An Improved Steganographic Method Based on Least-Significant-Bit Substitution and Modulus Pixel-Value Difference

Hsing-Han Liu, Yu-Fen Lo

Department of Information Management, National Defense University, Taiwan
liu.hansh@gmail.com, mrsc22188@gmail.com

Abstract

An image-domain steganographic method based on a least-significant-bit (LSB) substitution and modulus pixel-value differencing (MPVD) is proposed to hide data in greyscale images. This is a modification of the hybrid LSB/PVD (HLSB/PVD) method proposed by Jung [1]. The cover image is first partitioned into 3 × 3 sized blocks, followed by 5-bit LSB substitution on the central (base) pixel. The differences between the remaining and base pixel values are then computed. HLSB/PVD is applied, with the LSB method used to divide the pixels into higher and lower bits. Data embedding in the lower and higher bits is conducted using 2-bit LSB substitution and MPVD, respectively, leading to a high embedding capacity while retaining an adequate quality in a stego-image.

Embedding experiments conducted with Lena, Baboon, Goldhill, Boat, House, Airplane, Tiffany, and Peppers showed a mean embedding capacity of 1,087,274 bits, (3.46% higher than that of the HLSB/PVD method). At an embedding rate of 4.14 bpp, the peak signal-to-noise ratio of our method is 31.98 dB. Our method improves the embedding capacity and maintains the imperceptibility.

Keywords: Data hiding, LSB substitution, Pixel-value differencing

1 Introduction

Owing to the rapid development of computing technologies and the Internet, the means by which information is transmitted and processed has been completely transformed. For instance, it is possible to obtain nearly any information one might desire (e.g., the latest news, stock market updates, financial news, and shopping and travel information) using only a smartphone. These technological improvements have allowed all communications to be conducted within the shortest time possible, thus providing an immense level of convenience for everyday tasks. However, this convenience is also accompanied by a multitude of information security problems, which must be addressed with seriousness and urgency.

The development of secret communication techniques is a simple consequence of the necessities of everyday life in modern society. Cryptographic systems work by encrypting documents into unreadable ciphertext that are impossible to decrypt when intercepted, ensuring their security. Steganography on the other hand, is the technique of concealing data in ordinary, non-secret files to avoid detection by interested parties. Today, steganography is usually conducted by making minute alterations to multimedia data, thus concealing meaningful information in their insignificant bits. The main focus of this study is image steganography, that is, the use of cover images to hide data. This is achieved by using a steganographic algorithm to minutely alter the pixels of the cover image according to the secret data, thereby producing a stego-image. Owing to the limitations of the human vision system (HVS), it is difficult to detect minute changes of this type in a digital image. To extract the information embedded within a stego-image, the receiver must know the exact steganographic method used by the sender.

At present, image steganography may generally be categorized as spatial- or frequency-domain techniques [2]. Spatial-domain steganography is the most common type of image steganography and is applied by altering pixel values to hide data. These methods are characterized by their high data hiding capacity and low stego-image distortions, and include methods such as a least-significant-bit (LSB) substitution and pixel-value differencing (PVD). Frequency-domain steganography is conducted by converting the spatial-domain pixels of an image into frequency-domain coefficients, and then altering these coefficients to embed the secret information. The most commonly used spatial-domain method is the LSB substitution method proposed by Bender et al. [3], which hides data by exploiting the low sensitivity of the HVS to small changes in luminance. However, the hiding capacity of this method is rather limited because the cover image can be severely distorted if too many bits are used to

*Corresponding Author: Hsing-Han Liu; E-mail: liu.hansh@gmail.com
DOI: 10.53106/160792642021122207014

hide the data. Several modifications have since been made to the LSB technique to reduce the stego-image degradation. For instance, Chan and Cheng [4] proposed the optimal pixel adjustment process (OPAP), which improves the stego-image quality of a conventional LSB substitution by adjusting its non-message bits to reduce embedding errors. To increase the data-hiding capacity of spatial-domain steganography, Wu and Tsai [5] proposed PVD steganography, which uses the difference between two consecutive pixels to determine how many message bits can be embedded in a block. However, the data hiding capacity of this method is low, and it produces characteristic features that are vulnerable to steganalysis attacks. Wu et al. [6] subsequently proposed a high-capacity data hiding scheme based on PVD and LSB substitution, which uses either LSB substitution or PVD, depending on a threshold pixel-value difference value. Wang et al. [7] proposed a steganographic method based on PVD and modulus operations (i.e., the modulus PVD or MPVD method), which hides information by modifying the remainder between two pixels using a modulus operation. Furthermore, the remainder is altered in a way that minimizes image distortions. In 2009, Tsai et al. [8] proposed a reversible image hiding scheme based on histogram shifting for medical images. Compared to the histogram-based method, the quality of the stego image improved about 1.5 dB when the same amounts of secret data were embedded. In 2012, Liu et al. [9] proposed an improved MPVD to reduce its susceptibility to data extraction errors. In 2012, Khodaei and Faez [10] proposed a new adaptive method using LSB substitution and PVD. With this method, the cover image is first partitioned into consecutive non-overlapping 1× 3 sized blocks. Next, k-bits of secret data are then embedded into the central block through LSB substitution. Finally, the differences in pixel values between the central and end pixels of each block are calculated, and an improved PVD method is used to embed secret data. In 2015, Qin et al. [11] proposed a data hiding scheme with reversibility based on exploiting modification direction. Experimental results demonstrate that the scheme can achieve high hiding capacity and satisfactory visual quality. In 2015, Chang et al. [12] proposed a reversible data hiding scheme based on residual histogram shifting for the compressed images of block truncation coding. Experimental results reveal that the proposed scheme provides good image qualities of the embedded images and higher capacity. In 2018, Liu et al. [13] proposed a new data-hiding scheme based on pixel-value differencing (PVD) in which 3-by-3 blocks are used to hide data within nine-pixel groups. The PVD scheme and the side match method are combined to ultimately produce eight groups of pixel-value differences, enabling maximum hiding capacity while maintaining an acceptable peak signal-to-noise ratio (PSNR). Experimental results demonstrate that the hiding capacity of this scheme can reach a maximum of 808,760 bits with a PSNR value of 32.0283 dB, which is difficult to detect with human vision. In 2019, Hu et al. [14] proposed a reversible data hiding technique based on the residual histogram shifting technique. Experimental results demonstrate that the proposed technique not only provides good hiding capacity, but also maintains good image quality of the embedded image. In 2020, Su et al. [15] proposed a reversible data hiding in encrypted compressed images scheme based on AMBTC. Experimental results show that two proposed schemes are able to achieve average embedding rates as large as 0.6 bpp and 0.8 bpp when the block size is set to 2 × 2, respectively.

In 2018, Jung [1] proposed the hybrid LSB/PVD method, which was a high-capacity steganographic method that maintained a reasonable level of stego-image quality. However, with this method, an overflow tended to occur at the stego pixels, and the PVDs of the stego pixels and cover-image pixels often had different value ranges. Thus, the embeddable bits could not be taken out in the correct order. To overcome the problem, we present a method that retains the strengths of the method by Jung [1] and improves upon its weaknesses by applying an improved hybrid PVD-LSB method and an adjusted PVD range table here. In this way, we have created an image steganography method with a high embedding capacity, which maintains the stego-image quality within the range of acceptability of the HVS.

2 Literature Review

2.1 New Adaptive Steganographic Method Using LSB and PVD

Khodaei and Faez [10] proposed an adaptive steganographic method based on LSB substitution and PVD, which partitions the cover image into non-overlapping 1 × 3 sized blocks, and then embeds secret data into the central pixel of each block using LSB substitution and OPAP. The difference values between the central and end pixels of each block are then calculated, and an improved PVD method is used to embed the data, thus enhancing the embedding capacity while maintaining an acceptable level of image quality. We henceforth refer to this method as the new LSB-PVD (NLSB-PVD) method.

