Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **5**, September **2017**

An Efficient Steganography Scheme Based on Edge Detection for High Payload

Chin-Chen Chang

Department of Information Engineering and Computer Science
Feng Chia University
Taichung 40724, Taiwan, R.O.C.
alan3c@gmail.com

Thai-Son Nguyen

Department of Information Technology
Tra Vinh University
Tra Vinh Province, Vietnam
thaison@tvu.edu.vn

Tzu-Yi Chien

Department of Information Engineering and Computer Science
National Chung Cheng University
Chiayi 62102, Taiwan, R.O.C.
m80429@yahoo.com.tw

Received October, 2015; revised May, 2016

Abstract. *Steganography is widely used to embed a massive amount of secret information. Several methods focus on taking advantage of a sharp area in which to embed secret information due to its ability to contain more secret bits undetected than a smooth area. Consequently, we propose a scheme that presents a major distinction between sharp areas and smooth areas by utilizing hybrid edge detection. To reduce the additional payload, error in bit extraction, and enhance high capacity, we adopt a truncated value of pixel to detect whether the pixel is defined as an edge pixel or not. As a result, we can not only reduce error in bit extraction, but also make the most of space for embedding secret data and maintain the good quality of the stego image. Compared to the scheme proposed by Ioannidou et al., our scheme enhances much higher capacity.*

Keywords: Steganography, Hybrid edge detection, Data hiding, Image process, LSB

1. **Introduction.** With the abrupt advancement of science and technology, a massive amount of digital data in the form of images, and videos are transmitted over public channel in the Internet. Thus malevolent users can easily tamper with the original content. For this reason, the protection of data that is delivered to a receiver by a data hider becomes a significant issue. Researchers have been striving increasingly to find solutions to the problem of cyber security during the past years. The existing solutions are widely known as watermarking and data hiding. Watermarking is used to protect the copyright of data content, and data hiding is applied by embedding secret data, from data hider to receiver. Hence, these tools play an important role in embedding a massive amount of secure data in a natural image without easy notice by others. One of the method of technology to embed a great quantity of secrete data is steganography.

Steganography is the art of creating covered writing to hide a large amount of secret information in meaningful media. It can be traced back to the method proposed by Petitcolas et al. [1]. Because of the characteristics of a sharp area which can accommodate more secret data without being detectable by the human's vision, many studies have exploited this option within image steganography [3, 4, 6, 7, 8, 9, 10]. In 2003, an optimal replacement model for data hiding by utilizing dynamic programming strategy was proposed by Chang et al. [10]. However, their methods offered limited capacity and unsatisfied quality of stego image. In order to improve these drawbacks, Wu et al. exploited the difference value in two consecutive pixels and combined it with the LSB replacement method [7]. This scheme calculated the difference of two consecutive pixels. Then, they applied LSB substitution and tried to embed fewer secret bits into the smooth areas of the cover image with small difference value while embedding more secret bits into the edge area with large difference value. In principle, the edge areas are usually less sensitive to changes than the smooth areas. Based on this concept, Chen et al. presented a high payload steganography method by using the hybrid edge detector in 2010 [3]. Since different numbers of secret bits are concealed into each pixel, their scheme can resist to statistical analysis, meaning a relatively high level of security. In 2014, Tseng and Leng [11] extended the scheme [3] by combination of Canny edge detector and fuzzy logic edge detector, instead of using fuzzy complement edge detector. In [11], the fuzzy logic algorithm is applied to produce more weak edge pixels in order to achieve higher payload. Additionally, their scheme achieved minimal distortion by considering all possible embedding lengths for each block and then a best one is chosen from these cases for embedding data. However, the computational complexity is considerably high since it takes every possibility into account.

Recently, EL-Emam [2] presents a steganography method that involves embedding secret data in a color image. This paper adopts the weaknesses of the human visual system (HVS) to embed secret data in three color channels: red channel, green channel and blue channel. The capacity of each channel is different. This method not only achieves high capacity but also attains greater security.

The edge area is ambiguous due to the color changes. The pixel value with neighboring pixels may change rapidly. For this reason, Chen et al. [3] presents another steganography to achieve high capacity by using hybrid edge detection. Aimed at supporting the points described above, for more security and higher capacity, Ioannidou et al. [4] took advantage of the works of EL-Emam [2] and Chen et al. [3] to propose another method of steganography for color image. However, it exploited 8 bits per pixel to determine whether the pixel is an edge pixel or not. This practice may lead to the acquisition of fallacious data when the receiver extracts the secret data due to the change in pixel value. Therefore, the embedding capacity is not satisfied while the image quality is always smaller than 47dB. To further improve the performance of the scheme [4], we propose new steganography scheme for color image. Note that, the advantage and strength of the proposed scheme is obtained by embedding more secret bits in edge areas with small distortion. Thus, to exploit as many edge pixels as possible in the cover image, the hybrid edge detection is utilized, which is the combination of a Sobel filter/ Laplacian filter and a fuzzy detector. Experimental results demonstrated that the proposed scheme is superior to the scheme [4] in both embedding capacity and image quality.

2. **Related works.** The least significant bit (LSB) technology is the blandest method to be applied in steganography. Steganography is a method to hide secret data in a meaningful image. The highest priority is to maintain the quality of the stego image. Consequently, steganography is used to embed secret information in the least significant bit of every pixel. However, Fridrich [5] asserts that this technique cannot resist statistical

attacks. A few years later, some works of research [3, 4, 6, 7, 8, 9] proposed that under the condition which isn't detectable, more information can be embedded in sharp areas than in smooth areas. These methods are described as follows.

