A High Capacity Image Steganographic Model

Yeuan-Kuen Lee and Ling-Hwei Chen [1]

Department of Computer and Information Science
National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Corresponding Address:
Ling-Hwei Chen
Department of Computer and Information Science
National Chiao Tung University
1001 Ta Hsueh Rd., Hsinchu 30050, TAIWAN
(e-mail: lhchen@cc.nctu.edu.tw)

Yeuan-Kuen Lee
Department of Computer and Information Science
National Chiao Tung University
1001 Ta Hsueh Rd., Hsinchu 30050, TAIWAN
(e-mail: yklee@debut.cis.nctu.edu.tw)

[1]To whom all correspondence should be sent.

Abstract

Steganography is an ancient art of conveying messages in a secret way that only the receiver knows the existence of message. So, a fundamental requirement for a steganographic method is imperceptibility, this means that the embedded messages should not be sensible to human eyes. Besides, there are two other requirements, one is to maximize the embedding capacity, and the other is security. The least significant bit (LSB) insertion method is the most common and easy one to embed messages in an image. However, how to decide the maximal embedding capacity for each pixel is still an open issue. In this paper, we will propose an image steganographic model that is based on variable-sized LSB insertion to maximize the embedding capacity while maintaining the image fidelity. In the proposed model, for each pixel of a gray-scale image, at least 4 bits can be used for messages embedding. Three components are provided to achieve the goal. First, according to contrast and luminance characteristics, the capacity evaluation (CE) is provided to estimate the maximum embedding capacity of each pixel. Then, the minimum-error replacement (MER) method is adapted to find a gray-scale as close to the original one as possible. Finally, the improved gray-scale compensation (IGSC), which takes advantage of the peculiarities of human visual system (HVS), is used to eliminate the false contouring effect. Furthermore, two methods, pixelwise and bitwise, are provided to deal with the security issue when using the proposed model. Experimental results show effectiveness and efficiency of the proposed model.

1 Introduction

With the development of Internet technologies, digital media can be transmitted conveniently over the network. Therefore, How to protect secret messages during transmission becomes an important issue. Using the classic cryptography [1] only, the encrypted message becomes clutter data that can not pass the checkpoint on the network. Steganography [2] 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 agrees with that in [3]. A steganographic method will embed messages in a cover-media and create a stego-media. Some steganographic methods [4,5] use a stego-key to embed messages for achieving rudimentary security.

There are two kinds of image steganographic techniques: spatial-domain and frequency-domain based methods. Spatial-domain based methods [5-8] embed messages in the intensity of pixels of images directly. For frequency-domain based ones [9-12], images are first transformed to frequency domain, and then messages are embedded in the transform coefficients.

The most common and simplest steganographic method [13-15] is the least significant bit (LSB) insertion method, it embeds message in the least significant bit. For increasing the embedding capacity, two or more bits in each pixel can

be used to embed messages. At the same time, not only the risk of making the embedded statistically detectable increases but also the image fidelity degrades. So, how to decide the number of bit of each pixel used to embed message becomes an important issue of image steganography.

There are two types of LSB insertion methods, fixed-sized and variable-sized. The former embeds the same number of message bits in each pixel of the cover-image, Fig. 1 shows an example of fixed-sized LSB insertion. The cover-image, shown in Fig. 1(a), is entitled Lenna. There is a smooth area on the Lenna's shoulder. Fig. 1(b) shows the stego-image that is derived by directly embedding fixed 4 random bits in the 4 LSBs of each pixel. The embedding capacity is 50% of the cover-image size. In Fig. 1(b), we can see some false contours appearing on the shoulder of Lenna. The unwanted artifacts may arise suspicion and defeat the purpose of steganography. To treat this problem, either less bits are used for message embedding or a variable-sized method is applied.

For the variable-sized embedding method, the number of LSBs in each pixel used for message embedding depends on the contrast and luminance characteristics. How to adapt these local characteristics to estimate the maximum embedding capacity while maintaining the image fidelity is still an open issue. In this paper, we will propose a high capacity image steganographic model based on variable-sized LSB insertion. The embedding capacity will be over 50% of the cover-image size, and the image fidelity is better than 4-LSBs insertion. The most important advantage is that no artifacts appear in the stego-image with high embedding capacity.

In the next Section, we will describe the proposed image embedding/extracting

modules. In Section 3, two methods will be presented to achieve the security requirement. In Section 4, some experimental results are provided to show that the proposed embedding method is better than the 4-LSBs insertion method. Finally, conclusions will be presented.

