A novel method for random sequential high-capacity data hiding based on PVD and LSB

Wanqing Wu
Hebei University

Huili Li (lhlnie@163.com)
Hebei University

Research Article

Keywords: Information hiding, Image steganography, Pixel value difference (PVD), Least significant bit (LSB), Random sequence

Posted Date: October 25th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3469844/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Signal, Image and Video Processing on December 23rd, 2023. See the published version at https://doi.org/10.1007/s11760-023-02900-9.

A novel scheme for random sequential high-capacity data hiding based on PVD and LSB

Wanqing Wu[1,2] and Huili Li[1*]

[1*]School of Cyber Security and Computer, Hebei University, Baoding, 071002, China.
[2]Key Laboratory on High Trusted Information System in Hebei Province, Hebei University, Baoding, 071001, China.

*Corresponding author(s). E-mail(s): lhlnie@163.com;
Contributing authors: wuwanqing@hbu.com;

Abstract

In image information hiding, steganography capacity and image quality are the key indicators. Keeping the image intact with high embedding capacity is a major challenge in information hiding technology. This paper presents an adaptive block image steganography scheme based on pixel value difference (PVD) and least significant bit (LSB). The scheme achieves high capacity embedding while maintaining visually acceptable image quality, and the scheme is also resistant to regular and singular (RS) analysis and pixel difference histogram (PDH) analysis. The details are as follows: Firstly, the image is divided into 2×2 non-overlapping blocks, and a random sequence of integers 1-4 is generated by chaotic mapping as the embedding order. The first pixel in the embedding order is selected as the reference pixel for LSB embedding. Secondly, the difference between the remaining pixels in the block and the reference pixel is calculated, and the embedded bit is determined according to the range where the difference is located. The mixed method of PVD and LSB is utilized to embed the remaining pixels. Finally, the embedded pixels are adjusted to ensure that the differences before and after embedding are in the same range. Experimental results show that the scheme can achieve an average embedding capacity of 930,312 bits, with an average embedding bits per pixel of 3.55 and a peak signal to noise ratio (PSNR) of 35.51 dB. The comparison with other hiding schemes shows that it has a larger hiding capacity than other schemes while maintaining good visual quality.

Keywords: Information hiding, Image steganography, Pixel value difference (PVD), Least significant bit (LSB), Random sequence

1 Introduction

The rapid development of the Internet makes information exchange more convenient, and people can learn a lot of information through various channels at any time. The Internet carries out frequent data upload, download, transmission and other operations every day. However, when information is transmitted on the Internet, its security is difficult to ensure, which may lead to the disclosure of user privacy. Therefore, a reliable approach is needed to protect the security of private data. Common methods to protect information security include cryptography and information hiding. Traditional cryptography techniques encrypt information and then transmit it in an unreadable form, which will arouse the suspicion of illegal users and attack it. Information hiding achieves secure and

reliable transmission by embedding information into the carrier without being detected.

In information hiding, there are three elements: capacity, imperceptibility, and robustness. Information hiding includes steganography and digital watermarking. In steganography, more emphasis is placed on capacity and imperceptibility, since the main purpose of steganography is to avoid detecting the presence of messages. However, robustness is actually more important in watermarking technology because the purpose of watermarking is to provide copyright protection and authentication[1–3].

In the field of steganography, several types of image-based spatial domain information hiding techniques were proposed, such as least significant bit (LSB) [4] and pixel value difference (PVD) [5]. With the deepening of the research, researchers had improved on the basis of LSB and PVD methods to obtain better hiding effects [6–15].

Subsequently, researchers took full utilized the advantages of LSB and PVD methods and combined them to further improve the capacity and imperceptibility of images [16–26]. For example, Wu et al. [16] adopted 1×2 pixel blocks, used LSB substitution in the smooth areas of the image, and used PVD method in the edge areas to obtain greater embedding capacity. Khodaei et al. [17] combined the LSB and PVD methods and divided pixels into 1×3 pixel blocks, with the middle pixel as the reference pixel. The reference pixel was embedded using k-LSBs substitution, and other pixels in the block (the first pixel and third pixel) were embedded using PVD method. This approach achieved a high embedding rate and also maintained acceptable imperceptibility. Hussain et al.[26] utilized two novel methods of parity-bit pixel value difference (PBPVD) and improved rightmost digit replacement (iRMDR) to improve the embedding rate.

Khodaei et al. [18] proposed an adaptive PVD and LSB high-capacity hiding scheme. The scheme divided the image into 1×2 non-overlapping blocks and utilized LSB substitution based on the difference of adjacent pixels. Khodaei et al. method could increase the embedding capacity and maintain good image quality, but in some cases, it could not extract messages correctly. Jung [21] proposed a bit plane hiding method based on LSB and PVD. However, this scheme caused pixel values overflow in some cases and could not be extracted correctly.

Hussain et al. [24] improved the problem that Khodaei et al. [18] could not extract information perfectly, and re-classified the image pixel values to design a new range table, which could extract data correctly while maintaining good visual quality. Similarly, Sahu et al. [23] improved some boundary and incorrect extraction problems of Jung [21] scheme, and proposed an improved steganography scheme of LSB substitution and PVD. After the embedding of the 1×3 pixel blocks, the pixel values were shifted to adjust the overflow pixels.