2.1. EL-Emam's method.

Ioannidou et al.'s method [4] is based on the method of EL-Emam [2]. For this reason, we provide an introduction to these preliminary methods in this section. The technique focuses on the relationship per 8 bits of each channel, red channel, green channel, and blue channel, respectively. The main case whose relationship is defined by intensity of three colors is calculated by

$$MC = Int\left(\frac{ByteColor}{16}\right) + 1, \tag{1}$$

where the $ByteColor \in \{ByteRed, ByteGreen, ByteBlue\}$ corresponds to each color. Utilizing Eq. (1), the range of value of the main case lies between 1 and 16. Assume that MC^{color} = index and the MC of current pixel is C, X and Y are the rest of the colors in the same pixel.

To select a suitable color in which to embed information, EL-Emam's method [2] categorizes the relationship into 6 sub-cases.

$$SC = \left\{ \begin{array}{l} 1 \Leftrightarrow X, Y < C, \\ 2 \Leftrightarrow (X = C \& Y < C) \,|\, (X < C \& Y = C)\,, \\ 3 \Leftrightarrow X, Y = C, \\ 4 \Leftrightarrow (X > C \& Y < C) | (X < C \& Y > C), \\ 5 \Leftrightarrow (X > C \& Y = C) | (X = C \& Y > C), \\ 6 \Leftrightarrow X, Y > C, \end{array} \right\}. \tag{2}$$

Table 1. The relationship between the main case and 6 sub-cases, where CP represents the current pixel including red color, green color, or blue color

MC	The corresponding SC	Set description of SC
1	$\{SC_3, SC_5, SC_6\}$	$\{\forall SC \exists C = Sel_{color} \in CP \& RC_i \in CP \forall i = 1, 2 \exists C \leq RC_i\}$
2-15	$\{SC_1, SC_3, SC_5, SC_6\}$	$\{\forall SC \exists C = Sel_{color} \in CP \& RC_i \in CP \forall i = 1, 2 \exists C \leq RC_i \mid C \geq RC_i\}$
16	$\{SC_1, SC_3\}$	$\{\forall SC \exists C = Sel_{color} \in CP \& RC_i \in CP \forall i = 1, 2 \exists C \geq RC_i\}$

The 16 cases and the relationship of the 6 sub-cases are demonstrated below: The condition of selected color is held by

$$Sel_{color} = arg\ min_{\forall Color \in CP} \left\{MC_{Color}\right\}, \tag{3}$$

C_{mc} is selected by

$$C_{mc} = MC_{Sel_{Color}} \exists Color \in CP, \tag{4}$$

RC_1 and RC_2 are the rest colors except for the selected color C. Each sub-case has a different way of embedding secret data.

2.2. Chen et al.'s method.

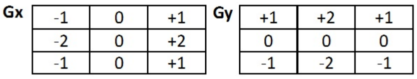
Chen et al. assumed a Canny edge detector and fuzzy detector to detect the edge area. The fuzzy detector is described in the proposed scheme. Hence, we introduce the Canny edge detector in this section. The reasons this method adopts Canny to detect edges are as follows: (1) No important edges should be missed, and there should be no false edges, while the error rate of detection should be kept low; (2) The distance between the actual and located positions of the edge should be minimal; and (3) There is only one response to a single edge. The characteristic of the Canny detector is classifying pixel to edge pixel if the gradient magnitude of the pixel is greater than the neighboring pixel in the direction of maximum intensity change. Application of the Gaussian mask in the image is the first step of implementation. Secondly, one uses a Sobel filter to determine the gradient direction and gradient magnitude of each pixel.

The Sobel filter is described in detail in the next section. Then, we examine the gradient magnitude of the pixel regarding whether it is greater than the gradient magnitude of the neighboring pixel. If the gradient magnitude of the pixel is greater than the others, it belongs to the edge pixel region. Finally, the weakness edge is removable by the threshold. The hybrid edge detection is combined with the result of the Canny detector with the fuzzy detector.

3. **Proposed scheme.** The technique presents majors in exploiting edge regions to embed more secret data to enhance the capacity in this paper. Ioannidou et al. [4] computed the actual edges to embed one bit in three channels of every pixel in order to obtain a higher capacity by using the hybrid edge detector. That is, it uses a Sobel filter combined with a fuzzy detector and Laplacian filter combined with a fuzzy detector to detect the actual edges in an image in order to find massive amounts of edges. The fuzzy detector is based on fuzzy logic. However, an inaccuracy may exist when the secret message is embedded in a pixel. Hence, the auxiliary file that records which pixels belong to an edge area must be created. For this reason, we present a method that enhances the capacity and reduction of inaccuracies, simultaneously. We use most significant bits (MSB) to examine whether the pixel belongs to a sharp area. Also, we use the hybrid edge detector in our method. A description of our methods to detect all edges is shown in the next section.

Preliminary

The Sobel filter approximates the derivatives in a specific direction by two 3×3 masks. Fig. 1 shows the 3×3 convolution masks, where Gx is used for horizon direction and Gy is used for vertical direction.



Gx

-1	0	+1
-2	0	+2
-1	0	+1

Gy

+1	+2	+1
0	0	0
-1	-2	-1

Figure 1. Two masks which is used to calculate the derivatives in Sobel filter

The gradient is computed by this formula for every pixel.

$$G = \sqrt{Gx^2 + Gy^2} \tag{5}$$

The gradient of direction of every pixel is computed by

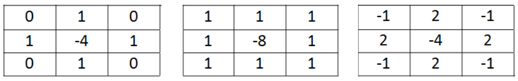
$$\theta = arctan(Gx/Gy). \tag{6}$$

The Laplacian filter is a second-order differential operator of an image. The formula of the Laplacian operator in two dimensions is given below:

$$\Delta f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}, \tag{7}$$

f indicates the identity of the pixel.

