Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

A reversible watermarking algorithm based on complementary embedding of diamond pixel pairs

Zhengwei Zhang*, Fenfen Li, Shenghua Jin

Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an, China

ARTICLE INFO

Article history:
Received 7 July 2025
Received in revised form
3 November 2025
Accepted 9 November 2025

Keywords:
Reversible watermarking
Complementary embedding
Diamond pixel pairs
Histogram shifting
Image distortion reduction

ABSTRACT

To improve both invisibility and embedding capacity, this study proposes a reversible watermarking algorithm based on diamond pixel pairs with complementary embedding. To avoid pixel value overflow, the carrier image is first preprocessed and divided into blocks. Each block is then separated into two half-planes, gray and white. The watermark is embedded in both horizontal and vertical directions. In the horizontal direction, the target pixel in a smooth block is compared with four reference pixels in its cross-shaped neighborhood to calculate their difference values and construct a histogram. The first embedding is carried out in the gray half-plane using the histogram shifting method, followed by a second embedding in the white half-plane. The same process is applied in the vertical direction to reduce the distortion introduced by horizontal embedding. Experimental results indicate that, under the same embedding capacity, the proposed method achieves an average PSNR (Peak Signal-to-Noise Ratio) that is 6.92 dB higher than that of the non-complementary embedding scheme. Moreover, compared with other related algorithms, this approach demonstrates superior embedding performance for the same PSNR value.

1. Introduction

With the rapid development of Internet technology, reversible watermarking has become a hot spot in information security (Lee et al., 2019; Zhang et al., 2020; Cao et al., 2020), especially for some fields with high precision requirements, such as medical, military, and chemical fields. After years of continuous development, many reversible watermarking algorithms have emerged. At present, large capacity and visual quality are one of the main research targets in reversible watermarking algorithms (Wang et al., 2024).

Tian (2003) proposed the DE (difference expansion) algorithm, which selects adjacent pixels to compute differences for watermark embedding. However, it easily causes pixel value overflow, degrading image quality. Based on Tian's algorithm, Alattar (2004) selected a group of adjacent pixels for watermark embedding. It fully utilizes the redundancy between adjacent pixels, which significantly increases the embedding capacity. But its modification of pixel differences is large, which may affect the image quality. Ni et al. (2008) proposed an HS (histogram shifting) with minimal pixel modification (≤ 1) for high visual quality but limited capacity. Subsequent advancements include: Yang and Tsai's (2010) interleaved prediction to enhance histogram peaks; Tai et al.'s (2009) binary tree structure for low-distortion embedding; Li et al.'s (2010) pixel difference sequence histogram; and Lee and Chen's (2012) strategy for adjustable prediction that eliminates the need for location maps. These methods all shared the common goal of increasing embedding capacity while maintaining visual quality.

To reduce image distortion, Arham and Nugroho (2024) introduced a novel method for reversible data hiding based on a combination of DE and a modulus function. This method enables the embedding of 3-bit data into 2-bit Least Significant Bit (LSB) difference values of pixel pairs formed in rectangular blocks. Zhang et al. (2023) applied reverse histogram translation to improve both capacity and visual quality, while Jia et al. (2019) reduced invalid pixel shifts during histogram shifting to enhance efficiency. Based on image texture, the algorithm reduces the invalid shift of pixels in the histogram shift. It has a high embedded rate and visual quality. Das and Sunaniya (2022) proposed a

* Corresponding Author.
Email Address: zzw49010650@sina.com (Z. Zhang)
https://doi.org/10.21833/ijaas.2025.12.002
Corresponding author's ORCID profile:
https://orcid.org/0000-0002-3207-0586

hybrid algorithm using improved ENMM (Enhanced Normalized Mean Modification) to reduce inter-pixel differences, thereby improving visual quality. Krishnamoorthi and Chinnababu (2022) employed a four-level watermarking technique to expand the payload while minimizing distortion. Zheng et al. (2019) proposed a new reversible watermarking scheme based on content-adaptive block size prediction. By using the content-adaptive block size of each predicted pixel, the prediction error is minimized and close to the difference between the original value and the predicted value, thus improving the visual quality of the image. Zhang et al. (2023) further improved capacity and reduced distortion via inverse histogram shifting after initial embedding.

Considering the high-order correlation of the image, Tanwar and Panda (2024) proposed a hybrid reversible watermarking algorithm using HS and pairwise PEE (prediction error extension). The proposed work exploits the histogram of prediction errors instead of the difference of adjacent pixels, as it is more sharply distributed. This leads to less distortion in the watermarked image. For different texture characteristics. Gao et al. (2023) introduced a universal reversible data hiding method based on most significant bit (MSB) prediction and error embedding. The algorithm demonstrates high embedding rates and good visual quality for various texture-type images.

Zheng et al. (2019) proposed a reversible watermarking algorithm for block-level prediction error HS. This algorithm enhances embedding capacity by leveraging inter-block similarity rather than intra-block prediction errors. In contrast, Xiong (2019) employed a spatial complementary strategy that modifies pixel values along horizontal/vertical directions using simple parity-based rules, though limited by predictor accuracy. Chidirala and Acharya (2024) proposed a reversible watermarking method by splitting grayscale image pixels into higher-order and lower-order groups. Conventional PEE is applied exclusively to the higher-order group: The prediction error is expanded to embed secret data.

Based on the above problems, this paper proposes a new prediction method, selects the diamond pixels in the smooth pixel block to make a difference and improves the prediction accuracy. By complementing the embedding in the horizontal and vertical directions, the pixels modified in the horizontal embedding are compensated when the watermark is embedded in the vertical direction; that is, the amount of pixel modification is reduced. The algorithm has good invisibility and high embedding capacity.