2 The Proposed Image Embedding and Extracting Modules

Fig. 2 shows the block diagram of the proposed steganographic model. The input messages can be images, texts, video, etc. Since some content-dependent patterns in the original messages may reveal the existence of the messages, and embedding more bits of messages will introduce more degradation, the compression module is used first to deal with these problems mentioned above. To raise the security level, a suitable encryption algorithm [1] is then applied on the compressed messages to conceal the meaning of messages. Note that the encryption module can conceal the content-dependent property, and the compression module must be performed prior to the encryption module for the benefit of entropy coding. The key-encryption module is optional. The stego-key is used to locate the embedding positions on the cover-image and the secret-key is used in the encryption module. If the sender and the receiver does not share both the stego-key and the secret-key, the sender can use the public-key of the receiver to encrypt these keys, then embeds these encrypted keys in fixed positions of the stego-image. When the receiver receives the stego-image, these keys are extracted and then decrypted using the private-key of the receiver.

In this Section, we will propose a high capacity gray-scale image embedding module and its corresponding extracting module. Fig. 3 illustrates the block diagram of the modules. In the embedding module, there are three components. The first one is to evaluate the capacity of each pixel of the cover image, the others aim to increase the image fidelity and eliminate the false contouring that may appear in the smooth regions of the stego-image. Note that the extracting module only uses the capacity evaluation component to extract the embedded messages.

2.1 The Embedding Module

In order to meet the requirement of imperceptibility and maximize the embedding capacity, three key concepts are used in the embedding module. First, the embedding capacity of each pixel must adapt to local image characteristics, such as contrast and luminance. Second, the new gray-scale of each embedded pixel should be as close to the original one as possible. Finally, the stego-image should not have any artifact. The embedding module consists of three major components: capacity evaluation (CE), minimum-error replacement (MER) and improved gray-scale compensation (IGSC).

First, for each pixel, the CE component uses the gray-scale variation of neighboring pixels and its intensity to evaluate its embedding capacity. Note that the local characteristics should not be changed after embedding message, then we can use the same characteristics to evaluate the embedded capacity in the extracting module.

Then, the MER component finds a replacing gray scale, based on the following

two requirements. First, the LSBs should be identical to the embedded message bits. Second, the gray-scale, which meets the first requirement, with minimum error to the original one should be taken. Using this gray-scale to replace the original one, the maximum embedding error can be restricted to $2^{(k-1)}$ while embedding k message bits in k LSBs. Note that reducing the embedding error can make us embed more messages in the cover-image.

Finally, the IGSC component compensates the embedding error from neighboring pixels to eliminate the false contouring without impairing the quality of image perception. In the following, the proposed embedding module will be described in detail.

2.1.1 The CE Component

Since the stego-image is viewed by human beings ultimately, it is worth exploring the characteristic of human visual system (HVS). HVS is insensitive to the noise component and the psychovisually redundant component in an image, thus these components can be used to embed messages.

For penetrating an image, we decompose the gray-scale of each pixel into 8 bits. The plane formed by the same bit of each pixel in a gray-scale image is called a bit-plane. Fig. 4 shows the 8 corresponding bit-planes of Fig. 1(a). Observing these bit-planes, we can see that some areas in the six least significant bit-planes are bestrewn with noise. HVS is insensitive to the value change in these areas. Thus, we can use these areas to embed messages. The main contribution of this paper is to locate these areas.

Generally speaking, the more significant bit-plane the noise area appears in, the larger variation of gray values among the neighboring pixels there will be, and then more bits could be used to embed messages. So, the first step in this module is based on the gray value variation of neighboring pixels to compute the number of embedding bits for each pixel.

The embedding module will be applied to each pixel from left to right and from top to bottom in an image sequentially. Assume that the gray scale of one pixel p at coordinates (x, y) is denoted by $f(x, y)$, the 8-neighbors of p are shown in Fig. 5. For p, f(x, y) will be modified according to its embedding capacity, which depends on its gray scale and the gray-scale variation of the upper and left neighbors (see the shaded pixels in Fig. 5). The advantage of using the upper and left neighbors to estimate the embedding capacity is that when or after the current pixel is processed, the gray scales of these upper and left neighbors will be never changed. Therefore, the embedding module and extracting module are synchronous when estimating the embedding capacity of each pixel.

Let

$$Max(x, y) = \max\{f(x-1, y-1), f(x-1, y), f(x-1, y+1), f(x, y-1)\},$$

$$Min(x, y) = \min\{f(x-1, y-1), f(x-1, y), f(x-1, y+1), f(x, y-1)\},$$

$$D(x, y) = Max(x, y) - Min(x, y),$$

Except for the boundary pixels in an image, the embedding capacity $Kn(x, y)$ of each pixel (x, y) is defined as

$$Kn(x, y) = \lfloor \log_2 D(x, y) \rfloor.$$

According to the HVS, the greater a gray-scale is, the more change of the gray-scale could be tolerated. Thus, the embedding capacity should be limited by the gray scale of current pixel. Here, an upper bound for embedding capacity at pixel (x, y) is defined as

$$U(x,y) = \begin{cases} 4, & \text{if } f(x,y) \leq t, \\ 5, & \text{otherwise.} \end{cases}$$

Note that t is set to be 191, the reason is described as follows. In the embedding module, the original gray scale is used to find $U(x, y)$, but in the extracting module, the gray scale has been changed. To make $U(x, y)$ consistent, the original gray scale and the modified one should appear in the same region ([0, t] or (t, 255]), only 191 meets this requirement.

