



PERGAMON

Pattern Recognition 34 (2001) 671–683

PATTERN RECOGNITION

THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Image hiding by optimal LSB substitution and genetic algorithm

Ran-Zan Wang^a, Chi-Fang Lin^{b,*}, Ja-Chen Lin^a

^aDepartment of Computer and Information Science, National Chiao Tung University, Hsinchu, 300, Taiwan, Republic of China

^bInstitute of Electrical Engineering and Computer Engineering and Science, Yuan-Ze University, 135 Yuan-Tung Road, Chungli, Taoyuan, 320, Taiwan, Republic of China

Received 29 September 1998; received in revised form 23 April 1999; accepted 24 November 1999

Abstract

In this paper, we develop a novel method to embed important data in the host image so that the interceptors will not notice about the existence of the data. The basic concept of the proposed method is by simple LSB substitution. To prevent illicit access of data and increase the system performance, the method of randomized process and the optimal LSB substitution are proposed. The effectiveness of the optimal LSB substitution in the worst case is proved. We also develop a genetic algorithm to solve the problem of hiding important data in the rightmost k LSBs of the host image, which may involve a huge computation time to find the optimal result when k is large. Finally, based on the concept of perceptual modeling, an improved hiding technique is developed to obtain a high-quality embedding result. Experimental results will reveal that the quality of the embedding result will not be seriously degraded, and is near optimal by utilizing the proposed method. © 2001 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Image encryption; Image hiding; LSB substitution; Genetic algorithm

1. Introduction

The popularity of the Internet offers a great convenience to the transmission of a large amount of data via networks. However, it also increases the risk of illegal access and unauthorized tampering with content while transmitting data. To safely transmit data through the Internet networks, some mechanisms must be provided to guard important data against illegal interception. In the past decade, numerous techniques were proposed to provide the environment for important data transmission. One of the most important techniques is by data encryption [1–4], in which the data are protected from illicit access by utilizing a particular cipher algorithm.

The data are transformed by the cipher algorithm into a ciphertext that looks like a stream of meaningless codes, and then sent to the receiver via the network. Upon receiving the ciphertext, the receiver can decrypt it using a cryptography key obtained from the transmitter to recover its original form. Another approach to transmitting data safely, however, whose goal is quite different from that of data encryption, is by data hiding [5]. Data hiding is the technique of embedding data into audio or video media called host signal such as music, images, and movies. It can be utilized in a wide variety of applications such as ownership identification, tamper proofing, caption, and secret data transmission. Unlike the goal of data encryption that prevents the data from illicit access and modification, the aim of data hiding is to make the data inaudible or invisible to the grabbers. That is, the grabbers will not notice the existence of the embedded data, though they may listen very hard to or look very carefully at the media where the important data are hidden.

*Corresponding author. Tel.: + 886-3463-8800; fax: + 886-3463-9357.

E-mail address: cscflin@saturn.yzu.edu.tw (C.-F. Lin).

In general, according to the applications to which the techniques of data hiding are applied, the types of data hiding can be classified into the following two categories, i.e., watermarking and image hiding. The first category is used to embed a distinguishable symbol, e.g., a signature or a trademark, into host signals to authorize the ownership of the signals. The size of the symbol in this category is usually small, ranging from one bit to thousands of bits to represent the symbol. The goal of watermarking focuses on the survival of the symbol after the host signal is manipulated by degrading operations such as filtering, resampling, lossy compression, etc. The second category that is the main subject studied in this paper is used to store and transmit important data, whose size tends to be very large, usually exceeds a quarter of the host signal. Instead of aiming at achieving high resistance to image manipulation, the goal of image hiding focuses on making the data invisible to viewers and obtaining high embedded data rate. A brief literature review of the above two categories of data-hiding techniques is given as follows.

Cox et al. [6] proposed a watermark that consists of k independently and identically distributed random numbers drawn from a standard normal distribution. To make the watermark robust to signal processing operations, they advocated that the watermark should be injected into the perceptually most significant spectral component of the host media. Their idea is analogous to spread spectrum communication systems [7], in which the frequency domain of the host media is viewed as a communication channel, and the watermark is viewed as a signal that is transmitted through it. The method is suitable to hide data with a tiny size into the host media. However, if it is desired to hide data with a huge size, the quality of the embedding result is not acceptable. Bender et al. [5] described two data hiding techniques in still images. The first is by a statistical method called patchwork, which embeds in a host image a specific statistic that has a Gaussian distribution. The method randomly chooses from the host image a pair of patches, and a procedure modifying the brightness of the patches is executed repeatedly to obtain a unique statistic that is used to indicate the presence or absence of a signature. The method is easy to implement. However, it yields very low embedded data rate, usually a single-bit signature per image, thus, limits its usefulness to low-bits applications such as watermarking. The second method is called “texture block coding”, which was implemented by hiding data in an area of a picture having similar texture of the data. This method does not work well on images that do not have large continuous texture areas such as text images. Koch and Zhao [8] proposed a labeling-based approach for copyright protection. They adopted a simple pulsing method to embed data at the specified locations that appear to be a natural part of the image, and utilized pulses with sufficient noise margins to pro-

tect hidden data from common image processing. The restriction of hiding data in specific positions also limits its application to data hiding low bit rate. Hsu and Wu [9] proposed a DCT-based data-hiding technique that can be utilized to hide signature in images. To effect a compromise between hiding the signature invisibility and robustness to image processing, they embedded the signature in the middle frequency band of the image. However, similar to all other frequency-domain-based image hiding techniques, the hiding capacity in this technique is also limited. In the past decade, there are numerous least-significant-bit-based (LSB-based) data hiding techniques, which find some special locations (pixels) in the host media, then embed data into the LSB of those pixels. Typical approaches include Adelson [10], Turner [11], and van Schyndel et al. [12].

The methods listed above devote themselves to make the embedded data difficult to be removed or degraded when the host image is processed by image operations. However, the hiding capacities in those methods are limited. In this paper, we develop a method for hiding important data with a large size. The proposed method is based on LSB substitution, i.e., the data is embedded in the LSB positions of the host image. To prevent illicit access of the data and obtain a better embedding result, a randomized embedding process and an optimal LSB substitution method as well as data hiding by human perceptual models are presented. It is proved in this study that in the worst case the embedding result obtained by the optimal LSB substitution is better than that obtained by the simple substitution method (i.e., the substitution without the optimal selection). To achieve the optimal result, the performance of each possible substitution must be evaluated, which may involve a large amount of computation time, and make the proposed method impractical. So we develop a genetic algorithm to solve the problem. The advantages of the proposed method include the following: (1) the size of the embedded data can be very large, and the quality of the embedding result is not degraded significantly; (2) the randomized process can make the embedded data meaningless to the grabbers, so the data can be protected well; and (3) the embedding result utilizing the proposed genetic algorithm is near optimal, and the processing time is acceptable. The experimental results will be given to reveal the advantages of the proposed method.

