An Adaptive Image Steganographic Model Based on Minimum-Error LSB Replacement

Yeuan-Kuen Lee* (李遠坤) and Ling-Hwei Chen (陳玲慧)

Department of Computer and Information Science
National Chiao Tung University, Hsinchu, 30050, Taiwan, R.O.C.
*Email: yklee@debut.cis.nctu.edu.tw, lhchen@cc.nctu.edu.tw

Abstract

Steganography is an ancient art of conveying message in a secret way that only the receiver knows the existence of message. Steganalysis is the art of detecting the message's existence and blockading the covert communication. The least-significant bit (LSB) insertion method, which uses fixed k LSBs in each pixel to embed secret message, is the most common and easy one to hide message in an image. However, it is easy to reveal a stego-image produced by the LSB insertion method. In this paper, we will first lead a bit-plane steganalysis on such stego-images. Then, an adaptive steganographic model will be proposed to take off the restriction of fixed embedding size in each pixel. The model will reduce the embedding error and provide higher embedding capacity. Moreover, to detect the message's existence will be very hard for those stego-images produced using the proposed model.
Keywords: steganography, steganalysis, covert communication, cover-image, stego-image.

1. Introduction

With the development of Internet technologies, digital media can be transmitted conveniently over the networks. Therefore, How to protect secret messages during transmission becomes an important issue. Using the classic cryptography only, the encrypted message becomes clutter data that can not pass the checkpoint on the network. Steganography provides another layer of protection on the secret message, which will be embedded in another media such that the transmitted data will be meaningful and innocuous to everyone. Compared with cryptography techniques attempting to conceal the content of messages, steganography conceals the existence of the secret messages.

The steganography terminology used in this paper is identical with that introduced in [1]. A steganographic method will embed messages in a cover-image and create a stego-image. Those stego-images produced using the LSB insertion method, which will be explained later, are called LSB-embedded images. Some steganographic methods use a stego-key to embed messages for achieving rudimentary security. Two of the most important requirements for a steganographic method are imperceptibility and statistical undetectability.

Steganalysis is the art of detecting the message's existence and blockading the covert communication. The goal of steganalysis tools is to automatically detect the existence of secret messages in media. In [2], Johnson and Jajodia pointed out that several steganographic tools [3] will generate some dependent patterns in the stego-images. By extracting these specific patterns, a steganalyst can break the steganographic system. In order to define attack techniques used for steganalysis, five levels of attacks on steganographic system are introduced in [2], i.e. stego-only, known cover, known message, chosen stego and chosen message attack.

Fig. 1 shows the block diagram of a secure steganographic system. Input messages can be images, texts, video, etc. Since embedding more bits of messages will introduce more degradation, a compression step is needed before embedding. Because some content-dependent patterns in the original message may reveal the existence of message, concealing these patterns before embedding is necessary. The encryption step is provided to achieve this task. A suitable encryption algorithm [4-5] will be applied on the messages, it will not only conceal the meaning of the messages but also the content-dependent properties of the messages. Note that a compression technique can also randomize these patterns, but a steganalyst may try to decompress the messages before analyzing and extracting these patterns by exhaustive search. This can be done because of the number of compression standards is small. Note also that the compression step must be performed before the encryption step for the benefit of entropy coding.

Just as other image processing methods [6], there are two kinds of steganographic techniques: frequency- and spatial-domain based. For frequency-domain based method, images are first transformed to frequency domain, and then messages are embedded in the coefficients of transform. Jpeg-Jstego [7] is a frequency-domain based steganography tool, which embeds 1-bit message in one DCT coefficient produced