On the other hand, according to the proposed IGSC component, which will be described later, the lower bound for embedding capacity could be set as 4 bits. So the embedding capacity $K(x, y)$ of each pixel can be computed by the following expression.

$$K(x,y) = \min\{\max\{Kn(x,y), 4\}, U(x,y)\}.$$

Fig. 6 shows the embedding result of applying the CE component on Lenna. The average embedding capacity of each pixel is 4.06 bits/pixel, and the RMS and PSNR values are 6.59 and $31.75dB$, respectively. We can see that some artifacts exist on the smooth regions of the image, these will be solved by the IGSC component.

2.1.2 The MER Component

In general, 8 bits are used to represent the intensity of each pixel in a gray-scale image. If we want to embed k ($k < 8$) bits in a pixel, then replacing the k-LSBs of

the pixel will introduce the smallest error than replacing any other k bits. In this case, the maximum embedding error introduced is $2^k - 1$. Considering the 256 gray-scales, there are $2^{(8-k)}$ gray levels whose k LSBs are identical to the k embedded bits. To achieve the highest quality, we can take the most similar gray-scale among these $2^{(8-k)}$ gray-scales to replace the original one. To reach the aim, a simple way to search the closest gray-scale is provided here.

Let $f(x, y)$ be the original gray-scale, $g(x, y)$ be the gray-scale obtained by embedding k LSBs directly, and $g'(x, y)$ be the gray-scale obtained by changing the value of the $(k+1)^{th}$ LSB of $g(x, y)$. The minimum-error gray-scale must be $g(x, y)$ or $g'(x, y)$. Let $e(x, y)$ be the error between $f(x, y)$ and $g(x, y)$, and $e'(x, y)$ be the error between $f(x, y)$ and $g'(x, y)$. If $e(x, y) < e'(x, y)$, then g(x, y) will be used to replace $f(x, y)$; otherwise $g'(x, y)$ is selected. Fig. 7 illustrates the replacing method, which contains two steps and is called minimum-error replacement (MER). Using this method, the maximum embedding error can be restricted to $2^{(k-1)}$.

Fig. 8 shows the result of applying the MER component on Fig. 6. It is quite clear that the image fidelity is increased. The RMS value is reduced from 6.59 to 5.76, and the PSNR value is increased from $31.75dB$ to $32.92dB$. However, some unwanted artifacts still appear in the smooth areas, such as the shoulder of Lenna. In the next section, we will propose a method to address this problem.

2.1.3 The IGSC component

As mentioned above, embedding too many bits in the smooth area of an image will cause the false contouring (see the face of Baby in Figs. 6 and 8). The same

phenomenon also appears in a quantized image, this is due to that an insufficient number of gray levels will not represent the smooth area of an image well. An efficient approach to eliminate these artifacts is known as improved gray-scale (IGS) quantization [16]. This concept is similar to the error diffusion method that is commonly used in conversion of true color images to palette-based color ones[17, 18]. One advantage of error diffusion is that the average image intensity values can be preserved.

In our IGSC component, the embedding error is evenly spread to the bottom and right neighboring pixels (the white neighboring pixels shown in Fig. 5). Let $e(x, y)$ denote the embedding error of pixel p at coordinates (x, y), these four bottom-right neighboring gray-scales are then modified according to the following expressions.

$$f(x, y+1) = f(x, y+1) + \frac{1}{4}\, e(x, y),$$
$$f(x+1, y-1) = f(x+1, y-1) + \frac{1}{4}\, e(x, y),$$
$$f(x+1, y) = f(x+1, y) + \frac{1}{4}\, e(x, y),$$
$$f(x+1, y+1) = f(x+1, y+1) + \frac{1}{4}\, e(x, y).$$

Fig. 9 illustrates the effectiveness of applying the IGSC component on Fig. 8. The embedding capacity is 4.06 bits/pixel and the RMS and PSNR values are 5.07 and $34.03dB$, respectively. Note that no false contouring appears in Fig. 9.

2.2 The Extracting Module

The extracting module in the proposed method is very simple. Using the same CE component as that in the embedding module to compute the embedded capacity

of each pixel, those embedding messages can be extracted directly.

3 Security Issue of the Proposed Modules

Up to now, we have proposed a high capacity embedding method, that could meet the imperceptible requirement. However, the security requirement has not been addressed. In this section, we will present two solutions, pixelwise and bitwise, to deal with the security issue.