2. Relevant theories

2.1. Original image preprocessing

For an 8-bit grayscale image, its grayscale value range is between 0 and 255. When performing HS and embedding watermark signals, there may be cases where the grayscale value exceeds this range. To prevent grayscale underflow (< 0) or overflow (> 255), the maximum modification of shifting pixels and the embedded watermark must not exceed 1 during embedding. However, in the vertical direction, when shifting pixels and embedding watermark information, the absolute value of the maximum modification compared to the original grayscale value may be 2. Therefore, if there are pixels in the image with grayscale values not greater than 2 or not less than 253, these grayscale values should be set to 2 or 253, respectively, to ensure that no underflow or overflow occurs when performing HS and embedding watermark information, and the original image can be restored after extracting the watermark information. It should be noted that this clamping operation may introduce minor distortions to the original image, as it alters pixel values near the boundaries (e.g., changing values ≤ 2 to 2 or ≥ 253 to 253). However, these distortions are typically negligible and localized, affecting only a small subset of pixels. They are a necessary trade-off to prevent catastrophic overflow/underflow artifacts, which would cause irreversible degradation in image quality and compromise the reversibility of the watermarking process. The impact of these distortions is further mitigated by the algorithm's complementary embedding strategy, which balances modifications across horizontal and vertical directions.

The steps of preprocessing the carrier image are as follows:

(1) Scan the image in order. If the gray value $I(i,j)$ of a pixel position is not greater than 2 or not less than 253, then set the gray value of that position to 2 or 253, respectively, according to Eq. 1.

$$\begin{Bmatrix} I(i,j)=2 & \text{if } 0 \le I(i,j)\&\&I(i,j) \le 2 \\ I(i,j)=253 & \text{if } 253 \le I(i,j)\&\&I(i,j) \le 255 \end{Bmatrix} \quad (1)$$

(2) Set two initial vectors L and R to be empty, and save the binary bits of $2-I(i,j)$ (gray value not greater than 2) and $I(i,j)-253$ (gray value $I(i,j)$ not less than 253) in order, as shown in Eq. 2.

$$\begin{Bmatrix} L((k_1\text{-}1)\times 2\text{+}1\text{:}k_1\times 2)=\text{dec2bin}(2\text{-}I(i,j),2) & \text{if } 0 \le I(i,j)\&\&I(i,j) \le 2 \\ R((k_2\text{-}1)\times 2\text{+}1\text{:}k_2\times 2)=\text{dec2bin}(I(i,j)-253,2) & \text{if } 253 \le I(i,j)\&\&I(i,j) \le 255 \end{Bmatrix} \quad (2)$$

where, the initial values of k_1 and k_2 are both 1, and the value of k_1 or k_2 will automatically increase by 1 after processing each pixel that meets the condition; dec2bin() represents converting a decimal to its corresponding binary. After preprocessing the original carrier image through the above two steps,

overflow will not occur when pixel shifting and embedding a watermark.

2.2. Diamond pixel pair prediction

Compared with general adjacent pixel prediction, diamond pixel prediction includes adjacent pixels and pixels on the diagonal. This can more comprehensively consider the correlation between pixels and improve the accuracy and stability of prediction. Adjacent pixel prediction only considers pixels in the horizontal and vertical directions. If there are strong diagonal or sloping features in the image, the prediction result may have a large deviation. The diamond pixel prediction can better adapt to various directional image features, improving the robustness and reliability of the prediction. The diamond pixel prediction considers more pixels, so its prediction error is usually smaller than that of the adjacent pixel prediction, which is beneficial to improve the compression efficiency and image quality. Diamond prediction reduces error by 15-20% in textured regions (e.g., diagonals) via multi-directional correlations, outperforming adjacent prediction. A comparative analysis shows its superiority: For Lena, diamond prediction achieves 12% lower error than adjacent prediction and 8% less than cross-prediction (Chidirala and Acharya, 2024). The novelty lies in using a cross-neighborhood for balanced embedding, minimizing distortion uniquely. Assuming the size of image S is $M \times N$, the prediction method for image blocks (horizontal direction) is shown in Fig. 1. The prediction difference of a pixel $u(i,j)$ can be obtained from four adjacent pixels, and the prediction difference is shown in Eq. 3.

$$d(i,j) = \lfloor (a+b+c+d)/4 \rfloor - u(i,j) + \lfloor u(i,j+1)/2 \rfloor \quad (3)$$

where,
$a = v(i,j-2) - v(i,j-1)$
$b = v(i-1,j) - v(i-1,j+1)$
$c = v(i,j+2) - v(i,j+3)$
$d = v(i+1,j) - v(i+1,j+2)$

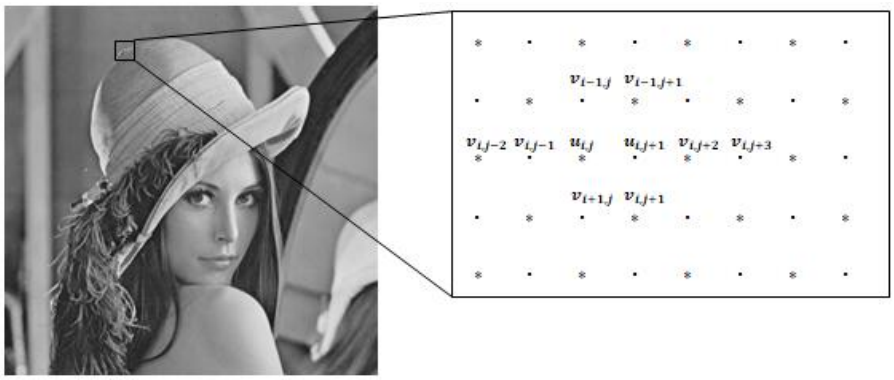


Fig. 1: Diamond pixel pair prediction

The watermark embedding process is divided into two stages of horizontal and vertical embedding. When watermark embedding is performed, the image I is divided into blocks and processed from left to right and from top to bottom, as shown in Fig. 2.