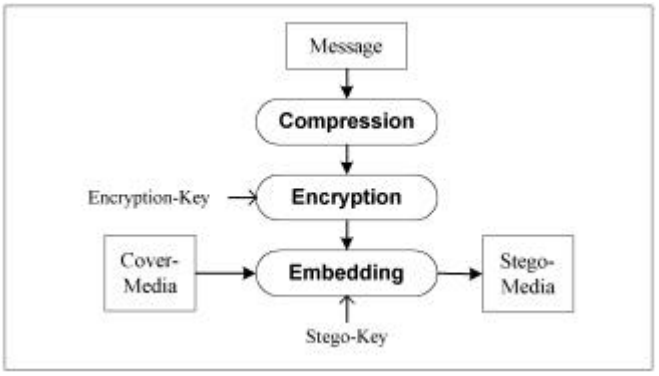


Fig. 1 The block diagram of a secure steganographic system.

by JPEG. JPEG standard use an integer to approach a DCT coefficient which is a real number, Jpeg-Jstego hides one-bit message in the coefficient by modulating the rounding choices either up or down to restrict the maximum error within 1. On spatial domain, the most common and simplest steganographic method [8] is the least significant bits (LSB) insertion method, it embeds message in k least significant bits with k fixed. The LSB insertion method is easy to be attacked, i.e. it is possible to judge if an image is a LSB-embedded image.

In this paper, we will first lead a steganalysis issue on the LSB insertion method, the issue will be described in the next section. Then, a concept to reduce error between the LSB-embedded image and the cover-image will be proposed and presented in Section 3. After that, an adaptive steganographic method will be presented in Section 4 for addressing the problems of embedding fixed size of message in each pixel of the cover-image. Finally, some experimental results are shown and conclusions are drawn in Section 5 and 6.

2. Bit-plane Steganalysis on LSB-embedded images

In general, we use 8 bits to store the intensity of each pixel on a grayscale image. The plane formed by the same bit of each pixel in a grayscale image is called a bit-plane. Fig. 2 shows the 8 corresponding bit-planes of a given grayscale image. From Fig. 2, we can see two phenomena. First, if a random texture appears on more significant bit-planes, the same position in less significant bit-planes will also appear a random texture. So, the area of "random texture" in each less significant bit-plane is bigger than that in each more significant bit-plane. If message is embedded in a certain bit-plane, not the least significant one, the phenomenon will vanish. Second, the texture random degree on a area changes gradually from the most significant bit-plane to the least significant bit-plane. However, when the k least significant bits in each pixel of a cover image are be used to embed message, these k least significant bit-planes will be bestrewn with random texture, and the texture random degree change from bit-plane k to bit-plane k-1 is abrupt. (See Figs. 3(e) and 3(f).) This means that the second phenomenon will also vanish. In the following, a transition density function will be proposed to measure these two phenomena.