The rest of this paper is organized as follows. The concept of image hiding by LSB substitution is presented in Section 2. In Section 3, the method of evaluating the performance for each LSB substitution is presented first. The effectiveness of the optimal substitution is proved, and a technique of image hiding by genetic algorithm is developed. Finally, an improved hiding technique by perceptual modeling is presented. Experimental results are given in Section 4, and a brief conclusion is made finally in Section 5.

2. Image hiding by LSB substitution

Before presenting the basic concept of image hiding by LSB substitution, the two constraints that are enforced in the proposed method are listed as follows.

(1) The quality of the embedding result cannot be degraded so significantly that the grabbers may notice the existence of the important data.

(2) The size of the important data can be large, ranging from a quarter to a half-size of the host image, and must be recovered from the host image without any distortion in quality.

There is a trade-off between the size of the embedded data and the quality of the host image after embedding, i.e., the more data to be hidden, the more degradation in quality of the host image will be obtained. A trade-off also exists between the data size and the immunity to image manipulations like filtering, cropping, resampling, etc. If the data size is very small, like a digital watermark, a high level of resistance to various forms of image manipulations is required. However, if the size of the data are very large, a low level of resistance to image modifications can be expected. As mentioned previously, the main objective of this study is to provide a method to hide important data with large size, therefore, the immunity to image operations is not considered in this paper. The proposed method is based on LSB substitution, and the concept is presented as follows.

Suppose we want to hide image E in image H , both are n -bits gray-scale images. To meet the requirements of the above constraints, a simple method is to hide E into the positions of the least significant bit (LSB) of H . More specifically, we can first retrieve the rightmost k LSBs from each pixel of H to form conceptually a k -bit gray-scale image R , named the residual image that has the same size of E (an assumption is made here that the size of E is identical to the power of 2, and the assumption is not required in Section 3.4). Next, convert E to a k -bit image by decomposing each pixel of E into several small k -bit units, and treating each unit as a single pixel. The resultant is denoted by E' . Finally, by replacing R by E' , the embedding result Z is obtained accordingly as depicted in Fig. 1.

The above method is simple, however, it may cause problems that are stated as follows. First, hiding data in the LSB of the host image will make it easy to be detected for grabbers by simply analyzing the bit-plane information of the embedding result. Second, the quality of the host image after embedding by the above method may not be acceptable, i.e., the host image may degrade seriously after embedding. For an illustrative example, assume that E and H in the above case are all 8-bit images with 256 gray scales, and E is embedded into the right-

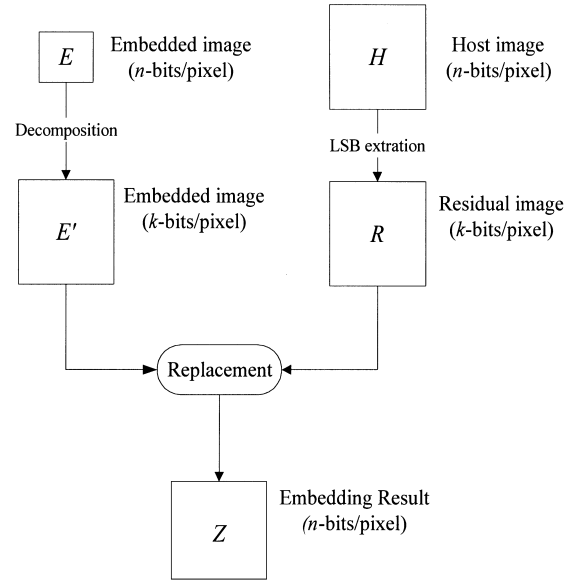


Fig. 1. The flowchart of embedding by simple LSB substitution.

most 4 LSBs of H . Theoretically, the $PSNR$ between Z and H can be obtained by

$$PSNR = 10 \times \log((2^n - 1)^2 / MSE) \\ = 10 \times \log(255^2 / MSE), \quad (1)$$

where n is the number of bits per pixel, and MSE is the mean square error defined as follows. Assume that z_i and h_i represent the pixel gray values of Z and H , respectively, then MSE is defined by

$$MSE = \frac{1}{m} \sum_{i=1}^m (z_i - h_i)^2, \quad (2)$$

where m is the image size of Z and H . It can be verified that in the above case the worst $PSNR$ calculated by Eq. (1) is identical to 24.61, which means that the host image is degraded drastically because the obtained $PSNR$ is very low. To solve the above two problems, a randomized embedding process and an optimal LSB substitution method are presented as follows.

A monoalphabetic substitution cipher algorithm [4] is adopted here to accomplish the randomized embedding process. Before replacing R by E' , the pixel location of E' is randomized by a bijection (i.e., one-to-one and onto) mapping function presented as follows. Assume that pixel locations in E' are numbered sequentially from 0 to $s - 1$, where s is the image size of E' . Pixel location x of E' is transformed to a new location $f(x)$ by

$$f(x) = (k_0 + k_1 \times x) \bmod s$$

and

$$\gcd(k_1, s) = 1, \quad (3)$$

where k_0 and k_1 are two constants which are used as keys, and $\gcd(\cdot)$ means the greatest common divisor. The above randomization process will transform E' into a meaningless picture, and prevent it from being detected unless the grabbers understand the cipher process and own the two keys (i.e., k_0 and k_1 .) The image obtained by applying the above procedure is denoted by E'' .

To achieve a better embedding result, a method of optimal substitution is presented in the following. A substitution matrix $S = \{s_{ij}\}$ is defined first in this study by

$$s_{ij} = \begin{cases} 1 & \text{gray value } i \text{ is replaced by gray value } j, \\ 0 & \text{do nothing.} \end{cases} \quad (4)$$

It is noted that exactly one element in each row and one element in each column of S can have a value of 1. So there are a total of $(2^k)!$ possible substitutions denoted by $S_1, S_2, \dots, S_{(2^k)!}$ that satisfy the above constraint for a k -bit image. According to each substitution matrix, E'' is further transformed into E''' , and the embedding result can be obtained by replacing R by E''' . To obtain the optimal embedding result, the simplest method is to calculate the $PSNR$ for each substitution, and select the one having the maximum $PSNR$ as the optimal result. For example, assume that the 2 LSBs (i.e., $k = 2$) are used for hiding data. Both of the residual image R and the disturbed image E'' mentioned above are all two-bit pictures. Hence, a total of $(2^2)! = 24$ substitution matrices can be applied to substitute image E'' , and 24 $PSNR$ calculations are thus required. Among them, the matrix having the largest $PSNR$ is selected as the best one. The flowchart of the optimal substitution is depicted in Fig. 2, in which E^* is used to represent the result obtained by applying the optimal substitution matrix to E'' .

3. Proposed methods

In Section 3.1, we present the method to calculate the $PSNR$ for each substitution matrix discussed in the last section. In Section 3.2, we show that the worst MSE of the optimal substitution is identical to half of the worst MSE of the simple substitution. In Section 3.3, a genetic algorithm is presented to solve the problem of hiding data in the k LSBs of the host image when k is large. Finally, in Section 3.4, a hiding technique by perceptual modeling is developed to improve the quality of embedding result.