The Laplacian filter contains three masks to compute the second derivatives by 3×3 masks. The three masks are given below:



0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

Figure 2. The three masks is utilized to calculate the second derivatives in Laplacian filter

Assume that image I is a gray-scale image and the size of image I is $W \times H$. The range γ of all pixels in image I lie between 0 and 255. That is to say, $\gamma \in [0, 255]$. We transform γ to μ_{mn} using the formula given below:

$$\mu_{mn} = \frac{x_{mu}}{x}, m \in [1, W], n \in [1, H], \tag{8}$$

x is the minimum value of $m \times n$.

Since we obtain μ_{mn} of the whole image, we can unify it as an array F. The form of array F is given below:

$$F = \bigcup_{m=1}^{W} \bigcup_{n=1}^{H} \mu_{mn}. \tag{9}$$

The simplest method to determine whether the pixel belongs to an edge area by using a fuzzy detector is a proper membership function

$$\overline{\mu}_{mn} = min\left(1, \frac{\imath}{w} \sum_{i} \sum_{j} min(\mu_{ij}, 1-\mu_{ij})^{\rho}\right)^{\frac{1}{\rho}}, \tag{10}$$

$$\mu_{mn} = \frac{\{max(x_{ij}) - min(x_{ij}) : i, j \in [1, w]\}}{x}, \tag{11}$$

where w is a $w \times w$ spatial window surrounding each pixel. The values of $\imath$ and w are defined by the user. However, the appropriate values of $\imath$ and w are 9 and 3, respectively.

3.1. **Embedding algorithm phase.**

Data embedding algorithm

Input: : Cover image I

Output: : Two stego image I' and I''

Step 1: : Select an image as a cover-image.

Step 2: : Each pixel of cover image $I(R, G, B)$ is erased in the last n LSB_s using Eq. (12), $n = 4$,

$$\begin{aligned} R' &= (R/2^n) * 2^n, \\ G' &= (G/2^n) * 2^n, \\ B' &= (B/2^n) * 2^n, \end{aligned} \tag{12}$$

where R', G' and B' are the values for the red channel, green channel, and blue channel of the truncated image, respectively.

Step 3: : After transforming the original image to the truncated image, user would obtain a grayscale image, which is a calculated truncated image,

$$I = 0.299 * R' + 0.587 * G' + 0.114 * B', \tag{13}$$

where R' is the red channel, G' is the green channel, and B' is the blue channel of the truncated image.

Step 4: : Detect the edges of the grayscale image I by using Eqs. (5)-(6) and Fig. 1. That is, detect the edges using Sobel filter. The user acquires the edge image E, which is detected by the Sobel filter. Also, the user would obtain the edge image by using Eq. (7) and Fig. 2. That is to say, the edge image E', which is detected by the Laplacian filter, is created.

Step 5: : Detect the edges of the grayscale image I using fuzzy detector (Eqs. (8)-(11)). The third edge image E'' would be obtained. Use the OR operator to combine edge image E and edge image E'', and use OR operator to combine edge image E' and edge image E''. The two edge images E''' (Sobel filter OR fuzzy detector) and E'''' (Laplacian filter OR fuzzy detector) are created with the hybrid edge detector.

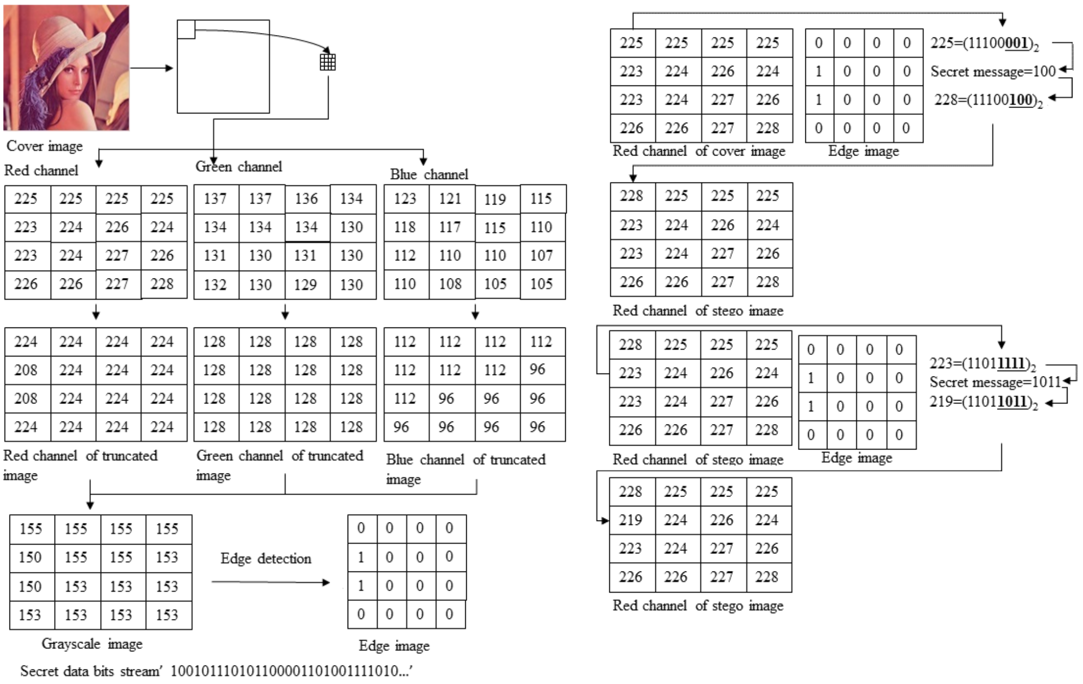
Step 6: : Examine whether the pixel of the cover image I belongs to the edge area by E''' or not.