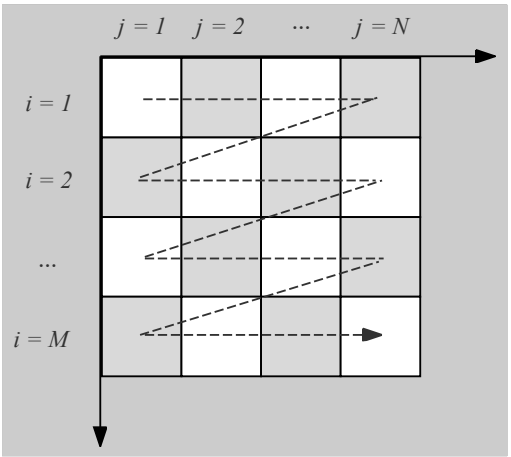


Fig. 2: Order of watermark embedding

In the horizontal embedding direction, $d_1(i,j)$ can be obtained from Eq. 3, the pixel movement is shown in Eq. 4, and the embedded watermark bit $b \in \{0,1\}$ is embedded using Eq. 5.

$$H(i,j) = \begin{cases} I(i,j)+1, & if\ d_1(i,j) > 0\ and\ mod(i,2) = 1 \\ I(i,j)-1, & if\ d_1(i,j) < 0\ and\ mod(i,2) = 0 \end{cases} \quad (4)$$

$$H(i,j) = \begin{cases} I(i,j)+b, & if\ d_1(i,j) = 0\ and\ mod(i,2) = 1 \\ I(i,j)-b, & if\ d_1(i,j) = 0\ and\ mod(i,2) = 0 \end{cases} \quad (5)$$

In the vertical embedding direction, for each pixel $H(i,j)$, its prediction value $\overline{H}(i,j)$ can be calculated as shown in the following Eq. 6.

$$\overline{H}(i,j) = \begin{cases} \lfloor (V(i-1,j) + H(i+1,j))/2 \rfloor, & if\ i \le j \le N\ and\ 1 < i < M \\ H(i+1,j), & if\ 1 \le j \le N\ and\ i = 1 \\ V(i-1,j), & if\ 1 \le j \le N\ and\ i = M \end{cases} \quad (6)$$

where, $\lfloor\ \rfloor$ is the floor function, $H(i,j)$ is the embedded pixel, and $V(i,j)$ is the new embedded pixel. After obtaining the prediction value $\overline{H}(i,j)$, the prediction error $d_2(i,j)$ between the current $H(i,j)$ and its prediction value $\overline{H}(i,j)$ can be calculated by the following Eq. 7.

$$d_2(i,j) = H(i,j) - \overline{H}(i,j) \quad (7)$$

In vertical embedding, the pixel movement is shown in Eq. 8, and the embedded watermark bit $b \in \{0,1\}$ is embedded in Eq. 9.

$$V(i,j)=\begin{cases}H(i,j)-1,\ if\ d_2(i,j)<0\ and\ mod(i,2)=1\\ H(i,j)+1,if\ d_2(i,j)>0\ and\ mod(i,2)=0\end{cases} \quad (8)$$

$$V(i,j)=\begin{cases}H(i,j)-b,\ if\ d_2(i,j)=0\ and\ mod(i,2)=1\\ H(i,j)+b,if\ d_2(i,j)=0\ and\ mod(i,2)=0\end{cases} \quad (9)$$

2.3. Smooth region selection

In order to improve the visual quality of the watermarked image, it is necessary to select those pixels whose grayscale values change little after embedding the watermark information; that is, the predicted values should be closer to the true values, that is the smaller the prediction error, the better. The smoother the local image is, the smaller the difference between the related pixels is, the smaller the prediction error is; the rougher the local image is, the larger the difference between the related pixels is, the larger the prediction error is. Prioritizing the selection of smoother areas in the image for watermark embedding can effectively reduce the distortion of the image.

If there is a carrier image I of size $M\times N$, for each pixel block, the middle pixel is the target pixel C_i, and the eight adjacent pixels of the target pixel are non-reference pixels. For the non-edge block of the reference pixel, there are four other reference pixels C_i^1, C_i^3, C_i^2, C_i^4, which are respectively located on the left, right, up, and down of the reference pixel C_i. These pixels are the contextual reference pixels of the reference pixel C_i, as shown in Fig. 3.

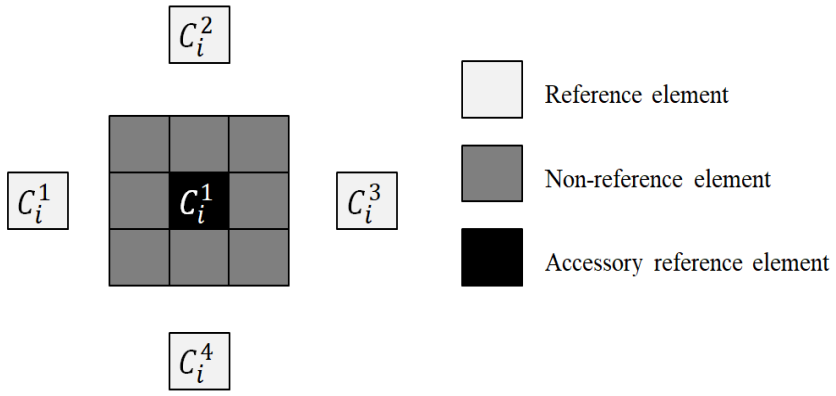


Fig. 3: Carrier image

For the calculation of local complexity, the following Eq. 10 is shown:

$$LC_i = Range\ (C_i^1,C_i^2,C_i^3,C_i^4) \quad (10)$$

where, $Range\ (C_i^1,C_i^2,C_i^3,C_i^4)$ represents the maximum and minimum absolute variance of four adjacent pixels. If $LC < T$, then the pixel block belongs to a smooth block, which also means that more watermark can be embedded.