3.1. PSNR calculation

Define an occurrence matrix $O = \{o_{jk}\}$ whose element o_{jk} represents the number of occurrences of gray value j in E''' and gray value k in R , when R is replaced by E''' .

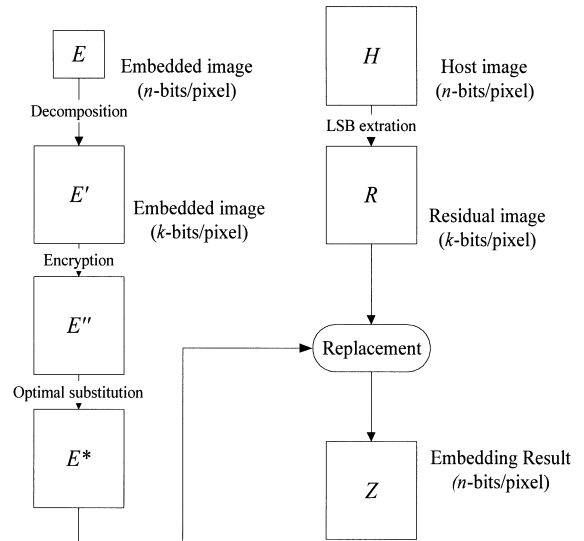


Fig. 2. The flowchart of embedding by the optimal LSB substitution.

Also define a weight matrix $W = \{w_{jk}\}$ whose element w_{jk} represents the square difference between the gray values j and k , i.e., $w_{jk} = (j - k)^2$. The cost of replacing pixels of R by pixels of E''' having gray value j is defined by

$$c_j = \sum_{k=0}^{2^k-1} o_{jk} w_{jk}. \quad (5)$$

The cost of embedding entire E''' in R is thus defined by

$$\begin{aligned} C &= \sum_{j=0}^{2^k-1} c_j \\ &= \sum_{j=0}^{2^k-1} \sum_{k=0}^{2^k-1} o_{jk} w_{jk}. \end{aligned} \quad (6)$$

If a substitution matrix S is given, the embedding cost corresponding to S is obtained by modifying the above equation as

$$C_S = \sum_{i=0}^{2^k-1} \sum_{j=0}^{2^k-1} \sum_{k=0}^{2^k-1} s_{ij} o_{jk} w_{jk} \quad (7)$$

and the MSE corresponding to S is given by

$$MSE_S = \frac{1}{m} C_S. \quad (8)$$

It should be noted that the value calculated by Eq. (8) is the MSE between images R and E''' , which is also identical to the MSE between images H and Z . By substituting the above equation into Eq. (1), the $PSNR$ of S can thus be obtained by

$$PSNR_S = 10 \times \log(m \times (2^n - 1)^2 / C_S). \quad (9)$$

3.2. Performance of the optimal substitution

Let MSE_{HZ} and $WMSE_{HZ}$ denote the MSE and the worst MSE between host image H and embedding result Z obtained by simple LSB substitution, respectively. Also let MSE_{HZ^*} and $WMSE_{HZ^*}$ denote the MSE and the worst MSE between H and Z obtained by optimal LSB substitution, respectively. In this section, we will show the following equation is true:

$$WMSE_{HZ^*} = \frac{1}{2} WMSE_{HZ}. \quad (10)$$

Before presenting the proof step by step, let $\bar{Z}$ denote the complement image of Z , i.e., the relation between pixel $\bar{z}$ in $\bar{Z}$ and the corresponding pixel z in Z is given by (both H and Z are all n -bit images with 2^n gray values)

$$\bar{z} + z = 2^n - 1. \quad (11)$$

It is easy to figure out that $WMSE_{HZ}$ is identical to $(2^n - 1)^2$. We now derive equations to prove the correctness of Eq. (10). As mentioned in the last section, since MSE_{HZ^*} is the optimum one as compared with that of the other substitutions, the following two equations are true:

$$MSE_{HZ^*} \leq MSE_{HZ} \quad (12)$$

and

$$MSE_{HZ^*} \leq MSE_{HZ}. \quad (13)$$

So the following equation can be obtained as

$$\begin{aligned} MSE_{HZ^*} &\leq \frac{1}{2} (MSE_{HZ} + MSE_{HZ}) \\ &= \frac{1}{2m} \sum_{i=1}^m ((h_i - z_i)^2 + (h_i - \bar{z}_i)^2) \\ &= \frac{1}{2m} \sum_{i=1}^m ((h_i - z_i)^2 + ((2^n - 1) - z_i - h_i)^2), \end{aligned} \quad (14)$$

where m is the image size, h_i , z_i , and $\bar{z}_i$ are the pixels of images H , Z and $\bar{Z}$, respectively. Let $|h_i - z_i| = \alpha_i$, so we have $h_i = z_i \pm \alpha_i$.

Case 1: When $h_i = z_i + \alpha_i$, rewrite Eq. (14) and obtain

$$\begin{aligned} MSE_{HZ^*} &\leq \frac{1}{2m} \sum_{i=1}^m (\alpha_i^2 + ((2^n - 1) - z_i - h_i)^2) \\ &= \frac{1}{2m} \sum_{i=1}^m (\alpha_i^2 + ((2^n - 1) - 2z_i - \alpha_i)^2). \end{aligned} \quad (15)$$

Because the following equation is true:

$$\begin{aligned} ((2^n - 1) - \alpha_i)^2 - ((2^n - 1) - 2z_i - \alpha_i)^2 \\ = ((2^n - 1) - \alpha_i + (2^n - 1) - 2z_i - \alpha_i) \\ \times ((2^n - 1) - \alpha_i - (2^n - 1) + 2z_i + \alpha_i) \end{aligned}$$

$$\begin{aligned} &= 4z_i((2^n - 1) - (\alpha_i + z_i)) \\ &= 4z_i((2^n - 1) - h_i) \\ &\geq 0, \end{aligned}$$

it can be concluded that

$$((2^n - 1) - 2z_i - \alpha_i)^2 \leq ((2^n - 1) - \alpha_i)^2. \quad (16)$$

The following equation can be derived from Eqs. (15) and (16).

$$\begin{aligned} MSE_{HZ^*} &\leq \frac{1}{2m} \sum_{i=1}^m (\alpha_i^2 + ((2^n - 1) - 2z_i - \alpha_i)^2) \\ &\leq \frac{1}{2m} \sum_{i=1}^m (\alpha_i^2 + ((2^n - 1) - \alpha_i)^2) \\ &\leq \frac{1}{2m} \sum_{i=1}^m (2^n - 1)^2 \\ &\leq \frac{1}{2} (2^n - 1)^2. \end{aligned} \quad (17)$$

Case 2: When $h_i = z_i - \alpha_i$, the proof is similar to the case discussed above. First, rewrite Eq. (14) and obtain the following equation:

$$\begin{aligned} MSE_{HZ^*} &\leq \frac{1}{2m} \sum_{i=1}^m (\alpha_i^2 + ((2^n - 1) - z_i - h_i)^2) \\ &= \frac{1}{2m} \sum_{i=1}^m (\alpha_i^2 + ((2^n - 1) - 2z_i - \alpha_i)^2). \end{aligned} \quad (18)$$

