High capacity image steganographic model

Y.K.Lee and L.H.Chen

Abstract: Steganography is an ancient art of conveying messages in a secret way that only the receiver knows the existence of a message. So a fundamental requirement for a steganographic method is imperceptibility; this means that the embedded messages should not be discernible to the human eye. There are two other requirements, one is to maximise the embedding capacity, and the other is security. The least-significant bit (LSB) insertion method is the most common and easiest method for embedding messages in an image. However, how to decide on the maximal embedding capacity for each pixel is still an open issue. An image steganographic model is proposed that is based on variable-size LSB insertion to maximise the embedding capacity while maintaining image fidelity. For each pixel of a grey-scale image, at least four bits can be used for message embedding. Three components are provided to achieve the goal. First, according to contrast and luminance characteristics, the capacity evaluation is provided to estimate the maximum embedding capacity of each pixel. Then the minimum-error replacement method is adapted to find a grey scale as close to the original one as possible. Finally, the improved grey-scale compensation, which takes advantage of the peculiarities of human visual system, is used to eliminate the false contouring effect. 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 classic cryptography [1] only, the encrypted message becomes clutter data that cannot 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 is 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 medium 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 methods [9–12] images are first transformed into the 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-size and variable-size. The former embeds the same number of message bits in each pixel of the cover image, Fig. 1 shows an example of fixed-size LSB insertion. The cover image, shown in Fig. 1*a*, is entitled 'Lena'. There is a smooth area on the Lena's shoulder. Fig. 1*b* shows the stego-image that is derived by directly embedding fixed four random bits in the four LSBs of each pixel. The embedding capacity is 50% of the cover-image size. In Fig. 1*b*, one can see some false contours appearing on the shoulder of Lena. 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-size method is applied.

For the variable-size 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 propose a high-capacity image steganographic model based on variable-size LSB insertion. The embedding capacity will be over 50% of the cover-image size, and the image fidelity is better than four-LSBs insertion. The most important advantage is that no artifacts appear in the stego-image with high embedding capacity.

IEE Proceedings online no. 20000341

DOI: 10.1049/ip-vis:20000341

Paper received 19th July 1999

The authors are with the Department of Computer and Information Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30050, Taiwan, Republic of China

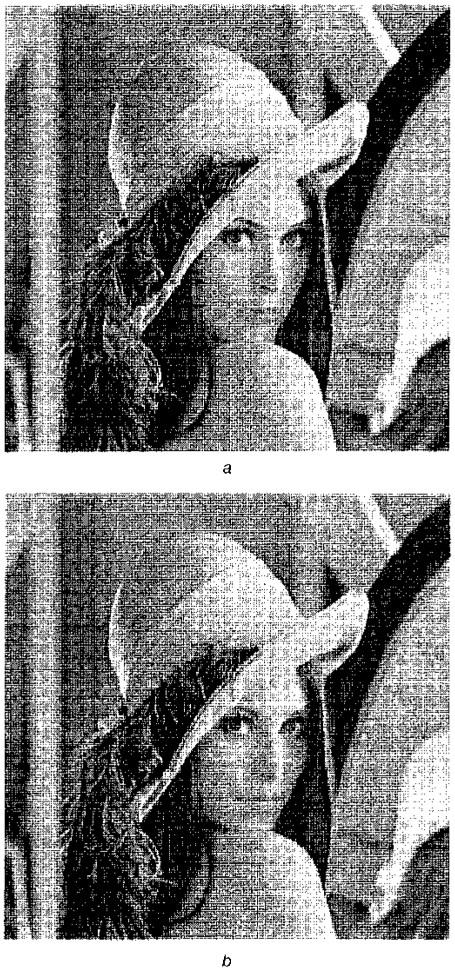

Fig. 1 *Experimental result of fixed four LSBs insertion method*
a Grey-scale cover image entitled 'Lena' (512 × 512)
b Stego image using fixed four LSBs insertion

2 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. 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 coverimage 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 propose a high-capacity grey-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 for evaluation of 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 Embedding module

To meet the requirement of imperceptibility and maximise 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. Secondly, the new grey scale of each embedded pixel should be as close to the original one