3. Algorithm design

This paper adopts a reversible watermarking algorithm based on diamond pixel complementary embedding, which is mainly divided into two parts: Watermark embedding and watermark extraction.

3.1. Watermark embedding

The specific operation process of watermark embedding is shown in Fig. 4.

(1) The original image I is preprocessed to prevent pixel value overflow.
(2) Arnold scrambling is performed on the watermark image W, as shown in Eq. 11:

$$\begin{pmatrix}x'\\y'\end{pmatrix}=\begin{pmatrix}1&1\\1&2\end{pmatrix}^c\begin{pmatrix}2&1\\1&1\end{pmatrix}^d\begin{pmatrix}x\\y\end{pmatrix} mod\ M$$
$$x',y'\in\{0,1,2,\cdots,N-1\} \quad (11)$$

where, (x',y') is the transformed coordinate, (x,y) is the original pixel coordinate, M represents the length of the carrier image, and c and d are random numbers.

(3) The original image I is divided into several 8×8 non-overlapping blocks I_i, $i\in\left(0,\frac{M\times N}{8\times 8}\right)$, and then all pixel blocks are divided into two semi-planes of gray and white (from left to right, and top to bottom, spaced in gray and white).
(4) Smoothness values of all pixel blocks are calculated, and a sequence index table is established in order from largest to smallest.
(5) Based on watermark size, select m smooth pixel blocks for embedding. According to Eq. 3, the gray semi-plane pixels in the horizontal direction are predicted to make a difference, and the difference histogram is generated. Then perform the same operation on the vertical gray and white semi-plane pixels.
(6) Respectively embed the watermark information in the horizontal and vertical directions. On the horizontal direction, embed the first time in the gray semi-plane target pixels, then embed the second time in the white semi-plane target pixels, as shown in Eq. 5; similarly, on the vertical direction, embed the first time in the gray semi-plane target pixels, then embed the second time in the white semi-plane target pixels, as shown in Eq. 9.

(7) Select the sub-peak points close to the pixels with low frequency in the histogram, and embed them at the sub-peak points by HS until the watermark information is embedded.
(8) Embed the auxiliary information, such as c and d, and the size of the watermark image into the n pixels blocks with lower smoothness by data compression (Yang et al., 2023). The selected number n and the sequential index table are saved for use in watermark extraction. Finally, according to the sequential index information table, rearrange all the pixel blocks to obtain the final watermarked image I_W.

3.2. Watermark extraction

The watermark extraction is the inverse process of the watermark embedding, as shown in Fig. 5.