Because the following equation is true,

$$\begin{aligned} ((2^n - 1) - \alpha_i)^2 - ((2^n - 1) - 2z_i - \alpha_i)^2 \\ = ((2^n - 1) - \alpha_i + (2^n - 1) - 2z_i - \alpha_i) \\ \times ((2^n - 1) - \alpha_i - (2^n - 1) + 2z_i - \alpha_i) \\ = 4((2^n - 1) - z_i)(z_i - \alpha_i) \\ = 4h_i((2^n - 1) - z_i) \\ \geq 0, \end{aligned}$$

we can conclude that

$$((2^n - 1) - 2z_i - \alpha_i)^2 \leq ((2^n - 1) - \alpha_i)^2. \quad (19)$$

The following equation can be derived from Eqs. (18) and (19):

$$\begin{aligned} MSE_{HZ^*} &\leq \frac{1}{2m} \sum_{i=1}^m (\alpha_i^2 + ((2^n - 1) - 2z_i - \alpha_i)^2) \\ &\leq \frac{1}{2m} \sum_{i=1}^m (\alpha_i^2 + ((2^n - 1) - \alpha_i)^2) \\ &\leq \frac{1}{2m} \sum_{i=1}^m (2^n - 1)^2 \\ &\leq \frac{1}{2} (2^n - 1)^2. \end{aligned} \quad (20)$$

According to the results implied by Eqs. (17) and (20), the correctness of Eq. (10) is proved.

3.3. Image hiding by genetic algorithm

The image hiding technique discussed in Section 3.1 works well if the secret data is embedded in the k LSBs of the host image, where k is less than or equal to 3. However, it works poorly if k is greater than 3 because the number of possible substitutions will grow exponentially as k increases. For example, assume that $k = 4$, there are a total of $(2^4)! = 16!$ possible substitutions (about 20,000 billions of substitutions) that can be utilized to embed data. Hence, it is very impractical and time consuming for us to compute the *PSNR* for each substitution. A genetic algorithm is thus developed in the paper to solve the above problem, and is presented as follows.

The genetic algorithm (GA) was introduced by Holland [13] in his seminal work “adaptation in natural and artificial systems”, and applied in diverse fields [14] such as image analysis, clustering, and scheduling in the search for optimal states of systems or functions. GA can be regarded as a randomized search procedure that is commonly used to solve the optimization problems. A solution in the problem domain corresponds to an individual in a GA, which is represented by a chromosome containing many genes. An objective function called the fitness function is used to evaluate the quality of each chromosome. In general, GA is mainly comprised of the following three operators, namely, (1) reproduction, (2) crossover, and (3) mutation. Reproduction retains the current chromosome’s genes, crossover assembles existing genes into new combinations, and mutation produces new genes. The procedure of GA is started by specifying an initial population in the first generation, and during each next generation, the individuals in the population undergo the activities of reproduction, crossover and mutation, to produce their offspring. Then a fitness function is applied to each offspring to determine its quality. The individuals with high quality will survive and form the population of the next generation. The process will repeat for many times until a predefined requirement is satisfied, or a constant number of iterations is exceeded.

In this study, a chromosome G in GA consisting of 2^k genes is described by a substitution matrix S defined in Section 2 as follows:

$$G = g_0 g_1 \dots g_{2^k-1}, \quad (21)$$

where g_0 represents the position of row 0 in S that has a value of 1, g_1 represents the position of row 1 in S that has a value of 1, and so on. A simple example is illustrated in Fig. 3. Note that the value of each of the genes is a constant ranging through 0 to $2^k - 1$. A chromosome G is said to be valid in this study if for any gene pair of G ,

	0	1	2	3
0	0	1	0	0
1	0	0	0	1
2	1	0	0	0
3	0	0	1	0

G	1	3	0	2
-----	---	---	---	---

Fig. 3. An example of S and the corresponding G for a 2-bit image with four gray values.

say, (g_i, g_j) , where $i \neq j$, implies $g_i \neq g_j$. The genetic operators of crossover and mutation are defined as follows.

Crossover. Given two chromosomes $G_1 = p_0 p_1 \dots p_{2^k-1}$ and $G_2 = q_0 q_1 \dots q_{2^k-1}$, the crossover of G_1 and G_2 is conducted by performing the following two steps: the step of fixed-point crossover and the step of chromosome validation. The first step is to split G_1 and G_2 into left-hand side and right-hand side parts with equal sizes, and replace their right-hand side parts with each other to get the new offspring $G'_1 = p_0 p_1 \dots p_{2^{k-1}-1} q_{2^{k-1}} \dots q_{2^k-1}$ and $G'_2 = q_0 q_1 \dots q_{2^{k-1}-1} p_{2^{k-1}} \dots p_{2^k-1}$, respectively. Although chromosomes G'_1 and G'_2 obtained by the above step may not be valid (i.e., the value of the genes may not be unique in each chromosome), they need to be adjusted so that the validation of chromosome can be achieved. Before presenting the adjusting method, 2^k flags indexed through 0 to $2^k - 1$ corresponding to the 2^k genes are prepared for each chromosome. The flags are used to check whether the genes in a chromosome have unique values or not. Initially, the flags are all marked as “off”. The method is comprised of two passes. In the first pass, the genes of the chromosome are scanned sequentially from left to right. If the value of the current gene is k , check the status of the corresponding flag (i.e., the k -indexed flag.) If the flag is marked as “off”, it is remarked as “on”; otherwise, the value of the current gene is set to a negative value, e.g., -1 . The process repeats for the next gene until the last one is checked. In the second pass, the genes of the chromosome are sequentially scanned again from left to right. If the value of the current gene is detected as -1 , its value is replaced by the index number of the flag scanned from left to right that is the first occurrence of “off” flag. The “off” flag is then remarked as “on”, and the process repeats until all the genes of

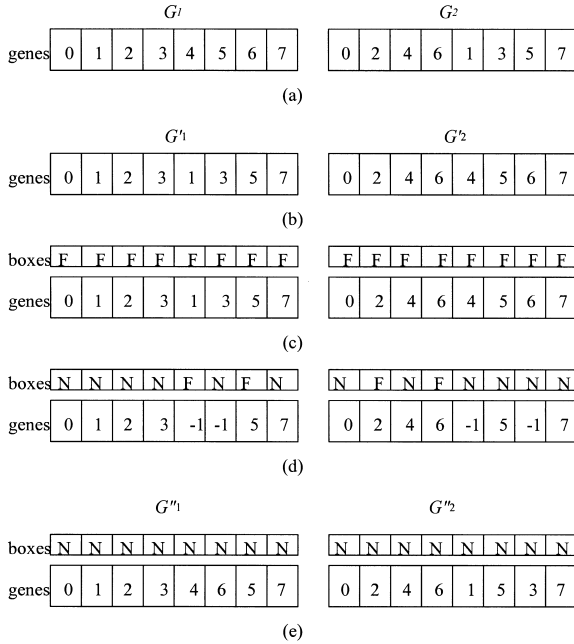


Fig. 4. An illustrative example of the crossover operation: (a) two chromosomes G_1 and G_2 , (b) the offspring of G_1 and G_2 obtained by the fixed-point crossover, (c) the initial values of the flags (N: on; F: off), (d) the status after the first pass of adjustment, (e) the status after the second pass of adjustment.

a chromosome are scanned. An example is given as follows. Assume that $G_1 = 01234567$ and $G_2 = 02461357$ are two chromosomes. The offspring of G_1 and G_2 created by the first step are $G'_1 = 01231357$ and $G'_2 = 02464567$, respectively, which are all invalid. By adopting the adjusting method, chromosomes $G''_1 = 01234657$ and $G''_2 = 02461357$ are obtained as shown in Fig. 4.