Subsequently, some researchers had expanded the pixel blocks to larger areas to reveal the correlation of adjacent pixels in the image and obtain a better look at the features between pixels. For example, Swain [19] expanded the pixel blocks size to 2×2, where the upper-left pixel was treated as the reference pixel, embedding secret information using a simple k-LSBs substitution. The remaining three pixels were embedded using the three-direction PVD method. Swain [20] then extended this scheme to 2×3 blocks and 3×3 blocks to enlarge capacity. Hameed [22] proposed the histogram of oriented gradient (HOG) algorithm to find the dominant edge direction of each 2×2 block. The pixels in the dominant direction were embedded with PVD method, while the other two remaining pixels were embedded with LSB substitution. Liao et al. [25] determined the level the block by calculating the average difference of four pixels in a 2×2 block, and then embedded the data with k-LSBs to make the differences before and after embedding belong to the same level.

Inspired by the above work, this paper proposes a novel adaptive LSB and PVD hybrid steganography scheme to achieve a larger embedding capacity, which can achieve completely correct data extraction. The experimental results show that the proposed method can maintain the image without distortion and resist the security analysis on the basis of realizing large capacity. The main contributions of this paper are:

1. Adopt random embedding order and make full use of the difference of pixel pairs in different directions to increase the embedding capacity. The level change of pixel value after embedding does not affect the correct extraction of data.

2. The proposed method improves the embedding capacity on the premise of maintaining

decent image quality and can resist steganographic attacks.

This paper is organized as follows. Section 2 introduces the related work. Section 3 gives a detailed introduction to the proposed scheme. The proposed scheme is analyzed and compared with other schemes In section 4. The section 5 is the conclusion.

2 Literature review

This section the high-capacity adaptive LSB and PVD scheme proposed by Khodaei et al. [18] and its error block problems.

2.1 High-capacity adaptive scheme

When using a hybrid method of PVD and LSB, it is necessary to adjust or limit the embedded pixel values to avoid the generation of error blocks.

The scheme [18] divides the image into 1×2 non-overlapping pixel blocks with pixel values set to p_i and p_{i+1}. The range interval $R_i = [L_i, U_i]$, L_i and U_i represent the lower and upper limit of the range, where i belongs 1 to 4. The range table is shown in Table 1, with four ranges: $R_1 = [0, 15]$, $R_2 = [16, 63]$, $R_3 = [64, 127]$, $R_4 = [128, 255]$. In the embedding process, the pixels are divided into two levels. *Level*-1: Pixels are less than 192. *Level*-2: At least one pixel is larger than 192. When the pixel pair satisfies *Level*-1, three bits of data are embedded in the p_i and p_{i+1} respectively to obtain p'_i and p'_{i+1}. When the pixel pair satisfies *Level*-2, calculate the difference $d_i = |p_i - p_{i+1}|$ between the two pixels to determine the embedding bits n. The number of bits embedded in each pixel is $k = n/2$. The k-bits secret data are embedded using LSB substitution on two pixels. The embedded pixel difference d'_i and the original difference d_i should in the same range. Otherwise, the pixel adjustment will be carried out according to the following steps:

Step1: Calculate the modified values of p_i and p_{i+1}

$$\begin{aligned} p''_i &= p'_i + 2^k \\ p'''_i &= p'_i - 2^k \\ p''_{i+1} &= p'_{i+1} + 2^k \\ p'''_{i+1} &= p'_{i+1} - 2^k \end{aligned} \tag{1}$$

Step2: Select the optimal pixel pair

$$(p'_i, p'_{i+1}) = optimal \begin{cases} (p'_i, p''_{i+1}) \\ (p'_i, p'''_{i+1}) \\ (p''_i, p'_{i+1}) \\ (p''_i, p''_{i+1}) \\ (p''_i, p'''_{i+1}) \\ (p'''_i, p'_{i+1}) \\ (p'''_i, p''_{i+1}) \\ (p'''_i, p'''_{i+1}) \end{cases} \tag{2}$$

Select the one optimal pixels pair that has a minimum difference between original pixels pair and steganographic (stego) pixels pair among other seven pixels pair combination. In addition, the new difference d'_i of the optimal pair of pixels should be in the same range as original difference d_i.

Table 1 Khodaei et al's range table

Ranges	R_1	R_2	R_3	R_4
The number of embedding (n)	8	10	12	14

2.2 Error block problems

The adaptive scheme proposed by Khodaei et al. can lead to pixel level problems that make it impossible to extract secret data correctly.

For example, given a pair pixels $p_i = 70$ and $p_{i+1} = 201$ that satisfy the *Level*-2 ($p_i \geq 192$ or $p_{i+1} \geq 192$). The difference $d_i = |p_i - p_{i+1}| = 131$ belongs to range R_4. Table 1 shows that $n = 14$ and $k = n/2 = 7$. Each pixel is embedded with 7 bits of secret data. Given binary sub-stream $S = (01101010010111)$. Embed $s_1 = (0110101)$ into $p_i = 70 = (01000110)_2$ to obtain $p'_i = (00110101)_2 = 53$ and $s_2 = (0010111)$ into $p_{i+1} = 201 = (11001001)_2$ to obtain $p'_{i+1} = (10010111)_2 = 151$. The new difference $d'_i = |53 - 151| = 98$ belongs to range R_3, whereas d_i belongs to R_4. Hence, the pixel adjustment operation is required. According to Eq.1, $p''_i = 53 + 2^k = 181$, $p'''_i = 53 - 2^k = -75$, $p''_{i+1} = 151 + 2^k = 279$, $p'''_{i+1} = 151 - 2^k = 23$. According to Eq.2, the optimal pixel pair $(181, 23)$ is obtained, and the new difference $d'_i = 158$ belongs to range R_4.

In the extraction process, the pixels $p'_i = 181$ and $p'_{i+1} = 23$ satisfy the *Level*-1 ($p_i < 192$ and $p_{i+1} < 192$). Therefore, the lower three bits of $p'_i = 181 = (10110101)_2$ and $p'_{i+1} = 23 = (00010111)_2$ are extracted respectively to obtain $s'_1 = (101)$ and $s'_2 = (111)$. As a result, the extracted data $S' = (101111)$ is different from the embedded data $S = (01101010010111)$.