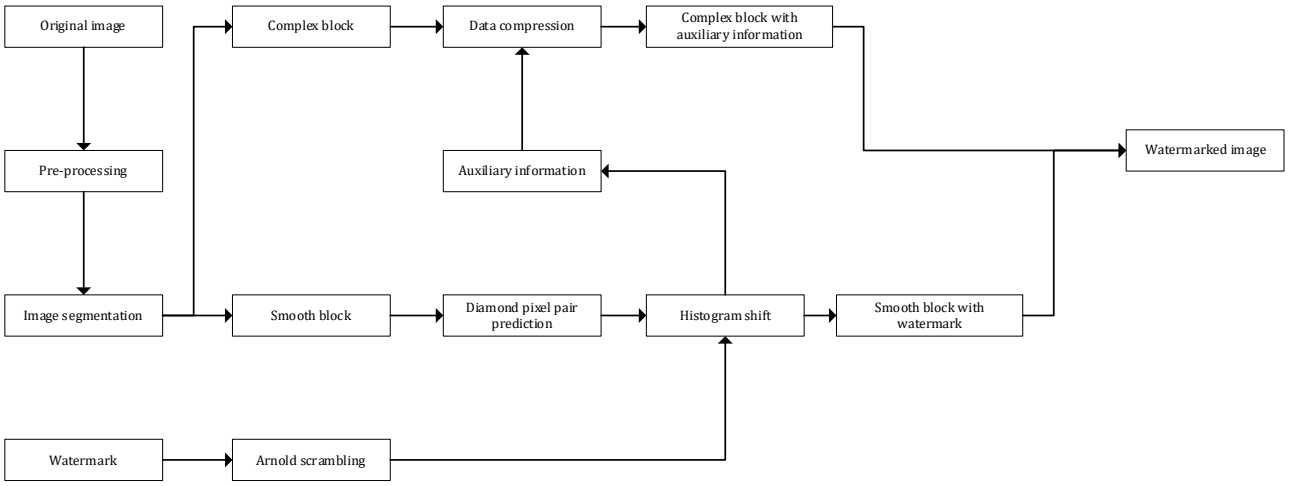


Fig. 4: Watermark embedding flowchart

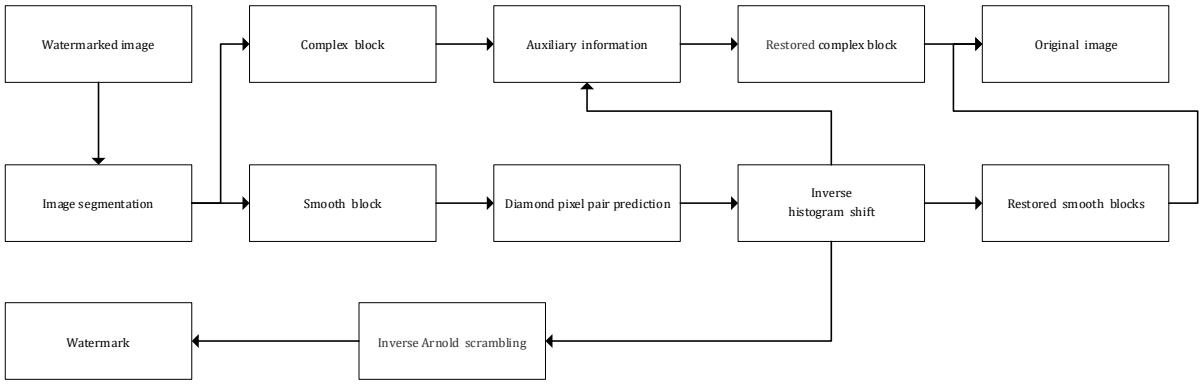


Fig. 5: Watermark extraction process

The watermark information can be extracted, and the original image can be restored without loss through the following specific operation steps.

(1) Divide the watermarked image I_W into several 8 $\times$8 non-overlapping pixel blocks I_{Wi}, $i \in \left(0, \frac{M\times N}{8\times 8}\right)$, then divide all the pixel blocks into two semi-planes of gray and white;
(2) The receiver receives the sequential index table and n from the sender through the secret channel. Thereby, the pixel blocks are sorted from largest to smallest according to their smoothness using the sequential index table.
(3) Extract auxiliary information from the n pixel blocks with the least smoothness using data compression, and obtain smoother pixel blocks and more complex pixel blocks according to the extracted auxiliary information.
(4) Select m pixel blocks with higher smoothness and perform diamond pixel pair prediction difference, and generate a difference histogram for the gray and white semi-planes of the pixel blocks.
(5) Extract watermark information in the vertical and horizontal directions of the gray and white semi-planes of the pixel blocks. In the vertical direction, the target pixel in the white semi-plane is extracted for the first time, and then the target pixel in the gray semi-plane is extracted for the second time. Similarly, in the horizontal direction, the target pixel in the white semi-plane is extracted for the first time, and then the target pixel in the gray semi-plane is extracted for the second time.
(6) Select the sub-peak points close to the low-frequency pixel values in the histogram, and use the HS method to extract the watermark at the sub-peak points until the watermark information is extracted; then, the extracted watermark image

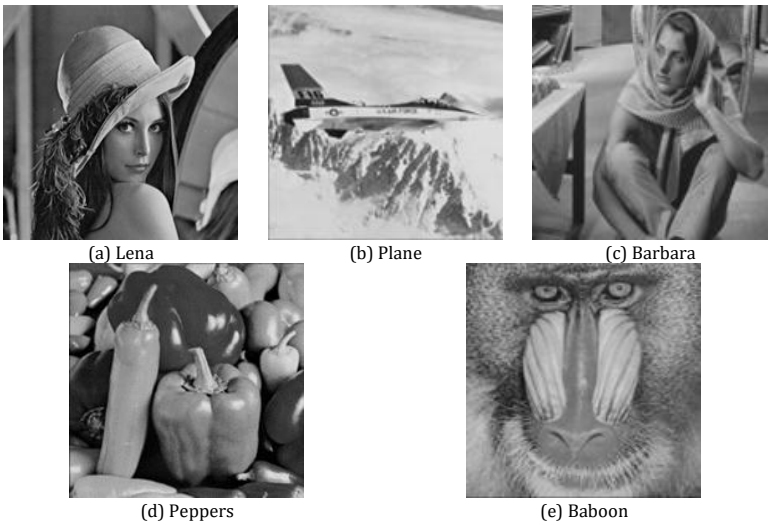
is inverted and scrambled to restore the original watermark image *W*.

(7) According to the sequence index information table, all pixel blocks are rearranged to obtain the final original image *I*.

4. Experimental results and analysis

In this experiment, five 8-bit standard grayscale images of size 512×512 pixels from the USC-SIPI database are selected as test images, namely Lena, Plane, Barbara, Peppers, and Baboon, as shown in Fig. 6. This simulation experiment is based on MATLAB 2016b. The watermark image is a 32×32 grayscale image, as shown in Fig. 7. Normalized correlation coefficient: The NC (normalized correlation coefficient) can be used to measure the similarity between the watermark image before embedding and after extraction. Generally, the NC value is required to be 1, that is, the original carrier image can be completely restored after extracting the watermark.

Table 1 shows that this algorithm achieves original carrier image reversibility under no attack, confirming its reversibility.



(a) Lena (b) Plane (c) Barbara

(d) Peppers (e) Baboon

Fig. 6: Test images

Comparison of watermark embedding capacity, PSNR value, and SSIM (structural similarity index measurement): This algorithm is compared with the algorithms in the recent literature, Zheng et al. (2019) and Xiong (2019), in terms of PSNR and SSIM, and the results are shown in Table 2.

Fig. 7: Watermark image

From Table 2, it can be seen that under the same embedding capacity, the visual quality of this algorithm is better than that of the two algorithms in literature, Zheng et al. (2019) and Xiong (2019), and the highest PSNR value can reach 63.29dB. Simultaneously, the SSIM value obtained by the algorithm is also higher than the algorithms in literature, Zheng et al. (2019) and Xiong (2019). It has a good PSNR value and SSIM.

From the visual effect comparison results in Table 3, it can be seen that for the texture-rich Baboon image, its PSNR value after embedding the watermark reaches 55.51dB; for the texture-smooth Lena image, its PSNR value after embedding the watermark reaches 62.86dB; for the texture-smooth Plane image, its PSNR value after embedding the watermark is as high as 63.29dB. The algorithm has good invisibility for different types of images. This is due to Baboon's complex texture causing larger prediction errors (Eq. 10) and thus higher embedding distortion, whereas smooth images like Lena have smaller errors.