Step 6.1: : If $E'''(I_{ij}) = 1$, embed the secret data bits stream in $I_{ij}(R, G, B)$ with n LSB_S, $n = 4, i, j \in [1, 128]$. That is to say, if a pixel belongs to the edge area, the user embeds the secret message to red channel, green channel, and blue channel of the pixel with 4 LSB_S.

Step 6.2: : Otherwise, $E'''(P_{ij}) = 0$, embed secret data bits stream in $P_{ij}(R, G, B)$ with n $LSB_S, n = 3, i, j \in [1, 128]$. That is to say, if the pixel belongs to a smooth area, the user embeds the secret message to the red channel, green channel, and blue channel of pixel with 3 LSB_S.

Step 7: : Repeat Step 6 through Step 6.2 until the stego image I' is created. Also, the method of stego image I'' is the same as the stego image I'.

Step 8: : Output the stego image I' and stego image I''.


Cover image
Red channel
Green channel
Blue channel
Red channel of truncated image
Green channel of truncated image
Blue channel of truncated image
Grayscale image
Edge detection
Edge image
Secret data bits stream' 1001011101011000011010011111010...'
Red channel of cover image
Edge image
225=(11100**001**)$_2$
Secret message=100
228=(11100**100**)$_2$
Red channel of stego image
223=(11011**111**)$_2$
Secret message=1011
219=(1101**1011**)$_2$
Edge image
Red channel of stego image
Red channel of stego image

Red channel

225	225	225	225
223	224	226	224
223	224	227	226
226	226	227	228

Green channel

137	137	136	134
134	134	134	130
131	130	131	130
132	130	129	130

Blue channel

123	121	119	115
118	117	115	110
112	110	110	107
110	108	105	105

Red channel of truncated image

224	224	224	224
208	224	224	224
208	224	224	224
224	224	224	224

Green channel of truncated image

128	128	128	128
128	128	128	128
128	128	128	128
128	128	128	128

Blue channel of truncated image

112	112	112	112
112	112	112	96
112	96	96	96
96	96	96	96

Grayscale image

155	155	155	155
150	155	155	153
150	153	153	153
153	153	153	153

Edge image

0	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0

Red channel of cover image

225	225	225	225
223	224	226	224
223	224	227	226
226	226	227	228

Edge image

0	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0

Red channel of stego image

228	225	225	225
223	224	226	224
223	224	227	226
226	226	227	228

Red channel of stego image

228	225	225	225
223	224	226	224
223	224	227	226
226	226	227	228

Edge image

0	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0

Red channel of stego image

228	225	225	225
219	224	226	224
223	224	227	226
226	226	227	228

FIGURE 3. The illustration indicates embedding secret data phase

3.2. **Extraction phase.** After the data embedding phase has been processed completely, the user would obtain two stego images. To enhance the security of the stego image, we encrypt the two stego images I' and I'' with Triple-DES algorithm. This application asks the receiver to insert the secret key, which is confirmed. Incidentally, the information table which contains the width and height of two stego images is compressed with stego images to ensure correct extraction. We can then send it to the receiver after the encryption processing completely.

Data extraction algorithm

Input: : Code streams CS in binary representation

Output: : Secret data bits streams SC

Step 1: : Extract the 16 front bits of the code stream CS to determine the width and height of the stego image.

Step 2: : Reconstruct the two stego images with the width, height, and the remainder code streams.

Step 3: : Erase the last n LSB_S of each pixel of the stego image $I'(R, G, B)$ by using Eq. (12), $n = 4$. The user obtains a truncated image, which is created by stego image I'.

Step 4: : Transform the truncated image into a grayscale image using Eq. (13).

Step 5: : Detect the edges by Eqs. (5)-(6) and Fig. 1. That is to say, detect the edges of the truncated image using the Sobel filter. The user then obtains edge image E.

Step 6: : Detect the edge image E with Eqs. (8)-(11). In other words, the user would detect the edge image E with a fuzzy detector. Then the user would obtain the other image E''.

Step 7: : Combine edge image E with E'' by using OR operator. After combining the edge image E with E'', the edge image E''' (Sobel OR fuzzy) based on hybrid edge detector is created.

Step 8: : Examine whether the pixel of stego image I' belongs to edge area by E''' or not.

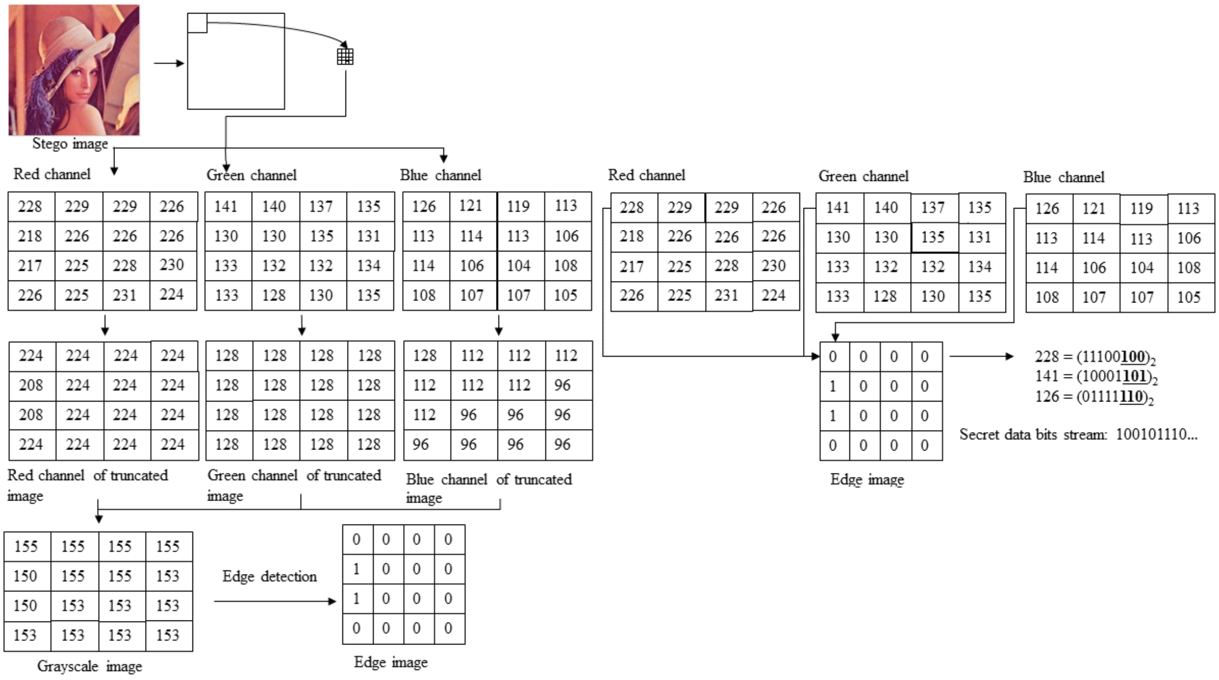
Step 8.1: : If $E'''(P_{ij}) = 1$, extract the secret data bits by using Eq. (14), $n = 4$. That is to say, the user can extract 4 LSB_S each from the red channel, green channel, and blue channel.