Consequently, Khodaei et al. scheme fails to extract the information correctly when the pixel pair level changes.

3 The proposed method

In this section, a novel hybrid PVD and LSB random embedding method is proposed, which avoids the incorrect extraction caused by pixel level changing. Firstly, the proposed method divides the cover image into 2×2 pixel blocks, and a pixel is selected as reference pixel for k-LSBs embedding according to the embedding order generated by the chaotic mapping. Further, the remaining pixels determine the number of embedded secret bits n by calculating the difference from the reference pixel, and utilize n-LSBs substitution. Finally, the remaining pixels are adjusted so that the differences before and after embedding are in the same range. The flowchart of the scheme is shown in Fig. 1.

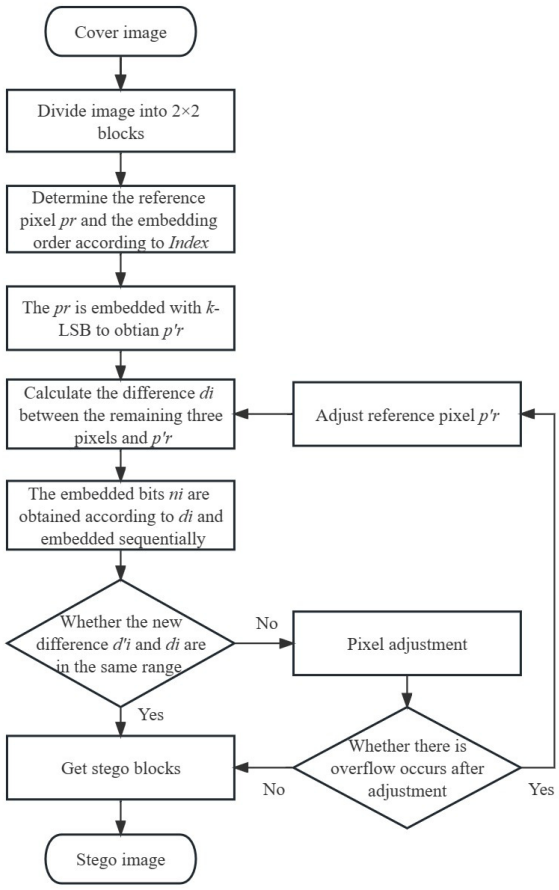


Fig. 1 Flowchart of the proposed method

3.1 Logistic mapping

Logistic mapping is a common nonlinear dynamic system, the main feature of this mapping model is simple and easy to use, and it is easy to produce chaos. Its form is:

$$x_{n+1} = \mu x_n(1 - x_n) \tag{3}$$

Where x_n represents the state variable iterated n times. It is obvious that when the initial value of the sequence x_0 is a real number between 0 and 1, all the numbers in the sequence are positive and less than 1. μ is the control parameter and must be $0 < \mu < 4$. When μ is near 2.58, x_n converges to a unique stable value at different initial values. However, when μ exceeds a certain threshold (about 3.57), the system will appear chaotic and show long-term unpredictable and random movement characteristics. The logistic mapping is used to generate random sequence that are used to select the reference pixel and the embedding order.

3.2 Difference range table

A novel difference range table proposed in this section divides pixel values $[0 - 255]$ into two levels to calculate the difference, as shown in Table 2. Given a pair of pixels p_i and p_{i+1}, calculate the difference $d_i = |p_i - p_{i+1}|$. The difference range table divides pixel values into two levels. For *Level*-1, when both p_i and p_{i+1} are less than 128, the difference d_i is divided into three ranges: $R_1 = [0, 15]$, $R_2 = [16, 63]$, $R_3 = [64, 127]$. For *Level*-2, when at least one of p_i or p_{i+1} is greater than or equal to 128, d_i is divided into two ranges: $R_4 = [0, 31]$, $R_5 = [32, 255]$. n in the table represents the number of embedded bits.

Table 2 The proposed method's range table

Level-1: $p_i < 128$ and $p_{i+1} < 128$			
Ranges	R_1	R_2	R_3
The number of embedding (n)	3	4	5
Level-2: $p_i \geq 128$ or $p_{i+1} \geq 128$			
Ranges	R_4	R_5	
The number of embedding (n)	4	5	

3.3 The proposed embedding steps

In order to protect the security of the secret data and make full use of the pixel value difference in different directions, a random embedding order is adopted. The embedding process is shown in Algorithm 1.

Algorithm 1 Embedding process

Input: Cover image I_c, Secret message S_m.
Output: Stego image I_s.

1: Divide I_c into 2×2 non-overlapping blocks.
2: Obtain embedding order $Index$ from Algorithm 2.
3: Select the first pixel as the reference pixel according to $Index$, denoted as p_r and perform k-LSBs embedding on p_r to obtain p'_r, where $k = 3$.
4: Calculate the difference between the p'_r and the other three pixels p_{r1}, p_{r2}, p_{r3} respectively in order of $Index$, obtain d_1, d_2, d_3.
5: Determine the level of pixel pairs. Referring to Table 2 to find the range R to which d_1, d_2 and d_3 belong, and obtain the number of bits n_1, n_2 and n_3 embedded in each group of pixels.
6: Obtain n_1, n_2 and n_3 bits from the S_m and embed them into p_{r1}, p_{r2}, p_{r3} using n-LSBs respectively, obtaining p'_{r1}, p'_{r2}, p'_{r3}.
7: Calculate new differences between the p'_r and p'_{r1}, p'_{r2}, p'_{r3} obtaining d'_1, d'_2, d'_3. If d'_i and d_i are in the same range, skip to step **9**, otherwise adjust p'_{ri} according to algorithm 3.
8: Check weather p'_{ri} has overflow. If there is overflow, adjust the reference pixel p'_r according to Eq.4 and then return to step **4**.

$$p'_r = \begin{cases} p'_r + 2^k, & \text{if} \quad p_1 < 0 \\ p'_r - 2^k, & \text{if} \quad p_1 > 255 \end{cases} \tag{4}$$

9: Follow the above steps to traverse all blocks for embedding operation.