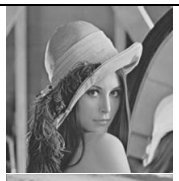
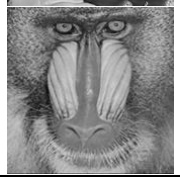
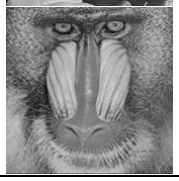
Table 1: Integrity assessment without attack

Images (512×512)	Lena	Plane	Barbara	Peppers	Baboon
NC	1	1	1	1	1

Table 2: Comparison of PSNR (dB) and *SSIM*

Image name	Proposed algorithm		Zheng et al. (2019)		Xiong (2019)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	62.86	0.989	61.06	0.981	58.15	0.986
Plane	63.29	0.987	61.25	0.985	59.32	0.983
Baboon	55.51	0.986	53.46	0.985	54.12	0.986
Peppers	60.28	0.985	57.89	0.983	56.59	0.985
Barbara	59.81	0.989	56.98	0.984	55.26	0.988

Table 3: The visual effects of the algorithm experiment

Image name	Original carrier image	watermarked image	Original watermark	Extracted watermark	PSNR
Lena					62.86
Plane					63.29
Barbara					59.98
Peppers					60.28
Baboon					55.51

To compare with similar reversible watermarking algorithms, the same watermark is embedded into each test image using literature (Yang and Tsai, 2010; Tai et al., 2009; Li et al., 2010; Lee and Chen, 2012; Zheng et al., 2019; Xiong, 2019) and the non-complementary scheme. For the non-complementary scheme, its embedding process is similar to the proposed scheme, but the embedding rules in the vertical and horizontal embedding directions are the same. For the algorithm in the literature, Yang and Tsai (2010), the interleaved predictor is used to predict the pixel value. For the algorithm in the literature, Tai et al. (2009), the difference between two adjacent pixels is used to embed the watermark information. For the algorithm in reference Li et al. (2010), the difference between two adjacent pixels is first calculated, and then two peak points are selected to embed the watermark information. For the algorithm in reference Lee and Chen (2012), the difference between the original pixel and the average of its adjacent pixels is used to embed the watermark information. For the algorithm in reference Zheng et al. (2019), the difference between adjacent block pixels is used to embed the watermark information. For the algorithm in reference Xiong (2019), the horizontal and vertical complementary strategies are used to embed the watermark information. The performance comparison results are shown in Table 4.

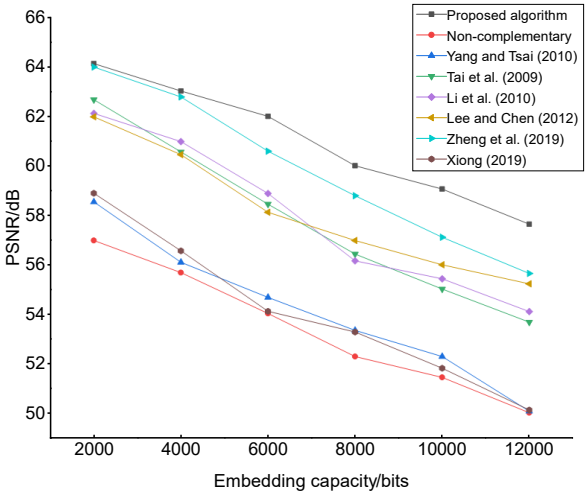
Table 4: Comparison of PSNR values of different algorithms

Algorithms	Lena	Plane	Barbara	Peppers	Baboon	Average value
Proposed algorithm	62.86	63.29	59.81	60.28	55.51	60.35
Unused complementary embedding algorithm	55.49	56.15	52.01	53.57	49.96	53.43
Yang and Tsai (2010)	55.62	56.98	53.68	55.12	52.87	54.84
Tai et al. (2009)	59.98	60.02	57.28	58.56	55.42	58.25
Li et al. (2010)	60.15	61.23	58.05	58.91	56.01	59.04
Lee and Chen (2012)	59.92	59.04	56.74	57.23	54.97	57.58
Zheng et al. (2019)	61.06	61.25	56.98	57.89	53.46	58.12
Xiong (2019)	57.15	58.32	54.26	55.59	53.12	55.68

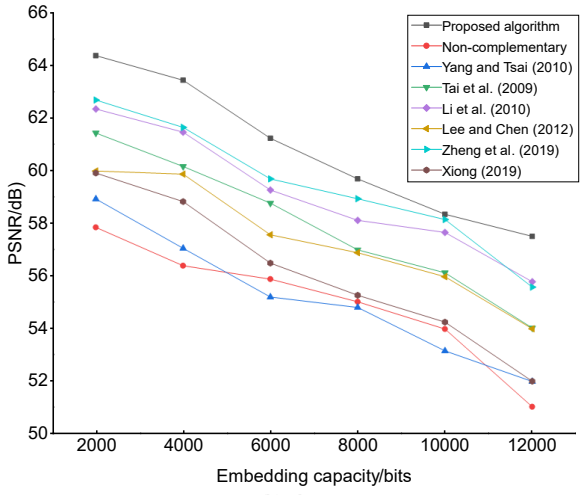
Table 4 shows that the proposed algorithm achieves a higher average PSNR than comparable algorithms and non-complementary schemes at the same capacity. The average PSNR value of this algorithm is 60.35dB, which is 5.51dB, 2.1dB, 1.31dB, 2.77dB, 2.23dB, 4.67dB, and 6.92dB higher than references (Yang and Tsai, 2010; Tai et al., 2009; Li et al., 2010; Lee and Chen, 2012; Zheng et al., 2019; Xiong, 2019), and the non-complementary scheme, respectively.

