



# High payload steganography mechanism using hybrid edge detector

Wen-Jan Chen<sup>a,\*</sup>, Chin-Chen Chang<sup>b,c</sup>, T. Hoang Ngan Le<sup>d</sup>

<sup>a</sup> Department of Computer Science and Information Engineering, Da-Yeh University, No. 168 University Rd., Dacun, Changhua 51591, Taiwan, ROC

<sup>b</sup> Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan, ROC

<sup>c</sup> Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan, ROC

<sup>d</sup> Department of Computer Science, Natural Science University, 227 Nguyen Van Cu, District 5, HCMC, Vietnam

## ARTICLE INFO

### Keywords:

LSB steganography  
Stego-image  
Image steganalysis  
Hybrid edge detection

## ABSTRACT

Steganography is the art and science of hiding data into information. The secret message is hidden in such a way that no one can apart from the sender or the intended recipient. The least significant bit (LSB) substitution mechanism is the most common steganographic technique for embedding a secret message in an image with high capacity, while the human visual system (HVS) would be unable to notice the hidden message in the cover image. In this paper, besides employing the LSB substitution technique as a fundamental stage, we take advantage of edge detection technique. The experimental results show that the proposed scheme not only achieves high embedding capacity but also enhances the quality of the stego image from the HVS by an edge detection technique. Moreover, based on that the secret message is replaced with different LSBs, our scheme can effectively resist the image steganalysis.

© 2009 Elsevier Ltd. All rights reserved.

## 1. Introduction

In the fields of information hiding, there is a visual requirements model, which is called magic triangle, given in Fig. 1 (Johnson, Duric, & Jajodia, 2001). The first requirement, called capacity or also embedding payload, is determined by the number of secret bits embedded in each cover pixel. A higher capacity allows much more the secret data to be inserted into the cover image. The second requirement, named imperceptibility, is usually calculated by peak signal-to-noise ratio (PSNR). When the difference between the cover image and the stego image is small, the PSNR value is high. Thus, the stego image quality is considered to be good with the imperceptibility is high. The last requirement called robustness which prevents the secret data from being attacked or stolen. Obviously, Fig. 1 is a convenient model for a visual representation of the required trade-offs between the capacity of the embedded data and the robustness to certain attacks, while keeping the perceptual quality of the stego image at an acceptable level.

Steganography is a technique which is used to transmit a secret message under the cover of digital media such as images. It first pays much more attention on embedding payload rather than robustness against intentional attacks compared with watermark which is used to protect the copyright. Moreover, imperceptibility, which is the second requirement, is carefully considered in the ste-

ganography algorithms. Thus, an effective steganography scheme should not cause any perceptible distortion and have to achieve high capacity as well.

In most steganographic techniques, although only the most insignificant components are altered, many analytical techniques can reveal existence of the hidden message by detecting statistical difference between the cover and stego objects. The following two measures may be taken in developing steganographic schemes to combat steganalysis:

- (1) Avoid conspicuous parts when embedding messages into the cover.
- (2) Improve embedding efficiency, i.e., embed more information per modification to the cover data.

This paper carefully regards to first two requirements described in Fig. 1 and both above measures. To achieve this goal, in this paper, we proposed a very simple and effective method that based on the LSB technique in combining with the edge detection mechanism. The LSB steganography, that replaces the LSBs of the cover image by the secret message, is a widely used technique with low computational complexity and high insertion capacity (Bender, Gruhl, Morimoto, & Lu, 1996; Böhme, & Westfeld, 2004; Farid, 2002; Fridrich, 2003). Although it has good perceptual transparency, it is vulnerable to steganalysis which is based on the statistical analysis. Many other steganography algorithms have been developed such as spread spectrum embedding. But the embedding capacity is not satisfied. To develop a new LSB steganography algorithm that can against statistical analysis, we apply the edge

\* Corresponding author. Tel.: +886 4 8511888x2403; fax: +886 4 8511350.

E-mail addresses: [cwj@mail.dyu.edu.tw](mailto:cwj@mail.dyu.edu.tw) (W.-J. Chen), [ccc@fcu.edu.tw](mailto:ccc@fcu.edu.tw), [ccc@cs.ccu.edu.tw](mailto:ccc@cs.ccu.edu.tw) (C.-C. Chang), [lthngan@fit.hcmuns.edu.vn](mailto:lthngan@fit.hcmuns.edu.vn) (T.H.N. Le).

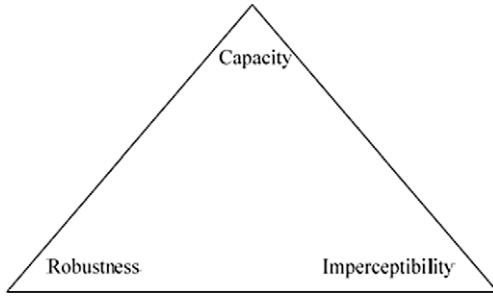


Fig. 1. Magic triangle-three requirements model.

detection mechanism in this proposed scheme. In each cover pixel, this mechanism allows selecting various numbers of LSBs which are used to replace with the secret message. Moreover, this mechanism helps improving not only the quality of the stego image but also the embedding payload.

Noticeably, the advantage and strength of our scheme is obtained by the edge detection. Thus, to exploit as many edge pixels as possible in the cover image, we use the combination of a fuzzy edge detector and a canny edge detector. The first edge detector helps detecting a large number of edge pixels in an image to provide clear and thick edge images. The second detector is designed to be an optimal edge detector which is considered as the most rigorously defined operator and is widely used. Moreover, the Canny edge detector can be attributed to its optimality according to the three criteria of good detection, good localization, and single response to an edge.