Algorithm 2 Embedding order

Input: Secret key SK, Length L.
Output: Embedding order $Index$.

1: Set Logistic initial value SK and L generate a random sequence, where L is the number of blocks.
2: Group the random sequence into groups of 4.
3: Convert each group of random numbers into integers 1-4 according to their size to obtain $Index$.

Algorithm 3 Adjustment pixel

Input: Reference pixel p'_r, Adjusted pixel p'_{ri}, Embed bits n_i, Difference value d_i.
Output: Optimal pixel p'_{ri}.

1: Calculate the modified value of p'_{ri}.

$$\begin{aligned} p''_{ri} &= p'_{ri} + 2^{n_i} \\ p'''_{ri} &= p'_{ri} - 2^{n_i} \end{aligned} \tag{5}$$

2: Calculate the difference value.

$$\begin{aligned} d''_i &= p''_{ri} - p'_r \\ d'''_i &= p'''_{ri} - p'_r \end{aligned} \tag{6}$$

3: Select the optimal pixel.

$$p'_{ri} = \begin{cases} p''_{ri}, & \text{if} \quad d''_i \in R(d_i) \\ p'''_{ri}, & \text{if} \quad d'''_i \in R(d_i) \end{cases} \tag{7}$$

In the process of pixel adjustment, some pixel values inevitably change around the pixel value boundary 128, resulting in changes in the initial conditions of the range table. The difference range table of this scheme makes full use of the relationship between the pixel variation range and the embedded bits, and does not need to restrict the embedded pixel level. Even if the adjusted pixel value changes around the level boundary of 128, it will not affect the extraction.

3.4 The proposed extracting steps

The extraction process is the inverse process of the embedding, and the extraction process of pixel blocks is shown in Algorithm 4.

Algorithm 4 Extraction process

Input: Stego image I_s.

Output: Secret message S_m.

1: Divide the stego image I_c into 2×2 non-overlapping blocks.
2: Obtain extraction sequence $Index$ from Algorithm 2.
3: Determine reference pixel according to $Index$ and extract its k-LSBs.
4: Calculate the difference between the p'_r and the other three pixels in order of $Index$ to obtain the number of bits to be extracted according to Table 2.
5: Extract the corresponding number of bits according to difference value separately.
6: Loop through 3-5 steps until all data extraction is complete.
7: Link the extracted data for each block to obtain S_m.

3.5 An example for proposed method

Fig. 2 shows an example of embedding and extracting of the proposed method. Select a 2×2 pixel block, given data $S = (0100011010101100)$, $Index = \{4, 2, 1, 3\}$. Select p_4 as the reference pixel p_r, with embedding order p_r, $p_{r1} = p_2$, $p_{r2} = p_1$, and $p_{r3} = p_3$. The pixel values of the original pixel block are $p_r = 110$, $p_{r1} = 153$, $p_{r2} = 129$, and $p_{r3} = 85$. Take 3-bits data sub-stream from the binary data stream and perform 3-LSBs embedding on p_r. Embed $s = (010)$ into $p_r = 110 = (01101110)_2$ to get $p'_r = (01101010)_2 = 106$. Calculate the differences between p'_r and the remaining pixels $d_1 = |153 - 106| = 47$, $d_2 = |129 - 106| = 23$, $d_3 = |85 - 106| = 21$. Both p_{r1} and p_{r2} are greater than 127, satisfying $Level$-2 in Table 2. Referring to the range of the difference, obtain the number of bits to be embedded $n_1 = 5$, $n_2 = 4$, $n_3 = 4$. Take the corresponding sub-streams $s_1 = (00110)$, $s_2 = (1010)$, $s_3 = (1100)$ from the binary data stream, obtaining $p'_{r1} = (10000110)_2 = 134$, $p'_{r2} = (10001010)_2 = 138$, $p'_{r3} = (01011100)_2 = 92$ after embedding. Calculate the new difference $d'_1 = 28$, $d'_2 = 32$, $d'_3 = 14$. Referring to Table 2, it is necessary to adjust p'_{r1}, p'_{r2}, p'_{r3} to obtain $p''_{r1} = 166$, $p'''_{r1} = 102$, $d''_1 = 60$, $d'''_1 = 4$. By comparison, it can be seen that d''_1 and d_1 belong to the same range, so adjust p'_{r1} to 166. For the same reason, p'_{r2} is adjusted to 122 and p'_{r3} is adjusted to 76.

In the extraction stage, according to $Index = \{4, 2, 1, 3\}$, the extraction order is p'_r, p'_{r1}, p'_{r2}, and p'_{r3}. Firstly, extract the lower three bits of the reference pixel $p'_r = 106 = (01101010)_2$ to obtain the data stream $s' = (010)$. Secondly, calculate the differences between p'_r and other pixels to obtain $d'_1 = 60$, $d'_2 = 16$, $d'_3 = 30$, and determine the level of the pixel pairs. Referring to Table 2, extract the low 5 bits of $p'_{r1} = 166 = (10100110)_2$, the low 4 bits of $p'_{r2} = 122 = (01111010)_2$, and the low 4 bits of $p'_{r3} = 76 = (01001100)_2$ to obtain $s'_1 = (00110)$, $s'_2 = (1010)$, $s'_3 = (1100)$. Finally, connect s'_1, s'_2 and s'_3 to obtain the data stream $S' = (0100011010101100)$ embedded in the block.