Mutation: Given a chromosome $G = g_0g_1 \dots g_{2^k-1}$, two genes of G are selected randomly and their values are replaced with each other. For example, assume that $G = 01234567$ and the two randomly selected genes are 1 and 5 as shown in Fig. 5, then the mutated chromosome is $G = 05234167$.

Finally, the fitness function F is defined in this study as

$$F(G) = \sum_{j=0}^{2^k-1} c_{g_j} = \sum_{j=0}^{2^k-1} \sum_{k=0}^{2^k-1} o_{g_j,k} w_{g_j,k}, \quad (22)$$

where $o_{g_j,k}$ and $w_{g_j,k}$ are defined in Section 3.1. The fitness function listed above calculates the *MSE* between images R and E''' , in which E''' is obtained by replacing the gray value 0 of E'' with gray value g_0 , the gray value 1 with g_1 , and so on.

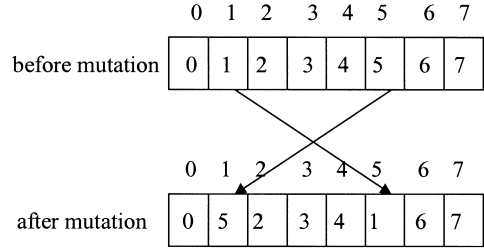


Fig. 5. A simple example of mutation operation.

To recover the original embedded image without any distortion, the optimal substitution matrix obtained by the above procedure should be further decomposed into a series of bit stream. Then each bit in the stream as well as the value of k are embedded into the $(k+1)$ th LSB position of the host image.

3.4. Image hiding by perceptual modeling

According to the research presented by Szu et al. [15], in which they proposed a “Johnson-like” criteria for video, the data embedded in a noisy picture is more difficult to be perceived than the data embedded in a smoothing picture. A simple example is illustrated in Fig. 6, in which images “Baboon” and “Lena” are used as host images, and the former looks noisier than the latter. After embedding the image “Jet” into the two host images, the results are depicted in Figs. 6(c) and (e). It can be seen from these figures that there are more perceived differences between Figs. 6(b) and (c) than those between Figs. 6(d) and (e). The concept is utilized in this study to improve the hiding result presented in the last section. Assume that the host image is a 256-gray-scale image (i.e., 8-bit per pixel.) The first four bits of each pixel counted from left to right are called the high byte, and the rest four bits are called the low byte. A pixel of the host image is defined in this study as a noisy pixel if less than half of its eight neighbors have the same content of high byte; otherwise, it is defined as a smoothing pixel. Due to the limitation of human video perception as stated above, it is suggested that a large scale of LSBs can be used to embed data for a noisy pixel, and a small scale of LSBs is used for a smoothing pixel. In this study, the rightmost four and two LSBs of noisy pixels and smoothing pixels, respectively, are used for hiding important data. In the next section, experiments are conducted to show the superiority of the modified embedding scheme.

4. Experimental results

Experimental results of the proposed method are presented and discussed in this section. The program was written in Visual C++, and ran on a personal

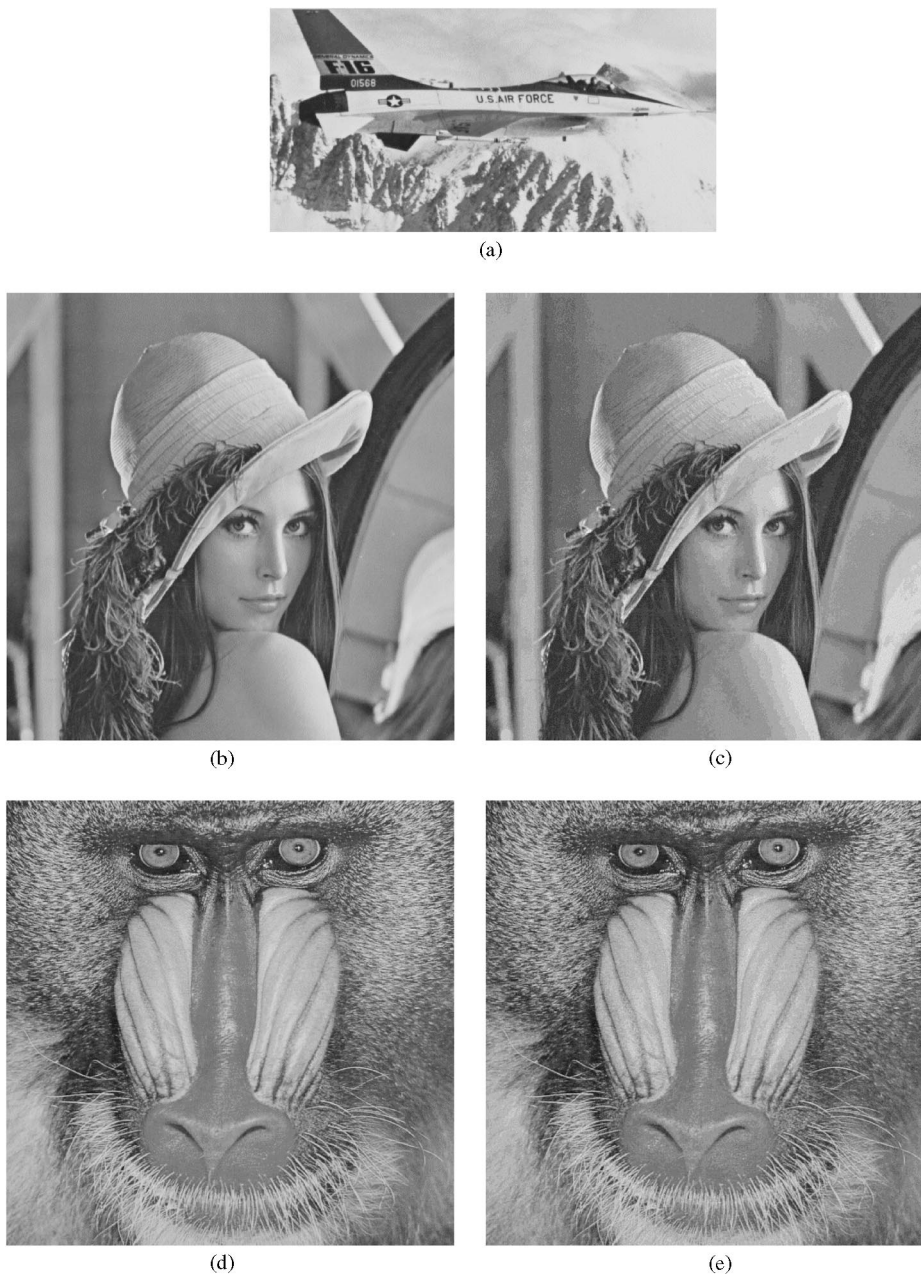


Fig. 6. A comparison of the embedding result between a noisy image and a smoothing image: (a) the embedding image named “Jet”, (b) the host image named “Lena”, (c) the result of embedding (a) into (b), (d) the host image named “Baboon”, (e) the result of embedding (a) into (d).

computer (PC) Pentium-233 with 64M RAM under the operation system of Win95.