Step 1: Partition the cover image into non-overlapping 1 × 3 sized blocks.

Step 2: Define the quantization range table and divide the difference values into lower and higher levels. A Type I range table (Table 1) provides high imperceptibility (i.e., high image quality), whereas a Type II range table (Table 2) gives a high embedding capacity.

Table 1. Type I range table for NLSB-PVD method

Range (R_i)	R_1	R_2	R_3	R_4	R_5
Range	0 to 7	8 to 15	16 to 31	32 to 63	64 to 255
Bits	3	3	3	4	4
Level	Lower level			Higher level	

Step 3: Apply LSB substitution on the central pixel of each block, p_{ic}. First, convert the k-rightmost LSBs of p_{ic} into a decimal value called LSB_i ($k \in \{3, 4, 5, 6\}$). Put the k-leftmost bits of the binary secret data into the k-rightmost LSBs of p_{ic} to obtain p'_{ic}. Finally, convert the k bits of binary secret data into a decimal value called s_{ic}.

Table 2. Type II range table for NLSB-PVD method

Range (R_i)	R_1	R_2	R_3	R_4	R_5
Range	0 to 7	8 to 15	16 to 31	32 to 63	64 to 255
Bits	3	3	4	5	6
Level	Lower level				Higher level

Step 4: Calculate the difference between LSB_i and s_{ic}, d_{ic}.

$$d_{ic} = LSB_i - s_{ic} \tag{1}$$

Step 5: Adjust the pixel value of p'_{ic} using the OPAP.

$$p'_{ic} = \begin{cases} p'_{ic} + 2^k, \text{if } d_{ic} > 2^{k-1} \text{ and } 0 \le p'_{ic} + 2^k \le 255 \\ p'_{ic} - 2^k, \text{if } d_{ic} > -2^{k-1} \text{ and } 0 \le p'_{ic} - 2^k \le 255 \\ p'_{ic}, \text{otherwise} \end{cases} \tag{2}$$

Step 6: Calculate d_{i1} and d_{i2}, which are the difference values between p'_{ic} and the two other pixels in the block, p_{i1} (the first pixel in the block) and p_{i2} (third pixel in the block). Find the ranges that d_{i1} and d_{i2} belong to in the range table to obtain the number of bits that can be hidden within their ranges.

Step 7: Convert the binary bit-stream of the secret data into decimal values, s_{i1} and s_{i2}, and calculate the new pixel-value differences, d'_{i1} and d'_{i2}.

$$\begin{aligned} d'_{i1} &= l_{i1} + s_{i1} \\ d'_{i2} &= l_{i2} + s_{i2} \end{aligned} \tag{3}$$

Step 8: Calculate the new p''_{i1} and p''_{i1} of the first pixel p_{i1} and the new p''_{i2} and p''_{i2} of the third pixel p_{i2}.

$$\begin{aligned} p''_{i1} &= p'_{ic} - d'_{i1} \\ p''_{i2} &= p'_{ic} + d'_{i1} \\ p''_{i2} &= p'_{ic} - d'_{i2} \\ p'''_{i2} &= p'_{ic} - d'_{i2} \end{aligned} \tag{4}$$

Step 9: Readjust the new values of p_{i1} and p_{i2} after the embedding process.

$$\begin{aligned} p'_{i1} &= \begin{cases} p''_{i1}, \text{if } |p_{i1} - p''_{i1}| < |p_{i1} - p'''_{i1}| \text{ and } 0 \le p''_{i1} \\ p''_{i1}, \text{otherwise} \end{cases} \\ p'_{i2} &= \begin{cases} p''_{i2}, \text{if } |p_{i2} - p''_{i2}| < |p_{i2} - p'''_{i2}| \text{ and } 0 \le p''_{i2} \\ p''_{i2}, \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

2.2 Improved Steganography Embedding Capacity Using Mixed PVD and LSB

Jung [1] proposed a steganographic technique based on a hybrid LSB/PVD approach. Previously, LSB and PVD were employed separately, or at best, combined in a complementary manner. Using the method developed by Jung [1], the pixels of the cover image are divided into lower and higher bit planes based on the LSB method. A 2-bit LSB substitution is then applied to the lower bit plane, whereas PVD is used for the 6 bits of the higher bit plane. This approach allows for a high embedding capacity while maintaining a high level of image fidelity. This method will henceforth be referred to as the hybrid LSB/PVD (HLSB/PVD) method. The procedures of this method are as follows:

Step 1: Partition the cover image into non-overlapping 1 × 2 sized blocks.

Step 2: Calculate the quotient (higher bits) and remainder (lower bits) of the pixel values and set the number of bits being replaced through LSB substitution into 2 bits ($k = 2$).

$$\begin{aligned} (P_i^m, P_{i+1}^m) &= (p_i \ div \ 2^k, p_{i+1} \ div \ 2^k) \\ (p_i^l, p_{i+1}^l) &= (p_i \ mod \ 2^k, p_{i+1} \ mod \ 2^k) \end{aligned} \tag{6}$$

Step 3: Calculate the difference between the quotients of two pixels, d_i^m.

$$d_i^m = |P_{i+1}^m - P_i^m| \tag{7}$$

Step 4: Define the quantization range table, as shown in Table 3.

Table 3. Range table of HLSB/PVD method

Range (R_i)	R_1	R_2	R_3	R_4
Range	0 to 7	8 to 15	16 to 31	32 to 63
Embeddable bits (n)	3	3	4	5

Step 5: Find the number of embeddable bits n from the range table, and then read n-bits from the binary bit-stream of the secret data. Convert these n-bits into the decimal b_i^m and calculate a new difference value, $d_i'^m$.

$$d_i'^m = l_i + b_i^m . \tag{8}$$

Step 6: Use Equation (9) to calculate the difference between the new and old difference values ($d_i'^m$ .and d_i^m), and then apply Equation (10) to calculate the values of the stego-pixels, $(p_i'^m, p_{i+1}'^m)$.

$$m = |d_i'^m, d_i^m| \tag{9}$$

$$(p_i'^m, p_{i+1}'^m) = \begin{cases} \left(p_i^m - \left\lceil \frac{m}{2} \right\rceil, P_{i+1}^m + \right), \\ \quad if\ d_i^m\ is\ odd \\ \left(p_i^m - \left\lfloor \frac{m}{2} \right\rfloor, P_{i+1}^m + \right), \\ \quad if\ d_i^m\ is\ even \end{cases} \tag{10}$$

Step 7: Embed k-bits of secret data into the LSBs of the pixels in each 1 × 2 sized block by reading k-bits from the binary bit-stream of the secret data and converting them into decimals, b_i^{l1} and b_i^{l2}. Finally, calculate the values of the stego-pixels (P_i', P_{i+1}').

$$(P_i', P_{i+1}') = (p_i'^m \times 2^k) + \sum_{i=0}^{n-1} b_i^{l1}, (P_{i+1}'^m \times 2^k) + \sum_{i=n}^{2n-1} b_i^{l2} \tag{11}$$

The corresponding extraction algorithm is as follows:

Step 1: Partition the stego-image into two consecutive, adjacent and non-overlapping blocks. Extract k-bits from the stego-pixels (P_i', P_{i+1}'), and calculate the new pixel values $(p_i'^m, p_{i+1}'^m)$.

