# An Image Steganographic Scheme Based on Edge Detection and Least Significant Bit Substitution

*Hsing-Han Liu*[1], *Sheng-Chih Ho*[2*], *Tai-Hsiu Wu*[1]

[1] *Department of Information Management, National Defense University, Taiwan*
[2] *Department of Management Information System, Takming University of Science and Technology, Taiwan*
*liu.hansh@gmail.com, hsz1028@gmail.com, wuashowshow@gmail.com*

## Abstract

This work presents a steganographic scheme based on Laplacian-of-Gaussian (LoG) edge detection and least significant bit (LSB) substitution. The cover image is first divided into continuous and non-overlapping 4×4-pixel blocks. The pixel at the top left corner of each block (first pixel) is defined as the reference pixel. After the LoG edge detection, the blocks are classified as edge or non-edge blocks, and this information is embedded in the reference pixels. Each non-reference pixel is then embedded with 5 bits and 4 bits of cipher text if it belongs to an edge block or non-edge block, respectively. Compared to the method of Tseng and Leng, Bai et al., and Ghosal et al., proposed method increased the capacity by 39.6%, 7.3%, and 42.7%, respectively, in the "Lena" cover image. 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. Compared to the aforementioned previous methods, our method improved the capacity by 33.9%, 2.7%, and 36.1%, respectively, while maintaining an acceptable stego-image quality. Finally, proposed method can resist the detection of RS, pixel difference histogram analysis and second order SPAM features.

**Keywords:** Information hiding, Least significant bit substitution, Edge detection, Laplacian of Gaussian

## 1 Introduction

Owing to the rapid development of information technologies, computers, and the Internet, multimedia signals such as sounds, text, and images have found widespread use through digitalization. Furthermore, the Internet has made it possible to propagate digital information with relative ease and incredible speed. However, this also creates security issues as secret information can be leaked and stolen when it travels through the Internet. Therefore, trustworthy information transfer is a basic requirement in current Internet technologies, and the value of some information is often commensurate to its level of information security. In military applications, information that is transmitted in an unguarded form through the Internet is likely to be unlawfully intercepted, spied on, or altered by an adversary. Secret information can be protected using cryptography, to provide the first layer of protection, and steganography techniques, which are gradually maturing. Although cryptography and steganography are both methods used for the protection and hiding of information, there are significant differences between these methods. Cryptography is a process that encrypts secret information (such as text or images) into a meaningless set of random numbers, which ensures that no one other than the intended receiver would be able to read the information. Steganography is a technique that embeds secret text into a cover image, in order to produce stego-image that contain secret information [1]. This prevents information transfers from being detected by adversaries, in order to protect the secret information.

As the human vision system (HVS) cannot detect minute changes in an image, it is possible to exploit this weakness to create difficult-to-detect stego-images and thus improve the security of information transfers. For example, when information is embedded in a cover image (to create a stego-image), the changes that occurred in the image are very difficult to detect using the naked eye. Therefore, even if the image is intercepted, it is unlikely that the intercepting party will realize that it contains secret information. After the transfer is completed, the intended receiver can simply use the pre-specified method to extract the secret information.

Steganography can be broadly classified into techniques that operate in the spatial or frequency domains [2]. Spatial domain steganography (the more common of the two approaches) is usually performed by embedding encrypted secret information in the least significant bit (LSB) [3] or by altering pixel values in the cover image. Frequency domain steganography is performed by converting spatial-domain pixel values into coefficients in the frequency domain, hiding secret information in selected coefficients, and then transforming them back into coefficients in the spatial domain.

An ideal steganography must fulfil three requirements [4]: imperceptibility, undetectable property, and capacity. However, these requirements are mutually incompatible with each other. A high capacity would inevitably reduce the image quality, which reduces the imperceptibility and undetectable property; conversely, improving stego-image quality would reduce capacity. Therefore, the majority of studies on steganography are focused either on image

*Corresponding Author: Sheng-Chih Ho; E-mail: hsz1028@gmail.com

DOI: 10.53106/160792642023092405002

quality or capacity, but not both. The aim of this study is to propose an algorithm that can simultaneously address both these requirements. The method proposed in this study has a high capacity, which facilitates its use for sending large amounts of cipher text while retaining an adequate stego-image quality. If the amount of cipher text is small, it would improve the stego-image quality instead of wasting space, thus ensuring imperceptibility and undetectable property.

Based on the aforementioned background and motivation, the aim of this study is to use the Laplacian of Gaussian (LoG) [5] edge-detection algorithm in tandem with LSB substitution to construct a well-rounded steganography that can enhance the capacity while maintaining an adequate image quality. To validate the cipher text embedding and extraction procedures of proposed method, it is implemented and tested in Python. Image-quality analysis is also performed on the stego-images to confirm the performance of the method in terms of capacity and image quality.

## 2 Literature Review

### 2.1 Least Significant Bit Substitution

The idea of LSB substitution was proposed by Bender et al. [6] in 1996. As the luminance of each pixel in a grayscale image is an 8-bit number, a pixel may take values between 0 and 255, where 0 is black and 255 is white. In LSB substitution, the luminance value of each pixel is first converted into the binary form. Secret information is then embedded in the LSB, which has little effect on the pixel. As the changes in color caused by small variations in luminance within a grayscale image are difficult to perceive by the HVS, information can be hidden in an image, as shown in Figure 1. Each pixel may be viewed as a luminance value, which is always given by 8 bits. The greatest possible change is from 0 (black) to 255 (white), as shown in Figure 2, where 0 goes to $2^8 - 1$. Each bit has a different effect on the luminance of a pixel; for instance, a change in the highest bit changes the luminance of a pixel by 128, but a change in the lowest bit only results in a change of 1. The lowest bit is called the LSB, while the highest bit is called the most significant bit (MSB). As it is very difficult for the HVS to detect changes in the lowest bit, it is possible to hide information in an image by altering the lowest bit—this is the basic principle of LSB substitution.

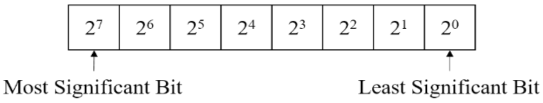


**Figure 1.** Binary representation of a pixel value

**Figure 2.** Illustration of pixel luminance changes in a grayscale image

### 2.2 Optimal Pixel Adjustment Process (OPAP)

The optimal pixel adjustment process (OPAP), which was proposed by Chan and Cheng [3], is a steganography that uses the high embedding capacity of LSB substitution. First, the pixels are classified into three intervals according to their embedding errors (i.e., the error caused by the embedding of secret information), and the pixels that cause the smallest embedding errors are used as stego pixels. This effectively reduces the loss of fidelity compared to that in the case of simple LSB substitution. Therefore, OPAP retains the high embedding capacity of LSB substitution while remaining largely imperceptible to the HVS. Let us suppose that $p_i$, $p_i$’, and $p_i$” are the pixel values of the *i*-th pixel in the cover image, the stego-image obtained from simple LSB substitution, and the stego-image obtained from OPAP, respectively. Let the embedding error be $d = p_i' - p_i$, and $k$ be the number of message bits to be hidden in each pixel. In OPAP, the pixels are divided into three intervals according to their value of $d$, and the $p_i$’ values are modified in a specific manner (depending on the pixel’s interval) to obtain $p_i$”, as shown below:

Interval 1: $2^{k-1} < d < 2^k$ : If $p_i^{'} \geq 2^k$, then $p_i^{''} = p_i^{'} \text{-} 2^k$

; otherwise = $p_i^{''} = p_i^{'}$

Interval 2: $-2^{k-1} \leq d \leq 2^{k-1}$: $p_i^{''} = p_i^{'}$

Interval 3: $-2^k < d < -2^{k-1}$: $p_i^{'} < 256 \text{-} 2^k$, then $p_i^{''} = p_i^{'} + 2^k$

; otherwise $p_i^{''} = p_i^{'}$

### 2.3 Edge Detection (ED)

An edge is a discontinuity between adjacent pixels, i.e., it is a visible inter-pixel difference. Therefore, edge detection (ED) can be used to understand changes in the grayscale values of an image, and it is usually performed by detecting abrupt changes in intensity (luminance or depth). Therefore, the aim of ED is to only extract the discontinuities of an image, which results in the extraction of a set of contours that describes the structure of the image, while discarding all unnecessary information. Edges may be quantitatively described as the set of pixels that exhibit significant local changes in grayscale value, which exceed some given threshold; they can also be described as long and thin image features that are perpendicular to sudden changes in an image. The goal of ED is to identify boundaries at which abrupt changes in grayscale value occur. Some of the more common methods for ED are the Canny, Sobel, Roberts, Prewitt, Laplacian, and LoG operators.