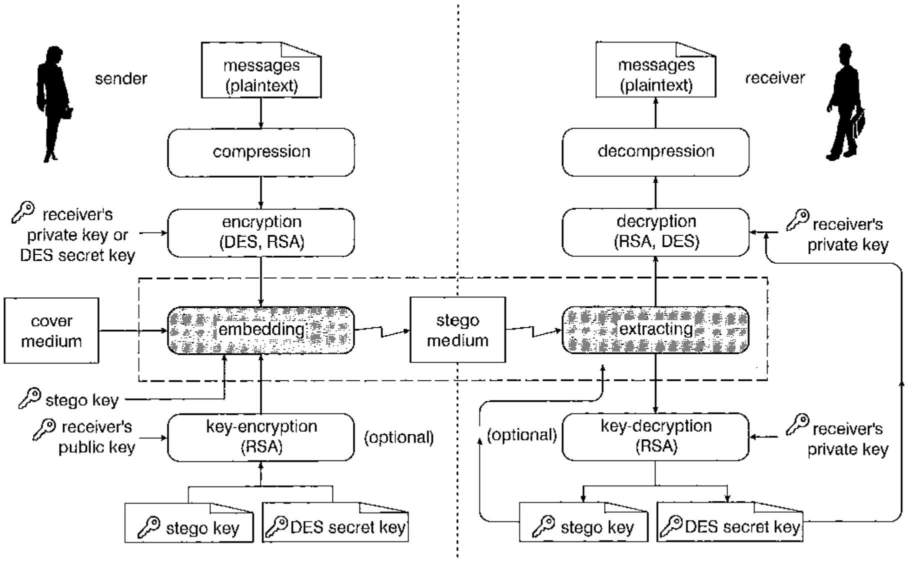

Fig. 2 *Block diagram of proposed steganographic model*

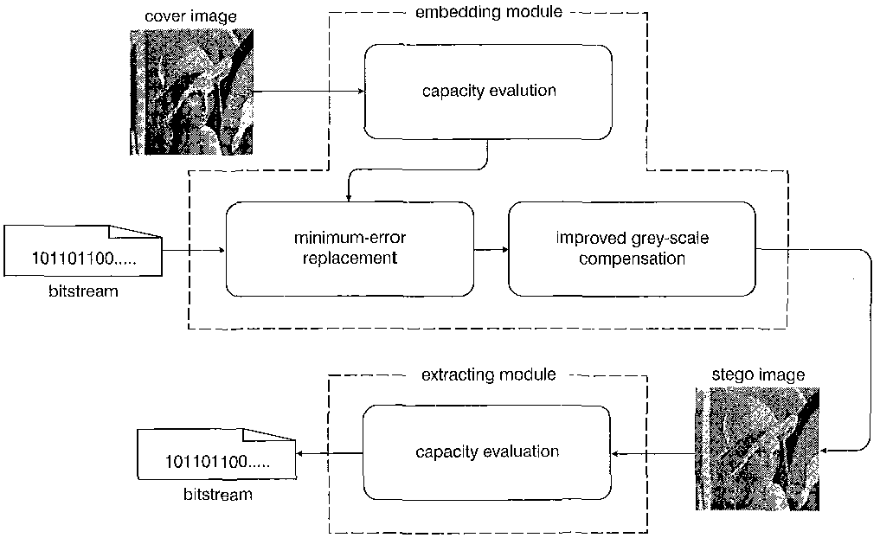

Fig. 3 *Block diagram of proposed image embedding and extracting modules*

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 grey-scale compensation (IGSC).

First, for each pixel, the CE component uses the grey-scale variation of neighbouring pixels and its intensity to evaluate its embedding capacity. Note that the local characteristics should not be changed after embedding the message, then the same characteristics can be used to evaluate the embedded capacity in the extracting module. Then the MER component finds a replacing grey scale, based on the following two requirements: The LSBs should be identical to the embedded message bits; and the grey scale, which meets the first requirement, with minimum error to the original one should be taken. Using this grey 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 lead to more messages in the cover image.