$$(p_i'^m, p_{i+1}'^m) = \begin{cases} (P_i' - (P_i' \bmod 2^k))\ div\ 2^k \\ (P_{i+1}' - (P_{i+1}'\ mod\ 2^k))\ div\ 2^k \end{cases} \tag{12}$$

Step 2: Calculate the difference between two adjacent pixels, and then find the range R_i, minimum range value l_i, and number of embedded bits n for that difference in the range table (Table 3). Finally, calculate the secret information b_i^m and convert it into a binary form to obtain the secret data.

$$\begin{aligned} d_i'^m &= |p_{i+1}'^m - p_i'^m| \\ b_i^m &= |d_i'^m - l_k| \end{aligned} \tag{13}$$

2.3 Improved Steganography Method Based on LSB and PVD

In 2020, Liu et al. [16] proposed a new steganographic method based on LSB substitution and PVD. First, the cover image is partitioned into 1 × 3 sized non-overlapping blocks, with the central pixel being designated as the base pixel. A 3-bit LSB substitution is conducted on the base pixel, and the differences between the values of the base pixel (after LSB substitution) and the two other pixels in the block are then calculated. An improved version of the MPVD method proposed by Wang et al. [7] is then used to embed the secret data. The procedures of this method, called the new LSB/MPVD method (NLSB/MPVD), are as follows:

Step 1: Partition the cover image C into 1 × 3 neighboring and non-overlapping pixel blocks.

Step 2: Designate the central pixel P_{ic} as the base pixel.

Step 3: Let the LSB substitution be 3 bits (k = 3) and substitute the lowest 3 bits of the binary base pixel with the 3-bit binary secret data to obtain a base pixel value P_{ic}' after embedding is performed. Further, transform the lowest 3 bits of the original base pixel P_{ic} into a decimal value LSB_i and also transform the secret data just embedded in the base pixel into a decimal value s_{ic}.

Step 4: Calculate the difference value d_i between LSB_i and s_{ic} as follows:

$$d_i = LSB_i - s_{ic} \tag{14}$$

Step 5: Use OPAP to adjust base pixel P_{ic}' by:

$$p_{ic}' = \begin{cases} P_{ic}' + 2^3, if\ d_i > 2^2\ and\ 0 \le P_{ic}' + 2^3 \le 255 \\ P_{ic}' - 2^3, if\ d_i > -2^2\ and\ 0 \le P_{ic}' - 2^3 \le 255 \\ P_{ic}', otherwise \end{cases} \tag{15}$$

Step 6: Compute the difference values d_{i1} and d_{i2} between the pixels P_{i1} and P_{i2} and the base pixel value after embedding P_{ic}'. Determine the interval R_i of the difference values (see Table. 4) and obtain the number of bits n needed to hide the secret information.

Table 4. Interval setting for pixel difference values

Range(R_i)	R_1	R_2
Range[l_i, u_i]	[0, 31]	[32, 255]
Embedded Bits (n)	4	5

Step 7: Calculate the sums of P_{ic}' with P_1 and P_2, respectively, and compute the remainders F_{remL} and F_{remR} using the mod 2^n operation as follows:

$$\begin{aligned} F_{remL} &= (P_1 + P_{ic}') \bmod 2^n \\ F_{remR} &= (P_2 + P_{ic}') \bmod 2^n \end{aligned} \tag{16}$$

Step 8: Denote the decimal values of the secret information embedded into pixels P_1 and P_2 as b_{iL} and b_{iR}. Compute the m_{ia} and m_{ib} values according to Equation (17), denoted as m_{iaL}, m_{iaR}, m_{ibL}, and m_{ibR}:

$$\begin{aligned} m_{iaL} &= |F_{remL} - b_{iL}| \\ m_{iaR} &= |F_{remR} - b_{iR}| \\ m_{ibL} &= 2^n - |F_{remL} - b_{iL}| \\ m_{ibR} &= 2^n - |F_{remR} - b_{iR}| \end{aligned} \tag{17}$$

Step 9: According to the values of F_{remL}, F_{remR}, m_{iaL}, m_{iaR}, m_{ibL}, and m_{ibR}, calculate the pixel values in the four cases, as shown in Equation (18) (F_{rem} is F_{remL} or F_{remR}, m_{ia} is m_{iaL} or m_{iaR}, m_{ib} is m_{ibL} or m_{ibR}, b_i is b_{iL} or b_{iR}, P_i is P_{i1} or P_{i2}, and P_i' is P_{i1}' or P_2'):

$$\begin{aligned} &Case\,1: F_{rem} > b_i \ and \ m_{ia} \le (2^n/2) \\ &\quad P_i' = P_i - m_{ia} \\ &Case\,2: F_{rem} > b_i \ and \ m_{ia} > (2^n/2) \\ &\quad P_i' = P_i + m_{ib} \\ &Case\,3: F_{rem} \le b_i \ and \ m_{ia} \le (2^n/2) \\ &\quad P_i' = P_i + m_{ia} \\ &Case\,4: F_{rem} \le b_i \ and \ m_{ia} > (2^n/2) \\ &\quad P_i' = P_i - m_{ib} \end{aligned} \quad \textbf{(18)}$$

Step 10: If pixel overflow occurs after embedding, adjust the values as follows:

$$p_i' = \begin{cases} P_i' + 2^n, if\ P_i \ge 0 \ and \ P_{ic}' \ge 0 \ and \ P_i' < 0 \\ P_i' - 2^n, if\ P_i \le 255 \ and \ P_i' \le 255 \ and \ P_i' > 255 \\ P_{ic}', otherwise \end{cases} \quad \textbf{(19)}$$

Step 11: After hiding the secret data as instructed in the formulas above, confirm whether the intervals of the pixel difference values are the same before and after embedding. If the intervals are unequal, adjust the values using Equations (20) and (21) to avoid errors in data extraction. For example, if the pixel difference value $d_i \in [0, 31]$ while the difference value after embedding is $d_i' > 31$, use Equation (20); if the pixel difference value $d_i \in [32, 255]$ while the difference value after embedding is $d_i' < 32$, use Equation (21).

$$P_i' = \begin{cases} P_i' - 2^n, if\ P_i' > P_{ic}' \\ P_i' + 2^n, otherwise \end{cases} \quad \textbf{(20)}$$

$$P_i' = \begin{cases} P_i' + 2^n, if\ P_i' \leqq P_{ic}' \\ P_i' - 2^n, otherwise \end{cases} \quad \textbf{(21)}$$

Step 12: Check for any overflow again. In the case of an overflow, adjust the values using Equation (22) to avoid extraction errors.

$$P_i'' = \begin{cases} P_i' + 2 \times 2^n, if\ P' < 0 \\ P_i' - 2 \times 2^n, if\ P' > 255 \end{cases} \quad \textbf{(22)}$$

The extraction process is less complex than that of embedding, and is explained as follows.

Step 1: Partition the embedded image into three consecutive pixel blocks.

Step 2: Designate the central pixel P_{ic}' as the base pixel and extract the secret data from it.

Step 3: Calculate the pixel difference values between the base pixel and the two neighboring pixels, respectively, and confirm whether the difference values fall into the intervals. Extract the decimal secret information b_i using Equation (23) and transform it into a binary value.

$$b_i = (P_i' + P_{ic}') \ mod \ 2^n \quad \textbf{(23)}$$

3 Research Methodology and Structure

The aim of this study is to maximize the embedding capacity while retaining an acceptable level of image quality in a stego-image. During our literature review, we found that the HLSB/PVD technique has room for improvement. To this end, we presented an improved steganographic technique based on HLSB/PVD. The remainder of this paper is organized as follows: First, we discuss the structure of this study, followed by the data embedding and extraction procedures of our method. Finally, we experimentally validate our steganographic technique.