In 1980, Marr and Hildreth [5] proposed the LoG operator as an extension of the Laplacian operator. In the LoG method, the image is first subjected to Gaussian filtering before the Laplacian operator is used to detect edges. As the Laplacian operator is a second derivative-based method, it is susceptible to noise. Therefore, in the LoG method, Gaussian filtering is first applied to reduce the noise, before using the zero-crossings of the second derivative to obtain edges. Thus, the image is smoothed before the ED is performed, as per Equation (1):

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1-(\frac{x^2+y^2}{2\sigma^2})\cdot e^{-\frac{x^2+y^2}{2\sigma^2}}\right]. \tag{1}$$

### 2.4 Image Steganographic Scheme Based on Edge Detection

In 2010, Chen et al. [7] proposed a hybrid ED method that combines the Canny and fuzzy ED methods. The cover image is first divided into blocks, with each block consisting of *n* pixels. In each block, the first pixel is responsible for storing the edge information of the edge and non-edge pixels. The status of each pixel is denoted as “1” or “0” if it is an edge or non-edge pixel, respectively. The embedding phase is controlled by three parameters: the number of pixels in each block (*n*), number of message bits to be embedded in non-edge pixels (*x*), and number of message bits to be embedded in edge pixels (*y*). However, not every pixel is embedded with a cipher text. As the first pixel of each block is always used as an index, the total number of pixels containing cipher text is reduced by 1/*n*.

In 2014, Tseng and Leng [8] extended the method of Chen et al. [7] with a steganographic method based on 4×4 pixel blocks. First, the minimum mean square errors (MSEs) associated with different LSB embedding lengths are calculated, and the length that provides the minimum MSE is selected. In contrast to the scheme of Chen et al., the method of Tseng and Leng only requires one parameter, i.e., the number of message bits embedded in non-edge pixels, *x*. The embedding length for the edge pixels is the length that results in the optimal MSE. However, the LSB embedding length of the edge pixels must be greater than *x*, i.e., [*x*, *x* + 1], [*x*, *x* + 2], [*x*, *x* + 3], or [*x*, *x* + 4], where the first and second elements represent the LSB embedding lengths of the non-edge and edge pixels, respectively. The case that provides the lowest MSE is selected. This method is an improvement over that of Chen et al. in terms of capacity and stego-image quality.

In 2014, Islam et al. [9] proposed a threshold-based ED method to enhance the security of stego-images. First, threshold selection is performed to find the Canny high threshold such that a sufficient number of edges is selected for the given payload. If a higher capacity is required, a weak threshold is selected such that more edges are selected. Thus, the strength of the threshold depends on the size of the payload.

One of the flaws of the methods proposed by Chen et al. [7] and Tseng et al. [8] is that some pixels are used to store edge information instead of message bits, which effectively reduces the capacity and image quality.

To address this problem, Bai et al. [10] proposed a data hiding method, wherein only three MSBs are retained, and the resulting “edge image” is then used for ED with the Canny, Sobel, and fuzzy edge detectors. LSB substitution is then performed on the last five LSBs to embed the secret message.

In the method of Ghosal et al. [11], only the MSB is used for ED with the LoG operator. Each pixel pair (Pi, Pi+1) is then classified into three categories: pair of non-edge pixels, pair of edge pixels, and pair of mixed pixels, and a specific embedding length is defined for each category. Embedding is then performed using Equations (2)-(6):

$$f = (N_s \times P_i + P_{i+1})\%N_s^2 . \tag{2}$$

$$D = (d - f) . \tag{3}$$

$$Q = (D / N_S) . \tag{4}$$

$$R = (D \% N_S) . \tag{5}$$

$$(P'_i, P'_{i+1}) = (P_i + Q, P_{i+1} + R) . \tag{6}$$

In these equations, $N_s$ is the number of embedded bits, $Q$ is the quotient of $D$ over $N_s$, and $R$ is the remainder of $D$ over $N_s$.

**Table 1.** Comparison of ED-based steganographic methods

| Method | Chen et al. [7] | Tseng and Leng [8] | Bai et al. [10] | Ghosal et al. [11] | Rezaei et al. [12] | Proposed |
|---|---|---|---|---|---|---|
| Edge detector | Fuzzy edge + Canny edge | Modified Fuzzy edge + Canny edge detection | Canny, Sobel, Fuzzy Logic | LoG | DBSCAN + enhanced Sobel | LoG |
| Results | • Maximum embedding bits of 2.76 bpp and PSNR of 32 dB<br>• If image quality is the priority, the capacity is 0.5 bpp, which corresponds to PSNR of 51.1 dB | • Maximum embedding bits of 3.16 bpp and PSNR of 33.58 dB<br>• If image quality is the priority, the capacity is 0.91 bpp, which corresponds to PSNR of 42.18 dB | • Canny: Maximum embedding bits of 4.11 bpp and PSNR of 30.10 dB<br>• Sobel: Maximum embedding bits of 4.05 bpp and PSNR of 30.69 dB<br>• Fuzzy: Maximum embedding bits of 3.79 bpp and PSNR of 34.32 dB | • Maximum embedding bits of 3.09 dB and PSNR of 38.45 dB.<br>• If image quality is the priority, the capacity is then 1.09 bpp, which corresponds to a PSNR of 51.48 dB | • The PSNR of 50k bit (0.2 bpp) is 61.19 dB. | • Maximum embedding bits of 4.41 dB and the PSNR was maintained above 30 dB. |

In 2022, Rezaei et al. [12] proposed the method that convolves the cover image using an enhanced Sobel operator prior to edge detection. For detecting edge areas, an adaptive clustering method based on the DBSCAN algorithm is proposed. The experimental results demonstrate that the method improves PSNR by 3.98db compared to other schemes. The aforementioned ED-based data hiding methods are compared using the "Lena" cover image, and the results are shown in Table 1.

In this study, the "Lena" cover image is used as the cover image and edge detectors such as Canny, Sobel, and LoG are used for the experiment testing the proposed method. As the experimental results (see Table 2) show that the LoG edge detector provides significantly higher capacities, the proposed method uses the LoG edge detector to detect edges.

From this comparison, it can be observed that it is possible to improve the capacity and stego-image quality in ED-based methods by adjusting the number of embedded message bits in smooth and edge regions. However, in the above methods, some pixels must be used to store edge information or are simply unusable for message embedding. Therefore, the aim of this study is to also utilize the pixels that store edge information for message embedding and to avoid having pixels that cannot be embedded with information in order to improve the capacity.

# 3 Method and Architecture

To avoid degradations in stego-image quality when embedding large quantities of cipher text, in our method, the cover image is divided into edge and non-edge blocks. Edge blocks are embedded with more secret information than non-edge blocks, which reduces the degradation of the stego-image quality.

## 3.1 Architecture

The cover image is divided into 4×4 non-overlapping blocks, and the LoG edge detector is used to identify edge pixels and thus identify the edge and non-edge blocks. The number of message bits for these blocks is then determined, which is followed by message embedding using the OPAP method to take advantage of the inherently high capacity of LSB substitution. It is expected that this approach will greatly improve the capacity whilst maintaining a good image quality. The architecture of proposed method is presented in Figure 3.

**Table 2.** Comparison of different edge detectors

| Metrics | Edge detector | | |
|---|---|---|---|
| | Canny | Sobel | LoG |
| PSNR | 35.05 | 34.10 | 30.50 |
| bpp | 3.81 | 3.89 | **4.41** |

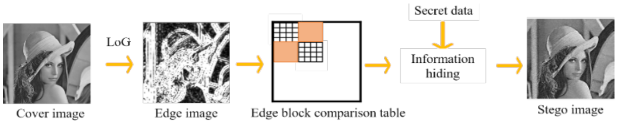


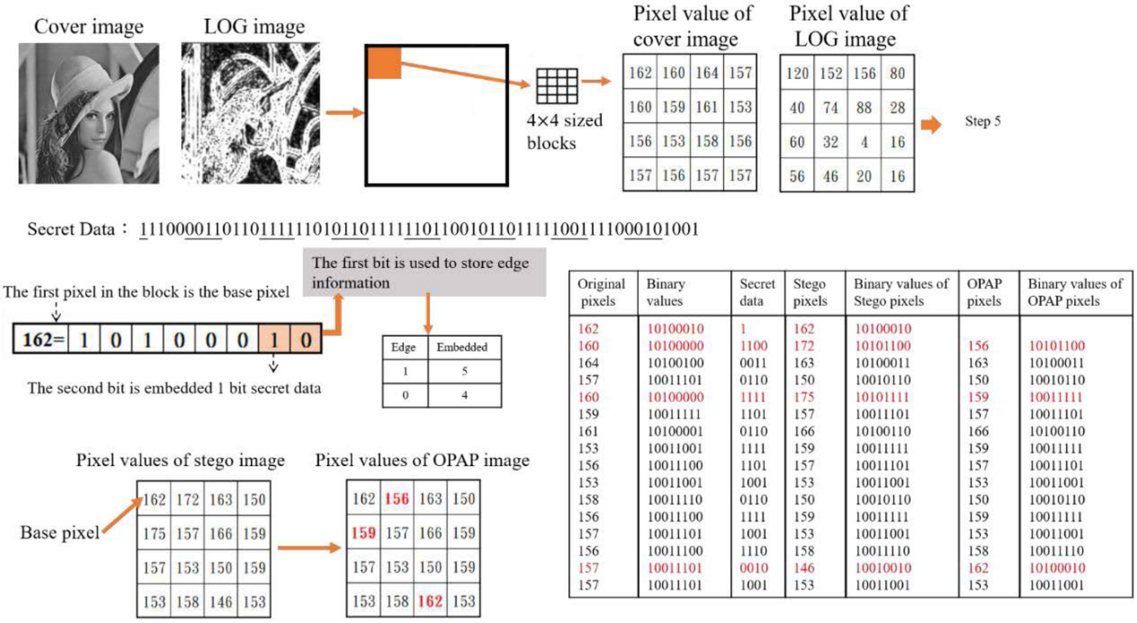
**Figure 3.** The system architecture of the study

## 3.2 Embedding Procedure

The embedding procedure of proposed method is described below and illustrated in Figure 4.