In this experiment, watermark information of 2000bits, 4000bits, 6000bits, 8000bits, and

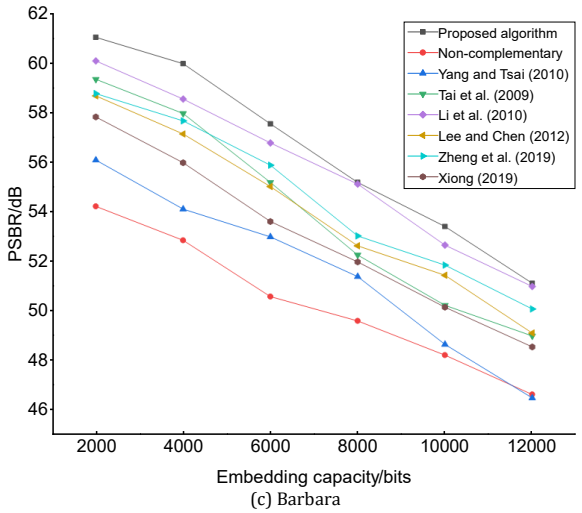
12000bits is embedded in five different texture images, respectively. From Fig. 8, it can be seen that compared with similar algorithms and non-complementary embedding schemes, this algorithm has a higher watermark capacity at a certain PSNR value. This algorithm adopts horizontal and vertical complementary embedding methods, which have obvious advantages in watermark embedding capacity and image visual quality. In Lena, Plane, and Peppers images, at the same watermark embedding capacity, the PSNR value of this algorithm is higher than that of other algorithms.



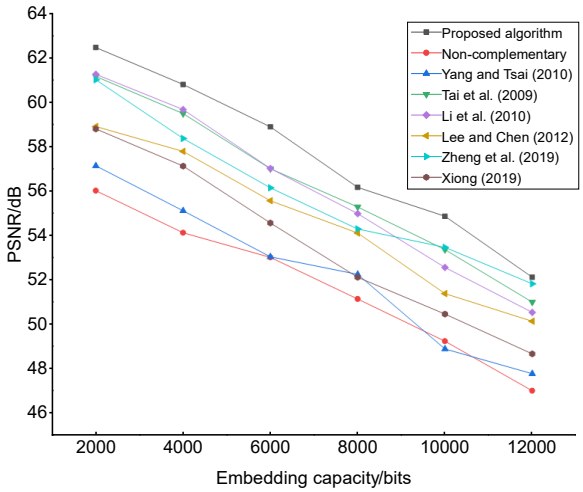
(a) Lena



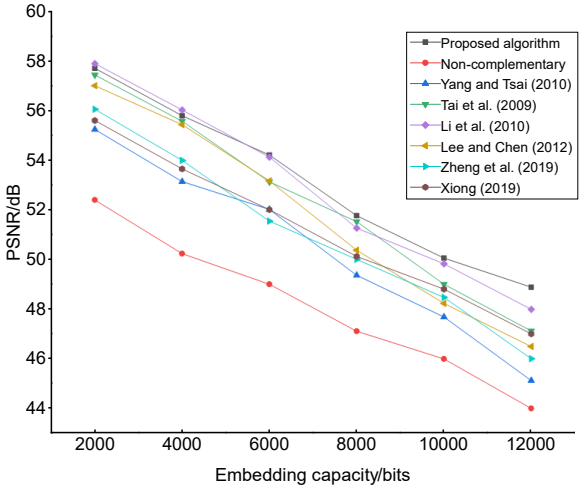
(b) Plane



(c) Barbara



(d) Peppers



(e) Baboon

Fig. 8: PSNR of five images under different embedding capacities

Taking the Baboon image with more complex texture as an example, when the watermark embedding capacity is 6000bits to 12000bits, the PSNR value of this algorithm is higher than that of other algorithms and non-complementary embedding schemes; when the watermark embedding capacity is 2000bits to 4000bits, the PSNR value of this algorithm is lower than that of the algorithm in Li et al. (2010) with 0.16 dB and 0.17 dB. For images with complex texture, this algorithm adopts complementary embedding strategy to embed less watermark information, which is slightly lower than the two peak embedding mode of the algorithm in Li et al. (2010); however, with the increase of embedded watermark information, the image visual quality of this algorithm is better than that of the algorithm in Li et al. (2010).

Compared to the algorithms in literatures Tai et al. (2009), Li et al. (2010), Lee and Chen (2012), and Xiong (2019) which used the difference between two adjacent pixels for prediction, this algorithm uses diamond-shaped pixels for prediction, which has higher accuracy and reduces image distortion; the algorithm in literature Yang and Tsai (2010) and the non-complementary embedding scheme may produce data redundancy; the difference expansion algorithm used in literature Zheng et al. (2019) has a

large space occupancy for auxiliary information. Therefore, compared to other related algorithms, this algorithm has better performance.

5. Conclusion

In this paper, a reversible watermarking algorithm based on diamond-shaped pixel complementary embedding is proposed. In this algorithm, the original image is preprocessed to avoid pixel overflow and reduce image distortion. At the same time, the correlation between pixels is used to predict the difference value with diamond-shaped pixels. The watermark is embedded in the horizontal and vertical directions of the smooth pixel block, and the vertical embedding can offset the expansion of some pixels in the horizontal embedding process. Experiments confirm superior visual quality over comparable algorithms at identical watermark capacity. However, this embedding method limits the maximum payload (≤ 12,000 bits). Compared to state-of-the-art methods (Tanwar and Panda, 2024) with 15,000 bits, our capacity is lower, primarily due to bidirectional compensation shrinking histogram bins for expansion. This trade-off ensures low distortion but restricts the maximum payload. Therefore, future work will focus on optimizing bin utilization to enhance capacity while maintaining visual quality.