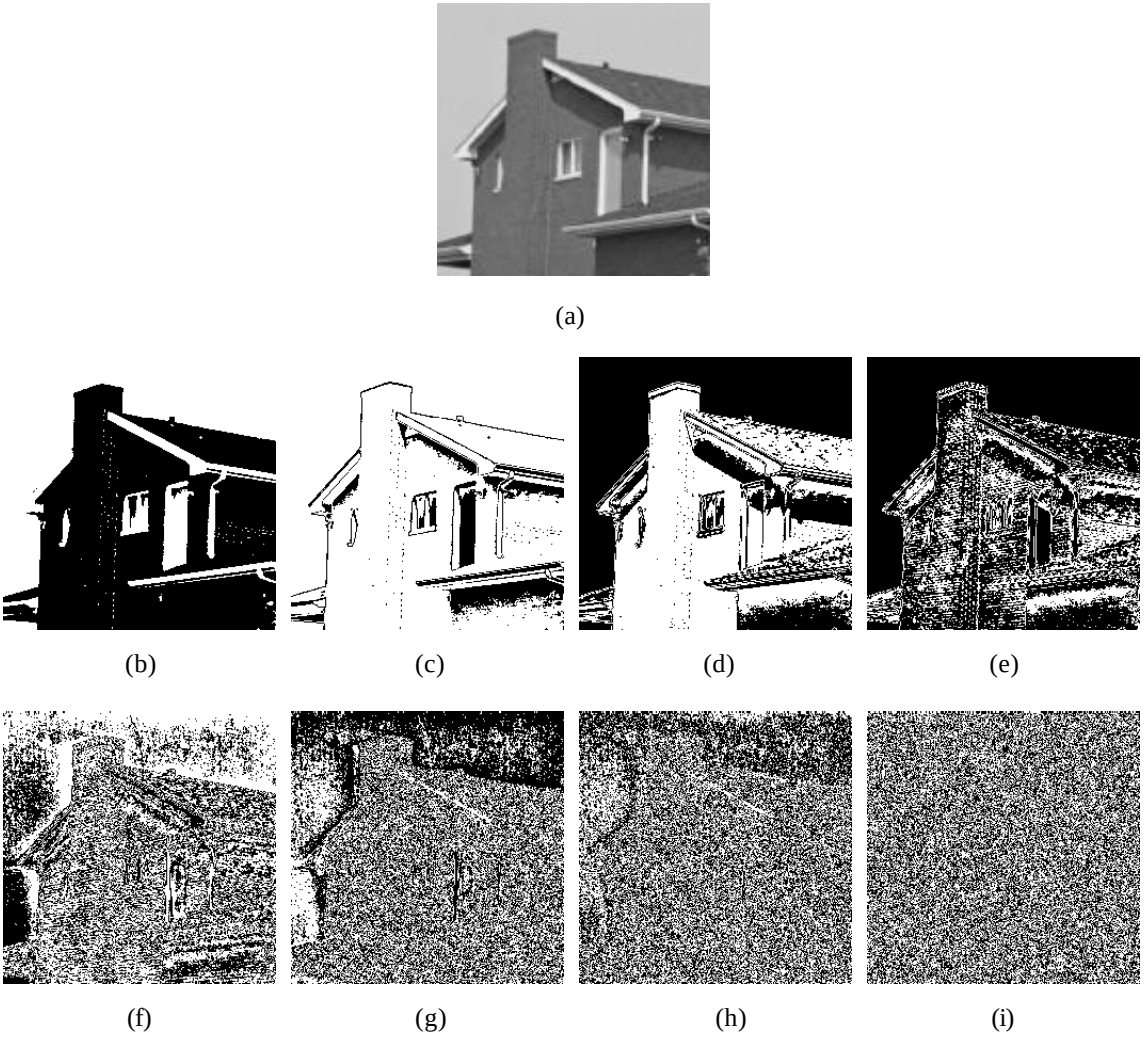
Based on the density function, a steganalysis model will be proposed to reveal the LSB-embedded image. Let W be a w * w window in a binary image. A transition means 1-to-0 or 0-to-1 change of two neighboring pixels. The transition density of W (TD) is defined to be the amount of horizontal and vertical transitions in W and described as follows.

$$TD = \sum_{(i,j)\in W} \left| x(i,j) - x(i,j+1) \right| + \sum_{(i.j)\in W} \left| x(i,j) - x(i+1,j) \right|,$$

where x(i,j) is the pixel value of (i,j). When w = 5, the TD belongs to [0, 40]. When W is a constant (all 0 or all 1) area, TD is 0. If the value of each pixel in W is given at random, TD approaches 20.

For each pixel P in a m*n binary image, take a W window centered at P and evaluate the TD of W, then a histogram of the TDs can be obtained. Figs. 4(a) and 4(b) show the TD histograms of Figs. 2(e) and 2(i), respectively. In Fig. 2(e), since some homogeneous areas exist, the mean of TDs is 8.2 that is far from 20, the standard derivation is 7.11, and the number of pixels with TD near-zero (TD<5) is 24150 (38%). On the other hand, Fig. 2(i) is the least significant bit-plane that is bestrewn with random texture, the mean of TDs is 20.32, the standard derivation is 3.49, and the number of pixels with TD near-zero is 2.