Step 1: An edge image is generated from the cover image using the LoG edge detector.

Step 2: The cover image is divided into continuous and adjacent 4×4-pixel blocks, and the cover image is scanned in a zig-zag fashion such that the blocks are non-overlapping (as shown in Figure 5).


Cover image
LOG image
4×4 sized blocks
Step 5
The first bit is used to store edge information
The first pixel in the block is the base pixel
The second bit is embedded 1 bit secret data
Base pixel

Pixel value of cover image

| | | | |
|---|---|---|---|
| 162 | 160 | 164 | 157 |
| 160 | 159 | 161 | 153 |
| 156 | 153 | 158 | 156 |
| 157 | 156 | 157 | 157 |

Pixel value of LOG image

| | | | |
|---|---|---|---|
| 120 | 152 | 156 | 80 |
| 40 | 74 | 88 | 28 |
| 60 | 32 | 4 | 16 |
| 56 | 46 | 20 | 16 |

Secret Data： 1110000110110111111010110111111011001011011111001111000101001

| 162= | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|

| Edge | Embedded |
|---|---|
| 1 | 5 |
| 0 | 4 |

Pixel values of stego image

| | | | |
|---|---|---|---|
| 162 | 172 | 163 | 150 |
| 175 | 157 | 166 | 159 |
| 157 | 153 | 150 | 159 |
| 153 | 158 | 146 | 153 |

Pixel values of OPAP image

| | | | |
|---|---|---|---|
| 162 | 156 | 163 | 150 |
| 159 | 157 | 166 | 159 |
| 157 | 153 | 150 | 159 |
| 153 | 158 | 162 | 153 |

| Original pixels | Binary values | Secret data | Stego pixels | Binary values of Stego pixels | OPAP pixels | Binary values of OPAP pixels |
|---|---|---|---|---|---|---|
| 162 | 10100010 | 1 | 162 | 10100010 | | |
| 160 | 10100000 | 1100 | 172 | 10101100 | 156 | 10101100 |
| 164 | 10100100 | 0011 | 163 | 10100011 | 163 | 10100011 |
| 157 | 10011101 | 0110 | 150 | 10010110 | 150 | 10010110 |
| 160 | 10100000 | 1111 | 175 | 10101111 | 159 | 10011111 |
| 159 | 10011111 | 1101 | 157 | 10011101 | 157 | 10011101 |
| 161 | 10100001 | 0110 | 166 | 10100110 | 166 | 10100110 |
| 153 | 10011001 | 1111 | 159 | 10011111 | 159 | 10011111 |
| 156 | 10011100 | 1101 | 157 | 10011101 | 157 | 10011101 |
| 153 | 10011001 | 1001 | 153 | 10011001 | 153 | 10011001 |
| 158 | 10011110 | 0110 | 150 | 10010110 | 150 | 10010110 |
| 156 | 10011100 | 1111 | 159 | 10011111 | 159 | 10011111 |
| 157 | 10011101 | 1001 | 153 | 10011001 | 153 | 10011001 |
| 156 | 10011100 | 1110 | 158 | 10011110 | 158 | 10011110 |
| 157 | 10011101 | 0010 | 146 | 10010010 | 162 | 10100010 |
| 157 | 10011101 | 1001 | 153 | 10011001 | 153 | 10011001 |

**Figure 4.** Embedding process of proposed method

Step 3: The first pixel of each block is set as the reference pixel $P_{i0}$, as shown in Figure 6.

Step 4: The edge values of the relative positioned pixels are compared, as shown in Figure 7.

Step 5: Generate the edge block table. After LoG ED, if a pixel block has more than 14 pixels (excluding the reference pixel) with pixel values equal to or greater than 32, it is an edge block, and the first bit of the reference pixel is marked as "1." Otherwise, the pixel block is a non-edge block and the first bit of the reference pixel is marked as "0." The number 14 is used as a threshold to determine whether a 4×4 pixel block is an edge block. In this study, different difference values were tested during the experiments. A difference of 32 is the value of the best experimental result that can be obtained.

Step 6: The pixel value of $P_{i0}$ is converted into binary, and the number of message bits to be embedded using LSB substitution is set to 1 bit. The edge information of the block is embedded in the first bit of $P_{i0}$, and the second bit is replaced with 1 bit of cipher text. This produces the embedded reference pixel $P'_{i0}$, as shown in Figure 8.

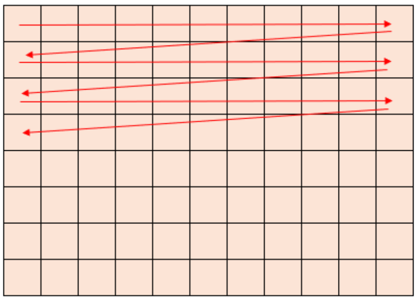
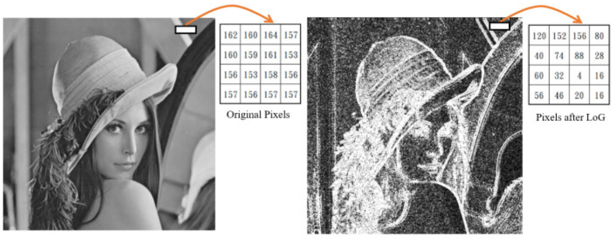


**Figure 5.** Zig-zag scanning of cover image

| | | | |
|---|---|---|---|
| $P_{i0}$ | $P_{i1}$ | $P_{i2}$ | $P_{i3}$ |
| $P_{i4}$ | $P_{i5}$ | $P_{i6}$ | $P_{i7}$ |
| $P_{i8}$ | $P_{i9}$ | $P_{i10}$ | $P_{i11}$ |
| $P_{i12}$ | $P_{i13}$ | $P_{i14}$ | $P_{i15}$ |

**Figure 6.** Selection of the reference pixel



| | | | |
|---|---|---|---|
| 162 | 160 | 164 | 157 |
| 160 | 159 | 161 | 153 |
| 156 | 153 | 158 | 156 |
| 157 | 156 | 157 | 157 |

Original Pixels

| | | | |
|---|---|---|---|
| 120 | 152 | 156 | 80 |
| 40 | 74 | 88 | 28 |
| 60 | 32 | 4 | 16 |
| 56 | 46 | 20 | 16 |

Pixels after LoG

**Figure 7.** Using the LoG edge detector to detect edge pixels

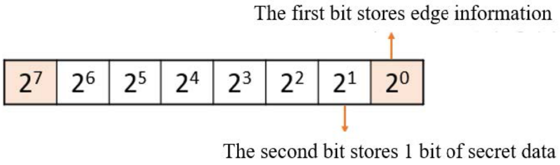

The first bit stores edge information
$2^7$
$2^6$
$2^5$
$2^4$
$2^3$
$2^2$
$2^1$
$2^0$
The second bit stores 1 bit of secret data

**Figure 8.** Bits of the reference pixel after its conversion into binary

Step 7: The first binary bit of the reference pixel is used to distinguish whether a block is an edge or a non-edge block ("1" for edge blocks; "0" for non-edge blocks).

Step 8: Based on the edge information provided by the reference pixel, LSB substitution is performed on the 15 other pixels in each pixel block, with an embedding length of $k$ bits for edge blocks and $(k - 1)$ bits for non-edge blocks.

Step 9: OPAP (Equation (7)) is used to adjust the embedded pixels.

$$p_i^{'} = \begin{cases} p_i^{'} + 2^k, \ if\ d_{ic} > 2^{k-1}\ and\ 0 \le p_i^{'} + 2^k \le 255 \\ p_i^{'} - 2^k, \ if\ d_{ic} < -2^{k-1}\ and\ 0 \le p_i^{'} - 2^k \le 255. \\ \quad p_i^{'},\ otherwise \end{cases} \quad \textbf{(7)}$$

Step 10: We return to Step 6 to embed cipher text in all other blocks, until all of the blocks have been embedded with cipher text.