$$\begin{array}{l} R' \ mod \ 2^n, \\ G' \ mod \ 2^n, \\ B' \ mod \ 2^n. \end{array} \tag{14}$$

Step 8.2: : If $E'''(I_{ij}) = 0$, extract the secret data bits by using Eq. (14), $n = 3$. In other words, the user can extract 3 LSB_S each from the red channel, green channel and blue channel.

Step 9: : Repeat Step 8 through Step 8.2 until all secret bits are extracted completely. Also, the extraction process of stego image I'' is the same as I'.

Step 10: : Output the secret data bits stream.



Red channel

228	229	229	226
218	226	226	226
217	225	228	230
226	225	231	224

Green channel

141	140	137	135
130	130	135	131
133	132	132	134
133	128	130	135

Blue channel

126	121	119	113
113	114	113	106
114	106	104	108
108	107	107	105

Red channel of truncated image

224	224	224	224
208	224	224	224
208	224	224	224
224	224	224	224

Green channel of truncated image

128	128	128	128
128	128	128	128
128	128	128	128
128	128	128	128

Blue channel of truncated image

128	112	112	112
112	112	112	96
112	96	96	96
96	96	96	96

Grayscale image

155	155	155	155
150	155	155	153
150	153	153	153
153	153	153	153

Edge image

0	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0

Red channel

228	229	229	226
218	226	226	226
217	225	228	230
226	225	231	224

Green channel

141	140	137	135
130	130	135	131
133	132	132	134
133	128	130	135

Blue channel

126	121	119	113
113	114	113	106
114	106	104	108
108	107	107	105

Edge image

0	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0

$228 = (11100\underline{\mathbf{100}})_2$
$141 = (10001\underline{\mathbf{101}})_2$
$126 = (01111\underline{\mathbf{110}})_2$

Secret data bits stream: 100101110...

FIGURE 4. The illustration indicates embedding secret data phase

4. **Experimental results.**

In this section, we show the results from our entire scheme. In order to evaluate the performance and compare the results with those of Chen et al. [3] and Ioannidou et al. [4], we adopt seven color images to conduct the experiments ;i.e., Lena, Building, Pepper, Tiffany, Baboon, Jet, and Scene. The size of these images we applied is 128 × 128. Fig. 5 shows seven test images.

FIGURE 5. Seven 128 × 128 color images we used

In order to make comparisons with Chen et al. [3], we divide the capacity (Bits) by 3 because the grayscale image is applied by Chen et al. [3] and calculate the PSNR value by Eq. (15).

The peak signal to noise ratio (PSNR) is statistically measured to apply in-image steganography to indicate the image quality PSNR that differs between the cover image and the stego image.

$$PSNR = 10\ log_{10}\left(\frac{255^2}{MSE}\right). \tag{15}$$

The mean square error (MSE) is calculated by

$$MSE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}(C_{i,j} - S_{i,j})^2. \tag{16}$$

H and W represent the height and width of cover image and stego image and $C_{i,j}$ and $S_{i,j}$ indicate the pixel values of cover image and stego image, respectively. The quality of the stego image indicates that it is difficult to recognize if the PSNR value is better. That is to say, the stego image may not be easily found by a malicious user.

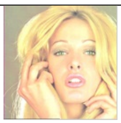
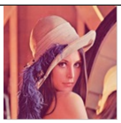
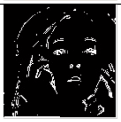
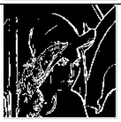
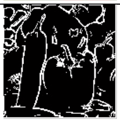
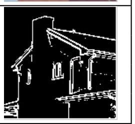
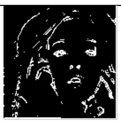
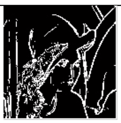
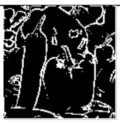
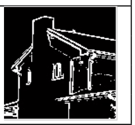
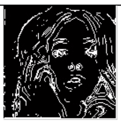
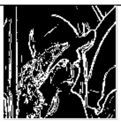
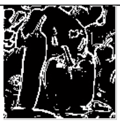
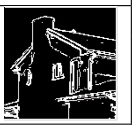
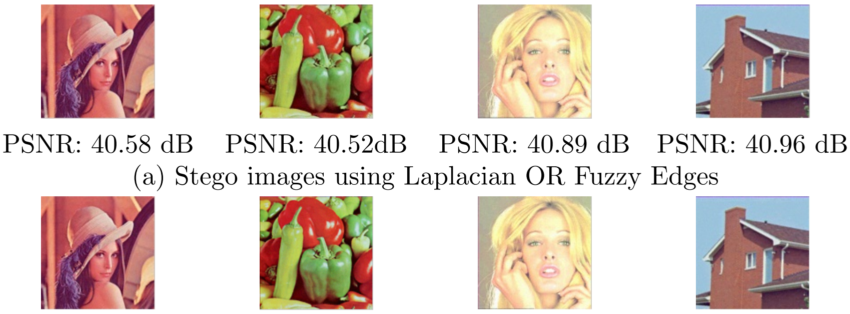
original image				
Edge image of original image				
The number of edge pixels	1199	2574	2688	2110
Edge image of truncated image using Sobel OR Fuzzy				
The number of edge pixels	1471	2750	2785	2270
Edge image of truncated image using Laplacian OR Fuzzy				
The number of edge pixels	2445	3375	3333	2686