This paper is organized as follows: In Section 2, the techniques of the classic LSB steganography, the fuzzy edge detection and the Canny edge detection are reviewed. In Section 3, we give a detailed description of the embedding and decoding phases while paying close attention to implementation issues. Section 4 shows the experimental results and analyzes the anti-statistical-detection ability of our scheme. Finally, Section 5 is our conclusion.

## 2. Related works

### 2.1. The classic LSB steganography

In the classic LSB steganography, the secret message is considered to be a bit stream and is embedded into the cover image by replacing the LSBs of the cover image with the secret message. Consider an 8-bit grayscale bitmap image where each pixel is presented by a byte corresponding to a grayscale value. Suppose the first three pixels  $\{P_1, P_2, P_3\}$  of the original image have the following grayscale values:  $[1\ 1\ 0\ 1\ 0\ 0\ 1\ 0]$ ,  $[0\ 1\ 0\ 0\ 1\ 0\ 1\ 0]$  and  $[1\ 0\ 0\ 1\ 0\ 1\ 1\ 1]$ , respectively. To hide the secret message  $S$  whose binary value is  $1\ 0\ 0$ , we replace the LSBs of  $P_1, P_2, P_3$  with bit stream  $S$ . These stego

pixels  $P_1, P_2$  and  $P_3$  have the following new grayscale values:  $[1\ 1\ 0\ 1\ 0\ 0\ 1\ 1]$ ,  $[0\ 1\ 0\ 0\ 1\ 0\ 1\ 0]$  and  $[1\ 0\ 0\ 1\ 0\ 1\ 1\ 0]$ . Fig. 2b shows the stego image which is created by replacing one LSB in each pixel of the cover image (Fig. 2a) with the secret message “Computer Sciences” embedded.

Obviously, the difference between the cover image and the stego image is hardly noticeable to the human eye. For increasing the embedding capacity, two or more LSBs in each pixel can be used to embed messages. However, there is a trade-off between the embedding payload and the quality of the stego image. The Fig. 3 demonstrates quality of the stego image when the number of LSBs called  $n$  are chosen from 1 to 5 and the cover image Lena sized is  $128 \times 128$ .

### 2.2. Edge detector

An edge is characterized by significant dissimilarity in gray levels being used to indicate the boundary between two regions in an image fragment. Edge detection is a significant area of the image processing and machine vision due to the fact that edges are considered to be the important features for analyzing the most essential information contained in images. Many classical edge operators such as Sobel, Prewitt, Laplacian and Canny operators are already available in the literature (Sonka, Hlavac, & Boyle, 1999). Among these edge detection methods proposed so far, the Canny edge detector is considered the most rigorously defined operator and is widely used. The popularity of the Canny edge detector can be attributed to its optimality according to the three criteria of good detection, good localization, and single response to an edge. It also has a rather simple approximate implementation, which is the subject of this paper.

Furthermore, in recent years, fuzzy techniques have also been used to develop new methods for edge detection due to the flexibility in dealing with the ambiguity and vagueness often present in digital images. Several fuzzy reasoning/inference and logic-based edge detection techniques have been reported (Kuo, Lee, & Liu, 1997; Russo, 1998; Tao, Thompson, & Taur, 1993; Tizhoosh, 2002). The proposed techniques, which are based on fuzzy reasoning/inference and logic, are flexible and robust, but are generally very expensive in computing in comparison to the classical methods. We use a simple fuzzy complement edge operator capable of detecting a large number of edge pixels in an image to provide clear and sharp edge images with less computational effort (Westfeld, & Pfitzmann, 1999).

With the combination of the Canny edge detector and the fuzzy edge detector, we can achieve many more edge pixels than if we had used either edge detector individually. The details of the fuzzy edge detector and the Canny edge detector are described in the following sub-sections.

#### 2.2.1. Fuzzy edge detector

The details of the edge operator based on the fuzzy complement is briefly presented as follows with the assumption that  $X$  is an image of dimension  $W \times H$ , and all the pixels in  $X$  are gray level from 0 to  $L$  (i.e.,  $x_{mn} \in [0, L]$ ):

*Step 1:* Obtain the membership grade value  $\mu_{mn}$  at position  $(m, n)$ .

The image  $X$  is first transformed into an array  $F$  of fuzzy singletons,  $\mu_{mn} \in [0, 1]$  with  $m \in [1, W]$  and  $n \in [1, H]$ . Note that in this domain, each value  $\mu_{mn}$  is called membership grade and indicates the degree of brightness of each pixel. The array  $F$  is a union of all  $\mu_{mn}$ 's and is determined as follows:

$$F = \bigcup_{m=1}^W \bigcup_{n=1}^H \mu_{mn} \quad (1)$$



(a). Cover image (b). Stego image

Fig. 2. The cover image and the stego image.



Fig. 3. The quality of the stego image when the number of LSBs changes from 1 to 5.

Let  $x$  be the biggest gray scale value in image  $X$ . Since all pixel values in image  $X$  have values of less than or equal to  $x$ ,  $x$  is obviously not larger than  $L$ . The membership grade value is obtained by a simple normalization Eq. (2):

$$\mu_{mn} = \frac{x_{mn}}{x} \quad (2)$$

Step 2: Determine the degree of edginess  $\bar{\mu}_{mn}$  at position  $(m, n)$ .