The proposed steganographic process is presented in Figure 9.

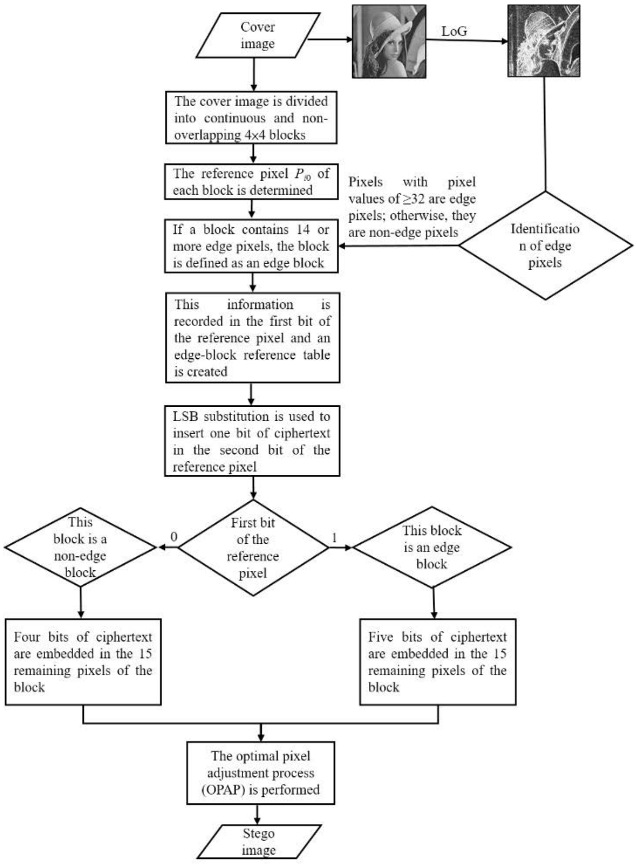

Cover image
LoG
The cover image is divided into continuous and non-overlapping 4x4 blocks
The reference pixel $P_{i0}$ of each block is determined
Pixels with pixel values of ≥32 are edge pixels; otherwise, they are non-edge pixels
Identification of edge pixels
If a block contains 14 or more edge pixels, the block is defined as an edge block
This information is recorded in the first bit of the reference pixel and an edge-block reference table is created
LSB substitution is used to insert one bit of ciphertext in the second bit of the reference pixel
This block is a non-edge block
0
First bit of the reference pixel
1
This block is an edge block
Four bits of ciphertext are embedded in the 15 remaining pixels of the block
Five bits of ciphertext are embedded in the 15 remaining pixels of the block
The optimal pixel adjustment process (OPAP) is performed
Stego image

**Figure 9.** Flowchart of the steganography method proposed in this study

### 3.3 Data Extraction Procedure

The data extraction procedure for the proposed method is described below and illustrated in Figure 10.

Step 1: The stego-image is divided into continuous, non-overlapping 4×4-pixel blocks.

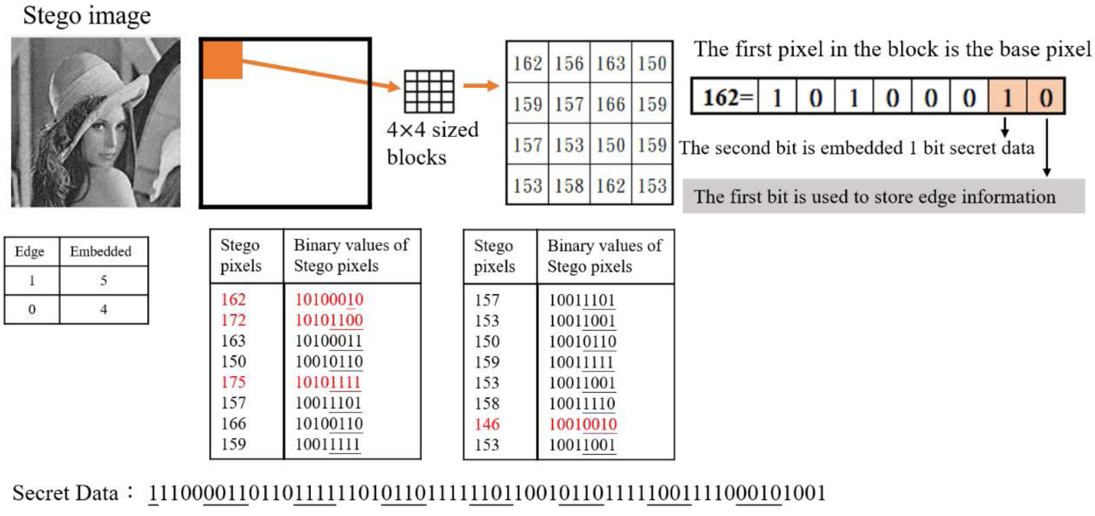
Step 2: The first pixel of each block, $P'_{i0}$, is set as the reference pixel.

Step 3: The 1-bit cipher text is extracted from the second bit of $P'_{i0}$.

Step 4: The first bit of $P'_{i0}$ is checked. If it is 1, the block is an edge block; if it is 0, the block is a non-edge block.

Step 5: If the block is an edge block, 5 bits of cipher text are extracted from the 15 other pixels; if the block is a non-edge block, four bits of cipher text are extracted from the 15 other pixels.

The data extraction process is presented in Figure 11.


Stego image
4×4 sized blocks
The first pixel in the block is the base pixel
162= 1 0 1 0 0 0 1 0
The second bit is embedded 1 bit secret data
The first bit is used to store edge information

|  |  |  |  |
|---|---|---|---|
| 162 | 156 | 163 | 150 |
| 159 | 157 | 166 | 159 |
| 157 | 153 | 150 | 159 |
| 153 | 158 | 162 | 153 |

| Edge | Embedded |
|---|---|
| 1 | 5 |
| 0 | 4 |

| Stego pixels | Binary values of Stego pixels |
|---|---|
| 162 | 10100010 |
| 172 | 10101100 |
| 163 | 10100011 |
| 150 | 10010110 |
| 175 | 10101111 |
| 157 | 10011101 |
| 166 | 10100110 |
| 159 | 10011111 |

| Stego pixels | Binary values of Stego pixels |
|---|---|
| 157 | 10011101 |
| 153 | 10011001 |
| 150 | 10010110 |
| 159 | 10011111 |
| 153 | 10011001 |
| 158 | 10011110 |
| 146 | 10010010 |
| 153 | 10011001 |

Secret Data： 1110000110110111111010110111111011001011011111001111000101001

**Figure 10.** Extraction process of proposed method

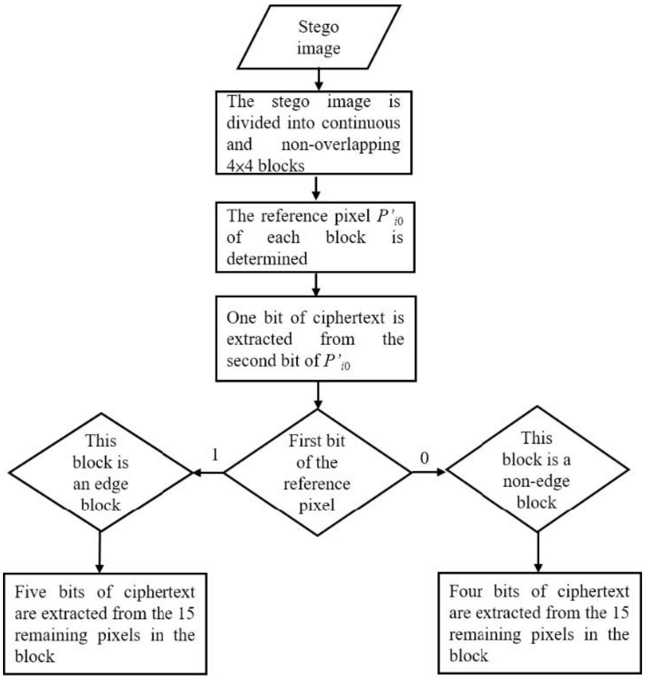

Stego image
The stego image is divided into continuous and non-overlapping 4x4 blocks
The reference pixel $P'_{i0}$ of each block is determined
One bit of ciphertext is extracted from the second bit of $P'_{i0}$
First bit of the reference pixel
1
0
This block is an edge block
This block is a non-edge block
Five bits of ciphertext are extracted from the 15 remaining pixels in the block
Four bits of ciphertext are extracted from the 15 remaining pixels in the block

**Figure 11.** Flowchart of cipher text retrieval in the steganography method proposed in this study

## 4 Experimental Results

The 1×7_edge5, 3×3_edge7, 4×4_edge13, and 4×4_edge14 block sampling approaches (see Table 3) were compared in terms of capacity and stego-image quality with the LoG, LSB, and OPAP methods using three standard grayscale images that illustrated in Figure 12. To test whether the proposed method is generalizable to all types of images, a steganography experiment was performed on 10,000 512×512 grayscale images from the BOSSBase database. The aforementioned block sampling approaches were compared in terms of stego-image quality to determine the approach with the best performance. After the optimal block sampling method was identified, the proposed method was compared to other data hiding method in terms of capacity and stego-image quality. Finally, the security analysis was used to perform stego-image detection to test whether the proposed method is resistant to steganalysis.