FIGURE 6. Experimental results which are generated by our proposed scheme

Fig. 6 indicates the results of hybrid edge detection by Tiffany, Lena, Pepper, and Building. Obviously, the edge regions are clear when we use a hybrid edge detector to detect the edge from Fig. 6. We consider the Lena image as an example. The number of edge pixels of Lena is 2,574 bits. In the same image, the number of edge pixels using a Laplacian filter combined with a fuzzy detector is 3,378 bits. Clearly, the performance of



PSNR: 40.58 dB PSNR: 40.52dB PSNR: 40.89 dB PSNR: 40.96 dB
(a) Stego images using Laplacian OR Fuzzy Edges

PSNR: 40.83 dB PSNR: 40.81 dB PSNR: 41.51 dB PSNR: 41.22 dB
(b)Stego images using Sobel OR Fuzzy Edges

Figure 7. Stego images which are generated by our proposed scheme

edge detection by using a hybrid edge detector is much better than using a single edge detector. As we know, the capacity is higher.

Fig. 7 indicates the quality of a stego image which is applied using hybrid edge detection to embed secret data. Fig. 7(a) shows the result of the quality of the stego image, which is generated by combining the Laplacian filter with a fuzzy detector. Also, Fig. 7(b) shows that the quality of the stego image is generated by combining the Sobel filter with a fuzzy detector. The PSNR value is much higher than 40 dB. Therefore, the quality of stego image is quite better. That is to say, it is almost indistinguishable between the corresponding cover image and the stego image.

Table 2. Comparison with Chen et al.'s result [3] when $x = 3$ and $x = 4$ on 'Lena' with image size 128 × 128

	Proposed scheme	Chen et al.'s scheme
	$x = 3$ and $y = 4$	$n = 3, x = 3$ and $y = 4$
PSNR (dB)	40.50	37.5
Payload (bpp)	3.21	2.1
	$x = 4$ and $y = 5$	$n = 3, x = 4$ and $y = 5$
PSNR (dB)	34.32	32.0
Payload (bpp)	4.21	2.76

Table 2 denotes that the performance of our scheme is better than Chen et al.'s scheme [3]. We experiment on the same condition when possible. Chen et al.'s scheme divided the cover image into several blocks. The size of each block denoted by n is 3×3, the non-edge area which is denoted by x is embedded 3 bits, and the edge area which is denoted by y is embedded 4 bits. As mentioned previously, we divide the payload by 3 because the grayscale image is applied in Chen et al.'s scheme. Therefore, the payload is 3.21 bpp and 4.21 bpp. However, these results are still better than those of Chen et al. (2.1 bpp and 2.76 bpp).

For comparison with Ioannidou et al.'s method [4], we adopt the same condition in our experiment. That is to say, we increase the range of secret data, which is embedded in the edge pixel from 1 bit to 6 bits in every RGB channel on Lena and Building. The range of secret data which is embedded in non-edge pixel is from 0 to 5 bits using the Sobel filter combined with the fuzzy detector and the Laplacian filter combined with the fuzzy detector. The results are demonstrated in Fig. 8. As we can see, the quality of the stego image is maintained as we embed the secret data from 1 bit to 4 bits. As EL-Emam [2] mentioned previously, the quality of the stego image is reduced significantly if the range lies between 5 bits and 6 bits. Consequently, we control the additional bits from 1 bit to 4 bits to examine the performance of the other images, Lena, Baboon, Tiffany, Pepper,