The images tested in our experiment are all 8-bit images with 256 gray levels. The first three of them with size 512×512 shown in Fig. 7 are treated as the host images. The other test images that are prepared to be embedded

into the host images are classified into two classes: the first class contains four images with size 256×256 as shown in Fig. 8, and the second class contains the rest four images with size 512×256 as shown in Fig. 9.

As mentioned in Section 2, before embedding into the host image, images depicted in Figs. 8 and 9 must be



As the computers are more and more integrated via the network, the distribution of digital media is becoming faster, easier, and requiring less effort to make exact copies. One of the major impediment is the lack of effective intellectual property protection of digital media to discourage unauthorized copying and distribution.

Conventionally, in analog world, a painting is signed by the artist to attest the copyright, an identity card is stamped by the steel seal to avoid forgery, and the paper money are identified by the embossed portrait. Such kind of hand-written signatures, seals and watermarks have been used from ancient times as a way to identify the source, creator of a document or a picture. For example, a priceless painting of the 11th century in National Palace Museum named "Travelers on a Mountain Path" had not been identified as the genuine work

(c)

Fig. 7. The three test images with size 512×512 are used as the host images. (a) Lena; (b) Baboon; (c) Text.

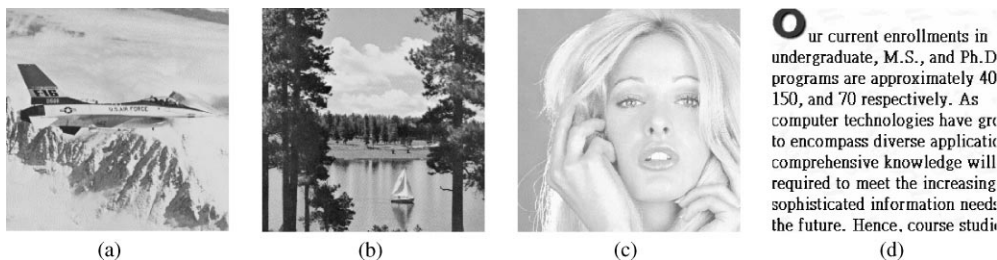


Fig. 8. The four test images with size 256×256 are used as the embedding data of the first class. (a) Jet; (b) Scene; (c) Tiff; (d) Text.

randomized by a cipher function. The cipher function adopted in this study is described by

$$f(x) = (1019 + 17x) \bmod 262144.$$

The first generation of the proposed GA is comprised of 10 chromosomes that are created by random numbers

to form the first population. After that, during each generation, the population is comprised of the following three kinds of chromosomes: (1) the chromosomes created by crossover; (2) the chromosomes created by mutation; and (3) the chromosomes created by reproduction. Chromosomes of the first kind are generated by randomly picking up 10 pairs of chromosomes from the

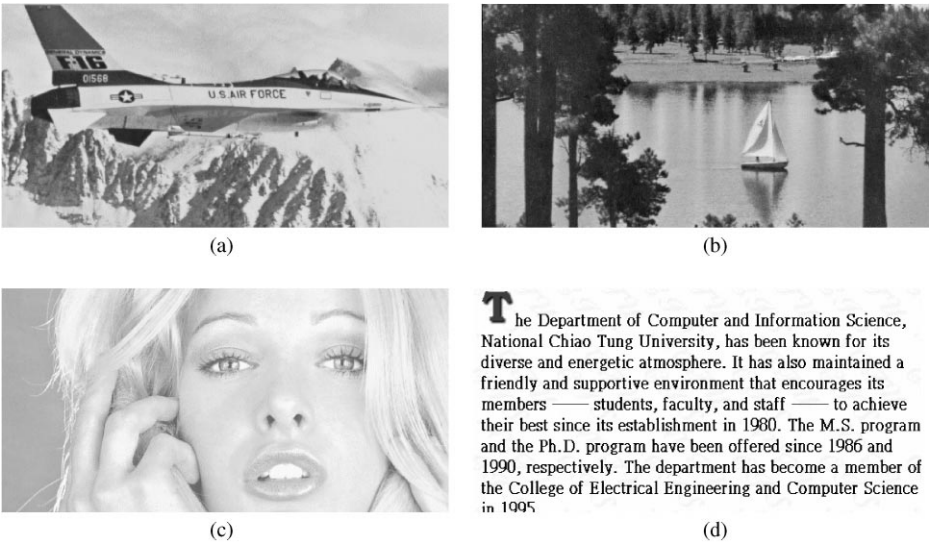


Fig. 9. The four test images with size 512×256 are used as the embedding data of the second class. (a) Jet; (b) Scene; (c) Tiff; (d) Text.

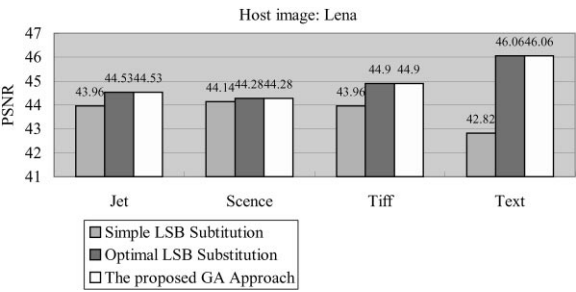


Fig. 10. The results of embedding the images shown in Fig. 8 into the host image labeled as “Lena”.

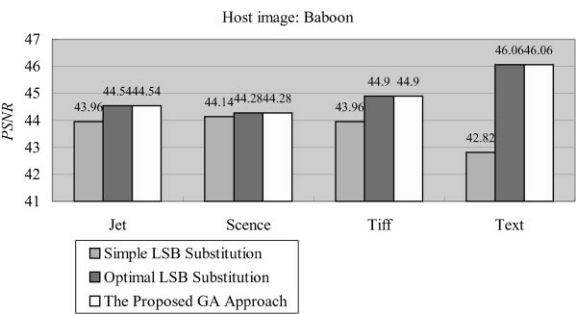


Fig. 11. The results of embedding the images shown in Fig. 8 into the host image labeled as “Baboon”.

population to undergo the crossover operation. Chromosomes of the second kind are generated by applying a mutation rate of 0.1 to each chromosome in the population. And chromosomes of the last kind are generated by reproducing (i.e., by copying) each chromosome in the

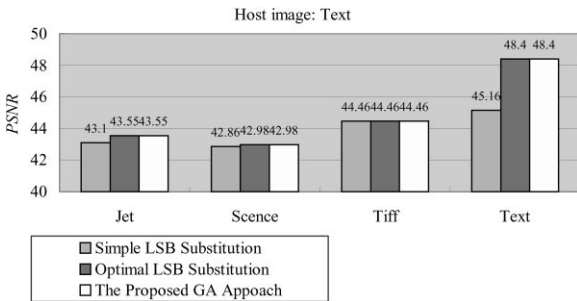


Fig. 12. The results of embedding the images shown in Fig. 8 into the host image labeled as “Text”.