**Table 3.** Experimental block definitions of this study

| No. | Method | Experimental block definitions |
|---|---|---|
| 1 | 1×7_edge5 | The cover image is first segmented into 1×7 non-overlapping blocks. LoG ED is then performed. If five or more pixels in a block have pixel values ≥32, the block is an edge block. |
| 2 | 3×3_edge7 | The cover image is first segmented into 3×3 non-overlapping blocks. LoG ED is then performed. If seven or more pixels in a block have pixel values ≥ 32, the block is an edge block. |
| 3 | 4×4_edge13 | The cover image is first segmented into 4×4 non-overlapping blocks. LoG ED is then performed. If thirteen or more pixels in a block have pixel values ≥ 32, the block is an edge block. |
| 4 | 4×4_edge14 | The cover image is first segmented into 4×4 non-overlapping blocks. LoG ED is then performed. If fourteen or more pixels in a block have pixel values ≥ 32, the block is an edge block. |

Lena Baboon Jet

**Figure 12.** Experimental images of this study

### 4.1 Experimental Environment

The software, hardware and experimental subjects used in this experiment are as follows:

- Hardware environment: Intel(R) Core(TM) i7-10510U CPU、RAM 8G.
- Software environment: The proposed method was coded in Python to test the correctness of the proposed procedure and to test its generalizability using different images.
- The steganography experiments were performed on standard 128×128 grayscale images of “Lena,” “Baboon,” and “Jet” and 10,000 512×512 grayscale images from the BOSSBase image database [13].
- The methods were assessed in terms of capacity, PSNR, and the structural similarity index (SSIM).

### 4.2 Experimental Results

Experiments were performed on the “Lena,” “Baboon,” and “Jet” standard grayscale images and 10,000 512×512 grayscale images from the BOSSBase image database using the 1×7_edge5, 3×3_edge7, 4×4_edge13, and 4×4_edge14 block sampling approaches. The embedding lengths are listed in Table 4. The results of each experiment were evaluated in terms of capacity (bits) and stego-image quality (PSNR, in units of dB).

**Table 4.** Embedding lengths

| Embedding length in each type of pixel | Embedding length in each type of block |
|---|---|
| Edge = 5;<br>non-edge = 4 | In edge blocks, 5 bits of cipher text were embedded in all pixels except the reference pixels, and 4 bits of cipher text were embedded in all non-reference pixels in the non-edge blocks. |

#### 4.2.1 Results with “Lena” Cover Image

With the “Lena” cover image and 5 bits (4 bits) of cipher text in the edge (non-edge) blocks, the 4×4_edge14 block sampling approach resulted in the highest PSNR of 30.50 dB. The 4×4_edge13 block sampling resulted in the highest capacity of 4.49 bpp. The 4×4_edge14 block sampling approach resulted in the best SSIM of 0.889 (as shown in Table 5 and Figure 13).

**Table 5.** Comparison between different block sampling approaches with the “Lena” cover image, in terms of overall steganographic performance

| Cover | 4×4_edge14 | 4×4_edge13 | 4×4_edge14 |
|---|---|---|---|
| | | | |
| | edge=5<br>non-edge=4 | edge=5<br>non-edge=4 | edge=5<br>non-edge=4 |
| | PSNR:<br>30.50 dB | Capacity:<br>4.49bpp | SSIM:0.889 |

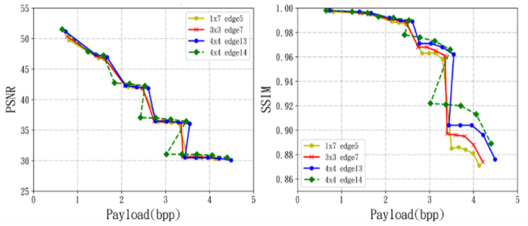
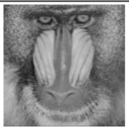
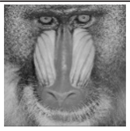
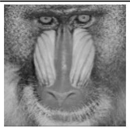
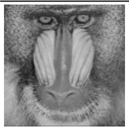


**Figure 13.** Comparison between block sampling approaches in terms of SSIM and steganographic performance with the “Lena” cover image

#### 4.2.2 Results with “Baboon” Cover Image

With the “Baboon” cover image and 5 bits (4 bits) of cipher text in the edge (non-edge) blocks, the 4×4_edge14 block sampling approach resulted in the highest PSNR of 29.97 dB. The 4×4_edge13 block sampling resulted in the highest capacity of 4.68 bpp. The 4×4_edge14 block sampling approach resulted in the best SSIM of 0.919 (as shown in Table 6 and Figure 14).

**Table 6.** Comparison between different block sampling approaches with the "Baboon" cover image, in terms of overall steganographic performance

| Cover Image | 4×4_edge14 | 4×4_edge13 | 4×4_edge14 |
|---|---|---|---|
|  |  |  |  |
|  | edge=5<br>non-edge=4 | edge=5<br>non-edge=4 | edge=5<br>non-edge=4 |
|  | PSNR:<br>29.97 dB | Capacity:<br>4.68bpp | SSIM: 0.919 |

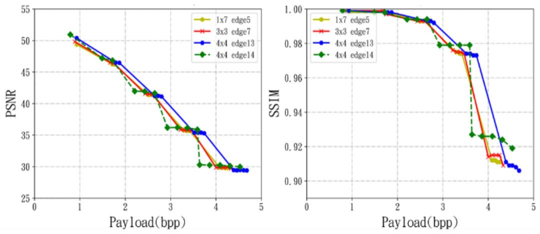


**Figure 14.** Comparison between block sampling approaches in terms of SSIM and steganographic performance with the "Baboon" cover image

### 4.2.3 Results with "Jet" Cover Image

With the "Jet" cover image and 5 bits (4 bits) of cipher text in the edge (non-edge) blocks, the 4×4_edge14 block sampling approach resulted in the highest PSNR of 30.71 dB. The 4×4_edge13 block sampling resulted in the highest capacity of 4.46 bpp. The 4×4_edge14 block sampling approach resulted in the best SSIM of 0.868 (as shown in Table 7 and Figure 15).

**Table 7.** Comparison between different block sampling approaches with the "Jet" cover image, in terms of overall steganographic performance

| Cover Image | 4×4_edge14 | 4×4_edge13 | 4×4_edge14 |
|---|---|---|---|
| | | | |
|  | edge=5<br>non-edge=4 | edge=5<br>non-edge=4 | edge=5<br>non-edge=4 |
|  | PSNR:<br>30.71 dB | Capacity:<br>4.46bpp | SSIM: 0.868 |

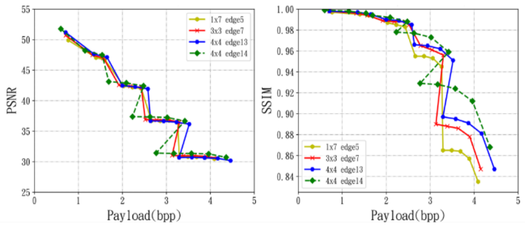


**Figure 15.** Comparison between block sampling approaches in terms of SSIM and steganographic performance with the "Jet" cover image

### 4.2.4 Generalizability Test with BOSSBase Image Database

The BOSSBase database [13] consists of 10,000 512×512 grayscale images, which comprise complex textures and smooth regions as well as photographs of persons, objects, scenery, and transportation tools. Therefore, the BOSSBase database is representative of the vast majority of digital images. The embedding parameters for these images were edge = 5 and non-edge = 4, i.e., 5 bits of cipher text in non-reference pixels in edge blocks, and 4 bits of cipher text in non-reference pixels in non-edge blocks.

Under these conditions, all four methods resulted in PSNR values greater than 31 dB; the 4×4_edge14 block sampling approach provided the best result (31.78 dB) while the 1×7_edge5 provided the worst result (31.15 dB). The best and worst performers in terms of SSIM were also the 4×4_edge14 (0.801) and 1×7_edge5 (0.763), respectively. All four methods had capacities greater than 4 bpp, with the best being 4×4_edge13 (4.24 bpp), which was followed by 4×4_edge14 (4.15 bpp). The worst performer was 1×7_edge5 (4.01 bpp). The results are listed in Table 8.