The simplest way to define a fuzzy edge detector is the determination of a proper membership function  $\bar{\mu}_{mn}$  for each pixel  $x_{mn}$  at the position  $(m, n)$  with a surrounding  $w \times w$  spatial window (Amarunnishad, Govindan, & Mathew, 2008). To do this, we need to sub-divide the image  $X$  into overlapping  $w \times w$  blocks. Let  $W(m, n)$  be a  $w \times w$  window, where  $x_{mn}$  is a center pixel.

Employing the fuzzy complement, the membership function can be defined for window  $w \times w$  as:

$$\bar{\mu}_{mn} = \min \left( 1, \frac{\tau}{w} \sum_i \sum_j \min(\mu_{ij}, 1 - \mu_{ij})^p \right)^{1/p} \quad (3)$$

or

$$\bar{\mu}_{mn} = \min \left( 1, \frac{\tau}{w} \sum_i \sum_j \mu_{ij} \times (\mu_{ij}, 1 - \mu_{ij})^p \right)^{1/p} \quad (4)$$

In this situation, the membership grade value  $\mu_{mn}$  is locally calculated inside each window  $w \times w$ :

$$\mu_{mn} = \frac{\{\max(x_{ij}) - \min(x_{ij}) | i, j \in [1, w]\}}{x} \quad (5)$$

where  $\frac{\tau}{w}$  is the scaling factor and  $\mu_{ij}$  is calculated by Eq. (2). Based on the experimental results, to attain the goal of high visual quality edge image, and to ensure a large number of detected edge pixels, Amarunnishad et al. (2008) suggested that the values of ' $\tau$ ' and the window size ' $w$ ' shall be 9 and 3, respectively.

Step 3: Obtain the edge image.

Let us assume that  $F'$  is an image containing all edges of image  $F$  in fuzzy domain. The image  $F'$  is also an array of fuzzy singleton  $\bar{\mu}_{mn}$  and is determined as in Eq. (6).

$$F' = \bigcup_{m=1}^M \bigcup_{n=1}^N \bar{\mu}_{mn} \quad (6)$$

In order to highlight the performance of the fuzzy edge detector, the experimental results of the test images "Tiffany", "Lena", "Pepper", "Building" are considered. Fig. 4 shows the visual quality of the edge images and the number of edge pixels which are generated by the fuzzy edge detector with index ' $p$ ' equal 1.

#### 2.2.2. Canny edge detector

The Canny edge operator has three characteristics (Nanning, 1998): (1) No important edges should be missed, and there should be no false edges, while the error detection rate should be kept low. (2) The distance between the actual and located position of the edge should be minimal. (3) There is only one response to a single edge.

The Canny edge detector is widely used in computer vision to locate sharp intensity changes and to find the object boundaries in an image. The Canny edge detector classifies a pixel as an edge if the gradient magnitude of the pixel is greater than those of the pixels on both sides of it in the direction of maximum intensity change. A typical implementation of the Canny edge detector (Trucco, & Verri, 1998; Jain, Kasturi, & Schunck, 1995) follows the following steps:

Step 1: The image is first smoothed by the Gaussian filter mask. At the beginning, we divide the image into a set of blocks. The size of each block is equal to the size of the Gaussian filter mask. The mask is applied in the image by convolution operation.

Grayscale image (I)				
Edge image (I <sub>1</sub> )				
The number of edge pixels	6151	15408	13937	8957

Fig. 4. The edge images are generated by the fuzzy edge detector.



Step 2: Determine gradient magnitude and gradient direction of each pixel. This step is done by using the Sobel operators. Basically, we use 2-D spatial gradient in which  $G_x$  and  $G_y$  is defined as follows:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Step 3: If the gradient magnitude at a pixel is greater than those of its neighbors in the gradient direction, mark the pixel as an edge. Otherwise, mark the pixel as the background.

Step 4: Remove the weak edges by hysteresis threshold.

In order to highlight the performance of the Canny edge detector, the experimental results of the test images “Tiffany”, “Lena”, “Pepper” and “Building” are considered. Fig. 5 shows the visual

Grayscale image ( $I$ )				
Edge image ( $I_2$ )				
The number of edge pixels	6103	5152	4854	4480

Fig. 5. The edge images are generated by the Canny edge detector.













Grayscale image ( $I$ )				
Edge image ( $I_1$ )				
The number of edge pixels	6151	15408	13937	8957
Edge image ( $I_2$ )				
The number of edge pixels	6103	15408	13937	8957
Edge image ( $I'$ )				
The number of edge pixels	10410	16581	15196	10660

Fig. 6. The edge images are generated by the hybrid edge detector.

quality of the edge images and the number of edge pixels which are generated by the Canny edge detector.

### 2.2.3. Hybrid edge detector

In this sub-section, a hybrid edge detector is constructed by a combination of the fuzzy edge detector and the Canny edge detector. Let us denote the grayscale image, the edge image generated by the fuzzy edge detector and the edge image generated by the Canny edge detector as  $I$ ,  $I_1$  and  $I_2$ , respectively. The edge image which is generated by the hybrid edge detector is called  $I'$ . Herein,  $I'$  is determined by performing the OR operation in  $I_1$  and  $I_2$ . This combination not only increases the number of edge pixels, but also clearly and precisely finds the object boundaries in the image. Fig. 6 shows the edge images and the number of edge pixels which are created from this strategy.

Obviously, the object boundaries achieved by our hybrid edge detector are more clear and precise than those obtained by either individual edge detector. Moreover, our scheme produces a greater number of edge pixels than the two edge detectors.