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

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

For penetrating an image we decompose the grey scale of each pixel into eight bits. The plane formed by the same bit of each pixel in a grey-scale image is called a bit-plane. Fig. 4 shows the eight corresponding bit-planes of Fig. 1*a*. Observing these bit-planes, we see that some areas in the six least-significant bit-planes are bestrewn with noise. The HVS is insensitive to the value change in these areas. Thus

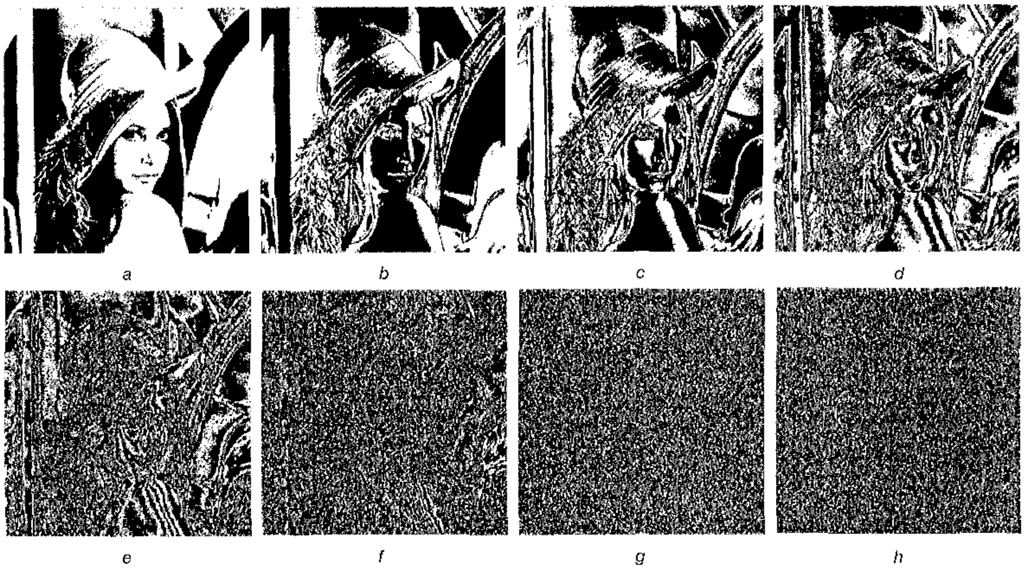

Fig. 4 *Eight bit-planes of 'Lena'* (512 × 512)
a–h Eight bit-planes from most-significant to least-significant bit

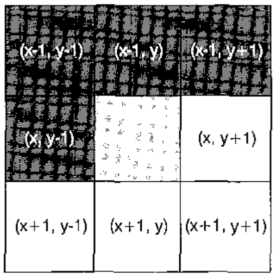

Fig. 5 *Eight neighbours of pixel p at co-ordinates (x, y)*

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 grey values among the neighbouring 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 grey-value variation of neighbouring 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 grey scale of one pixel p at co-ordinates (x, y) is denoted by $f(x, y)$, the eight-neighbours of p are shown in Fig. 5. For p, $f(x, y)$ will be modified according to its embedding capacity, which depends on its grey scale and the grey-scale variation of the upper and left neighbours (see shaded pixels in Fig. 5). The advantage of using the upper and left neighbours to estimate the embedding capacity is that when or after the current pixel is processed, the gray scales of these upper and left neighbours will be never changed. Therefore the embedding module and extracting module are synchronous when estimating the embedding capacity of each pixel. Let

$$\begin{aligned} Max(x,y) &= \max\{f(x-1,y-1), f(x-1,y),\\ &\qquad f(x-1,y+1), f(x,y-1)\}\\ Min(x,y) &= \min\{f(x-1,y-1), f(x-1,y),\\ &\qquad f(x-1,y+1), f(x,y-1)\}\\ D(x,y) &= Max(x,y) - Min(x,y) \end{aligned}$$

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 grey scale is, the more change of the grey scale could be tolerated. Thus, the embedding capacity should be limited by the grey scale of current pixel. Here, an upper bound for embedding capacity at pixel (x, y) is defined as

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

Note that t is set to be 191, the reason is as follows. In the embedding module, the original grey scale is used to find $U(x, y)$ but in the extracting module, the grey scale has been changed. To make $U(x, y)$ consistent the original grey 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, described later, the lower bound for embedding capacity could be set as four bits. So the embedding capacity $K(x, y)$ of each pixel can be computed by the following expression:

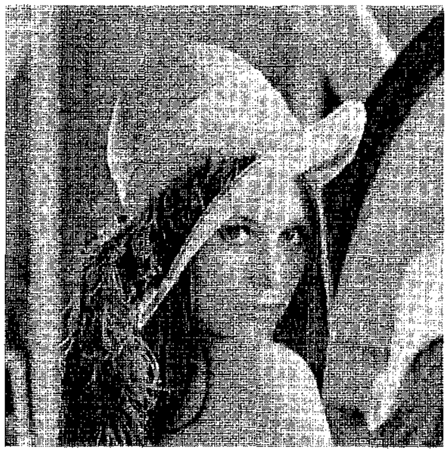

Fig. 6 *Embedding result of applying CE component of 'Lena'*

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

Fig. 6 shows the embedding result of applying the CE component on 'Lena.' The average embedding capacity of each pixel is 4.06 bits per pixel, and the RMS and PSNR values are 6.59 and 31.75 dB, respectively. Some artifacts exist on the smooth regions of the image; these will be solved by the IGSC component.

2.1.2 MER component: In general, eight bits are used to represent the intensity of each pixel in a grey-scale image. If we want to embed k ($k < 8$) bits in a pixel, 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 grey scales, there are $2^{(8-k)}$ grey levels whose k LSBs are identical to the k embedded bits. To achieve the highest quality we can take the most similar grey scale among these $2^{(8-k)}$ grey scales to replace the original one. To reach the aim, a simple way to search the closest grey scale is provided here.

Let $f(x, y)$ be the original grey scale $g(x, y)$ be the gray scale obtained by embedding k LSBs directly, and $g'(x, y)$ be the grey scale obtained by changing the value of the $(k+1)$th LSB of $g(x, y)$. The minimum-error grey 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

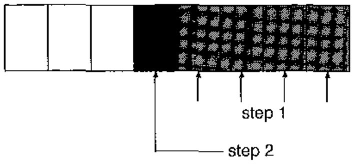

Fig. 7 *Two steps of MER component*
Step 1: Embed k ($k=4$) message bits in k LSBs
Step 2: Adjust the value of the $(k+1)$th LSB

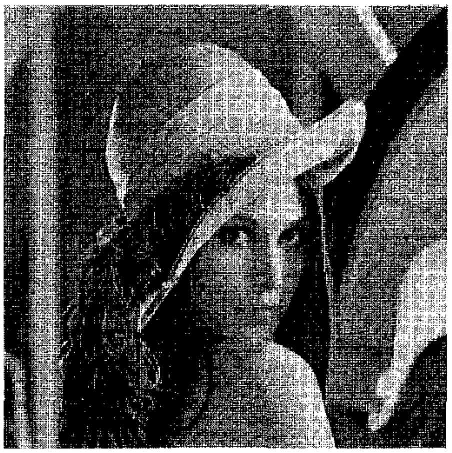

Fig. 8 *Result of applying MER component on Fig. 6*

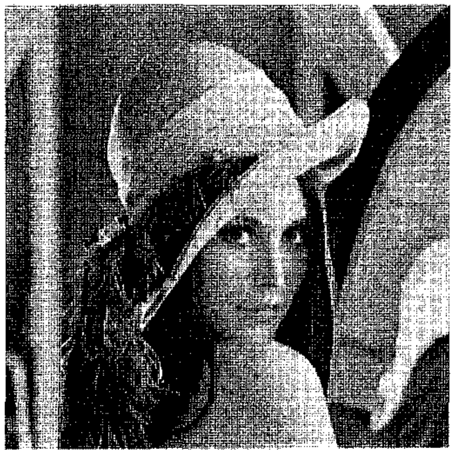

Fig. 9 *Result of applying IGSC component on Fig. 8*

increased. The RMS value is reduced from 6.59 to 5.76, and the PSNR value is increased from 31.75 to 32.92 dB. However, some unwanted artifacts still appear in the smooth areas, such as the shoulder of Lena. In the following Section we propose a method to address this problem.

2.1.3 IGSC component: Embedding too many bits in the smooth area of an image will cause the false contouring (see the face in Figs. 6 and 8). The same phenomenon also appears in a quantised image, this is because 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 grey-scale (IGS) quantisation [16]. This concept is similar to the error diffusion method that is commonly used in conversion of true colour images to palette-based colour 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 neighbouring pixels (the white neighbouring pixels shown in Fig. 5). Let $e(x, y)$ denote the embedding error of pixel p at co-ordinates (x, y), these four bottom-right neighbouring 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 per pixel and the RMS and PSNR values are 5.07 and 34.03 dB, respectively. Note that no false contouring appears in Fig. 9.