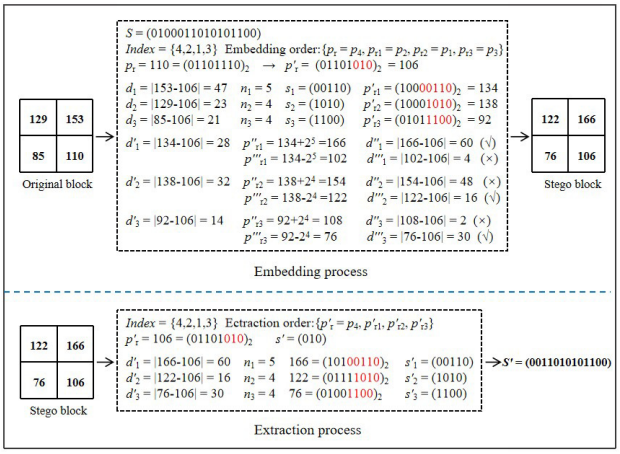
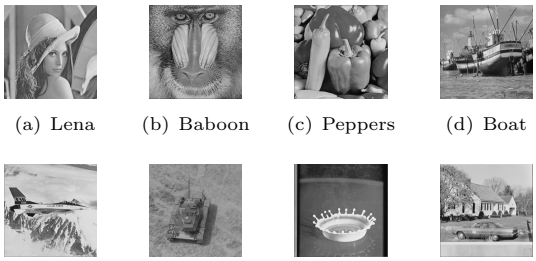


Fig. 2 An embedding and extraction example of a pixel block

4 Experimental results and analysis

The proposed method is simulated by using MATLAB software. USC-SIPI[27] and BOWS2[28]image datasets are used for the experimental images. In the experiment, eight 512×512 gray scale images are selected, and the secret data were generated by pseudo-random number generator. Sample images are shown in Fig. 3.

This section is divided into two parts, the first part is to analyze the safety of the proposed method, the second part is to analyze the experimental results.



(a) Lena (b) Baboon (c) Peppers (d) Boat

(e) F16 (f) Tank (g) Splash (h) House

Fig. 3 Sample images

4.1 Safety analysis

This section analyzes the security of the proposed method, including error block problems analysis, RS analysis and PDH analysis.

4.1.1 Error block problems analysis

The error block problems are mainly divided into two situations: pixel value level changing and pixel value overflow. In the difference range table (Table 2) proposed in this paper, the intermediate pixel value 128 is selected as the boundary, and the interval range is greater than twice the change of pixel value range caused by number of embedded bits. This design can correctly obtain the number of embedded bits to extract data even if the problem of pixel value level changing occurs. The proposed method modify the reference pixel to eliminate the overflows of the pixel value. If the adjusted pixel value overflows upward, reduce the reference pixel, and if it overflows downward, increase the reference pixel. Repeat the embedding process until no overflow occurs. The following are two examples to illustrate the processing of two situations.

(i) The problem of pixel level changing is shown in Fig.2. The pixel values of p_{r2} and p'_r in the original pixel pair are 129 and 106 respectively, belonging to *Level*-2. However, p'_{r2} becomes 122 after embedding, indicating that the pixel pair belongs to *Level*-1. According to the difference range of *Level*-1 during extraction, the correct number of embedded bits $n_1 = 4$ can still be obtained, which will not affect the correct extraction of data.

(ii) Pixel overflow problem (take underflow as an example) As shown in Fig. 4. In the pixel adjustment in Algorithm 4, there is an underflow situation in p'_{r1}. In this case, the reference pixel p'_r is increased to 31, and the embedding process is performed again, so that overflow will not occur and the data can be extracted correctly.

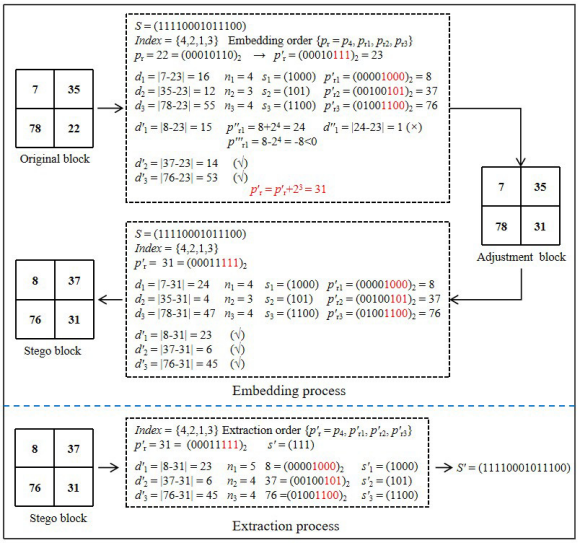


Fig. 4 A pixel value overflow example

From the above analysis, it can be shown that the method proposed in this paper can successfully avoid the error block problems.