Based on the capacities (bpp), PSNR, and SSIM values obtained using 10,000 images from the BOSSBase database, it was found that the 4×4_edge14 provides the best PSNR and SSIM values. Therefore, in subsequent comparisons with steganographic techniques proposed by other authors, the 4×4_edge14 block sampling approach was used in our method.

**Table 8.** Comparison between different block sampling approaches in terms of steganographic performance and SSIM

| Method | PSNR | SSIM | bpp | Capacity |
|---|---|---|---|---|
| 1×7_edge5 | 31.15 | 0.763 | 4.01 | 1,051,885.28 |
| 3×3_edge7 | 31.32 | 0.772 | 4.06 | 1,064,218.34 |
| 4×4_edge13 | 31.21 | 0.779 | **4.24** | **1,112,514.34** |
| 4×4_edge14 | **31.78** | **0.801** | 4.15 | 1,088,073.67 |

## 4.3 Experimental Comparison with Other Method

### 4.3.1 Comparison with the Method of Tseng and Leng

In the method of Tseng and Leng [8], ED is performed using improved fuzzy and Canny edge detectors, followed by LSB substitution. MSE minimization is then used to select the embedding length. The method of Tseng and Leng was compared to our proposed method in terms of PSNR (dB), with the same payload (bpp) and using the "Lena," "Baboon," and "Jet" images. The results of this comparison are listed in Table 9. Figure 16 compares the steganographic performance of these methods.

It is shown above that our method outperformed that of Tseng and Leng in terms of PSNR (dB) with the same payload and the "Lena," "Baboon," and "Jet" cover images. When the PSNR was maintained above 30 dB, the maximum payloads of the method of Tseng and Leng were 3.16 bpp, 3.32 bpp, and 3.15 bpp, respectively, for the aforementioned cover images. The corresponding maximum payloads of our method were 4.41 bpp, 4.31 bpp, and 4.36 bpp, respectively. Hence, our method provides significantly higher capacities (as listed in Table 10).

**Table 9.** Comparison between our method and that of Tseng and Leng

| Experimental images | Payload (bpp) | Tseng and Leng | Proposed method |
|---|---|---|---|
| | | PSNR (dB) | |
| | 0.91 | 42.18 | 50.53 |
| | 1.66 | 41.03 | 46.76 |
| | 2.41 | 38.18 | 42.31 |
| | 3.16 | 33.58 | 36.75 |
| | 1.06 | 41.47 | 48.68 |
| | 1.80 | 40.22 | 46.50 |
| | 2.56 | 37.04 | 41.82 |
| | 3.32 | 32.47 | 36.20 |
| | 0.90 | 42.10 | 50.56 |
| | 1.65 | 41.02 | 46.81 |
| | 2.40 | 38.16 | 42.59 |
| | 3.15 | 33.60 | 37.08 |

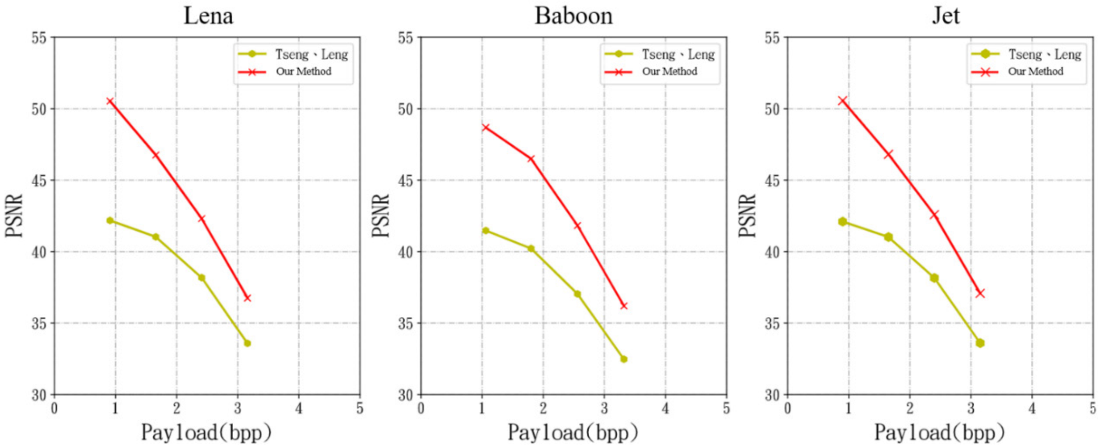


**Figure 16.** Steganographic performance of our method and that of Tseng and Leng

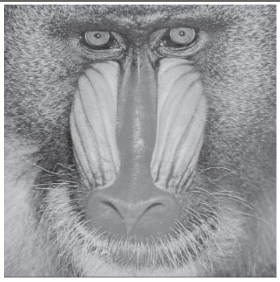
**Table 10.** Maximum capacities of our method and the method of Tseng and Leng

| | Lena | Baboon | Jet |
|---|---|---|---|
| Tseng and Leng | 3.16 bpp | 3.32 bpp | 3.15 bpp |
| Proposed method | 4.41 bpp | 4.31 bpp | 4.36 bpp |
| Percentage increase (%) | **+39.6%** | **+29.8%** | **+38.4%** |

### 4.3.2 Comparison with the Method of Bai et al.

In the method of Bai et al. [10], the pixel values are first converted into binary, and only the three MSBs are retained to create an edge image. The ED is then performed using the Canny, fuzzy, and Sobel edge detectors, which divides the pixels into edge and non-edge pixels. Finally, LSB substitution is performed on the pixels. As Bai et al. found that the Canny edge detector provides the best capacity, the results that were obtained using the Canny edge detector were used for the comparison. Our method was compared to that of Bai et al. in terms of PSNR, with the same payload, as shown in Table 11 and Figure 17.

**Table 11.** Comparison between our method and that of Bai et al.

| Experimental images | Payload (bpp) | Bai et al. | Proposed method |
|---|---|---|---|
| | | PSNR (dB) | |
| | 1.22 | 47.942 | 47.969 |
| | 2.11 | 42.458 | 42.803 |
| | 3.22 | 33.176 | 36.768 |
| | 4.11 | 30.095 | 30.788 |
|  | 1.19 | 47.813 | 48.283 |
| | 2.19 | 42.268 | 42.048 |
| | 3.19 | 37.255 | 36.363 |
| | 1.91 | 46.961 | 46.180 |
| | 2.91 | 39.921 | 37.372 |
| | 3.91 | 33.856 | 31.292 |

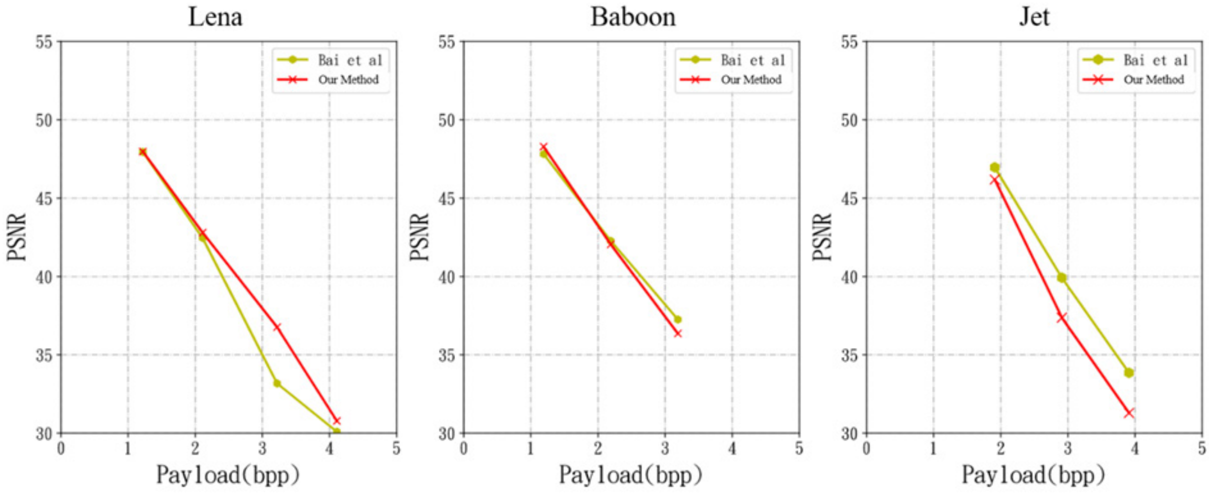


**Figure 17.** Steganographic performance of our method and that of Bai et al.

**Table 12.** Maximum capacities of our method and the method of Bai et al.

| | Lena | Baboon | Jet |
|---|---|---|---|
| Bai et al. | 4.11 bpp | 4.08 bpp | 4.10 bpp |
| Proposed method | 4.41 bpp | 4.31 bpp | 4.36 bpp |
| Percentage increase (%) | **+7.3%** | **+5.6%** | **+6.3%** |

In the case of the "Lena," "Baboon," and "Jet" cover images, our method outperforms that of Bai et al. in terms of PSNR, for the same payload. When the PSNR was maintained above 30 dB, the maximum payloads of the method of Bai et al. were 4.11 bpp, 4.08 bpp, and 4.10 bpp, respectively, for the aforementioned cover images. The corresponding maximum payloads of our method were 4.41 bpp, 4.31 bpp, and 4.36 bpp, respectively. Hence, our method provides significantly higher capacities (as listed in Table 12).