Assume that the embedded message is a random pattern consisting of "0" and "1". If we insert the



(a)

(b) (c) (d) (e)

(f) (g) (h) (i)

Fig. 2 An example to illustrate the 8 bit-planes of a gray-scale image. (a) A grayscale image "House" of size 256*256. (b)-(i) 8 bit-planes from the most significant bits to the least significant bits of pixels in (a).

message into a certain bit-plane, random texture will bestrew on the bit-plane. The histogram of TDs will look like Fig. 4(b). In a nature image, two or three least significant bit-planes may be bestrewn with random texture. Thus it would draw suspicion for the transmission of a hidden message, if the more significant bit-planes has high mean of TDs.

3. Minimum-Error LSB Replacement Method (MELSBR)

As mentioned above, there are total 256 levels to represent the intensity of each pixel of a grayscale image. If we were to embed k ($k<8$) bits of message in a pixel, directly replacing the k-LSBs of the pixel will introduce less error than replacing any other k-bits, and the maximum error is 2^k-1.

In the total 256 gray levels, there are $2^{(8-k)}$ gray levels with the same value in the k least significant bits as the k message bits. To reduce the embedding error, we should select the one that has minimum-error with the original gray level to replace the pixel level. To reach the aim, a simple way is provided. It will adjust the $(k+1)^{th}$ LSB, and check it's embedding error. And then select the gray-scale with less embedding error to replace the original ones. Fig. 5 illustrates the adjusting method, which contains two steps and is called minimum-error LSB replacement method (MELSBR). Using the MELSBR method, the maximum error can be restricted to $2^{(k-1)}$.