## 3. Proposed scheme

In this section, a novel LSB steganography scheme, which uses hybrid edge detection, is thoroughly presented. Like other data hiding schemes, the proposed scheme consists of two procedures: the embedding procedure and the extracting procedure.

### 3.1. Embedding procedure

This procedure contains three phases, and the flowchart of these phases is illustrated in Fig. 7.

**Phase 1:** Applying the hybrid edge detector, we obtain the edge image  $I'$  from the grayscale image  $I$ .

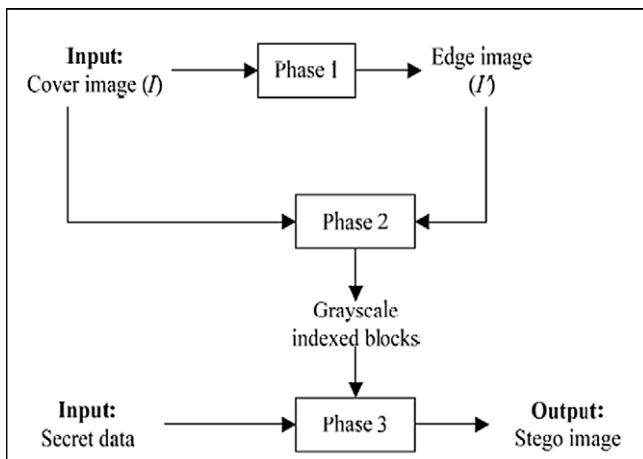


Fig. 7. Flowchart of the embedding procedure.

**Phase 2:** Divide the edge image  $I'$  into a set of blocks. Each block contains  $n$  pixels and is called  $n$ -pixel block. The  $n$  pixels are indexed as  $P_1, P_2, \dots, P_n$ . Herein, we use  $P_1$  to store the status of the remaining pixels. The status of each pixel,  $P_i$ , is defined as '1' if  $P_i$  is an edge pixel. Otherwise, the status of each pixel,  $P_i$ , is defined as '0'. The status of pixels from  $P_2$  to  $P_n$  is stored inside  $P_1$  by an LSBs substitution operation.

For example, take a block  $A = [P_1, P_2, P_3]$ , with  $n = 3$ . In this example, assume that  $P_1$  and  $P_3$  are edge pixels. Thus, the status of the pixels  $P_2$  and  $P_3$  is '01'. And, we will replace two LSBs in the pixel  $P_1$  with '01'.

In this phase, whether a pixel is considered to be an edge pixel or not is based on the edge image. This edge image is created in Phase 1 using the hybrid edge detector.

In our approach, pixel  $P_1$  is considered to be the index of  $n$ -pixel block. Because the values of the LSBs in  $P_1$  are changed by the status of pixels  $P_2, P_3, \dots, P_n$ , the length of block is carefully considered. If there are  $n$  pixels in each block, we need to use  $(n - 1)$  bits to represent the status of the pixels  $P_2, P_3, \dots, P_n$ . Thus, we need to change  $(n - 1)$  LSBs in the pixel  $P_1$ . To preserve the quality of pixel  $P_1$  as well as to increase the embedding payload, based on the experimental results, we suggest assigning the values of  $n$  as 3, 4 or 5.

**Phase 3:** To embed the secret message bits into an  $n$ -pixel block, we separate the  $n$ -pixel block into two categories corresponding to non-edge pixels category and edge pixels category. Each cover pixel in the first category contains 'x' secret message bits using the LSBs substitution technique. Each cover pixel in the second category contains 'y' secret message bits using the LSBs substitution technique. Obviously, to maintain the quality of the stego image, the value of  $x$  here is 1 or 2, as usual. The experimental results show that we can choose the value of 'y' as 3, 4, or 5 without causing any perceptible distortion. This phase can be explained using the Fig. 8.

For example, let us consider an image  $A$  having four pixels as  $\{[1\ 0\ 1\ 0\ 1\ 0\ 1\ 0], [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0], [1\ 1\ 1\ 1\ 1\ 1\ 0\ 0], [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1]\}$  corresponding to  $P_1, P_2, P_3$  and  $P_4$  with the secret message  $S = '0\ 1\ 1\ 0\ 1\ 0\ 1'$ . The image  $A$  is considered to be a four-pixel block.

Let us assume that based on the hybrid edge detector, we determine that  $P_2$  and  $P_4$  are edge pixels. Obviously, the status of  $P_2, P_3$  and  $P_4$  is '101'. Replace 3 LSBs in pixel  $P_1$  with '101'. Thus, the pixel  $P_1$  receives the new value of  $[1\ 0\ 1\ 0\ 1\ 1\ 0\ 1]$  and becomes pixel  $P'_1$ .

Let us assume that the values of parameters 'x' and 'y' are 1 and 3, respectively. Herein, we replace three LSBs in pixel  $P_2$  with three secret message bits. Also, we replace one LSB in pixel  $P_3$  with one secret message bit. Similarly, we replace three LSBs in pixel  $P_4$  with three secret message bits. The new values of pixels  $P_2, P_3$  and  $P_4$  are  $[1\ 0\ 0\ 0\ 0\ 0\ 1\ 1]$ ,  $[1\ 1\ 1\ 1\ 1\ 1\ 0\ 0]$  and  $[0\ 0\ 0\ 0\ 1\ 1\ 0\ 1]$ , respectively. Thus, the new value of the image  $A$ , which is called stego image  $A'$ , is  $\{[1\ 0\ 1\ 0\ 1\ 1\ 0\ 1], [1\ 0\ 0\ 0\ 0\ 0\ 1\ 1], [1\ 1\ 1\ 1\ 1\ 1\ 0\ 0], [0\ 0\ 0\ 0\ 1\ 1\ 0\ 1]\}$ .