### 4.3.3 Comparison with the Method of Ghosal et al.

In the method of Ghosal et al. [11], only the MSB is retained, which is followed by ED using the LoG edge detector. Each pixel pair is then classified as an edge, mixed, or non-edge pair of pixels. Here, our method was compared to that of Ghosal et al. in terms of PSNR, with the same payloads. The results thus obtained are presented in Table 13 and Figure 18.

The results show that, for the same payloads, our method outperforms that of Ghosal et al. in terms of PSNR in the case of the "Lena," "Baboon," and "Jet" cover images.

**Table 13.** Comparison between our method and that of Ghosal et al.

| Experimental images | Payload (bpp) | Ghosal et al. | Proposed method |
|---|---|---|---|
| | | PSNR (dB) | |
| | 0.09 | 61.51 | 59.52 |
| | 0.38 | 47.83 | 53.77 |
| | 1.28 | 42.81 | 48.26 |
| | 3.09 | 38.45 | 36.94 |
| | 0.13 | 60.05 | 58.65 |
| | 0.55 | 43.13 | 52.55 |
| | 1.41 | 43.14 | 47.46 |
| | 3.13 | 37.64 | 36.19 |
| | 0.08 | 62.05 | 60.45 |
| | 0.34 | 46.24 | 54.11 |
| | 1.25 | 44.32 | 48.31 |
| | 3.08 | 38.65 | 37.13 |

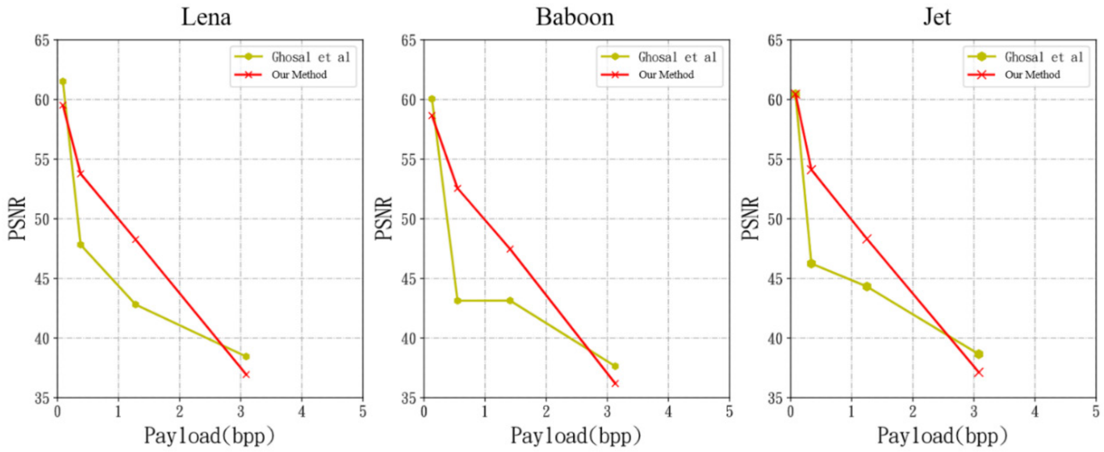


**Figure 18.** Steganographic performance of our method and that of Ghosal et al.

In the case of the "Lena," "Baboon," and "Jet" cover images, our method outperforms that of Ghosal et al. in terms of PSNR, for the same payload. When the PSNR was maintained above 30 dB, the maximum payloads of the method of Bai et al. were 3.09 bpp, 3.13 bpp, and 3.08 bpp, respectively, for the aforementioned cover images. The corresponding maximum payloads of our method were 4.41 bpp, 4.31 bpp, and 4.36 bpp, respectively. Hence, our method provides significantly higher capacities (as listed in Table 14).

**Table 14.** Maximum capacities of our method and the method of Ghosal et al.

| | Lena | Baboon | Jet |
|---|---|---|---|
| Ghosal et al. | 3.09 bpp | 3.13 bpp | 3.08 bpp |
| Proposed method | 4.41 bpp | 4.31 bpp | 4.36 bpp |
| Percentage increase (%) | **+42.7%** | **+37.7%** | **+41.6%** |

### 4.3.4 Comparison with the Method of Rezaei et al.

To compare this method with that of Rezaei et al. [12], the "Jet" image was used as the cover image. The PSNR of the method of Rezaei et al. is 61.19 dB at 0.2 bpp, compared to the PSNR of 60.45 dB at 0.08 bpp in the proposed method, which is inferior. The method proposed by Rezaei et al. [12] emphasizes the low distortion of stego-images, whereas the proposed method focuses on the maximum capacity of stego-images that can be achieved while maintaining acceptable image quality. The steganographic requirements of the proposed method thus differ from those of Rezaei et al.

### 4.3.5 Generalizability Test with BOSSBase Image Database

In this experiment, our method was compared to those of Tseng and Leng, Bai et al., and Ghosal et al., with each method being configured to maximize the capacity. The method of Tseng and Leng resulted in an average PSNR, capacity, and SSIM of 30.77 dB, 3.10 bpp, and 0.75, respectively. The method of Bai et al. resulted in an average PSNR, capacity, and SSIM of 31.38 dB, 4.04 bpp, and 0.77, respectively. The method of Ghosal et al. resulted in an average PSNR, capacity, and SSIM of 36.37 dB, 3.05 bpp, and 0.92, respectively. Our method (with the 4×4_edge14 block sampling method) provided an average capacity of 4.15 bpp, which was higher than all the aforementioned methods (as shown in Table 15). The PSNR of this study is inferior to that of Ghosal et al. because the maximum hiding capacity of Ghosal et al. is only 3.05 bpp, while that of this method is up to 4.15 bpp.

**Table 15.** Averaged results from BOSSBase image database

| Method | PSNR | SSIM | bpp |
|---|---|---|---|
| Tseng and Leng [8] | 30.77 | 0.75 | 3.10 |
| Bai et al. [10] | 31.38 | 0.77 | 4.04 |
| Ghosal et al. [11] | **36.37** | **0.92** | 3.05 |
| Proposed method | 31.78 | 0.80 | **4.15** |

### 4.3.6 Security Analysis of the Proposed Method

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. To this end, RS detection, pixel difference histogram (PDH) analysis and second-order SPAM features will be conducted on stego-images produced by proposed method.

#### 4.3.6.1 RS Detection

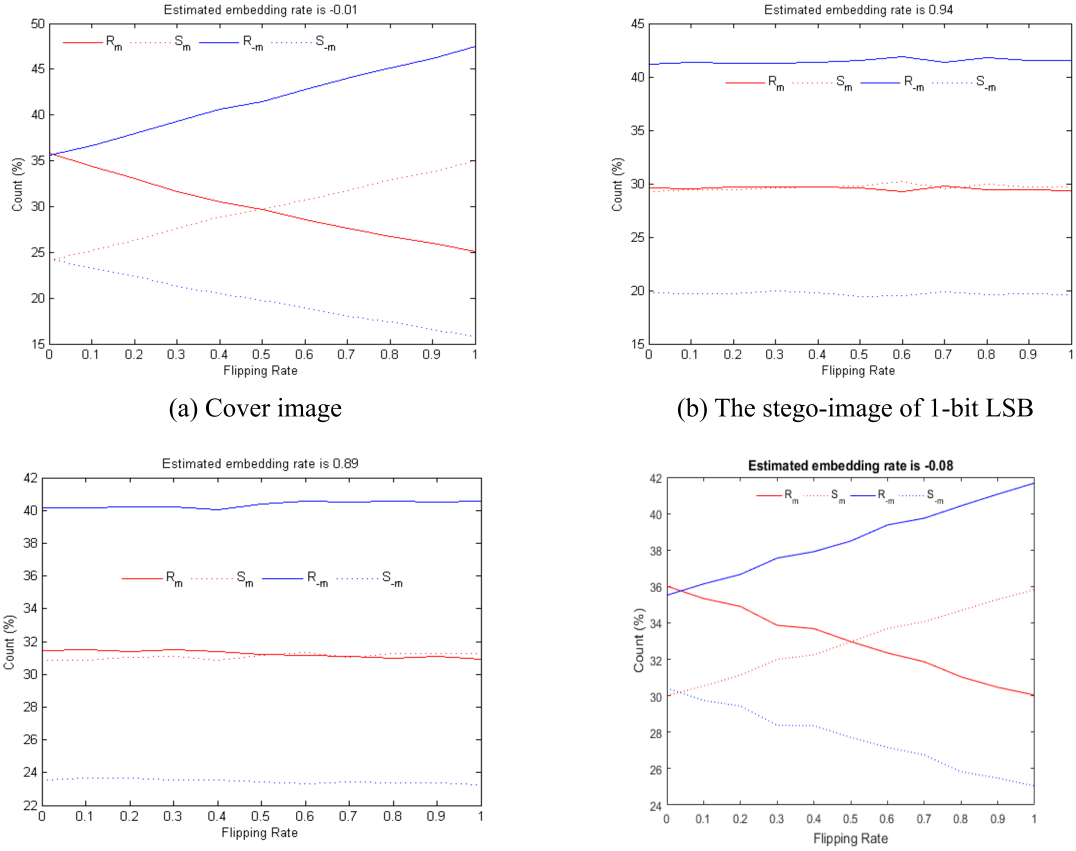
The security of the proposed method against the statistical RS detection technology [14] is depicted in Figure 19. Figure 19(a) shows the results for the case wherein RS detection technology is used for analysis of the Lena cover image. Figure 19(b) and Figure 19(c), respectively, show the results for the cases wherein 1-bit LSB and 3-bit LSB stego-images were used for the analyses. Figure 19(d) shows the result for the case in which the stego image generated using the proposed method was used for the analysis. From the RS detection results shown in Figure 19(a) to Figure 19(d), we can see that the evaluated embedding rates were -0.01, 0.94, 0.89, and -0.08, 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 OPAP and edge detection, is effective and robust against RS detection technology.