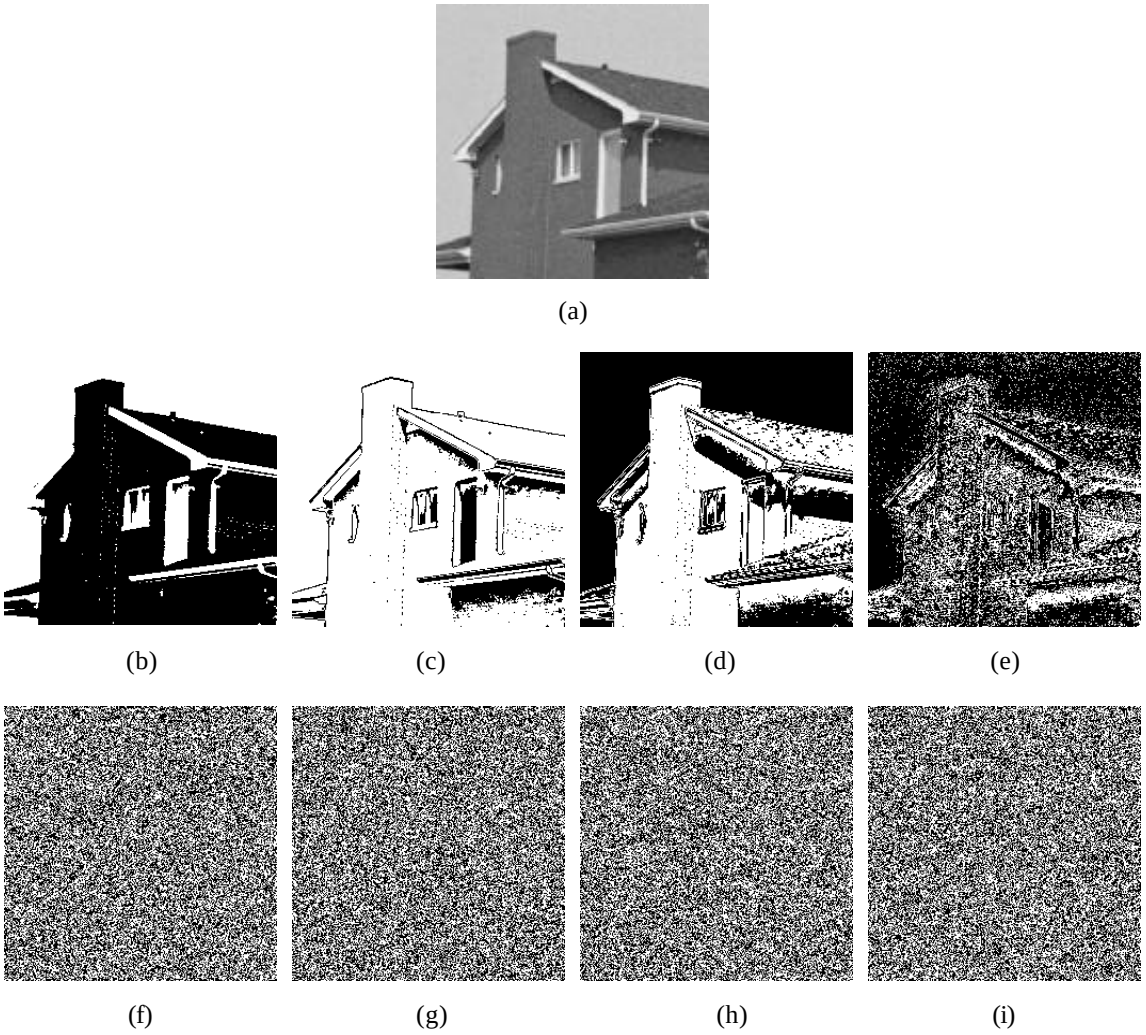


Fig. 3 An example of LSB-image with 4 LSBs insertion and its 8 corresponding bit-planes. (a) The stego-image "HouseLSB4".(b)-(i) 8 bit-planes from the most significant bits to the least significant bits of pixels in (a).

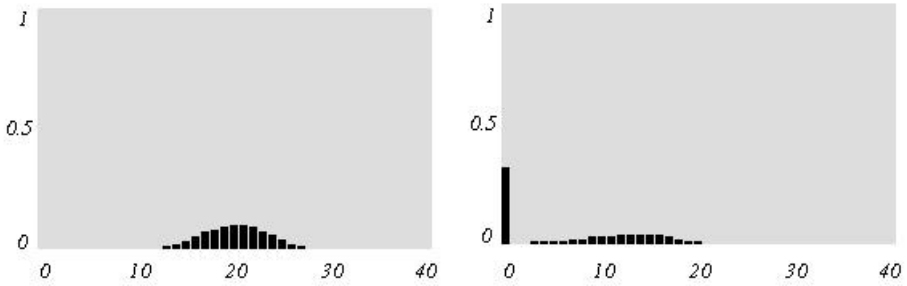


Fig.4 Two histograms from Figs. 2(e) and 2(i). (a) The TD histogram from Fig. 2(e). (b) The TD histogram from Fig. 2(i).

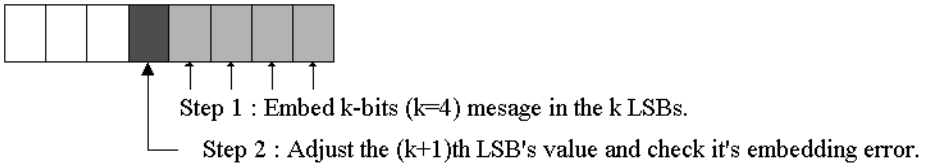


Fig. 5 Two steps of MELSBR method.

4. Adaptive MELSBR Method

To avoid changing the properties of cover-images, the message must be embedded in those "random texture" areas of each bit-plane. For taking advantage of local characteristics, an adaptive steganographic method based on the MELSBR method is proposed. First, the upper bound of embedding capacity for each pixel in the cover-image is evaluated. If the amount of message to be embedded is less than the total embedding capacity provided by the cover-image, to avoid embedding all message in a local area, a scattering method is provided to treat this problem.