For each pixel, the pixelwise method will generate a random number in [0,1] to decide whether the pixel is used to embed message. A stego-key is used as the seed of the random number generator. The sender must select an embedding ratio, which determines the number of pixels used for message embedding, as a threshold value. If a random number is smaller than the embedding ratio, the corresponding pixel will be used for message embedding. Note that the receiver must know the embedding ratio to locate the embedded messages. One available way to tackle this problem is to embed the embedding ratio in the cover-image. So, the prerequisite for the receiver to extract the embedded messages is the stego-key.

The bitwise method is similar to the pixelwise one, a random number is generated for each bit, which is originally considered to embed message. Assuming that the embedding capacity of a pixel is k, k random numbers will be generated for the least significant k bits, respectively. An example, shown in Fig. 10, is given to illustrate the bitwise method. Suppose that the embedding ratio T is 0.5, and the embedding capacity k is 4 for some pixel, then 4 random numbers will be generated. Suppose that these random numbers are (0.81, 0.47, 0.25, 0.63). Since

the second and the third random numbers are smaller than 0.5, the second and the third LSBs will be used for message embedding. Assuming that the original gray-scale is 142 (0x10001110) and the two embedded bits are 0 and 1. Since the gray-scale 139 (0x10001011) is the closest one with the third and the second LSB being 0 and 1,respectively. The original gray-scale 142 will be replaced by 139. Finally, the embedding error will be uniformly spread to neighboring pixels via IGSC component.

4 Experimental Results

We have tested the proposed embedding module on a number of gray-scale images. First, to test the image fidelity in the worst case, in each cover-image the maximum amounts of random messages were embedded using the proposed method. Note that all the embedded random messages used in our experiments were obtained by applying DES encryption algorithm [1] on each image. Two objective fidelity criteria, the root-mean-square (RMS) error and the peak signal-to-noise ratio (PSNR), are used to evaluate the performance of our method and that of the 4-LSBs insertion. Then, we embed from 90% to 10% of maximal capacity to compare the image fidelity between pixelwise and bitwise methods.

On the average case, our proposed method can embed 4.025 bits in each pixel, the embedding capacity is a little more than 4 bits. Furthermore, the RMS and PSNR are listed in Table 1. From Table 1, we can see that the performance of the proposed method is better than that of the 4-LSBs insertion. In addition, the great benefit of our proposed method is that no false contours appear in the smooth area.

Fig. 11(a) shows a gray-scale image, entitled Maraho, obtained from a scanner, its background is near white color. Fig. 11(b) shows the result of embedding the maximum capacity (1043351 bits, 4.011 bits/pixel) in Fig. 11(a). The RMS and PSNR measure for Fig. 11(b) are 6.34 and $32.09dB$, respectively. From the viewpoint of human eyes, these two images are almost undistinguishable.

From the description in previous section, we can see that more random numbers are needed and a sequential search is necessary in the bitwise method. So the pixelwise method is more efficient than the bitwise one in the issue of computational speed. Now, we embed random messages with embedding capacity ranging from 90% to 10% of maximal capacity to compare the stego-image fidelity between these two methods. Fig. 12 illustrates the RMS and PSNR curves on the average case. Clearly, both RMS and PSNR measurements of bitwise method are better than those of pixelwise one. From the fidelity issue, the bitwise method is a better choice. It is worthy of noting that the fidelity of all stego-images of applying these two methods is acceptable to human eyes. Therefore, the sender can choose different methods alternatively to increase the difficulty of steganalysis on these stego-images. This is the major benefit of supporting these two security methods in the proposed model.

5 Conclusions

In this paper, we have introduced an image steganographic model and have proposed a new high capacity embedding/extracting module that is based on the variable-sized LSB insertion. In the embedding part, based on the contrast and luminance property, we use three components to maximize the capacity, minimize

the embedding error and eliminate the false contours. Using the proposed method, we can embed at least 4 message bits in each pixel while maintaining the imperceptible requirement. For the security requirement, we have presented two different ways to deal with the issue. The major benefit of supporting these two ways is that the sender can use different methods in different sessions to increase difficulty of steganalysis on these stego-images. Only using the stego-key, which is used as the seed of the random number generator, the receiver can extract the embedded messages exactly. Experimental results show that the proposed model is effective and efficient.

6 Acknowledgment

This research was supported in part by the National Science Council, R.O.C., under Contract NSC 88-2213-E-009-063.

References

[1] SCHNEIER, B.: 'Applied cryptography' (John & Wiley, New York, 1996) 2nd edn.