4.1.2 Regular and singular (RS) analysis

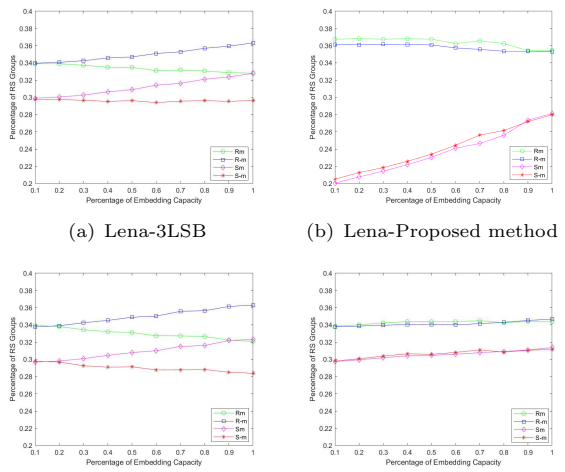
RS analysis [29] mainly detects the LSB substitution steganography algorithm, and its main process is to judge whether the image contains secret information according to the texture complexity of image sub-blocks. For a set of pixels $(x_1, x_2, ..., x_n)$, defines the smoothness of the pixel as follows:

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n-1} |x_i - x_{i+1}| \tag{8}$$

Where f is the smoothness function of the image and describes the smoothness of the pixel group.

In RS analysis, the image is divided into several pixel groups, and each pixel group is flipped non-negatively under the mask operator M. If the f value becomes larger after flipping, the pixel group is called a regular group. If the f value

becomes smaller after flipping, the pixel group is called a singular group. If it does not change, it is called a useless group. The proportion of the regular groups in all groups is R_M, and the proportion of the singular groups in all groups is S_M, where $R_M + S_M \leq 1$. The ratios of regular groups and singular groups obtained by flipping under the mask -M are R_{-M} and S_{-M}. Once the condition of $R_M \approx R_{-M} > S_M \approx S_{-M}$ is satisfied, it is indicated that the steganography method has successfully passed the RS analysis. If $R_{-M} - S_{-M} > R_M - S_M$, it indicates that the steganography method has been detected by RS analysis. Experimental results in Fig. 5 show that the proposed method can successfully pass the RS analysis.



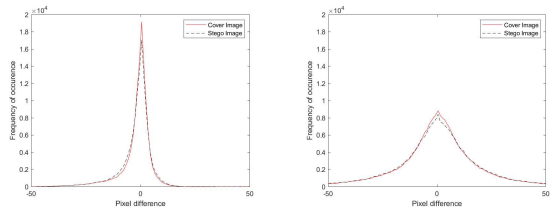
(a) Lena-3LSB (b) Lena-Proposed method

(c) Baboon-3LSB (d) Baboon-Proposed method

Fig. 5 The RS analysis of proposed method

4.1.3 Pixel difference histogram (PDH) analysis

The pixel difference histogram method is a steganographic analysis method based on random modulation[30]. Standard steganographic methods such as PVD are generally not resistant to PDH analysis. PDH is achieved by obtaining a histogram of the difference between adjacent pixels, which can be between -255 and 255. The PDH curves of cover image and stego image are shown in Fig. 6. The results indicate that both PDH curves are smooth, and there is no jagged appearance. Therefore, the proposed method is safe for PDH analysis.



(a) Lena (b) Baboon

Fig. 6 The PDH analysis of proposed method

4.2 Analysis of experimental results

The proposed scheme is compared with other schemes in terms of peak signal to noise ratio (PSNR), embedding capacity (EC), bits per pixel (BPP) and error blocks (EB).

4.2.1 Experimental results of the proposed method

The mean square error (MSE) represents the mean of the difference between the pixel values of the cover image and the steganographic image. MSE is further used to calculate the PSNR, defined as follows:

$$MSE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} [I_1(x,y) - I_2(x,y)]^2 \quad (9)$$

Where W×H is the dimension of the image, $I_1(x,y)$ and $I_2(x,y)$ are the pixel values in row x and column y in the cover image and stego image, respectively.

PSNR is a quality measure between two images, indicating how much of one image is present in the other image after processing and how much noise has been added. A higher PSNR means that the stego image is largely similar to the composition of the cover image. PSNR is defined as follows:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (10)$$

EC refers to the maximum number of bits that can be embedded in an image. However, due to

the differences in the size of the images compared, as well as the difference in color and gray scale, the capacity can be measured using the size of the number of bits embedded in each pixel – bits per pixel (BPP). For images of size W×H, the BPP is calculated as follows:

$$BPP = \frac{\text{Embedding Capacity}}{W \times H} \tag{11}$$

The EB is whether there is a pixel overflow or a pixel level change in the scheme that prevents the correct extraction of data. Here Y means there are error blocks and N means there are no error blocks.

The experiment is carried out according to the embedding steps introduced in Section 3. The 8 representative grayscale images in Fig. 3 are tested, and the results are shown in Table 3.

Table 3 Experimental results of the proposed method

Cover Images	EC(bits)	PSNR(dB)	BPP
Lena	919,520	35.97	3.51
Baboon	963,107	32.23	3.67
Peppers	900,125	36.40	3.43
Boat	946,825	35.22	3.61
F16	962,588	35.04	3.67
Tank	944,153	35.37	3.60
Splash	919,520	38.18	3.20
House	967,371	34.64	3.69
Average	930,312	35.51	3.55

The average embedding capacity EC of the scheme is 930,312 bits, the average PSNR is 35.51dB, and the number of embedded bits per pixel is up to 3.55. When the embedding capacity EC reaches a maximum of 967,352 bits, the PSNR value is 34.64dB, which can be accepted by human visual system (HVS).

4.2.2 Comparison with existing methods

In order to illustrate the effectiveness of the algorithm, EC, PSNR, BPP and EB are compared with other methods in this paper. The comparison results are shown in Table 4.

Khodaei et al. [18] method provides a new idea for the combination scheme of PVD and LSB. Although it cannot extract data correctly, its PSNR with an average embedding capacity of 811,476 bits and 35.77 dB is acceptable.