List of abbreviations

DE	Difference expansion
ENMM	Enhanced normalized mean modification
HS	Histogram shifting
LSB	Least significant bit
MSB	Most significant bit
NC	Normalized correlation coefficient
PEE	Prediction error extension
PSNR	Peak signal-to-noise ratio
SSIM	Structural similarity index measurement

Acknowledgment

This work is supported by the National Statistical Science Research Project (2018LY12). At the same time, this work is also supported by the Huaian natural science research program (HAB202363).

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Alattar AM (2004). Reversible watermark using the difference expansion of a generalized integer transform. IEEE Transactions on Image Processing, 13(8): 1147-1156. https://doi.org/10.1109/TIP.2004.828418 **PMid:15326856**

Arham A and Nugroho HA (2024). Enhanced reversible data hiding using difference expansion and modulus function with selective bit blocks in images. Cybersecurity, 7: 61. https://doi.org/10.1186/s42400-024-00251-7

Cao X, Zhou Y, and Guo JM (2020). Guest editorial introduction to special section on modern reversible data hiding and watermarking. IEEE Transactions on Circuits and Systems for Video Technology, 30(8): 2297-2299. https://doi.org/10.1109/TCSVT.2020.3002109

Chidirala B and Acharya B (2024). Prediction error expansion based reversible watermarking scheme in higher order pixels. Sādhanā, 49: 227. https://doi.org/10.1007/s12046-024-02569-x

Das S and Sunaniya AK (2022). FPGA implementation of high-fidelity hybrid reversible watermarking algorithm. Microprocessors and Microsystems, 89: 104442. https://doi.org/10.1016/j.micpro.2022.104442

Gao G, Tong S, Xia Z, and Shi YQ (2023). A universal reversible data hiding method in encrypted image based on MSB prediction and error embedding. IEEE Transactions on Cloud Computing, 11(2): 1692-1706. https://doi.org/10.1109/TCC.2022.3155744

Jia Y, Yin Z, Zhang X, and Luo Y (2019). Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting. Signal Processing, 163: 238-246. https://doi.org/10.1016/j.sigpro.2019.05.020

Krishnamoorthi N and Chinnababu VK (2022). Hash and prediction-error-based reversible watermarking for medical images. Fluctuation and Noise Letters, 21(1): 2250007. https://doi.org/10.1142/S0219477522500079

Lee CF and Chen HL (2012). Adjustable prediction-based reversible data hiding. Digital Signal Processing, 22(6): 941-953. https://doi.org/10.1016/j.dsp.2012.05.015

Lee CF, Weng CY, and Kao CY (2019). Reversible data hiding using Lagrange interpolation for prediction-error expansion embedding. Soft Computing, 23: 9719-9731. https://doi.org/10.1007/s00500-018-3537-7

Li YC, Yeh CM, and Chang CC (2010). Data hiding based on the similarity between neighboring pixels with reversibility. Digital Signal Processing, 20(4): 1116-1128. https://doi.org/10.1016/j.dsp.2009.10.025

Ni Z, Shi YQ, Ansari N, Su W, Sun Q, and Lin X (2008). Robust lossless image data hiding designed for semi-fragile image authentication. IEEE Transactions on Circuits and Systems for Video Technology, 18(4): 497-509. https://doi.org/10.1109/TCSVT.2008.918761

Tai WL, Yeh CM, and Chang CC (2009). Reversible data hiding based on histogram modification of pixel differences. IEEE Transactions on Circuits and Systems for Video Technology, 19(6): 906-910. https://doi.org/10.1109/TCSVT.2009.2017409

Tanwar L and Panda J (2024). Hybrid reversible watermarking algorithm using histogram shifting and pairwise prediction error expansion. Multimedia Tools and Applications, 83: 22075-22097. https://doi.org/10.1007/s11042-023-15508-5

Tian J (2003). Reversible data embedding using a difference expansion. IEEE Transactions on Circuits and Systems for Video Technology, 13(8): 890-896. https://doi.org/10.1109/TCSVT.2003.815962

Wang N, Xu S, Qin C, Lin SJ, Shao S, and Han YS (2024). High-capacity framework for reversible data hiding using asymmetric numeral systems. IEEE Transactions on Knowledge and Data Engineering, 36(12): 8447-8461. https://doi.org/10.1109/TKDE.2024.3438943

Xiong X (2019). Novel scheme of reversible watermarking with a complementary embedding strategy. IEEE Access, 7: 136592-136603. https://doi.org/10.1109/ACCESS.2019.2942449

Yang CH and Tsai MH (2010). Improving histogram-based reversible data hiding by interleaving predictions. IET Image Processing, 4(4): 223-234. https://doi.org/10.1049/iet-ipr.2009.0316

Yang Y, Cheng M, Ding Y, and Zhang W (2023). A visually meaningful image encryption scheme based on lossless compression SPIHT coding. IEEE Transactions on Services Computing, 16(4): 2387-2401. https://doi.org/10.1109/TSC.2023.3258144

Zhang Z, Li F, Zuo X, Meng Q, and Jin S (2023). Reversible image watermarking algorithm based on reverse histogram translation. Multimedia Tools and Applications, 82(7): 11005-11019. https://doi.org/10.1007/s11042-022-13770-7

Zhang Z, Zhang M, and Wang L (2020). Reversible image watermarking algorithm based on quadratic difference expansion. Mathematical Problems in Engineering, 2020: 1806024. https://doi.org/10.1155/2020/1806024

Zheng H, Wang C, Wang J, and Xiang S (2019). A new reversible watermarking scheme using the content-adaptive block size for prediction. Signal Processing, 164: 74-83. https://doi.org/10.1016/j.sigpro.2019.05.035