptive pairwise modification for efficient reversible data hiding. *IEEE Trans. Inf. Forens. Secur.* 15 (2020), 2306–2319.

[134] Tian-Cong Zhang, Tan-Shuai Hou, Shao-Wei Weng, Fu-Min Zou, Hong-Chao Zhang, and Chin-Chen Chang. 2022. Adaptive reversible data hiding with contrast enhancement based on multi-histogram modification. *IEEE Trans. Circ. Syst. Vid. Technol.* 32, 8 (2022), 5041–5054.

[135] Weiming Zhang, Xiaocheng Hu, Xiaolong Li, and Nenghai Yu. 2013. Recursive histogram modification: Establishing equivalency between reversible data hiding and lossless data compression. *IEEE Trans. Image Process.* 22, 7 (2013), 2775–2785.

[136] Xinpeng Zhang. 2011. Reversible data hiding in encrypted image. *IEEE Sign. Process. Lett.* 18, 4 (2011), 255–258.

[137] Xinpeng Zhang. 2012. Separable reversible data hiding in encrypted image. *IEEE Trans. Inf. Forens. Secur.* 7, 2 (2012), 826–832.

[138] Xinpeng Zhang. 2013. Reversible data hiding with optimal value transfer. *IEEE Trans. Multimedia* 15, 2 (2013), 316–325.

[139] Xiaoya Zhang, Yuanzhi Yao, and Nenghai Yu. 2021. Convolutional neural network-driven optimal prediction for image reversible data hiding. In *Proceedings of the IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP'21)*. IEEE, 1–6.

[140] Hongchang Zheng, Chuntao Wang, Junxiang Wang, and Shijun Xiang. 2019. A new reversible watermarking scheme using the content-adaptive block size for prediction. *Sign. Process.* 164 (2019), 74–83.

Received 6 February 2023; revised 17 December 2023; accepted 31 January 2024