[2] KAHN, D.: 'The history of steganography', First Workshop of Information Hiding Proceedings, May 30 - June 1, 1996, Cambridge, U.K., pp. 1-5, Lecture Notes in Computer Science, Vol. 1174 (Springer-Verlag)

[3] PFITZMANN, B.: 'Information hiding terminology', First Workshop of Information Hiding Proceedings, May 30 - June 1, 1996, Cambridge, U.K., pp. 347-350, Lecture Notes in Computer Science, Vol. 1174 (Springer-Verlag)

[4] ANDERSON, R.J., and PETITCOLAS, F.A.P.: 'On the limits of steganography', IEEE Journal on Selected Areas in Communications (J-SAC), Special Issue on Copyright & Privacy Protection, 1998, 16(4), pp. 474-481

[5] KUTTER, M., JORDAN, F., and BOSSEN, F.: 'Digital signature of color images using amplitude modulation', J. of Electronic Imaging, 1998, 7(2), pp. 326-332

[6] LANGELAAR, G.C., LUBBE, J.C.A., and BIEMOND, J.: 'Copy protection for multimedia data based on labeling techniques', 17th Symposium on Information Theory in the Benelux, 30-31 May 1996, Enschede, The Netherlands.

[7] DARMSTAEDTER, V., DELAIGLE, J.-F, QUISQUATER, J.J., and MACQ, B.: 'Low cost spatial watermarking', Comput. & Graphics, 1998, 22(4), pp. 417-424

[8] SWANSON, M.D., KOBAYASHI, M., and TEWFIK, A.H.: 'Multimedia data embedding and watermarking technologies', Proceedings of the IEEE, 1998, 86(6), pp. 1064-1087

[9] COX, I.J., KILIAN, J., LEIGHTON, T., and SHAMOON, T.: "Secure spread spectrum watermarking for multimedia', IEEE Trans. on Image Processing, 1997, 6(12), pp. 1673-1687

[10] WOLFGANG, R.B., PODILCHUK, C.I., and DELP, E.J.: 'Perceptual watermarks for digital images and video', Proceedings of SPIE/IS&T International Conference on Security and Watermarking of Multimedia Contents, San Jose, CA, January 25-27, 1999, Vol. 3657, pp. 44-51

[11] KOCH, E. and ZHAO, J.: 'Towards robust and hidden image copyright labeling', IEEE Workshop on Nonlinear Signal and Image Processing, June 20-22, 1995, Neos Marmaras, Halkidiki, Greece, pp. 452-455

[12] XIA, X.-G., BONCELET, C. G., and ARCE, G.R.: 'A multiresolution watermark for digital images', IEEE International Conference on Image Processing, Oct. 26-29, 1997, Santa Barbara, CA, Vol. 3, pp. 548-551

[13] LIN E. T., and DELP, E. J.: 'A review of data hiding in digital images', Proceedings of the Image Processing, Image Quality, Image Capture Systems Conference, PICS '99, April 25-28, 1999, Savannah, Georgia, pp. 274-278

[14] JOHNSON, N.F., and JAJODIA, S.: 'Steganography: seeing the unseen', IEEE Computer, February 1998, pp. 26-34

[15] BENDER, W., GRUHL D., MORIMOTO, N. and LU, A.: 'Techniques for data hiding', IBM System Journal, 1996, 35(3&4), pp. 313-336

[16] GONZALEZ, R.C., and WOODS, R.E.: 'Digital Image Processing' (Addison-Wesley, New York, 1992)

[17] MINTZER, F.C., GOERTZEL, G. and THOMPSON, G.R.: 'Display of images with calibrated color on a system featuring monitors with limited color palettes', 1992 SID International Symposium Digest Tech. Papers, pp. 377-380

[18] YEUNG, M.M., and MINTZER F.C.: 'Invisible watermarking for image verification', J. of Electronic Imaging, 1998, 7(3), pp. 578-591

(a) (b)

Figure 1: An experimental result of fixed 4 LSBs insertion method. (a) A gray-scale cover-image, entitled Lenna ($512 * 512$). (b) The stego-image using fixed 4 LSBs insertion.