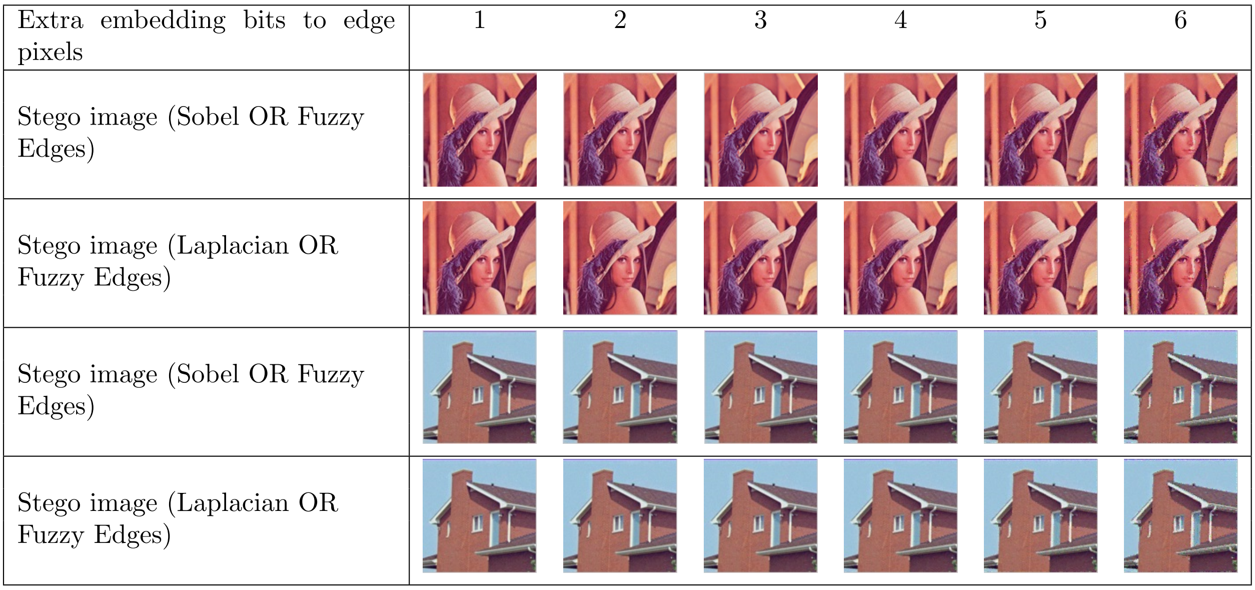

Extra embedding bits to edge pixels
1
2
3
4
5
6
Stego image (Sobel OR Fuzzy Edges)
Stego image (Laplacian OR Fuzzy Edges)
Stego image (Sobel OR Fuzzy Edges)
Stego image (Laplacian OR Fuzzy Edges)

Figure 8. Stego image results for embedding secret data in edge pixels

Jet and Scene. To examine the performance of our scheme rigorously, we experiment not only on Lena but also on Baboon, Tiffany, Pepper, Jet and Scene. The conditions of experiments are similar to Chen et al.'s experimental results [3]. That is to say, we control the range of secret data which is embedded in non-edge area lies between 1 bit and 4 bits and the number of edge area lies between 2 bits and 5 bits. Fig. 9 demonstrates the results. As we can see, the PSNR value exceeds 30 dB in Fig. 9. The stego image and the cover image are almost indistinguishable by human eyes. Therefore, as we know, the quality of the stego image is favorable and our scheme is suitable for embedding secret data. Comparison of results for the proposed method and the method in (Ioannidou et al. and Chen et al. when $x = 1$ and $y = 3$)

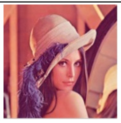
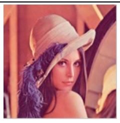
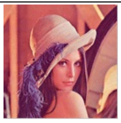
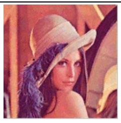
Table 3. The comparison of results for Chen et al.'s method, Ioannidou et al.'s method and the proposed method for Lena

	Chen et al.'s method	Ioannidou et al.'s method		
		Laplacian OR Fuzzy edges		
		RNG with Steps 1, 2, 3	RNG with Steps 1,2	NO RNG used
Capacity	10,662 (bits)	15,295 (bits)	20,596 (bits)	30,987 (bits)
	0.65 (bpp)	0.93 (bpp)	1.26 (bpp)	1.89 (bpp)
PSNR	47.1	46.88	46.88	45.12

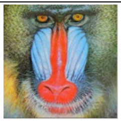
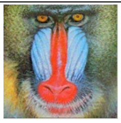
	Ioannidou et al.'s method			Proposed scheme
	Sobel OR Fuzzy edges			Laplacian OR Fuzzy edges
	RNG with Steps 1, 2, 3	RNG with Steps 1, 2	NO RNG used	RNG with Steps 1, 2, 3
Capacity	15,213 (bits)	20,490 (bits)	30,811 (bits)	15,171 (bits)
	0.93 (bpp)	0.93 (bpp)	1.88 (bpp)	0.93 (bpp)
PSNR	45.19	46.88	44.45	59.00

	Proposed scheme		Proposed scheme	
	Laplacian OR Fuzzy edges		Sobel OR Fuzzy edges	
	RNG with Steps 1, 2	NO RNG used	RNG with Steps 1, 2, 3	RNG with Steps 1, 2
Capacity	19,602 (bits)	59,151 (bits)	14,727 (bits)	18,858 (bits)
	1.20 (bpp)	3.61 (bpp)	0.90 (bpp)	1.15 (bpp)
PSNR	58.10	53.34	59.33	58.54

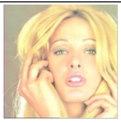
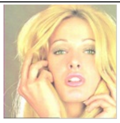
In our proposed scheme, every pixel is used to embed a secret message. The red channel, green channel, and blue channel of every pixel are embedded with 3 bits. That is to say,

$x = 1$	$x = 2$	$x = 3$	$x = 4$
			

PSNR = 53.26 dB
Payload = 3.62 bpp
	PSNR = 46.75 dB
	Payload = 6.62 bpp
		PSNR = 40.56 dB
		Payload = 9.62 bpp
			PSNR = 34.32 dB
			Payload = 12.62 bpp

PSNR = 52.81 dB
Payload = 3.77 bpp
	PSNR = 46.40 dB
	Payload = 6.77 bpp
		PSNR = 40.11 dB
		Payload = 9.77 bpp
			PSNR = 34.02 dB
			Payload = 12.77 bpp

PSNR = 53.88 dB
Payload = 3.45 bpp
	PSNR = 47.20 dB
	Payload = 6.45 bpp
		PSNR = 40.94 dB
		Payload = 9.45 bpp
			PSNR = 34.57 dB
			Payload = 12.45 bpp

PSNR = 53.34 dB
Payload = 3.61 bpp
	PSNR = 46.75 dB
	Payload = 6.61 bpp
		PSNR = 40.53 dB dB
		Payload = 9.61 bpp
			PSNR = 34.47 dB
			Payload = 12.61 bpp