3.1 Research Structure

To enhance the embedding capacity and stego-image quality, the cover image is partitioned into 3 × 3 sized blocks. A 5-bit LSB substitution is then applied to the central pixel, which will no longer be altered after this step. Next, HLSB/PVD is conducted. The embedding of data in the higher bit plane is applied by calculating the difference between the central pixel Q_{ic} and the remaining pixels in the block (Q_{i1}–Q_{i8}), and then using the MPVD method of Liu et al. [16], with alterations being conducted in a way that minimizes image distortions. A 2-bit LSB substitution is then applied on R_{i1}–R_{i8} to embed secret data in the lower bit plane. This approach allows 17 sets of secret data to be embedded in each block, i.e., one set in the base pixel and 16 sets in the higher (Q_{i1}–Q_{i8}) and lower (R_{i1}–R_{i8}) bit planes of P_{i1}–P_{i8}.

3.2 Embedding and Extraction Process

The procedures of the steganographic method proposed in this work are as follows:

Step 1: Partition the cover image into consecutive non-overlapping 3 × 3 sized blocks.

Step 2: Define the central pixel, P_{ic}, as the base pixel (see Figure 1). Apply a 5-bit LSB substitution on P_{ic} and convert its value into a decimal called LSB_i. The 5-bit secret data that are to be embedded in P_{ic} are also converted into a decimal, called s_i. Calculate the difference between LSB_i and s_i, i.e., d_{ic}.

$$d_{ic} = LSB_i - s_i \quad \textbf{(24)}$$

P_{i1}	P_{i2}	P_{i3}
P_{i4}	P_{ic}	P_{i5}
P_{i6}	P_{i7}	P_{i8}

Figure 1. Illustration of a pixel block

Step 3: Calculate P'_{ic}, which is the stego-pixel value of P_{ic}.

$$P'_{ic} = P_{ic} - (P_{ic} \ mod \ 2^5) + s_i \quad \textbf{(25)}$$

Step 4: Apply the OPAP to P'_{ic}.

$$P'_{ic} = \begin{cases} P'_{ic} + 2^5, if \ d_{ic} > 2^4 \ and \ 0 \le P'_{ic} + 2^5 \le 255 \\ P'_{ic} - 2^5, if \ d_{ic} > -2^4 \ and \ 0 \le P'_{ic} - 2^5 \le 255 \\ P'_{ic}, otherwise \end{cases} \quad \textbf{(26)}$$

Step 5: Calculate the quotient (higher bits) and remainder (lower bits) between pixels P_{i1}–P_{i8} and P'_{ic}. Define the number of bits to be exchanged in an LSB substitution as 2 (k = 2).

$$\begin{aligned} &Q_i = P_i \ div \ 2^k, i \in [1\text{-}8] \\ &Q'_i = P'_{ic} \ div \ 2^k \\ &Q'_{ic} = P'_{ic} \bmod 2^k, R_i = P_i \bmod 2^k, i \in [1\text{-}8] \end{aligned} \quad \textbf{(27)}$$

Step 6: Calculate the pixel-value differences between Q'_{ic} and Q_{i1}–Q_{i8}.

$$d_i = |Q'_{ic} - Q_i|, i \in [1\text{-}8] \quad \textbf{(28)}$$

Step 7: Define the quantization range table to ascertain the number of secret data bits n that can be embedded in each range (see Table 5).

Table 5. Range table for proposed method

Range	R_1	R_2
Range $[l_i, u_i]$	[0, 7]	[8, 63]
Bits	2	3

Step 8: Calculate the sum between Q_{i1}–Q_{i8} and P'_{ic}, and compute their remainders using the mod 2^n operation

$$F_{rem} = (Q_i + Q'_{ic}) \bmod 2^n, i \in [1\text{-}8] \quad \textbf{(29)}$$

Step 9: Read n-bits from the secret data and convert these data into decimals called b_i. Calculate m_{ia} and m_{ib}.

$$\begin{aligned} &m_{ia} = |F_{rem} - b_i|, i \in [1\text{-}8] \\ &m_{ib} = 2^n - |F_{rem} - b_i|, i \in [1\text{-}8] \end{aligned} \quad \textbf{(30)}$$

Step 10: Compute the pixel values Q'_i based on the values of m_{ia} and m_{ib} (where F_{rem} represents F_{rem1}–F_{rem8}, m_{ia} represents m_{ia1}–m_{ia8}, m_{ib} represents m_{ib1}–m_{ib8}, b_i represents b_{i1}–b_{i8}, Q_i represents Q_{i1}–Q_{i8}, and Q'_i represents $Q'_{i1} - Q'_{i8}$).

$$\begin{aligned} &Case\,1: F_{rem} > b_i \ and \ m_{ia} \le (2^n / 2) \\ &\quad Q'_i = Q_i - m_{ia} \\ &Case\,2: F_{rem} > b_i \ and \ m_{ia} > (2^n / 2) \\ &\quad Q'_i = Q_i + m_{ib} \\ &Case\,3: F_{rem} \le b_i \ and \ m_{ia} \le (2^n / 2) \\ &\quad Q'_i = Q_i + m_{ia} \\ &Case\,4: F_{rem} \le b_i \ and \ m_{ia} > (2^n / 2) \\ &\quad Q'_i = Q_i - m_{ib} \end{aligned} \quad \textbf{(31)}$$

Step 11: If Q'_i overflows after the embedding process, adjust Q'_i using Equation (32).

$$Q'_i = \begin{cases} Q'_i + 2^n, if \ Q_i \ge 0 \ and \ Q'_{ic} \ge 0 \ and \ Q'_i < 0 \\ Q'_i - 2^n, if \ Q_i \le 63 \ and \ Q'_{ic} \le 63 \ and \ Q'_i > 63 \end{cases} \quad \textbf{(32)}$$

Step 12: Check whether the pixel-value differences before and after the embedding of secret data belong to the same range. If they belong to different ranges, adjust the value of Q'_{ic} according to the following: Use Equation (33) if the original pixel value difference is $d_i \in [0, 7]$ and the post-embedding pixel value difference is $d'_i > 7$, and use Equation (34) if the original pixel value difference is $d_i \in [8, 63]$ and the post-embedding pixel value difference is $d'_i < 8$.

$$Q'_i = \begin{cases} Q'_i - 2^n, if \ Q'_i > Q'_{ic} \\ Q'_i + 2^n, otherwise \end{cases} \quad \textbf{(33)}$$

$$Q'_i = \begin{cases} Q'_i + 2^n, if \ Q'_i > Q'_{ic} \\ Q'_i - 2^n, otherwise \end{cases} \quad \textbf{(34)}$$

Step 13: If a pixel value overflow still occurs, adjust the pixel values using Equation (35).

$$Q''_i = \begin{cases} Q'_i + 2 \times 2^n, if \ Q'_i < 0 \\ Q'_i - 2 \times 2^n, if \ Q'_i > 63 \end{cases} \quad \textbf{(35)}$$

Step 14: Read k bits from the binary secret data bit-stream and convert these bits into the decimal R'_i. Finally, calculate the stego-pixel value P'_i.

$$P'_i = Q'_{ic} \times 2^k + R'_i|, i \in [1\text{-}8] \quad \textbf{(36)}$$

The corresponding data extraction procedure is as follows:

Step 1: Partition the stego-image into consecutive non-overlapping 3 × 3 sized blocks.