The entire embedding procedure in this example can be represented in the Fig. 9.

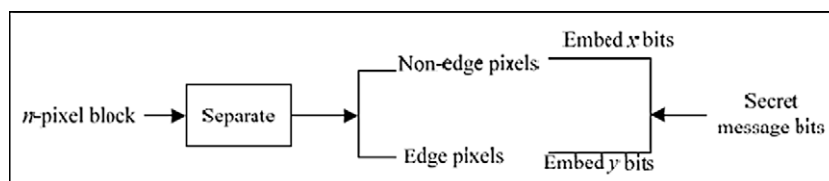


Fig. 8. The details of Phase 3.

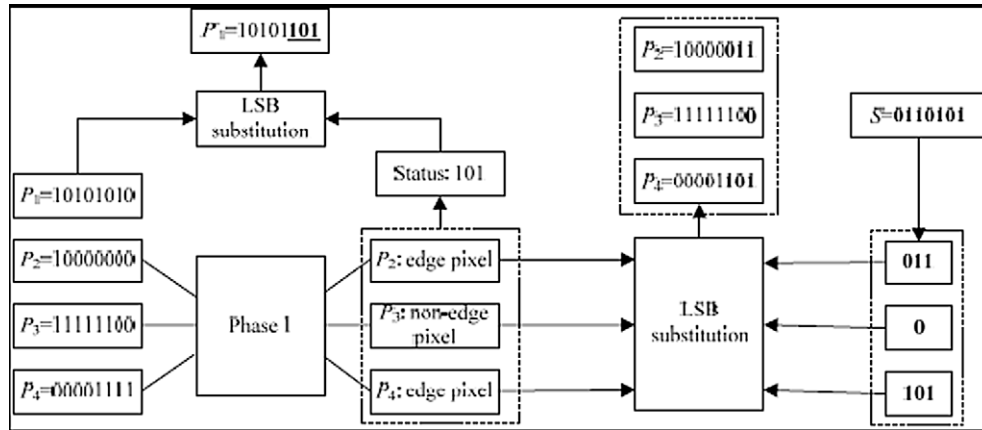


Fig. 9. Example of the embedding procedure.

### 3.2. Extracting procedure

The flowchart of this extracting procedure, having two phases, is illustrated in Fig. 10.

**Phase 1:** Similar to the dividing operation presented in the previous procedure. Here, we divide the stego image into a set of blocks, each block contains  $n$  pixels and is called  $n$ -pixel block. The  $n$  pixels in each block are indexed as  $P'_1, P'_2, \dots, P'_n$ .

**Phase 2:** Based on the  $(n-1)$  LSBs in pixel  $P'_1$ , we obtain the status of the remaining pixels from  $P'_2$  to  $P'_n$ . From this status value, we can identify two categories corresponding to the non-edge pixels category and the edge pixels category. To extract the secret message bits, we get  $y$  LSBs from the first category and  $x$  LSBs from the second category. The secret message is generated by appending all of the LSBs from the above two categories.

For example, take a stego image  $A'$  having four pixels as  $\{[10101101], [10000011], [11111100], [00001101]\}$  corresponding to four pixels  $P'_1, P'_2, P'_3$  and  $P'_4$ .

Obtain  $(n-1) = 3$  LSBs in the first pixel, we get three bits as '101'. Thus the second and the fourth pixels are edge pixels. And, the third pixel is a non-edge pixel. Based on the assumption of the embedding procedure, we will extract three LSBs from the pixel  $P'_2$  and the pixel  $P'_4$ . Also, we extract one LSB from the pixel  $P'_3$ . The extracted bits from the pixel  $P'_2$  are '011'. The extracted bit from the pixel  $P'_3$  is '0'. The extracted bits from the pixel  $P'_4$  are '101'. By appending these extracted bits, we obtain the secret message as '0010101'.

## 4. Experimental results and analysis

The experimental results presented in this section demonstrate the performance of our proposed scheme. To conduct our experiments, we used four  $128 \times 128$  grayscale images, "Tiffany", "Lena", "Pepper" and "Building". These test images are shown in Fig. 11.

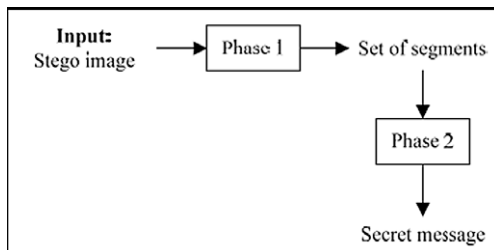
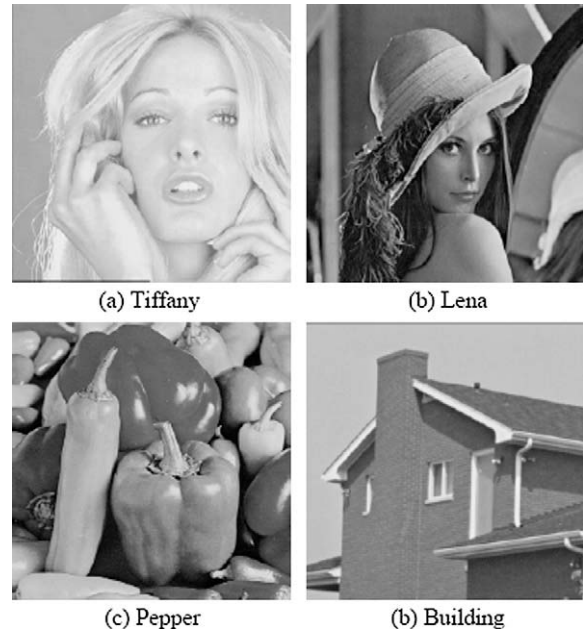


Fig. 10. The flowchart of the extracting procedure.