PSNR = 50.08 dB
Payload = 3.69 bpp
	PSNR = 46.51 dB
	Payload = 6.69 bpp
		PSNR = 40.33 dB
		Payload = 9.69 bpp
			PSNR = 34.17
			Payload = 12.69 bpp

PSNR = 52.67 dB
Payload = 3.84 bpp
	PSNR = 46.16 dB
	Payload = 6.84 bpp
		PSNR = 40.0 dB
		Payload = 9.84 bpp
			PSNR = 33.83 dB
			Payload = 12.84 bpp

FIGURE 9. The experimental results whose edge pixels increased by 2 bits to 5 bits and non edge pixels increased by 1 bit to 4 bit on six images 'Lena', 'Baboon', 'Tiffany', 'Pepper', 'Jet' and 'Scene' sized 128 × 128

	Proposed scheme	Proposed scheme		
	Sobel OR Fuzzy edges	Laplacian OR Fuzzy edges		
	NO RNG used	RNG with Steps 1, 2, 3	RNG with Steps 1, 2	NO RNG used
Capacity	57,402 (bits)	27,459 (bits)	35,703 (bits)	108,429 (bits)
	3.50 (bpp)	1.68 (bpp)	2.18 (bpp)	6.62 (bpp)
PSNR	53.67	52.42	51.48	46.72

	Proposed scheme		
	Sobel OR Fuzzy edges		
	RNG with Steps 1, 2, 3	RNG with Steps 1, 2	NO RNG used
Capacity	27,015 (bits)	34,698 (bits)	106,554 (bits)
	1.65 (bpp)	2.14 (bpp)	6.50 (bpp)
PSNR	52.70	51.83	47.01

	Proposed scheme		
	Laplacian OR Fuzzy edges		
	RNG with Steps 1, 2, 3	RNG with Steps 1, 2	NO RNG used
Capacity	39,747 (bits)	51,858 (bits)	157,581 (bits)
	2.43 (bpp)	3.17 (bpp)	9.62 (bpp)
PSNR	46.32	45.31	40.52

	Proposed scheme		
	Sobel OR Fuzzy edges		
	RNG with Steps 1, 2, 3	RNG with Steps 1, 2	NO RNG used
Capacity	39,303 (bits)	51,114 (bits)	155,706 (bits)
	2.40 (bpp)	3.12 (bpp)	9.50 (bpp)
PSNR	46.58	45.60	40.91

every pixel is embedded with 9 bits at least. Therefore, in Table 3, our scheme can achieve 9.50 (bpp). However, there exist smooth areas and sharp areas in an image. The major technique in our proposed scheme makes a distinction between smooth areas and sharp areas. If the pixel is determined to belong to an edge area, one bit is embedded in every channel additionally. So, the pixel which belongs to the edge area would be embedded with 12 bits.

5. Conclusions.

Our scheme offers advantages in two aspects. Firstly, it utilizes the MSB bits to detect the edge region. As a result, it wouldn't acquire error bits when the receiver extracts the secret data, which is embedded in the stego image. Second, it doesn't need the auxiliary files to record which pixel belongs to the edge area anymore. That is to say, the payload would decrease rapidly. The data hider can embed even more secret bits in the cover image. In other words, our scheme has an advantage of higher capacity. Simultaneously, the quality of the stego images still reaches above 40 dB when we achieve an even higher capacity. That is to say, it is still secure when the stego images are transmitted over the public channel or the Internet.

However, on the other hand, the secret data is much easier to be extracted than EL-Emam's scheme [2] and Ioannidou et al.'s scheme [4], since the number of secret data which is embedded in the stego image is the same. Inevitably, the security may decrease for this reason. Also, today, the method has exhibited a tendency toward reversible data hiding. More and more researches have focused on it. Hence, the security and reversible data hiding (RDH) should be a future area of study.

REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, Information hiding a survey, *Proceedings of the IEEE*, vol. 87, no. 7, pp. 106 2-1078, 1999.

[2] N. N. EL-Emam, Hiding a large amount of data with high secret using steganography, *Journal of Computer Science*, vol. 3, no. 4, pp. 223-232, 2007.

[3] W. J. Chen, C. C. Chang and T. H. N. Te, High payload steganography mechanism using hybrid edge detector, *Expert System with Applications*, vol. 37, no. 4, pp. 3292-3301, 2010.

[4] A. Ioannidou, S. T. Halkidis and G. Stephanides, A novel technique for image steganography based on a high payload method and edge detection, *Expert System with Applications*, vol. 39, no. 14, pp. 11517-11524, 2012.

[5] J. Fridrich, M. Goljan and R. Du, Detecting LSB steganography in color and gray-scale images, *IEEE Multimedia*, vol. 8, no. 4, pp. 22-28, 2001.

[6] C. C. Chang and H. W. Tseng, A steganographic method for digital images using side match, *Pattern Recognition Letters*, vol. 25, no. 12, pp. 1431-1437, 2004.

[7] D. C. Wu and W. H. Tsai, A steganographic method for images by pixel-value differencing, *Pattern Recognition Letters*, vol. 25, no. 9-10, pp. 1613-1626, 2003.

[8] C. M. Wang, N. I. Wu, C. S. Tsai and M. S. Hwang, A high quality steganographic method with pixel-value differencing and modulus function, *Journal of Systems and Software*, vol. 81, no. 1, pp. 150-158, 2008.

[9] C. C. Thien and J. C. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, *Pattern Recognition*, vol. 36, no. 12, pp. 2875-2881, 2003.

[10] C. C. Chang, J. Y. Hsiao and C. S. Chan, Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy, *Pattern Recognition*, vol. 36, no. 7, pp. 1583-1595, 2003.

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