Step 2: Convert P'_{ic} into binary values, extract the five lowest bits, and then compute Q'_{ic} and $Q'_{i1} - Q'_{i8}$.

$$Q'_i = P'_i \, div \, 2^k, i \in [1\text{-}8]$$
$$Q'_{ic} = P'_{ic} \, div \, 2^k \quad (37)$$

Step 3: Use Equation (38) to extract the decimal secret data b_i and convert it into a binary value.

$$b_i = (Q'_i + Q'_{ic}) \, mod \, 2^n \quad (38)$$

Step 4: Extract the two lowest bits of pixels P'_{i1} and P'_{i2} by converting their values into binary values.

3.3 Description of Embedding and Extraction Example

The embedding procedure of our steganographic method is illustrated in Figure 2. First, the cover image is partitioned into 3 × 3 sized blocks. Suppose the pixel values of the block are $P_{i1} = 70$, $P_{i2} = 72$, $P_{i3} = 70$, $P_{i4} = 68$, $P_{i5} = 66$, $P_{i6} = 64$, $P_{i7} = 64$, $P_{i8} = 68$, and $P_{ic} = 65$, and the to-be-embedded secret data are s = $(11110110011000111110110110101100000110)_2$. Let k = 2. In addition, a 5-bit LSB substitution is first performed on the base pixel $P_{ic} = 65 = (01000001)_2$, which gives $P'_{ic} = 62$. Equation (24) is then used to calculate the difference between LSB_i and s_{ic}, where $LSB_i = (00001)_2$ and $s_{ic} = (11110)_2 = 30$. The pixel quotients are calculated using Equation (27), which yields $Q_{i1} = 17$, $Q_{i2} = 18$, $Q_{i3} = 17$, $Q_{i4} = 17$, $Q_{i5} = 16$, $Q_{i7} = 16$, $Q_{i8} = 16$, and $Q'_{ic} = 15$. The differences between Q'_{ic} and $Q_{i1} - Q_{i8}$ are then $d_{i1} = 1$, $d_{i2} = 2$, $d_{i3} = 1$, $d_{i14} = 2$, $d_{i15} = 1$, $d_{i16} = 1$, $d_{i17} = 1$, and $d_{i8} = 1$. Based on Table 7, 2 bits will be embedded in each pixel. The to-be-embedded binary secret data is first converted into decimals, thus yielding $b_{i1} = (11)_2 = 3$, $b_{i2} = (00)_2 = 0$, $b_{i3} = (11)_2 = 3$, $b_{i4} = (00)_2 = 0$, $b_{i5} = (01)_2 = 1$, $b_{i6} = (11)_2 = 3$, $b_{i7} = (11)_2 = 3$, and $b_{i8} = (01)_2 = 1$. The remainders are then calculated using Equation (29), which gives $F_{rem1} = (17 + 15) \bmod 2^2 = 0$, $F_{rem2} = 1$, $F_{rem3} = 0$, $F_{rem4} = 0$, $F_{rem5} = 3$, $F_{rem6} = 3$, $F_{rem7} = 3$, and $F_{rem8} = 3$. Equation (30) yields $m_{ia1} = |0 - 3| = 3$, $m_{ia2} = 1$, $m_{ia2} = 1$, $m_{ia3} = 3$, $m_{ia4} = 0$, $m_{ia5} = 2$, $m_{ia6} = 0$, $m_{ia7} = 0$, $m_{ia8} = 2$, and $m_{ib1} = 2^2 - |0 - 3| = 1$, $m_{ib2} = 3$, $m_{ib3} = 1$, $m_{ib4} = 4$, $m_{ib5} = 2$, $m_{ib6} = 4$, $m_{ib7} = 3$, and $m_{ib8} = 2$. Because $F_{rem1} \le b_1$ and $m_{ia1} > 2$, it follows from Equation (31) that $Q'_{i1} = 17 - 1 = 16$. We then check the pixel value differences to see if they are still within the same range after the embedding of secret data. No change is required because they are within the same range ($Q'_{i2} = 17$, $Q'_{i3} = 16$, $Q'_{i4} = 17$, $Q'_{i5} = 14$, $Q'_{i6} = 16$, $Q'_{i7} = 16$, and $Q'_{i8} = 14$). Here, 2 bits of secret data are embedded into the lower bits of all pixels in the block, which are then converted into decimal form: $R'_{i1} = (10)_2 = 2$, $R'_{i1} = (10)_2 = 2$, $R'_{i2} = (11)_2 = 3$, $R'_{i3} = (01)_2 = 1$, $R'_{i4} = (01)_2 = 1$, $R'_{i5} = (10)_2 = 2$, $R'_{i6} = (00)_2 = 0$, $R'_{i7} = (01)_2 = 2$, and $R'_{i8} = (10)_2 = 2$. Finally, the stego-pixel values are calculated using Equation (36): $P'_{i1} = Q'_1 \times 2^2 + R'_{i1} = 16 \times 4 + 2 = 66$, $P'_{i2} = 17 \times 4 + 3 = 71$, $P'_{i3} = 65$, $P'_{i4} = 69$, $P'_{i4} = 58$, $P'_{i6} = 64$, $P'_{i7} = 65$, $P'_{i8} = 58$, and $P'_{ic} = 62$.

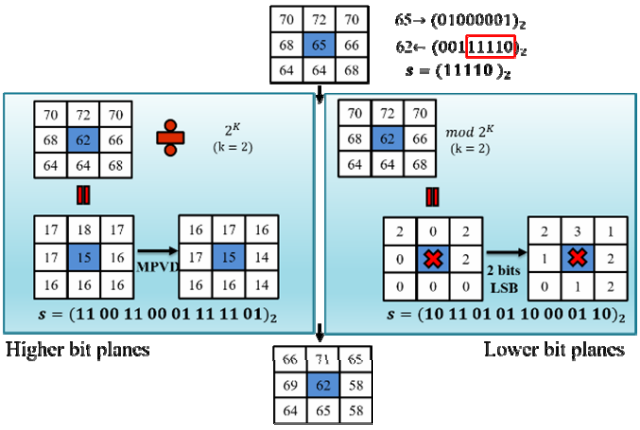


Figure 2. Embedding process of proposed method

An example of a data extraction is shown in Figure 3. Suppose that the stego-pixel values are $P'_{i1} = 66$, $P'_{i2} = 71$, $P'_{i3} = 65$, $P'_{i4} = 69$, $P'_{i5} = 58$, $P'_{i6} = 64$, $P'_{i7} = 65$, $P'_{i8} = 58$, and $P'_{ic} = 62$. To extract the secret data, begin by converting P'_{ic} into a binary form, $(00111110)_2$. Take the 5 lowest bits, i.e., $(11110)_2$. Calculate the quotients of the pixel values using Equation (37), which gives $Q'_{i1} = 66/4 = 16$, $Q'_{i2} = 17$, $Q'_{i3} = 16$, $Q'_{i4} = 17$, $Q'_{i5} = 14$, $Q'_{i6} = 16$, $Q'_{i7} = 16$, $Q'_{i8} = 14$, and $Q'_{ic} = 15$. Use Equation (38) to calculate the secret data, $b_{i1} - b_{i8}$: $b_{i1} = (15 + 16) \bmod 2^2 = 3 = (11)_2$, $b_{i2} = (17 + 15) \bmod 4 = 0 = (00)_2$, $b_{i3} = 3 = (11)_2$, $b_{i4} = 0 = (00)_2$, $b_{i5} = 1 = (01)_2$, $b_{i6} = 3 = (11)_2$, $b_{i7} = 3 = (11)_2$, and $b_{i8} = 1 = (01)_2$. Now take the 2 lowest bits of $R'_{i1} - R'_{i8}$, which gives $(11\ 00\ 11\ 00\ 01\ 11\ 11\ 01)_2$. Finally, merge the secret data to obtain the whole secret message, s = $(11110110011000111110\ 1011010110000110)_2$.