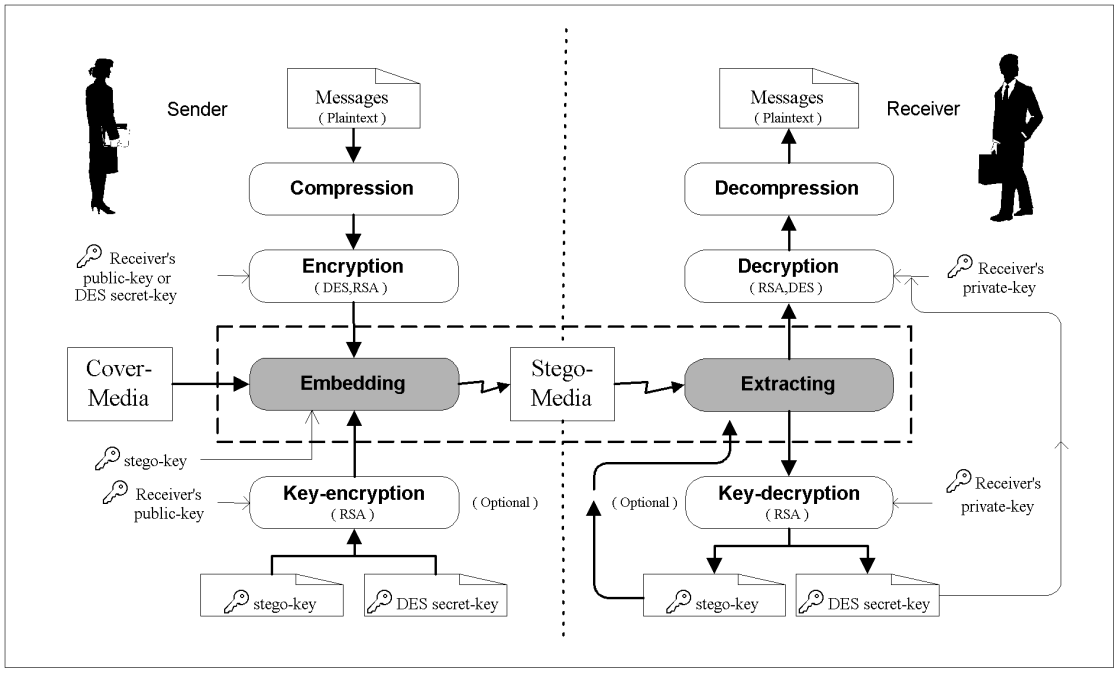


Figure 2: The block diagram of the proposed steganographic model.

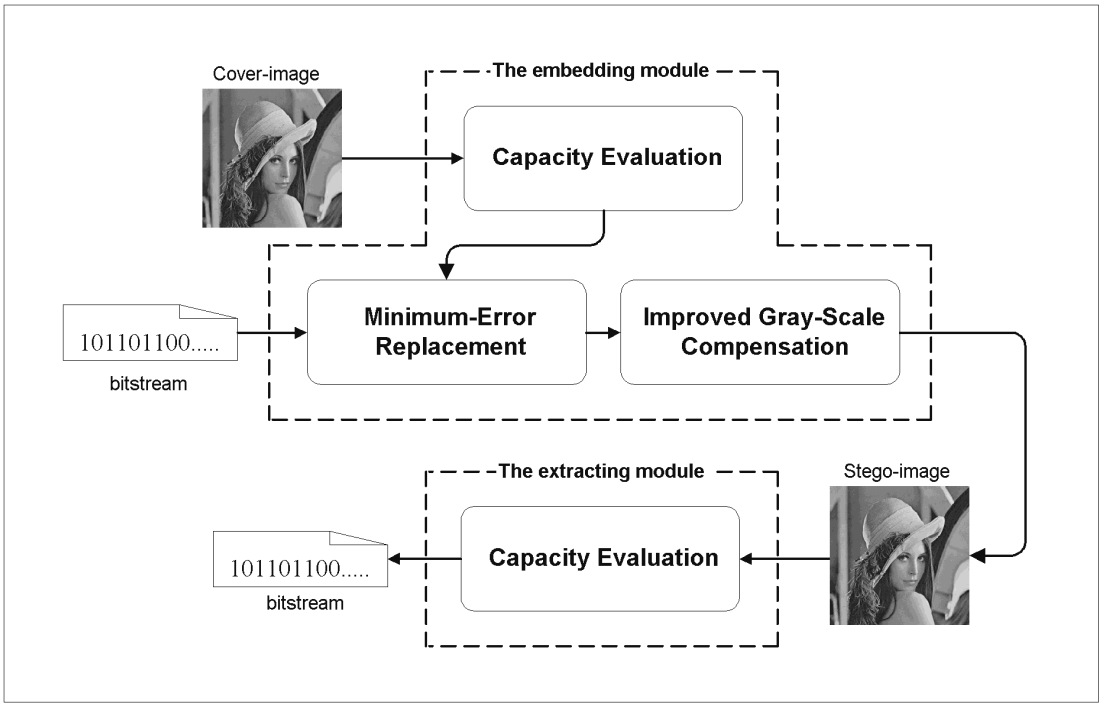


Figure 3: The block diagram of proposed image embedding and extracting modules.