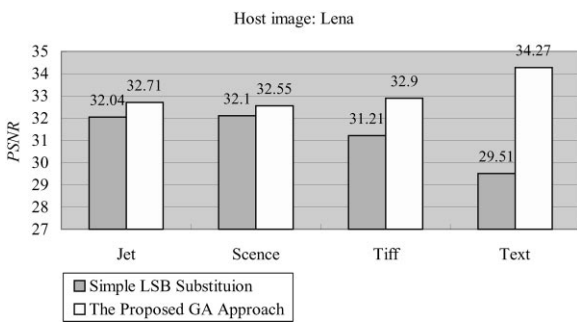


Fig. 13. The results of embedding the images shown in Fig. 9 into the host image labeled as “Lena”.

population. These chromosomes are then evaluated by the fitness function listed in Eq. (22), and sorted by the *PSNR* values calculated from the function. The chromosomes that own the top-10 highest *PSNR* values will survive, and form the population of the next generation.

The iterative number is set to 8 in our experiment. Finally, in the last generation, the chromosome that owns the highest *PSNR* value is selected to serve as the substitution function.

The results of embedding images in the first class are shown through Figs. 10–12, in which three different operations are conducted, including the simple substitution, the optimal substitution, and the proposed GA ap-

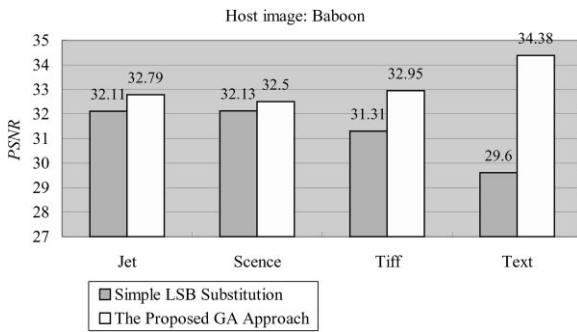


Fig. 14. The results of embedding the images shown in Fig. 9 into the host image labeled as “Baboon”.

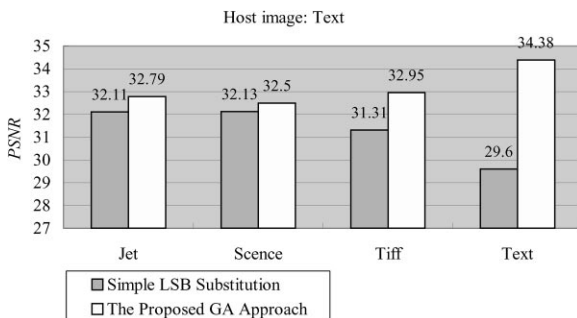


Fig. 15. The results of embedding the images shown in Fig. 9 into the host image labeled as “Text”.

proach. The data is embedded into the rightmost two LSBs of the host image, so the total number of substitution ways is $(2^2)! = 24$. Due to the small amount of searching space, the search procedure conducted by the GA approach is actually an exhaustive search, hence yields the same results as compared with that obtained by the optimal LSB substitution. The results of embedding images in the second class are shown in Figs. 13–15, in which the data are now embedded into the rightmost four LSBs of the host image and only two different operations are conducted, including the simple substitution and the GA approach. The optimal substitution cannot be executed for images in the second class because it involves a huge number of substitution operations. It can be seen from these figures that (1) the *PSNR* obtained by the GA approach is very high, which means that the quality of the embedding result can be acceptable; and (2) the GA approach can achieve similar results as the optimal substitution does.

The following experiment is to compare the embedding results obtained by the simple substitution method and the proposed GA approach. In this experiment, images shown in Figs. 9(d) and 7(a) are selected as the embedding data and the host image, respectively. To see more clearly the difference between the two methods, the data shown in Fig. 9(d) are not randomized by the cipher function given above, and the results obtained by the two methods are depicted in Fig. 16. As compared with the original host image shown in Fig. 7(a), it can be seen from this figure that the result obtained by the simple substitution method looks more noisy than that obtained by the GA approach. (The embedded data can be slightly perceived as seen from Fig. 16(a).) The *PSNR* values of the two methods are 29.51 and 34.27, respectively, which also indicates that the quality of the host image is less degraded if the GA approach is utilized. A comparison of the processing time by the two methods is summarized in Tables 1 and 2. The processing time of the proposed

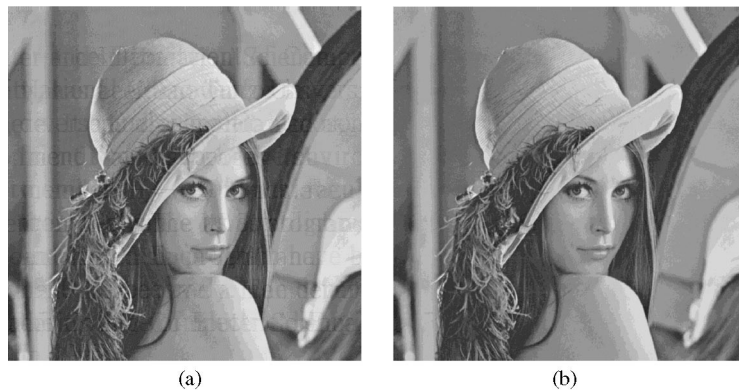


Fig. 16. The comparison of the embedding results obtained by the simple substitution method and the GA approach: (a) the result obtained by the simple substitution method, (b) the result obtained by the proposed GA approach.