Since the stego-image is viewed by human beings ultimately, it is worth exploring the characteristic of human visual system (HVS). Some properties of HVS have been applied for lossy image compression, these are also suited for image steganographic techniques. Human eyes are insensible to the change of high-frequency components. And high-frequency components characterize edges and other sharp details in an image. From this characteristic, we can see that the embedding capacity of each pixel is dependent on the gray level variation of the pixel's neighbors. Here, a spatial mask shown in Fig. 6 is used to evaluate the gray level variation D in neighbors of pixel x.

For each pixel x, define D as follows.

$$D = max\{a, b, c, d\} - min\{a, b, c, d\},$$

where a, b, c and d are the gray levels of x's neighborhood as shown in Fig. 6. The embedding capacity K of pixel x is defined as the minimum number of bits to store the value D minus 1. Thus K can be expressed as

$$K = \ddot{e} log_2(D) \hat{u} . \qquad (1)$$

Note that the amount of embedded message will rise in proportion as the logarithmic variation degree increases.

According to the luminance property of HVS, the greater a gray value is, the more change of the gray value we can make. Based on this property, we set the upper bound U of embedding capacity for each pixel as

$$U = \ddot{e} log_2(X) \hat{u} - 1, \qquad (2)$$

where X is the gray value of pixel x.

In order to scatter the message, a random number with value in [0,1] is generated for each pixel to decide whether the pixel is used to embed message. The embedding ratio P is defined as

$$P = AM / C,$$

where AM is the amount of message that would be embedded, and C is the predictive embedding capacity of a cover-image. If the random number is less than P, the pixel is used to embed message. In defense of known cover-image attack, we can randomly embed some random bits in those pixels which are not used to embed message.

a	b	c
d	x	

Fig.6 The mask for evaluating the gray variation in the neighbors of pixel "x".

Note that some information used in the embedding step and called hiding-information is needed in the extraction step. These include threshold, message type, height and width of message image, encryption scheme, compression scheme, etc. We divide pixels of cover-image into two parts. One is used to embed the hiding-information, the other is used to embed the message. The hiding-information part is selected by a random sequence, the message-embedded part is processed by row major sequentially. A benefit of such a scheme is that any digital signature scheme can be applied to the message-embedded part and embeds the signature in the hiding-information area. Using the digital signature scheme, the receiver can verify if the message is the exact one sent.

The whole steps of the proposed adaptive MELSBR algorithm is described as follows:

A-MELSBR Algorithm

Step 1. Predict the embedding capacity C of the cover-image.

Step 2. Compute the embedding ratio with P=AM/C, AM is the amount of embedded message.

Step 3. Randomly select the hiding-information area from the cover-image, and embeds the hiding-information.

Step 4. Scan the cover image from the top-left to the bottom-right. For each pixel x in the message-embedded part, perform the following steps:

Step 4.1 Generate a random number r with $0 \le r \le 1$

Step 4.2 Using Eqs.(1) and (2) to evaluate the embedding capacity (K) and the embedding upper bound (U) of x. Take K* = min(K,U)

If ($r \le p$)

Embed K*-bit message in the K*-least significant bits of x

Adjust the $(K+1)^{th}$ bit and check it's embedding error.