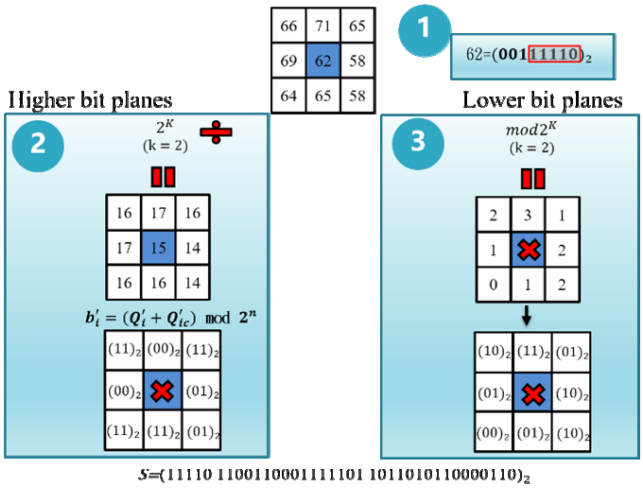


Figure 3. Extraction process of proposed method

4 Experimental Results

In this section, experimental validation of the improved HLSB/MPVD steganographic method proposed in this study is described, and the results are compared to existing steganographic methods in terms of the embedding capacity and stego-image quality. The security of our steganographic method tested using steganalysis attacks, including analyses of the pixel difference histogram and content-selective residuals, is then provided.

4.1 Experimental Environment

Eight 512 × 512 sized standard grayscale images (Baboon, Boat, House, Airplane, Goldhill, Lena, Tiffany, and Peppers) were used to evaluate our steganographic method (see Figure 4). These images include both smooth and rough textures. The secret information embedded in these images were strings comprised of random 0s and 1s, and the results were evaluated in terms of the embedding capacity, peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM).

Figure 4. Experimental images of this study

4.2 Analysis of Experimental Results

Table 6 compares the embedding capacities of the original PVD method [5], NLSB/PVD method [10], side match/PVD method [13], HLSB/PVD method [1], and NLSB/MPVD method [16]. The PSNR and SSIM of these methods are shown in Table 7.

Table 6. Embedding capacities of several steganographic methods (in bits)

Method / Image	Original PVD (2003)	NLSB/PVD (2012)	Side Match PVD (2018)	HLSB/PVD (2018)	NLSB/MPVD (2020)	Our method
Lena	409,807	809,966	712,112	1,049,742	962,452	1,078,559
Airport	409,834	809,262	717,511	1,050,973	963,471	1,082,035
Boat	420,638	820,391	735,413	1,051,124	965,764	1,089,173
Goldhill	411,896	813,968	720,574	1,049,093	962,633	1,082,662
Baboon	457,105	886,516	808,760	1,054,327	977,179	1,122,606
Tiffany	407,365	806,847	709,764	1,049,513	961,484	1,076,290
House	420,123	818,580	729,785	1,051,474	965,871	1,088,510
Peppers	407,643	802,228	713,062	1,050,571	961,563	1,078,357
Average	**418,051**	**820,970**	**730,873**	**1,050,852**	**965,052**	**1,087,274**

Table 7. PSNR and SSIM comparison among several steganographic methods

Method / Image	Original PVD (2003)	NLSB/PVD (2012)	Side Match PVD (2018)	HLSB/PVD (2018)	NLSB/MPVD (2020)	Our method
Lena	41.18 (0.978)	37.63 (0.936)	36.70 (0.929)	33.21 (0.805)	35.35 (0.896)	32.45 (0.815)
Airport	40.20 (0.973)	37.53 (0.931)	36.19 (0.923)	33.19 (0.787)	35.33 (0.882)	32.34 (0.797)
Boat	39.71 (0.981)	36.53 (0.941)	34.97 (0.941)	32.84 (0.846)	34.91 (0.918)	31.87 (0.852)
Goldhill	41.00 (0.983)	37.55 (0.944)	36.23 (0.944)	32.54 (0.859)	35.31 (0.920)	32.14 (0.862)
Baboon	36.96 (0.987)	36.29 (0.934)	32.04 (0.934)	31.74 (0.927)	33.93 (0.956)	30.63 (0.919)
Tiffany	40.89 (0.974)	37.79 (0.922)	36.84 (0.922)	32.47 (0.780)	34.76 (0.876)	31.87 (0.781)
House	39.15 (0.977)	36.44 (0.937)	35.47 (0.937)	32.73 (0.838)	34.96 (0.908)	31.90 (0.838)
Peppers	40.61 (0.978)	37.97 (0.927)	34.83 (0.927)	33.55 (0.806)	34.89 (0.900)	32.03 (0.813)
Average	**39.96 (0.978)**	**37.21 (0.932)**	**35.40 (0.932)**	**32.78 (0.831)**	**34.93 (0.907)**	**31.98 (0.838)**

It can be seen that the embedding capacity of the original PVD method is relatively low, which may be attributed to the fact that it only uses one steganographic technique. The NSLB/PVD, side match/PVD, HLSB/PVD, and NLSB/MPVD methods, by contrast, have higher embedding capacities than the original PVD method because they combine two distinct steganographic techniques. With our method, the cover image is partitioned into 3 × 3 sized blocks, which can be embedded with 17 sets of secret data. The biggest difference between our method and the HLSB/PVD approach is that the former uses MPVD for the higher bits and an adjusted PVD range table, which allows our method to simultaneously increase the embedding capacity while ensuring an acceptable level of stego-image quality.

Table 7 compares the PSNR and SSIM (which is shown in brackets) of the stego-images produced by each of the aforementioned methods. It can be observed that the original PVD method, NLSB/PVD method, and side match/PVD methods have high PSNR/SSIM values (image quality) owing to their low embedding capacities. The steganographic method of this study has a mean PSNR of 31.98 dB and mean SSIM of 0.838, which are approximately 0.8 dB and 0.007 lower than those of the HLSB/PVD method. However, our method has a higher embedding capacity (3.46% higher, or 36,422 bits) and significantly less pixel overflow. Our method also has an acceptable level of stego-image quality (31.98 dB on average).

To test the generalizability of our method, an embedding capacity and image quality test were conducted using 10,000 512×512 sized greyscale images from the BossBase dataset. The results shown in Table 8 indicate that our method has an average embedding capacity of 1,082,046 bits and a mean PSNR of 31.698 dB. As compared to the HLSB/PVD method, the embedding capacity of our method is 3.08% larger (32,401 more bits), whereas its PSNR and SSIM are 2.3% (0.777 dB) and 0.008 lower, respectively. Therefore, it may be concluded that our method has a larger embedding capacity than all other steganographic methods, and is able to maintain an acceptable level of stego-image quality.

Table 8. Averaged results from BossBase image database

Method / Image	Original PVD (2003)	NLSB/PVD (2012)	Side Match PVD (2018)	HLSB/PVD (2018)	NLSB/MPVD (2020)	Our method
Embedding capacity	409,780	805,717	719,708	1,049,645	962,718	1,082,046
PSNR	40.858	35.146	35.502	32.475	34.906	31.698
SSIM	0.969	0.919	0.909	0.783	0.878	0.775

4.3 Security Analysis

Because the main purpose of steganography is to transmit secret messages through the Internet without being noticed, the distortions caused by steganography must be imperceptible to the HVS and common steganalysis techniques. The resistance of our steganographic method to steganalysis is investigated in this section. To this end, two different steganalysis attacks, i.e., a pixel difference histogram (PDH) analysis and content-selective residuals, will be conducted on stego-images produced by our method.