Fig. 11. Four  $128 \times 128$  grayscale images.

The stego image quality is considered from two viewpoints. First, we use the peak signal-to-noise ratio (PSNR) measurement to evaluate the difference between the stego and cover images. Second, we compare the quality of the stego image to that of the cover image as seen by the human visual system (HVS).

The PSNR formula is defined as:

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \text{ (dB)} \quad (7)$$

where  $MSE$  is the mean square error between the cover and stego images. For a cover image whose width and height are  $W$  and  $H$ ,  $MSE$  is defined as







$$MSE = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H (I_{ij} - I'_{ij})^2 \quad (8)$$

where  $I_{ij}$  and  $I'_{ij}$  are the pixel values of the cover and stego images, respectively. A higher PSNR indicates that the quality of the stego image is better and that it is more similar to the cover image.

Table 1 shows the quality of the stego image generated by our proposed scheme in the case that: (1) we replace one LSB in each



























**Table 1**The performance of our scheme on  $128 \times 128$  Lena when  $x = 1$ ,  $n = 2$  and  $y \geq 1$ .

$x = 1, n = 2$						
$y$	1	2	3	4	5	6
Stego image						
	PSNR = 51.1 dB	PSNR = 50.0 dB	PSNR = 47.1 dB	PSNR = 42.3 dB	PSNR = 36.9 dB	PSNR = 31.3 dB
Payload	0.5 bpp	0.575 bpp	0.65 bpp	0.73 bpp	0.80 bpp	0.87 bpp
Ratio	(8192 bits)	(9427 bits)	(10,662 bits)	(11,897 bits)	(13,132 bits)	(14,367 bits)

**Table 2**

The stego image quality with the classic LSB steganography scheme.

$x$	1	2	3	4	5	6
Stego image Tiffany						
	PSNR = 51.1 dB	PSNR = 44.7 dB	PSNR = 37.9 dB	PSNR = 31.8 dB	PSNR = 25.0 dB	PSNR = 18.5 dB
Stego image Lena						
	PSNR = 51.1 dB	PSNR = 44.1 dB	PSNR = 38.0 dB	PSNR = 31.9 dB	PSNR = 25.7 dB	PSNR = 19.8 dB
Stego image Pepper						
	PSNR = 51.2 dB	PSNR = 44.3 dB	PSNR = 37.9 dB	PSNR = 31.2 dB	PSNR = 25.8 dB	PSNR = 19.9 dB
Stego image Building						
	PSNR = 51.1 dB	PSNR = 44.6 dB	PSNR = 38.0 dB	PSNR = 31.7 dB	PSNR = 25.7 dB	PSNR = 20.2 dB
Payload	1 bpp 16,384 (bits)	2 bpp 32,768 (bits)	3 bpp 49,152 (bits)	4 bpp 65,536 (bits)	5 bpp 81,920 (bits)	5 bpp 98,304 (bits)

non-edge pixel by the secret message bit, (2) there are two cover pixels in each block, and (3) the number of LSBs, which are inserted into each edge pixel, are chosen from 1 to 6.

From Table 1, we can see that even when replacing six LSBs in each edge pixel by the secret image, not only is the PSNR high, but the stego image quality is also good as seen by the HVS. To ensure that our scheme offers a greater advantage than the classic LSB steganography scheme, we will perform the classic LSB steganography scheme on the test images. Table 2 gives the the stego

image quality when the number of LSBs in each pixel is chosen from 1 to 6.

From the HVS, we can see that the quality of the stego image is completely acceptable when one or two LSBs in each cover pixel are replaced with the secret message. However, the quality of the stego image is unacceptable in cases where the number of LSBs is greater than 4. In cases where values of  $x$  are 3 and 4, we need to carefully consider both the PSNR values and the image quality under the HVS. Table 3 shows the experimental results of the four

**Table 3**

The stego image quality generated by the classic LSB steganography scheme.

$x$	3	4
Lena	 PSNR = 38.0 dB	 PSNR = 31.9 dB
Tiffany	 PSNR = 37.9 dB	 PSNR = 31.8 dB
Pepper	 PSNR = 37.9 dB	 PSNR = 31.2 dB
Building	 PSNR = 38.0 dB	 PSNR = 31.7 dB

test images when we use the secret message bits to replace three and four LSBs in each cover pixel.

As seen in Table 3, when three LSBs in each cover pixel are replaced by the secret message bits, the PSNR values of the stego images are still high (greater than 37 dB). However, the stego image quality under the HVS is not good. In other words, by the HSV, the stego images are neither smooth nor sharp enough, and we can clearly see some noises. Thus, the classic LSB steganography scheme is only able to perform well if there are one or two LSBs replaced with secret message bits.