2.2 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 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 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, then k random numbers will be generated for the least significant k bits, respectively. An example, shown in Fig. 10 and Table 1, is given to illustrate the bitwise method. Suppose that the embedding ratio T is 0.5, and the embedding capacity k is four for some pixel, then four random numbers will be generated.

Table 1: Grey scales for the example in Fig. 10

Grey scale		
Decimal	Binary	Search order
137	(0 × 10001001)	
138	(0 × 10001010)	
139	(0 × 10001011)	5 new grey-scale
140	(0 × 10001100)	3
141	(0 × 10001101)	1
142	(0 × 10001110)	0 original grey-scale
143	(0 × 10001111)	2
144	(0 × 10010000)	4
145	(0 × 10010001)	
146	(0 × 10010010)	

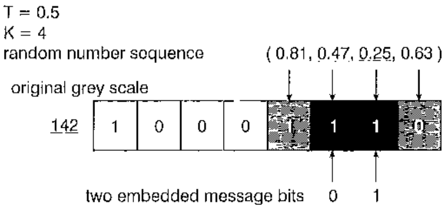

Fig. 10 *Example of applying bitwise method*

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 grey scale is 142 (0 × 10001110) and the two embedded bits are 0 and 1. Since the grey scale 139 (0 × 10001011) is the closest one with the third and the second LSB being 0 and 1, respectively. The original grey scale 142 will be replaced by 139. Finally, the embedding error will be uniformly spread to neighbouring pixels via the IGSC component.

Table 2: Experimental results of 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

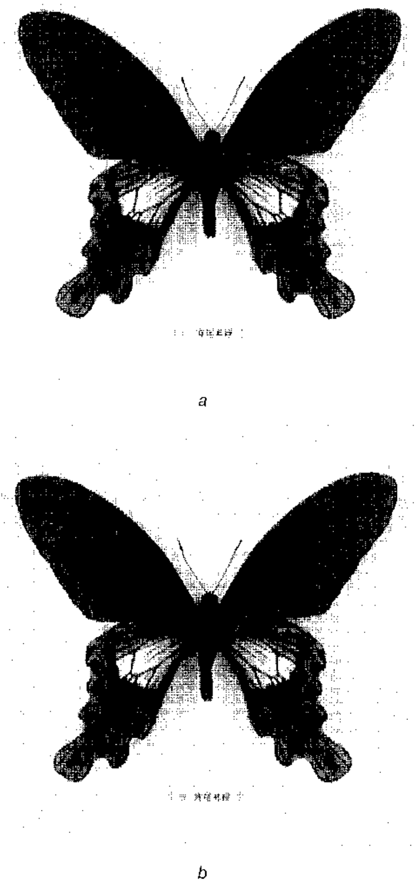

Fig. 11 *Experimental result of proposed method*
a Cover image entitled 'Maraho' (512 × 512)
b Stego-image of proposed method (RMS = 6.34, SNR = 32.09 dB)

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 RMS error and the peak signal-to-noise ratio (PSNR), are used to evaluate the performance of our method and that of the four-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 four bits. Furthermore, the RMS and PSNR are listed in Table 2. From Table 2 one can see that the performance of the proposed method is better than that of the four-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 grey-scale image entitled 'Maraho', obtained from a scanner; its background is near-white colour. Fig. 11*b* shows the result of embedding the maximum capacity (1043351 bits, 4.011 bits per pixel) in Fig. 11*a*. The RMS and PSNR measure for Fig. 11*b* are 6.34 and 32.09 dB, respectively. From the viewpoint of human eyes these two images are almost indistinguishable.