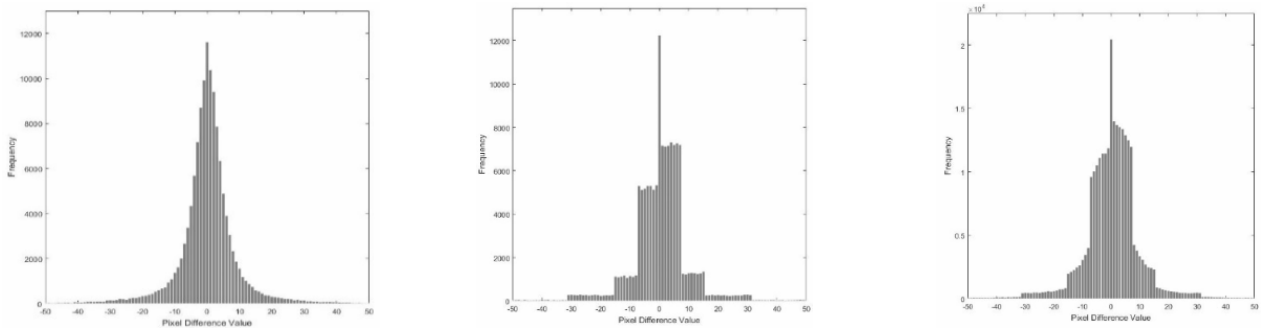
4.3.1 Pixel Difference Histogram Analysis

The shades of grey of the pixels in an ordinary grayscale image are usually represented by 8-bit values (0-255). In a PDH analysis, the number of times a certain adjacent-pixel difference value appears in an image is counted. The PDH of an ordinary image is shown in Figure 5(a), where the horizontal and vertical axes are the pixel difference values and their corresponding frequencies, respectively. An ordinary image will have a normal (Gaussian) distribution, whereas a stego-image might exhibit non-normal, stepped distributions (Figures 5(b), (c), and (d)). Hence, significant deviations from a normal distribution are opportunities for a steganalysis, and can be used to determine whether an image contains secret information.

In this regard, this study performs a security analysis on the original PVD, the new Side Match/PVD, the mix PVD/LSB, the new LSB/PVD, and our method, by examining the change in the features of the PDH after embedding the cover image (see the PDH of the cover image and that of the embedded image in Figure 5). The PDH of the cover image has a distribution that is nearly normal. However, as shown in Figure 5 (b)(c)(d), the height and width of the PDHs change after embedding and some are no longer normally distributed. Such changes in PDH features after embedding constitute an opportunity for steganalysis. In comparison, as shown in (f) of Figure 5, the PDH generated from our method still conform to a normal distribution, and are thus relatively successful at defending against steganalysis based on the PDH features.

4.3.2 RS Analysis

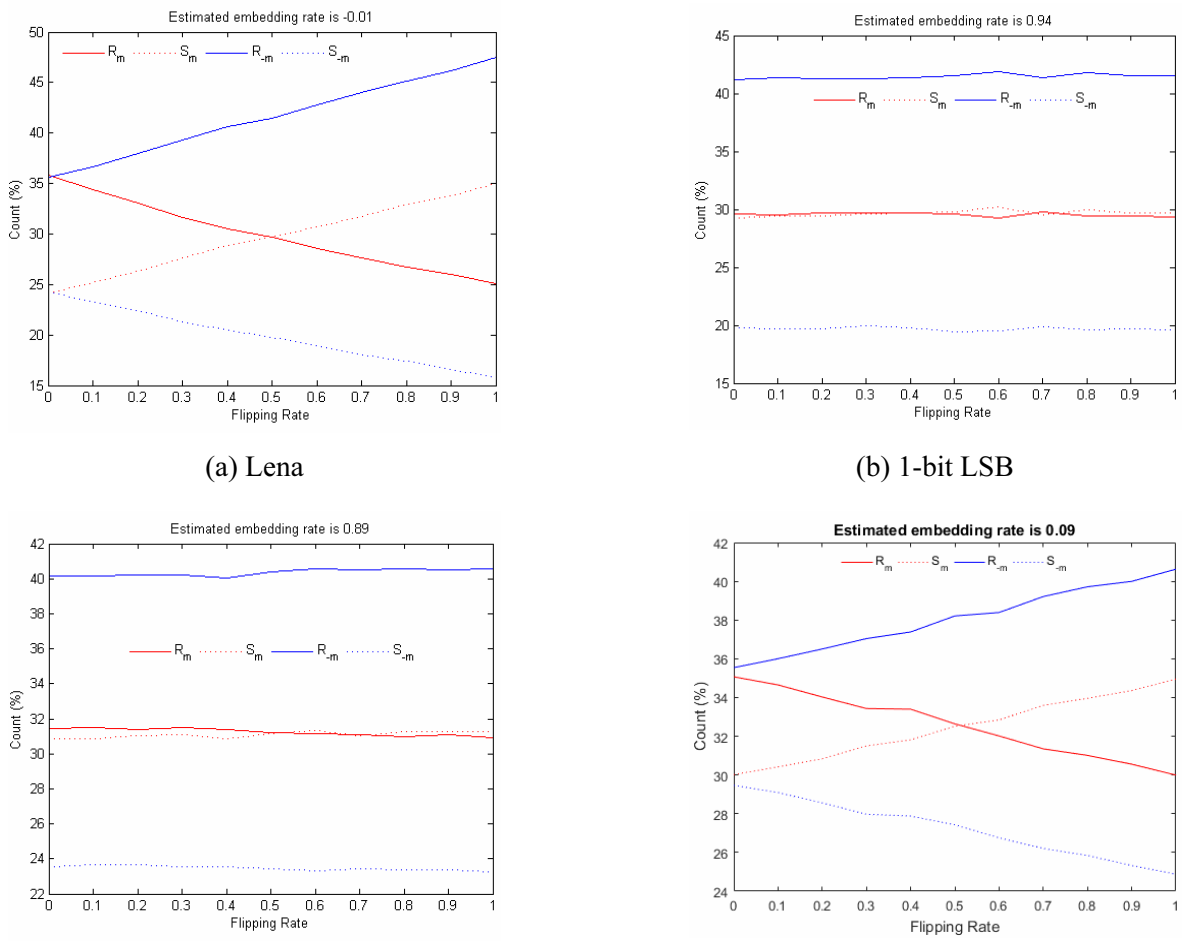
The security of the proposed method against the statistical RS detection technology [17] is depicted in Fig. 6. Figure 6(a) shows the results for the case wherein RS detection technology is used for analysis of the Lena raw image. Figures 6(b) and (c), respectively, show the results for the cases wherein 1-bit LSB and 3-bit LSB Lena Stego images were used



(a) Cover image (b) Original PVD (c) Side Match/PVD

(d) HLSB/PVD (e) NLSB/MPVD (f) Proposed method

Figure 5. PDH of pixel difference values with cover image and different steganographic methods



(a) Lena (b) 1-bit LSB

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

Figure 6. Results of security analysis of the proposed method

for the analyses. Figure 6(d) shows the result for the case in which the Lena Stego image generated using the proposed method was used for the analysis. From the RS detection results shown in Figs. 6(a)-(d), we can see that the evaluated embedding rates were –0.01, 0.94, 0.89, and 0.09, respectively. This shows that RS detection technology can effectively detect LSB steganography; however, it cannot effectively detect the Stego images generated using the proposed method. This shows that the stenography method proposed in this study, which combines LSB and modulus pixel-value differencing, is effective and robust against RS detection technology.