Swain [19] selects the reference pixel in the upper left corner of the 2×2 block and utilizes the LSB substitution method, while the remaining pixels utilize the PVD method, making full use of the texture area of the image, and maintaining a good image quality (38.90 dB) when the capacity reaches 832,675 (3.18 BPP) bits. However, when this method is applied to 3×3 pixel blocks [20], the image quality decreases (35.94 dB) when the capacity is increased to 864,616 (3.3 BPP) bits. Based on the scheme proposed by Jung[21], Sahu[23] solves the error block problems caused by pixel adjustment, and obtains a capacity of 823,717 (3.14 BPP) bits and an image quality of 34.50 dB. Based on scheme [18], Hussain[24] proposes a new range table and graded pixel pairs to solve the problem of error blocks. The capacity of 842,364 (3.21 BPP) bits and 35.76 dB is obtained.

It can be seen from Table 4 that the scheme proposed in this paper has the highest average hiding capacity, obtaining an embedding capacity of up to 930,312 bits and a PSNR value of 35.51 dB. A maximum capacity of 967,371 bits with 34.64 dB is achieved in the "House" image. Compared with the scheme [18], this scheme increases the EC by 14.6 % and decreases the PSNR by only 0.7 %, which is an excellent data. Compared with other schemes, the embedding capacity of this scheme is improved, and the visual quality of the image is maintained at a good level. In addition, there is no error block problems in the comparison scheme except [18], and the pixel adjustment method proposed in this scheme successfully avoids the occurrence of error blocks. Fig. 7 shows the comparison between this scheme and the schemes in Table 4 in terms of EC and PSNR. It can be intuitively seen that this scheme has obtained prominent results in EC and PSNR.

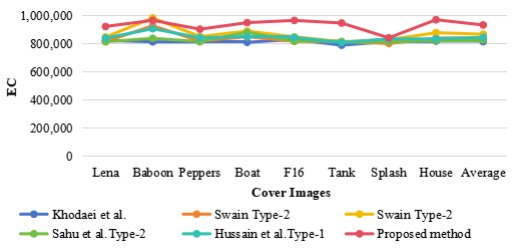
5 Conclusion

In this paper, an adaptive block image steganographic algorithm based on PVD and LSB is proposed. Divide the image into 2×2 blocks, select a reference pixel in the block, and embed the secret data by calculating the difference between the remaining pixels and the reference pixel. Using random embedding order can not only improve the security of data, but also make full use of

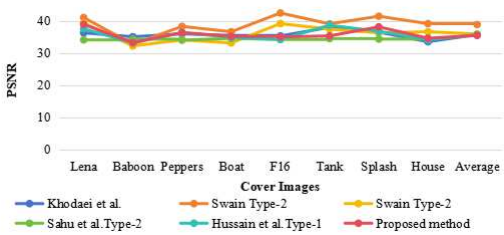
Table 4 Comparison with existing methods

Cover Images 512×512	Khodaei et al. [18]				Swain [19] Type-2				Swain [20] Type-2			
	EC	PSNR	BPP	EB	EC	PSNR	BPP	EB	EC	PSNR	BPP	EB
Lena	820,307	36.32	3.13	N	812,567	41.40	3.10	N	844,517	39.38	3.22	N
Baboon	810,517	35.10	3.09	Y	924,182	32.76	3.53	N	979,792	31.32	3.74	N
Peppers	811,841	36.01	3.10	Y	815,912	38.33	3.11	N	848,130	34.10	3.24	N
Boat	807,021	35.51	3.08	Y	846,510	36.66	3.23	N	886,598	33.17	3.38	N
F16	827,363	35.35	3.16	Y	814,497	42.51	3.11	N	846,267	39.20	3.23	N
Tank	786,499	38.03	3.00	N	812,682	39.12	3.10	N	811,033	37.48	3.09	N
Splash	812,299	36.54	3.10	N	798,261	41.50	3.05	N	825,326	36.33	3.15	N
House	815,957	33.12	3.11	Y	836,791	39.19	3.19	N	875,268	36.66	3.34	N
Average	811,476	35.77	3.10	-	832,675	38.90	3.18	-	846,616	35.94	3.30	-
Cover Images 512×512	Sahu et al. [23] Type-2				Hussain et al.[24] Type-1				Proposed method			
	EC	PSNR	BPP	EB	EC	PSNR	BPP	EB	EC	PSNR	BPP	EB
Lena	811,053	34.78	3.09	N	836,627	37.57	3.19	N	919,520	35.97	3.51	N
Baboon	835,355	34.12	3.19	N	903,441	33.21	3.45	N	963,107	33.23	3.67	N
Peppers	812,123	34.35	3.10	N	839,765	36.42	3.20	N	900,135	36.40	3.43	N
Boat	877,900	34.05	3.30	N	847,483	34.95	3.23	N	946,825	35.22	3.61	N
F16	814,926	34.58	3.09	N	841,971	34.25	3.21	N	962,588	35.04	3.67	N
Tank	809,180	35.24	3.09	N	805,677	38.68	3.07	N	944,153	35.37	3.60	N
Splash	810,633	34.52	3.09	N	831,377	36.81	3.17	N	838,796	38.18	3.20	N
House	818,727	34.38	3.12	N	832,567	34.15	3.18	N	967,371	34.64	3.69	N
Average	823,717	34.50	3.14	-	842,364	35.76	3.21	-	930,312	35.51	3.55	-

Measurement unit: EC(bits), PSNR(dB).



(a) The EC comparison with existing methods



(b) The PSNR comparison with existing methods

Fig. 7 Data comparison

the difference of pixels in different directions to achieve greater embedding capacity. This method utilizes a mixed PVD and LSB embedding scheme for two kind of levels of pixel values, avoiding the problem of pixel values changing at the pixel level boundary 128. Moreover, after adjustment, it does not need to limit the pixel level and can correctly extract data. In the experimental phase, the scheme achieved a high embedding capacity of 930,312 bits on average and a PSNR value of 35.51 dB. The method is also simulated on a variety of database images, and the results show that it can improve the embedding capacity, maintain a good imperceptibility, can be accepted by HVS, and can resist RS analysis and PDH analysis.