To prove that our approach can improve the stego image quality in a case where more than three LSBs in each cover pixel are replaced, we perform the proposed scheme on the  $128 \times 128$  Lena

with the values of parameters chosen as:  $x = 3$ ,  $n = 3$  and  $y = 3, 4, 5$ . The experimental results are given in Table 4. Note that  $x$  and  $y$  correspond to the number of LSBs in each non-edge pixel and in each edge pixel which are replaced by secret message bits. Also note that  $n$  is the length of each block divided from the cover image. From this table, we can see that the stego images, which are generated by our scheme, have good quality under the HVS even when the PSNR values are lower than 39 dB.

**Conclusion 1:** With the same PSNR value, our scheme gives better stego image quality than the classic steganography scheme as seen by the HVS.

Obviously, the advantage of the proposed scheme depends on the three parameters  $x$ ,  $y$  and  $n$ . To prove that our scheme can achieve better stego image quality and obtain a high embedding payload, we perform the proposed scheme with various values of  $x$ ,  $y$  and  $n$ . Tables 5–9 present the relationship among these parameters and the stego image quality.

Tables 5–9 show that the larger values of parameters  $x$ ,  $y$  and  $n$  will yield a higher embedding payload but lower stego image quality. Take Tables 5 and 7 for example, with the same values of  $x$  and  $y$ , the PSNR will decrease when we increase value  $n$ . Also, as seen in Tables 8 and 9, with the same values of  $n$ , the PSNR decreases with larger values of  $x$  or  $y$ . However, within the same table, from the HVS, we see that the quality of the stego image is not greatly changed when increasing the value of  $y$ . However, based on the experimental result, we see that the stego image quality is preserved as seen by the HVS even when the value of parameter  $y$  is 6. Moreover, the PSNRs remain about 28 dB when six LSBs in each edge pixel are replaced with the secret message.

**Conclusion 2:** The change in the edge-pixel does not seriously affect the quality of the stego image as seen by the HVS.




Furthermore, from these tables shown above, we can see that the quality of the stego image is still acceptable even when the values of the parameters  $x$ ,  $y$  and  $n$  are chosen as 4, 6 and 5, respectively. The experimental results show that with these values, we can achieve very high embedding payload. For example, in Table 5, the embedding payload is 0.87 bpp with  $x = 1$  and  $n = 2$ ; in Table 6, the embedding payload is 1.30 bpp with  $x = 2$  and  $n = 2$ ; in Table 9, the embedding payload is 2.86 bpp with  $x = 4$  and  $n = 3$ .

**Conclusion 3:** The proposed scheme preserves high embedding payload which reaches 2.86 bpp while the quality of the stego image remains good.

Aside from generating a high stego image quality and preserving high embedding payload, our scheme is robust against some steganalysis systems. Indeed, by taking this approach, we can obtain the following two results:







- (1) The secret message is inserted into various numbers of LSBs in different cover pixels. In other words, there are  $x$

**Table 4**The stego image quality generated by our scheme when  $x = 3$ ,  $n = 3$  and  $y = 3, 4, 5$ .






$x = 3, n = 3$			
$y$	3	4	5
Stego image	 PSNR = 38.8 dB	 PSNR = 37.5 dB	 PSNR = 34.4 dB









**Table 5**The performance of our scheme on  $128 \times 128$  Lena when  $x = 1$ ,  $n = 2$  and  $y \geq 1$ .

$x = 1, n = 2$						
$y$	1	2	3	4	5	6
Stego image						
Payload	PSNR = 51.1 dB 0.5 bpp	PSNR = 50.0 dB 0.575 bpp	PSNR = 47.1 dB 0.65 bpp	PSNR = 42.3 dB 0.73 bpp	PSNR = 36.9 dB 0.80 bpp	PSNR = 31.3 dB 0.87 bpp
Ratio	(8192 bits)	(9427 bits)	(10,662 bits)	(11,897 bits)	(13,132 bits)	(14,367 bits)







**Table 6**The performance of our scheme on  $128 \times 128$  Lena when  $x = 2$ ,  $n = 2$  and  $y \geq 2$ .

$x = 2, n = 2$					
$y$	2	3	4	5	6
Stego image					
Payload	PSNR = 46.3 dB 1 bpp	PSNR = 44.9 dB 1.07 bpp	PSNR = 41.6 dB 1.15 bpp	PSNR = 36.6 dB 1.22 bpp	PSNR = 31.1 dB 1.30 bpp
Ratio	(16,384 bits)	(17,619 bits)	(18,854 bits)	(20,089 bits)	(21,324 bits)

**Table 7**The performance of our scheme on  $128 \times 128$  Lena when  $x = 1$ ,  $n = 5$  and  $y \geq 1$ .



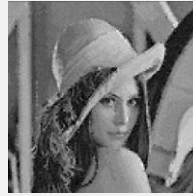
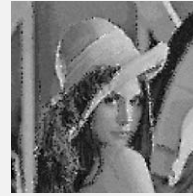

$x = 1, n = 5$						
$y$	1	2	3	4	5	6
Stego image						
Payload	PSNR = 37.9 dB 0.8 bpp	PSNR = 36.8 dB 0.92 bpp	PSNR = 36.5 dB 1.03 bpp	PSNR = 35.5 dB 1.15 bpp	PSNR = 33.1 dB 1.27 bpp	PSNR = 28.6 dB 1.39 bpp
Ratio	(13,104 bits)	(15,028 bits)	(16,952 bits)	(18,876 bits)	(20,800 bits)	(22,724 bits)

**Table 8**The performance of our scheme on  $128 \times 128$  Lena when  $x = 3$ ,  $n = 3$  and  $y \geq 3$ .

$x = 3, n = 3$						
$y$	3	4	5	6	7	8
Stego image						
Payload	PSNR = 38.8 dB 1.99 bpp	PSNR = 37.5 dB 2.1 bpp	PSNR = 34.4 dB 2.2 bpp	PSNR = 29.6 dB 2.3 bpp	PSNR = 24.1 dB 2.4 bpp	PSNR = 19.6 dB 2.5 bpp
Ratio	(32,766 bits)	(34,381 bits)	(35,996 bits)	(37,611 bits)	(39,226 bits)	(40,841 bits)

**Table 9**

The performance of our scheme on  $128 \times 128$  Lena when  $x = 4$ ,  $n = 3$  and  $y \geq 4$ .