From the description in the previous Section, 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

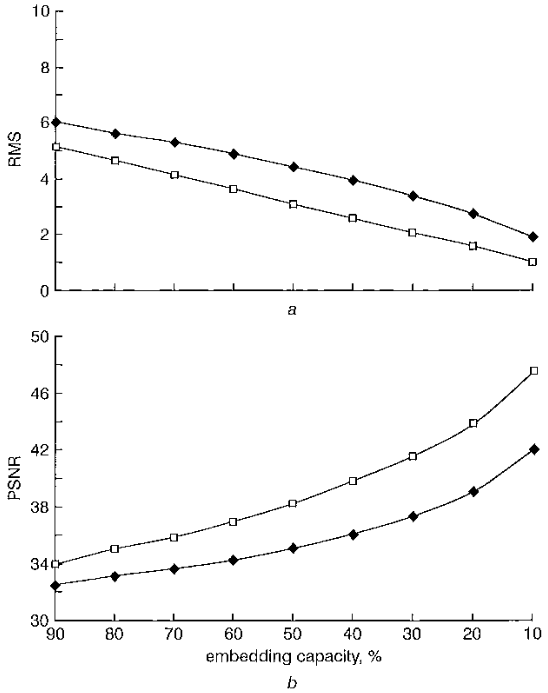

Fig. 12 *Comparison of pixelwise and bitwise methods*
-◆- pixelwise
-□- bitwise
a RMS
b PSNR

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. The fidelity of all stego-images through applying these two methods is acceptable to human eyes. Therefore the sender can choose different methods 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

We have introduced an image steganographic model and have proposed a new high-capacity embedding/extracting module that is based on the variable-size LSB insertion. In the embedding part, based on the contrast and luminance property, we used three components to maximise the capacity, minimise the embedding error and eliminate the false contours. Using the proposed method we embeded at least four message bits in each pixel while maintaining the imperceptibility 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. Using only 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, ROC, under contract NSC 88-2213-E-009-063.

7 References

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

2 KAHN, D.: 'The history of steganography'. Proceedings of the first workshop on *Information hiding*, 30 May–1 June 1996, Cambridge, UK, pp. 1–5 (*Lect. Notes Comput. Sci.* (Springer-Verlag), **1174**)

3 PFITZMANN, B.: 'Information hiding terminology'. Proceedings of the first workshop on *Information hiding*, 30 May–1 June 1996, Cambridge, UK, pp. 347–350, (*Lecture Notes Comput. Sci.* (Springer-Verlag), **1174**)

4 ANDERSON, R.J., and PETITCOLAS, F.A.P.: 'On the limits of steganography', *IEEE J. Sel. Areas Commun.*, 1998, **16**, (4), pp. 474–481

5 KUTTER, M., JORDAN, F., and BOSSEN, F.: 'Digital signature of color images using amplitude modulation', *J. Electron. 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'. Presented at the 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', *Proc. IEEE*, 1998, **86**, (6), pp. 1064–1087

9 COX, I.J., KILIAN, J., LEIGHTON, T., and SHAMOON, T.: 'Secure spread spectrum watermaking for multimedia', *IEEE Trans. Image Process.*, 1997, **6**, (12), pp. 1673–1687

10 WOLFGANG, R.B., PODILCHUK, C.I., and DELP, E.J.: 'Perceptual water-marks for digital images and video', *Proc. SPIE. - Int. Soc. Opt. Eng.*, 1999, **3657**, pp. 44–51

11 KOCH, E., and ZHAO, J.: 'Towards robust and hidden image copyright labeling'. Proceedings of IEEE workshop on *Nonlinear signal and image processing*, 20–22 June 1995, Neos Marmaras, Halkidiki, Greece, pp. 452–455

12 XIA, X.-G., BONGELET, C.G., and ARCE, G.R.: 'A multiresolution water-mark for digital images'. Proceedings of IEEE international conference on *Image Processing*, 26–29 Oct 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 conference on *Image processing, image quality, image capture systems* PICS '99, 25–28 April 1999, Savannah, Georgia, pp. 274–278

14 JOHNSON, N.F., and JAJODIA, S.: 'Steganography: seeing the unseen', *IEEE Comput.*, 1998, 26–34

15 BENDER, W., GRUHL, D., MORIMOTO, N., and LU, A.: 'Techniques for data hiding', *IBM syst. J.*, 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'. SID international symposium digest of technical papers, 1992, pp. 377–380

18 YEUNG, M.M., and MINTZER, F.C.: 'Invisible watermarking for image verification', *J. Electron. Imaging*, 1998, **7**, (3), pp. 578–591