ELSE

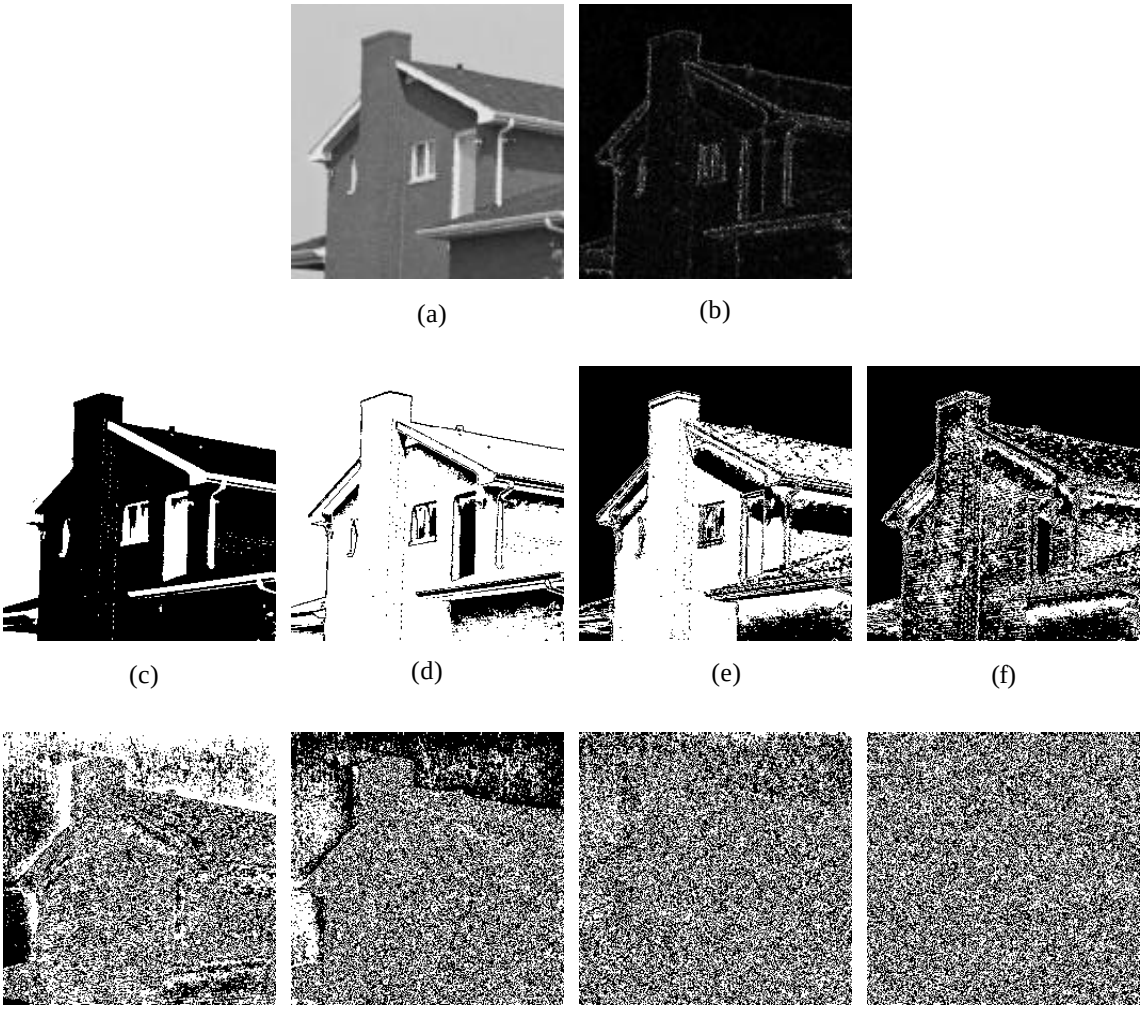
Embed ((r*100) mod K*)-bits random message.

The extraction step in our proposed model is very simple. Using the same stego-key, we can find these pixels that are used to embed message bits. And then we use the same mask to compute their K* values and extract the embedded message bits directly.

5. Experimental Results

We have used the proposed model to embed different kinds of messages in different images, Fig. 7 shows an example to embed the full capacity (160947-bit) message in the cover-image "House"(see Fig. 2(a)). The message is the forepart of the cipher-image "BomberC" (see Fig. 8(b)), which is the result of encrypting "BomberBase" (see Fig. 8(a)) using DES algorithm [4]. The embedded message occupies about 31% of size of cover-image "House" (524888 bits). The difference between the stego-image shown in Fig. 7(a) and the cover-image "House" is shown in Fig. 7(b), the root-mean-square (RMS) error is 6.48 and PSNR is 31.89. In order to view the embedding position clearly, we multiply the amplitude of each pixel in the error-image by 5. Note that the properties of random texture in Figs. 2(b)-2(i) are kept in Figs. 7(c)-7(j), so the stego-image will not draw the attention to the hidden message.