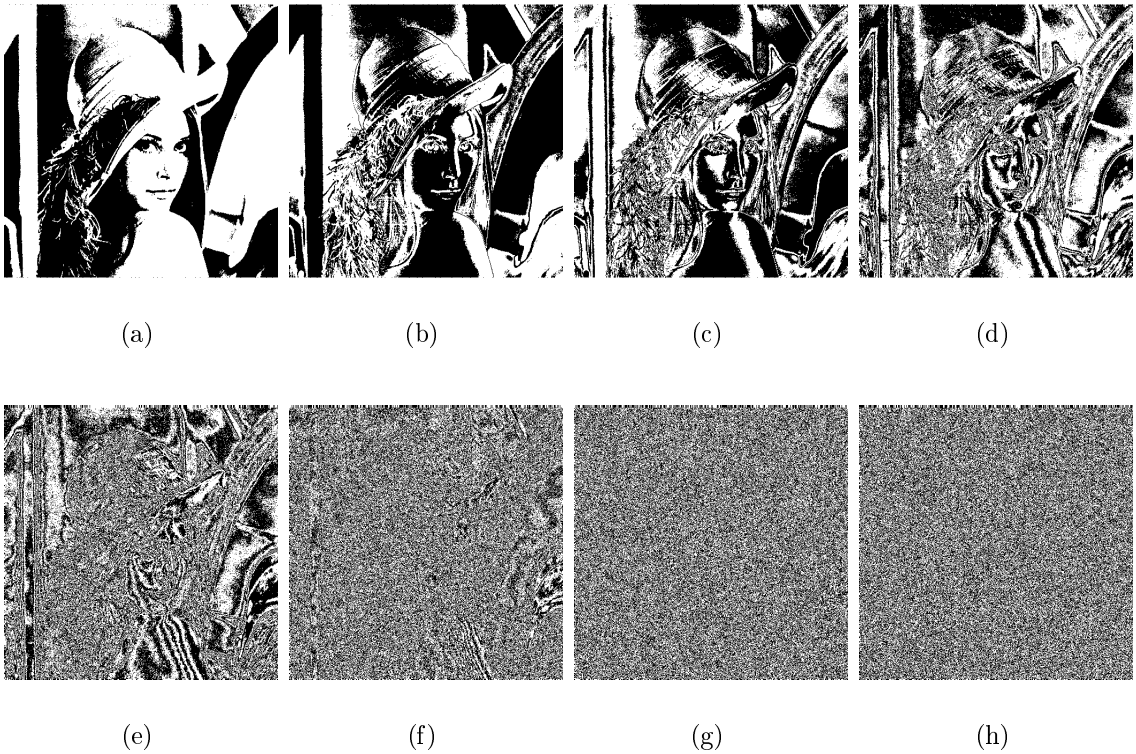


Figure 4: 8 bit-planes of Lenna $(512 * 512)$. (a)-(h) 8 bit-planes from the most significant bit to the least significant bit.

(x-1,y-1)	(x-1,y)	(x-1,y+1)
(x,y-1)	(x,y)	(x,y+1)
(x+1,y-1)	(x+1,y)	(x+1,y+1)

Figure 5: The 8-neighbors of pixel p at coordinates (x, y).

Figure 6: The embedding result of applying the CE component on Lenna.

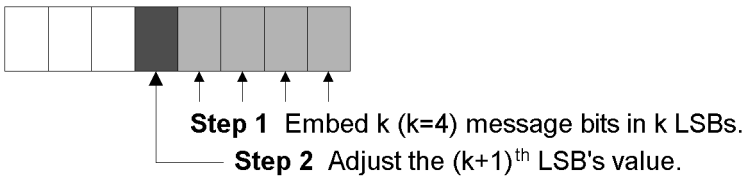


Figure 7: Two steps of the MER component.

Figure 8: The result of applying MER component on Fig. 6.

Figure 9: The result of applying IGSC component on Fig. 8.