4.3.3 CSR Analysis

Content-Selective Residuals or CSR analysis [18], which was proposed by Denemark et al. in 2014, can be used to detect the 1183 features that are generated using spatial domain steganography. To prove that our method is robust against CSR detection, we analyze the features after embedding using 10,000 512×512 8-bit grayscale images from the BOSSBase Database as test images. We evaluate the accuracy with AC (see Equation 39). The lower the value, the more security our method provides. Moreover, TP represents the number of embedded images that are correctly identified, TN represents that of cover images correctly identified, FP represents that of images falsely identified as embedded images, and FN represents that of images falsely identified as cover images.

$$\text{Accuracy}=(TN+TP)/(TN+TP+FN+FP)\times 100\% \quad \textbf{(39)}$$

Table 9's experimental results show the detection accuracy of 54.22%, 53.1%, 56.09%, 56.59% and 59.36% for the NLSB/PVD, side match/PVD, HLSB/PVD, NLSB/MPVD and our method, respectively. Because the embedding capacity of the proposed method was higher than that of other methods, the detection rate of CSR analysis was also higher than that of other methods, which was only slightly higher than the probability of random judgments (50%). This implies that the CSR steganalysis failed to detect the difference between the features of cover images and those of embedded images using our method. Thus, we have solid evidence that our method is robust against CSR detection.

Table 9. Experimental results of CSR detection

Method \ Indicator	TP	TN	FP	FN	Accuracy (%)
NLSB/PVD	4999	423	4577	1	54.22%
Side Match PVD	5000	310	4690	0	53.10%
HLSB/PVD	4997	612	4388	3	56.09%
NLSB/MPVD	5000	659	4341	0	56.59%
Our method	5000	936	4064	0	59.36%

5 Conclusion

In this study, we improved on the HLSB/PVD method developed by Jung [1] to create a steganographic method with a high embedding capacity (the highest of all known methods). However, increasing the embedding capacity will usually reduce the imperceptibility and security of the steganographic method. A new method was devised after numerous trials. In the final method, the cover image is partitioned into non-overlapping 3 × 3 sized blocks, and the embedding of data in the central (base) pixel is conducted using a 5-bit LSB substitution. Pixel-value difference calculations are then conducted among the remaining pixels in the block and the embedded base pixel, followed by data embedding with the improved HLSB/PVD method. The MPVD method, which hides data by altering the remainder between two consecutive pixels in a way that minimizes image distortions, is used to embed data into a higher bit-plane. In this way, 17 sets of secret data can be embedded in each block. This results in an enhanced embedding capacity, as well as an acceptable level of stego-image quality.

Although the primary intent of the proposed method is to maximize the embedding capacity, only 2 pixel-value difference ranges (with embedding capacities of 2 and 3 bits) were defined to ensure that the stego-image is sufficiently imperceptible for the HVS. In the future, we will attempt to combine this method with other steganographic techniques, and investigate the possibility of dynamically varying the embedding capacity based on a variety of image characteristics (e.g., luminance and contrast) to enhance the steganographic security.

References

[1] K.-H. Jung, Data Hiding Scheme Improving Embedding Capacity using Mixed PVD and LSB on Bit Plane, *Journal of Real-Time Image Processing*, Vol. 14, No. 1, pp. 127-136, January, 2018.

[2] N.-F. Johnson, S. Jajodia, Exploring Steganography: Seeing the Unseen, *Computer*, Vol. 31, No. 2, pp. 26-34, February,

1998.

[3] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for Data Hiding, *IBM Systems Journal*, Vol. 35, No. 3-4, pp. 313-336, 1996.

[4] C.-K. Chan, L.-M. Cheng, Hiding Data in Images by Simple LSB Substitution, *Pattern Recognition*, Vol. 37, No. 3, pp. 469-474, March, 2004.

[5] D.-C. Wu, W.-H. Tsai, A Steganographic Method for Images by Pixel Value Differencing, *Pattern Recognition Letters*, Vol. 24, No. 9-10, pp. 1613-1626, June, 2003.

[6] H.-C. Wu, N.-I. Wu, C.-S. Tsai, M.-S. Hwang, Image Steganographic Scheme Based on Pixel-value Differencing and LSB Replacement Methods, *IEE Proceedings - Vision, Image and Signal Processing*, Vol. 152, No. 5, pp. 611-615, October, 2005.

[7] C.-M. Wang, N.-I. Wu, C.-S. Tsai, M.-S. Hwang, A High Quality Steganographic Method with Pixel-value Differencing and Modulus Function, *Journal of Systems and Software*, Vol. 81, No. 1, pp. 150-158, January, 2008.

[8] P. Tsai, Y.-C. Hu, H.-L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, *Signal Processing*, Vol. 89, No. 6, pp. 1129-1143, June, 2009.

[9] J.-L. Liu, T.-H. Lai, Y.-H. Lee, Pixel-value Difference Steganography against Histogram Attacks, *2012 Conference on Information Technology and Applications in Outlying Islands*, Magong, Penghu, Taiwan, 2012, pp. 119-125.

[10] M. Khodaei, K. Faez, New Adaptive Steganographic Method Using Least-Significant-Bit Substitution and Pixel-value Differencing, *IET Image Processing*, Vol. 6, No. 6, pp. 677-686, August, 2012.

[11] C. Qin, C.-C. Chang, T.-J. Hsu, Reversible data hiding scheme based on exploiting modification direction with two steganographic images, *Multimedia Tools and Applications*, Vol. 74, No. 15, pp. 5861-5872, August, 2015.

[12] I-C. Chang, Y.-C. Hu, W.-L. Chen, C.-C. Lo, High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding, *Signal Processing*, Vol. 108, pp. 376-388, March, 2015.

[13] H.-H. Liu, Y.-C. Lin, C.-M. Lee, A Digital Data Hiding Scheme based on Pixel-value Differencing and Side Match Method, *Multimedia Tools and Applications*, Vol. 78, No. 9, pp. 12157-12181, May, 2019.

[14] Y.-C. Hu, Y.-H. Lin, C.-C. Lo, C.-M. Wu, Implementation of Block-Based Hierarchical Prediction for Developing an Error-Propagation-Free Reversible Data Hiding Scheme, *Symmetry*, Vol. 11, No. 9, Article No. 1146, September, 2019.

[15] G.-D. Su, C.-C. Chang, C.-C. Lin, A High Capacity Reversible Data Hiding in Encrypted AMBTC-Compressed Images, *IEEE Access*, Vol. 8, pp. 26984-27000, January, 2020.

[16] H.-H. Liu, P.-C. Su, M.-H. Hsu, An Improved Steganography Method Based on Least-Significant-Bit Substitution and Pixel-Value Differencing, *KSII Transactions on Internet and Information Systems*, Vol. 14, No. 11, pp. 4537-4556, November, 2020.

[17] J. Fridrich, M. Goljan, R. Du, Detecting LSB steganography in color, and gray-scale images, *IEEE Multimedia*, Vol. 8, No. 4, pp. 22-28, October-December, 2001.

[18] T. Denemark, J. Fridrich, V. Holub, Further study on the security of S–UNIWARD, *SPIE Electronic Imaging, Media Watermarking, Security, and Forensics, International Society for Optics and Photonics*, Vol. 9028, pp. 902805, February, 2014.

Biographies

Hsing-Han Liu received the Ph.D. degree in department of electrical and electronic engineering from National Defense University, Taoyuan, Taiwan, R.O.C., in 2013. Currently, he is an associate professor in the department of information management, National Defense University, Taipei, Taiwan. His current research interests include information hiding and steganalysis.

Yu-Fen Lo received M. S. degree in department of information management from National Defense University, Taipei, Taiwan. His current research interests include information hiding and information security.