Declarations

- **Data availability statement** The data during this study are available in the submitted article.

References

[1] Fridrich, J. *Steganography in Digital Media: Principles, Algorithms, and Applications* (Cambridge University Press, Britain, 2009).

[2] Cheddad, A., Condell, J., Curran, K. & Mc Kevitt, P. Digital image steganography: Survey and analysis of current methods. *Signal Process* **90**, 727–752 (2010).

[3] Kumar, A., Rani, R. & Singh, S. A survey of recent advances in image steganography. *Security and Privacy.* **6**, e281 (2022).

[4] Bender, W., Gruhl, D., Morimoto, N. & Lu, A. Techniques for data hiding. *IBM Syst J* **35**, 313–336 (1996).

[5] Wu, D. C. & Tsai, W. H. A steganographic method for images by pixel-value differencing. *Pattern Recogn Lett* **24**, 1613–162 (2003).

[6] Yang, C. H., Weng, C. Y., Tso, H. K. & Wang, S. J. A data hiding scheme using the varieties of pixel-value differencing in multimedia images. *J Syst Softw* **84**, 669–678 (2011).

[7] Setiadi, D. I. M. Improved payload capacity in lsb image steganography uses dilated hybrid edge detection. *Journal Of King Saud University-computer And Information Sciences* **34**, 104–114 (2022).

[8] Abdulhammed, O. Y. A novel approach of steganography by using strong edge detection and chaos theory. *Multimed Tools Appl* **81**, 17875–17888 (2022).

[9] Lee, Y. P. *et al.* High-payload image hiding with quality recovery using tri-way pixel-value differencing. *Inform Sciences* **191**, 214–225 (2012).

[10] Paul, G., Saha, S. K. & Burman, D. A pvd based high capacity steganography algorithm with embedding in non-sequential position. *Multimed Tools Appl* **79**, 13449–13479 (2020).

[11] Prasad, S. & Pal, A. K. An rgb colour image steganography scheme using overlapping block-based pixel-value differencing. *Roy Soc Open Sci* **4**, 161066 (2017).

[12] Chan, C. K. & Cheng, L. M. Hiding data in images by simple lsb substitution. *Pattern Recogn* **37**, 469–474 (2020).

[13] Halder, T., Karforma, S. & Mandal, R. A block-based adaptive data hiding approach using pixel value difference and lsb substitution to secure e-governance documents. *J Inf Process Syst* **15**, 261–270 (2019).

[14] Xu, W. L., Chang, C. C., Chen, T. S. & Wang, L. M. An improved least-significant-bit substitution method using the modulo three strategy. *Displays* **42**, 36–42 (2016).

[15] Mielikainen, J. Lsb matching revisited. *IEEE Signal Proc Let* **13**, 285–287 (2006).

[16] Wu, H. C., Wu, N. I., Tsai, C. S. & Hwang, M. S. High-payload image hiding with quality recovery using tri-way pixel-value differencing. *IEE P-Vis, Image and Sign* **152**, 611–615 (2005).

[17] Khodaei, M. & Faez, K. New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing. *IET Image Process* **6**, 677–686 (2012).

[18] Khodaei, M., Bahram, B. S. & Faez, K. Adaptive data hiding, using pixel-value-differencing and lsb substitution. *Cybernet Syst* **47**, 617–628 (2016).

[19] Swain, G. A steganographic method combining lsb substitution and pvd in a block. *Procedia Computer Science* **85**, 39–44 (2016).

[20] Swain, G. High capacity image steganography using modified lsb substitution and pvd against pixel difference histogram analysis. *Secur Commun Netw* **2018** (2018).

[21] Jung, K. H. Data hiding scheme improving embedding capacity using mixed pvd and lsb

on bit plane. *J Real-Time Image PR* **14**, 127–136 (2018).

[22] Hameed, M. A., Hassaballah, M., Aly, S. & Awad, A. I. An adaptive image steganography method based on histogram of oriented gradient and pvd-lsb technique. *IEEE Access* **7**, 185189–185204 (2019).

[23] Sahu, A. K. & Swain, G. in *An improved method for high hiding capacity based on lsb and pvd* (ed.Hassaballah, M.) *Digital Media Steganography* 41–64 (Elsevier, Amsterdam, 2020).

[24] Hussain, M., Riaz, Q., Saleem, S., Ghafoor, A. & Jung, K. H. Enhanced adaptive data hiding method using lsb and pixel value differencinge. *Multimed Tools Appl* **80**, 20381–20401 (2021).

[25] Liao, X., Wen, Q. Y. & Zhang, J. A steganographic method for digital images with four-pixel differencing and modified lsb substitution. *J Vis Commun Image R* **22**, 1–8 (2011).

[26] Hussain, M., Wahab, A. W. A., Ho, A. T. S., Javed, N. & Jung, K. H. A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement. *Signal Process-Image* **50**, 44–57 (2017).

[27] Weber, A. G. The usc-sipi image database. (2006). Available online http://sipi.usc.edu/database/. Accessed on 17 Mar 2023.

[28] Bas, P. & Furon, T. The bows2 image dataset. (2008). Available online http://bows2.ec-lille.fr/. Accessed on 17 Mar 2023.

[29] Fridrich, J., Goljan, M. & Du, R. Detecting lsb steganography in color, and gray-scale images. *IEEE Multimedia* **8**, 22–28 (2001).

[30] Zhang, X. P. & Wang, S. Z. Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security. *Pattern Recogn Lett* **25**, 331–339 (2004).