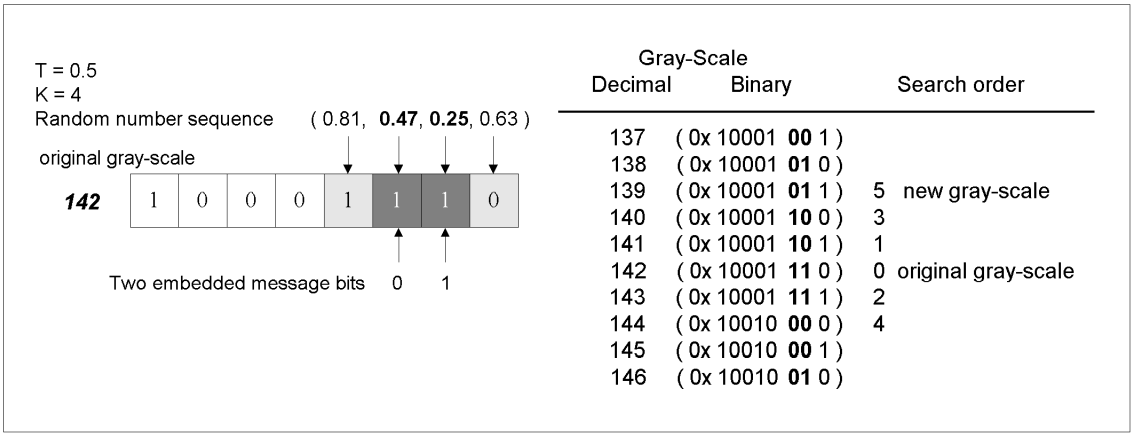
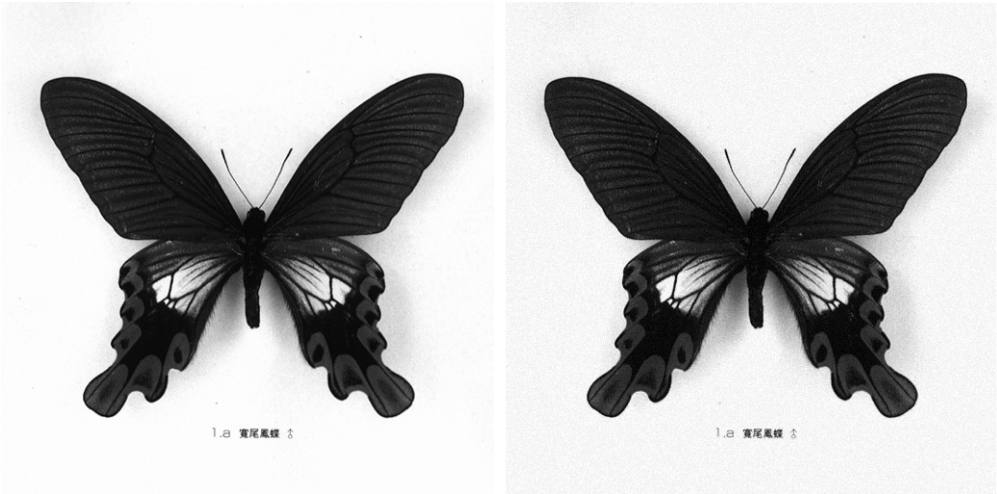
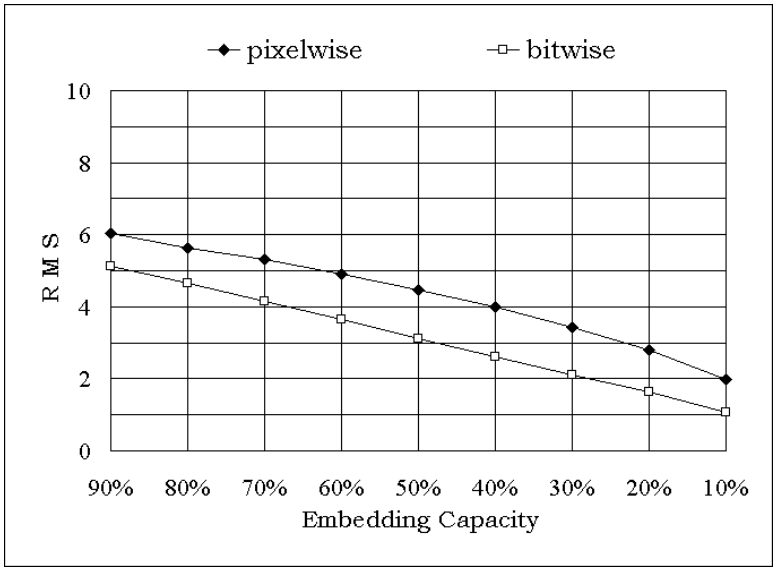


Figure 10: An example of applying the bitwise method

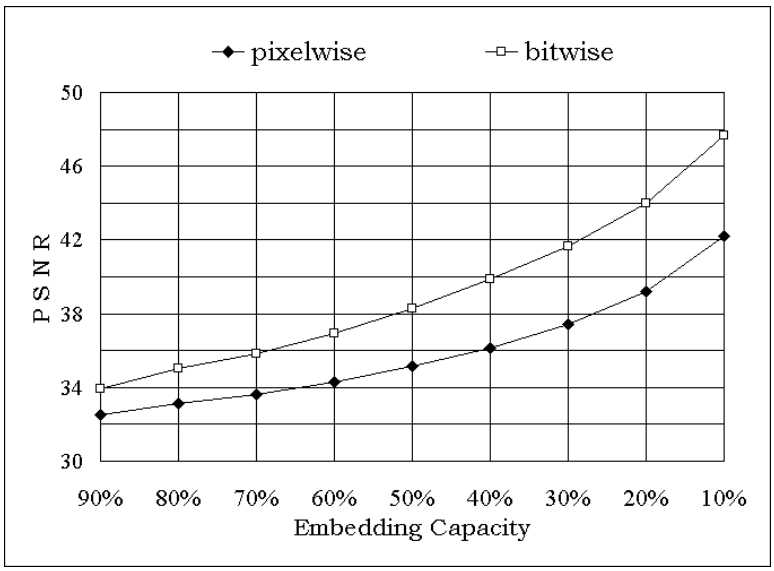


(a) (b)

Figure 11: An experimental result of the proposed method. (a) A cover-image, entitled Maraho ($512 * 512$). (b) The stego-image of the proposed method ($RMS = 6.34, PSNR = 32.09dB$).



(a)



(b)

Figure 12: The comparison of the pixelwise and bitwise methods. (a) RMS (b) PSNR

Table 1: Experimental results of the average case.

Embedding Method	Capacity	RMS	PSNR	Artifacts
4 LSBs Insertion	4 bits/pixel	6.61	31.71	Yes
Proposed Method	4.025 bits/pixel	6.02	32.57	No