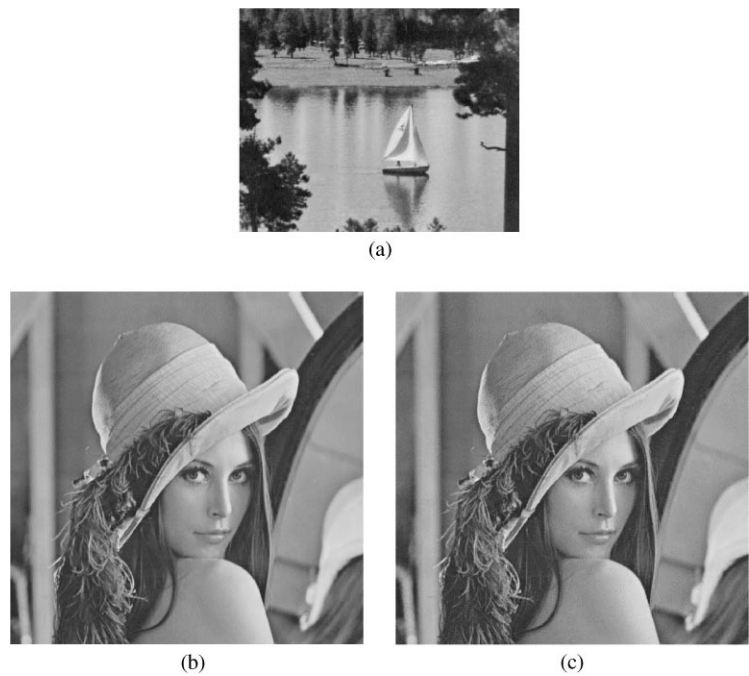


Fig. 17. A high quality of the embedding result obtained by the method presented in Section 3.4: (a) the hidden data with size 320×256 , (b) the host image labeled as “Lena” with size 512×512 , (c) the result of embedding (a) into (b) using the modified embedding scheme.

Table 1
The computation time of the simple LSB substitution method (unit: s)

Host	Embedded			
	Jet	Scene	Tiff	Text
Lena	0.28	0.33	0.28	0.33
Baboon	0.27	0.28	0.27	0.27
Text	0.33	0.33	0.33	0.28
Average		0.30		

GA approach is approximately twice of that by the simple substitution method as indicated in Tables 1 and 2. Both of them are acceptable for most applications, and can be further improved if the program is optimized or runs on a fast computer like a workstation or a Pentium III PC.

Finally, shown in Fig. 17 is the experiment conducted for demonstrating the high quality obtained by the method presented in Section 3.4. It is noted that the size of the hiding image is larger than a quarter of the size of the host image, so it cannot be entirely embedded in the rightmost two LSBs of the host image. If the hiding image is embedded in the rightmost four LSBs of the host image, the quality of the result may not be accepted. The method presented in Section 3.4 provides a good solution

Table 2
The computation time of the LSB substitution via GA approach (unit: s)

Host	Embedded			
	Jet	Scene	Tiff	Text
Lena	0.49	0.50	0.49	0.55
Baboon	0.50	0.61	0.55	0.60
Text	0.49	0.61	0.55	0.55
Average		0.58		

for the above problem. The rightmost four and two LSBs of the noisy pixels and smoothing pixels, respectively, of the host image are used to hide data. The reason is stated previously in Section 3.4, and the result is depicted in Fig. 17, which exhibits a good visual quality.

5. Conclusion

The major goal of image hiding is to embed skillfully the important data in the host image so that the interceptors will not notice the existence of the data. It is not like the goal of data encryption, which restricts the regular access to the embedded data, and also dissimilar to the goal of watermarking, which stresses

on the survival of the embedded data after image operation.

The basic concept of the proposed method is by simple LSB substitution. To increase the system performance, the method of optimal LSB substitution is presented. The effectiveness of the optimal LSB substitution under the worst condition is proved. We develop a genetic algorithm to solve the problem of hiding important data in the rightmost k LSBs of the host image when k is large, which may involve a huge computation time to find the optimal result. An improved hiding technique by perceptual modeling is also presented, and experimental results reveal the practicability and superiority of the proposed method.

References

- [1] J.K. Jan, Y.M. Tseng, On the security of image encryption method, *Inform. Process. Lett* 60 (1996) 261–265.
- [2] N. Bourbakis, C. Alexopoulos, Picture data encryption using scan patterns, *Pattern Recognition* 25 (6) (1992) 567–581.
- [3] H.J. Highland, Data encryption: a non-mathematical approach, *Comput. Security* 16 (1997) 369–386.
- [4] M.Y. Rhee, *Cryptography and Secure Communication*, McGraw-Hill Book Co, Singapore, 1994.
- [5] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, *IBM Systems J* 35 (3&4) (1996) 313–336.
- [6] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, *IEEE Trans. Image Process* 6 (12) (1997) 1673–1687.
- [7] R.L. Pickholtz, D.L. Schilling, L.B. Millstein, Theory of spread-spectrum communications — a tutorial, *IEEE Trans. Commun* 30 (1982) 855–884.
- [8] E. Koch, J. Zhao, Toward robust and hidden image copyright labeling, *IEEE Workshop on Nonlinear Signal and Image Processing*, Neos Marmaras, Greece, 1995, pp. 452–455.
- [9] C.T. Hsu, J.L. Wu, Hidden digital watermarks in images, *IEEE Trans. Image Process* 8 (1) (1999) 58–68.
- [10] E. Adelson, Digital signal encoding and decoding apparatus, U.S. Patent, No. 4,939,515, 1990.
- [11] L.F. Turner, Digital data security system, Patent IPN WO 89/08915, 1989.
- [12] R.G. van Schyndel, A.Z. Tirkel, C.F. Osborne, A digital watermark, *International Conference on Image Processing*, 1994, pp. 86–90.
- [13] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
- [14] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading MA, 1989.
- [15] H. Szu, P. Cox, C. Hsu, Video Johnson-like criterion for moving vehicle identification, *J. Electron. Imaging* 7 (4) (1998) 781–784.

About the Author—RAN-ZAN WANG was born in 1972 in Fukien, Republic of China. He received his B.S. degree in Computer Engineering and Science in 1994 and M.S. degree in Electrical Engineering and Computer Engineering and Science in 1996, both from Yuan-Ze University. Since 1996 he has been studying toward the Ph.D. degree and he is currently a Ph.D. candidate in the Computer and Information Science Department of National Chiao Tung University. His current research interests include image processing, pattern recognition, and information security.

About the Author—CHI-FANG LIN was born in 1960 in Taiwan, Republic of China. He received the B.S. degree in Transportation Engineering and Management in 1983, and the M.S. and the Ph. D. degrees in Institute of Computer Engineering in 1986 and 1991, all from National Chiao Tung University. During his study at National Chiao Tung University, he worked in the school library as a system engineer from 1983 to 1986. In the academic year 1987–1989, he was an instructor in Lien Hu Junior College of Technology. He is an assistant professor in the Department of Computer Engineering and Science at Yuan-Ze University. He is now the chairman of Department of Information Networking Technology, and is also the director of Information Technology Research Center. His field interest includes image processing, pattern recognition, and computer vision.

About the Author—JA-CHEN LIN was born in 1955 in Taiwan, Republic of China. He received his B.S. degree in computer science in 1977 and M.S. degree in applied mathematics in 1979, both from National Chiao Tung University, Taiwan. In 1988 he received his Ph.D. degree in mathematics from Purdue University, USA. In 1981–1982, he was an instructor at National Chiao Tung University. From 1984 to 1988, he was a graduate instructor at Purdue University. He joined the Department of Computer and Information Science at National Chiao Tung University in August 1988, and is currently a professor there. His recent research interests include pattern recognition and image processing. Dr. Lin is a member of the Phi-Tau-Phi Scholastic Honor Society.