#### 4.3.6.2 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 pixel difference histogram (PDH) analysis, the number of times a certain adjacent-pixel difference value appears in an image is counted. In this regard, this study performs a security analysis on the cover image, original PVD and our method, by examining the change in the features of the PDH after embedding the cover image. The PDH of the cover image has a distribution that is nearly normal, as shown in Figure 20(a). However, as shown in Figure 20(b), 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 (c) of Figure 20, 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.

#### 4.3.6.3 Second-Order SPAM Features

Penvy et al. [15] presented second-order SPAM features for detection of steganographic methods that embed in the spatial domain. To prove the security of the proposed method against the second-order SPAM features, 5000 cover images were retrieved from the BOSSBase image database [13] and the corresponding stego-images of the proposed method (4×4_edge14) were used to carry out the following experiment. The training image sets consisted of 3000 cover images and 3000 stego-images for the proposed steganography algorithm. The remaining cover images and stego-images were used for test image sets. The first step was to extract the 686 features of SPAM of training images. Furthermore, the stego-images and cover images were given diverse labels. The purpose of the different labels used in the Back-Propagating Neural Network (BPNN) training stage was to obtain the relationship between feature sets and classification categories. The second step was to use a more flexible classifier, BPNN, which was employed to discriminate between cover images and the stego-images. Finally, according to the classification results, the detection accuracy was calculated. The security of the proposed method against second-order SPAM features is presented in Table 16. From the SPAM detection results shown in Table 16, we can see that the accuracy was 62.8%. Table 16 shows that the SPAM features cannot effectively identify the stego-images generated using the proposed method and the cover images. This verifies that the stenography method proposed in this study is effective against second-order SPAM features.



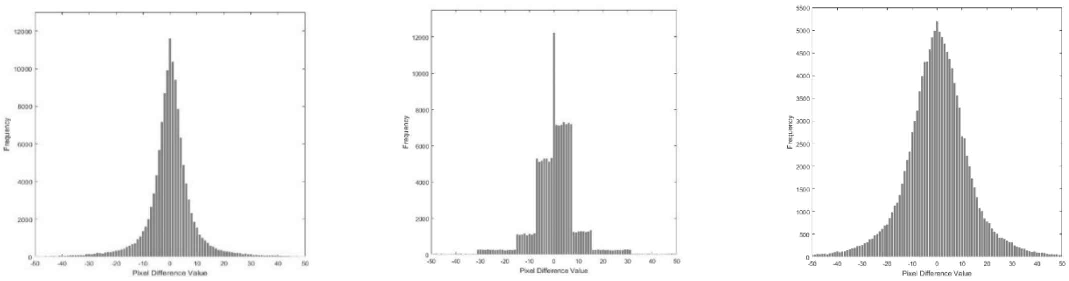
(a) Cover image

(b) The stego-image of 1-bit LSB

(c) The stego-image of 3-bit LSB

(d) The stego-image of proposed method

**Figure 19.** Results of the RS detection



(a) Cover image

(b) Original PVD

(c) Proposed method

**Figure 20.** PDH of pixel difference values with cover image and different steganographic methods

**Table 16.** Security of the proposed method against second-order SPAM features

| TP | FN | TN | FP | Accuracy |
|---|---|---|---|---|
| 935 | 1065 | 1577 | 423 | 62.8% |

## 5 Conclusion

In steganography, the two most important concerns are steganographic capacity and stego-image quality, which are contradictory requirements. The aim of this study was to develop a method that maximizes the steganographic capacity while maintaining an adequate level of stego-image quality. In the proposed method, the LoG edge detector is first used to classify the pixels of the cover image into edge and non-edge pixels. The cover image is then divided into 4×4 non-overlapping blocks; if a block contains 14 or more edge pixels, it is considered an edge block. The pixels are then embedded with cipher text, with the embedding length depending on whether the block it belongs to is an edge or non-edge block. When the PSNR is required to be >30 dB, our method provided an average PSNR, SSIM, and capacity of 31.78 dB, 0.801, and 4.15 bpp, respectively, with the BOSSBase image database. In addition to having a larger capacity than other comparable steganographic techniques, the proposed method also performs well in terms of PSNR and SSIM. Finally, proposed method can resist the detection of RS, pixel difference histogram analysis and second order SPAM features.

## References

[1] A. Westfeld, F5-A Steganographic Algorithm: High Capacity Despite Better Steganalysis, in: I. S. Moskowitz, I.S. (Eds.), *Proc. International Workshop Information Hiding*, *Lecture Notes in Computer Science, Vol. 2137*, Springer, Berlin, Heidelberg, 2001, pp. 289-302.

[2] K. L. Chung, *Image processing and computer vision*, Tung Hua Book Co., Ltd., Taipei, Taiwan, 2009.

[3] 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.

[4] T. Morkel, J. H. P. Eloff, M. S. Olivier, An Overview of Image Steganography, *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005)*, Sandton, South Africa, 2005, pp. 1-11. (Published electronically)

[5] D. Marr, E. Hildreth, Theory of edge detection, *Proceedings of the Royal Society of London. Series B. Biological Sciences*, Vol. 207, No. 1167, pp. 187-217, February, 1980.

[6] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, *IBM systems journal*, Vol. 35, No. 3.4, pp. 313-336, 1996.

[7] W. J. Chen, C. C. Chang, T. H. N. Le, High payload steganography mechanism using hybrid edge detector, *Expert Systems with applications*, Vol. 37, No. 4, pp. 3292-3301, April, 2010.

[8] H. W. Tseng, H. S. Leng, High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion, *IET Image Processing*, Vol. 8, No. 11, pp. 647-654, November, 2014.

[9] S. Islam, M. R. Modi, P. Gupta, Edge-based image steganography, *EURASIP Journal on Information Security*, Vol. 2014, No. 1, 1-14, April, 2014.

[10] J. Bai, C. C. Chang, T. S Nguyen, C. Zhu, Y. Liu, A high payload steganographic algorithm based on edge detection, *Displays*, Vol. 46, pp. 42-51, January, 2017.

[11] S. K. Ghosal, J. K. Mandal, R. Sarkar, High payload image steganography based on Laplacian of Gaussian (LoG) edge detector, *Multimedia Tools and Applications*, Vol. 77, No. 23, pp. 30403-30418, May, 2018.

[12] A. Rezaei, Y. Ahmadiadli, L. Farzinvash, M. Asadpour, Low distortion and adaptive image steganography by enhancing DBSCAN, Sobel operator, and XOR coding, *Journal of Information Security and Applications*, Vol. 70, Article No. 103343, November, 2022.

[13] BOSS Break Oure Steganographic System, *BOSSBases (v0.93)* [Online], Available: http://agents.fel.cvut.cz/boss/index.php?mode=VIEW&tmpl=materials, Accessed on: March 2, 2022.

[14] 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.

[15] T. Pevný, P. Bas, J. Fridrich, Steganalysis by subtractive pixel adjacency matrix, *IEEE Transactions on Information Forensics and Security*, Vol. 5, No. 2, pp. 215-224, June, 2010.

## 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 security, information management, information hiding and steganalysis.

**Sheng-Chih Ho** received the Ph.D. degree in department of electrical and electronic engineering from National Defense University, Taoyuan, Taiwan, R.O.C., in 2012. Currently, he is an assistant professor in the Department of Management Information System, Takming University of Science and Technology, Taipei, Taiwan. His current research interests include information security and data science.

**Tai-Hsiu Wu** 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.