Fig. 9 shows the result of another experiment. The color scheme of the color-image "Waterlily" (see Fig. 9(a)) is very complex, the full capacity is 456201 bits (about 3.91 bits/pixel), it occupies 49% of the cover-image size. The stego-image and error-image are shown in Figs. 9(b) and 9(c), respectively. Note that the PSNR value is only 29.87dB, but it does not make sense from the viewpoint of human eyes. This is due to that the proposed model takes advantage of local characteristics of cover images.

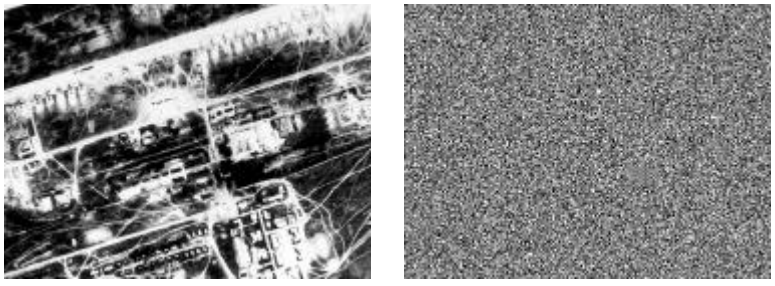


(a) (b)

(c) (d) (e) (f)

(g) (h) (i) (j)

Fig. 7 An experimental result of the proposed method. (a) The stego-image, entitled "HouseXC". (b) The error-image (RMS=6.48, PSNR=31.89dB). (c)-(j) 8 bit-planes of (a).



(a) (b)

Fig. 8 An example of image encryption using DES standard. (a) The message-image "BomberBase" of size 350*263. (b) The cipher-image of (a).

(a) (b) (c)

Fig. 9 Another experimental result. (a) The cover-image "WaterLily" of size 350*333. (b) The stego-image "WaterLilyXC" (c) The error-image (RMS=8.18 and PSNR=29.87dB).

6. Conclusions

We have described the general steganographic system and indicated some problems of embedding techniques based on LSB Insertion. A new steganalysis issue has been indicated, it uses random properties of the bit-planes to reveal the LSB-embedded images. Our proposed steganography model can treat these problems, it takes advantage of local characteristics of a cover-image to embed maximal amount of message in the cover-image and maintain the imperceptible alteration. Furthermore, it is hard to break the model to reveal the stego-image.

Acknowledgement

This work was supported in part by the National Science Council of the R.O.C. under contract NSC87-2213-E-009-006.

References

[1] Birgit Pfitzmann, "Information Hiding Terminology", First Workshop of Information Hiding Proceedings, Cambridge, U.K. May 30 - June 1, 1996. Lecture Notes in Computer Science, Vol.1174, pp 347-350. Springer-Verlag (1996).

[2] Neil F. Johnson and Sushil Jajodia, "Steganalysis of Image Created Using Current Steganography Software", Workshop of Information Hiding Proceedings, Portland Oregon, USA,15-17 April,1998. Lecture Notes in Computer Science, Vol.1525, Springer-Verlag (1998).

[3] A. Brown, S-Tools, Shareware ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/s-tools4.zip (Version 4)

[4] Bruce Schneier, "Applied Cryptography", 2nd Edition, John Wiley & Sons, (1996)

[5] Tuomas Aura, "Practical Invisibility in Digital Communication", First Workshop of Information Hiding Proceedings, Cambridge, U.K. May 30 - June 1, 1996. Lecture Notes in Computer Science, Vol.1174, Springer-Verlag (1996).

[6] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Addison-Wesley, 1992

[7] D. Upham, Jpeg-Jstego, Modification of the Independent JPEG Group. JPEG software (release 4) for 1-bit steganography in JFIF output file. ftp://ftp.funet.fi/pub/crypt/steganography

[8] Neil F. Johnson and Sushil Jajodia, "Steganography: Seeing the Unseen", IEEE Computer, February 1998, pp 26-34.