$x = 4, n = 3$					
$y$	4	5	6	7	8
Stego image					
	PSNR = 33.5 dB	PSNR = 32.0 dB	PSNR = 28.6 dB	PSNR = 23.7 dB	PSNR = 18.4 dB
Payload	2.67 bpp	2.76 bpp	2.86 bpp	2.97 bpp	3.06 bpp
Ratio	(43,688 bits)	(45,303 bits)	(46,918 bits)	(48,533 bits)	(50,148 bits)

secret message bits replaced with  $x$  LSBs of each non-edge pixel and  $y$  secret message bits are inserted into each edge pixel.

- (2) The secret message is only embedded into a part of the cover image. In our proposed scheme, because the first pixel of each block acts as the index, there is no secret message bit embedded into this pixel. Therefore, the LSBs of every stego pixel do not contain the secret message.

Based on the above two properties, our scheme shows that it can resist steganalysis systems (Provos, 2001; Fridrich, & Goljan, 2002; Fridrich, Goljan, & Du, 2001) which are based on statistical analysis.

**Conclusion 4:** Our scheme shows can resist steganalysis systems which are based on statistical analysis.

## 5. Conclusion

In this paper, we have proposed a novel steganography scheme which is based on the LSB steganography mechanism and employs a hybrid edge detector which combines the fuzzy edge detector with the Canny edge detector. The hybrid edge detector assists the new scheme in generating a better quality stego image. Indeed, compared with other steganography schemes which generate the same PSNR of stego image, the new scheme produces higher quality stego images under the HVS due to the use of the hybrid edge detector. Experimental results confirm that the proposed scheme is successful in not only achieving a high embedding payload, but also in obtaining a stego image of satisfactory quality. Moreover, it can resist steganalysis systems which are based on statistical analysis.

## References

- Amarunnishad, T. M., Govindan, V. K., & Mathew, A. T. (2008). Improving BTC image compression using a fuzzy complement edge operator. *Signal Processing*, 88, 2989–2997.

- Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Techniques for data hiding. *IBM Systems Journal*, 35, 313–336.
- Böhme, R., & Westfeld, A. (2004). Statistical characterisation of MP3 encoders for steganalysis. In *Proceedings of the multimedia and security workshop* (Vol. 1, pp. 25–34).
- Farid, H. (2002). Detecting hidden messages using higher-order statistical models. *Proceedings of International Conference on Image Processing*, 2, 905–908.
- Fridrich, J. (2003). Quantitative steganalysis of digital images: Estimating the secret message length. *ACM Multimedia Systems Journal*, 9, 288–302.
- Fridrich, J., & Goljan, M. (2002). Practical steganalysis of digital images-state of the art. In *Proceedings of SPIE photonics imaging 2002, security and watermarking of multimedia contents* (Vol. 4675, pp. 1–13).
- Fridrich, J., Goljan, M., & Du, R. (2001). Detecting LSB steganography in color and gray-scale images. *IEEE Multimedia*, 8, 22–28.
- Jain, R., Kasturi, R., & Schunck, B. G. (1995). *Machine vision*. McGraw-Hill.
- Johnson, N., Duric, Z., & Jajodia, S. (2001). *Information hiding: Steganography and watermarking-attacks and countermeasures*. Boston, MA: Kluwer Academic Publishers.
- Kuo, Y. H., Lee, C. S., & Liu, C. C. (1997). A new fuzzy edge detection method for image enhancement. In *Proceedings of the 6th IEEE international conference on fuzzy systems* (Vol. 2, pp. 1069–1074).
- Nanning, Z. (1998). *Computer visualization and pattern recognition*. National Defense Industry Press.
- Provos, N. (2001). Defending against statistical steganalysis. In *Proceedings of the 10th conference on USENIX security symposium* (Vol. 10, p. 24).
- Russo, F. (1998). Edge detection in noisy images using fuzzy reasoning. *IEEE Transactions on Instrumentation and Measurement*, 47, 1102–1105.
- Sonka, M., Hlavac, V., & Boyle, R. (1999). *Image processing, analysis, and machine vision*. Thomson Brooks/Cole.
- Tao, C. W., Thompson, W. E., & Taur, J. S. (1993). A Fuzzy if-then approach to edge detection. In *Proceedings of the second IEEE international conference on fuzzy systems* (Vol. 2, pp. 1356–1360).
- Tizhoosh, H. R. (2002). Fast fuzzy edge detection. In *Proceedings of the annual meeting of the North American fuzzy information processing society* (pp. 239–242).
- Trucco, E., & Verri, A. (1998). *Introductory techniques for 3-D computer vision*. Prentice-Hall.
- Westfeld, A., & Pfitzmann, A. (1999). Attacks on steganographic systems. In *Proceedings of information hiding – The third international workshop, lecture notes in computer science* (Vol. 1768, pp. 61–76).