Image Hiding Using Digit Number Transformation

Trees-Juen Chuang (莊樹諄), Ja-Chen Lin (林志青), and Wen-Hsiang Tsai (蔡文祥)*

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

Abstract

A new, simple, and efficient technique for image hiding by digit number transformation is proposed. Images of different sizes can be embedded into a counterfeit image by using different digit number systems. The given counterfeit image is first partitioned into non-overlapping fixed-size subimages, and the pixels of a given image are then embedded into the different subimages of the counterfeit image by certain mapping functions called embedding functions. In order to increase the quality of the embedding results, an enhanced method is also proposed. It is found the peak signal to noise ratio (PSNR) values of the proposed method are high. Neither counterfeit image preprocessing nor lookup table construction is required, so the method has the advantage of efficiency. The theory behind the proposed method and some experimental results are also provided.

1. Introduction

To prevent illicit access, special or important images need protection before being transmitted. Examples include military secrets, medical images, copyright, banking information, and so on. There are two types of image protection, namely, image encryption and image hiding, and their goals are different. Image encryption emphasizes that the encrypted images must be unrecognizable and that it is difficult for unauthorized users to guess the encryption key correctly [1-5]. On the other hand, image hiding focuses on embedding an image (i.e. say, the main image) into a counterfeit image (a normal image, e.g., a scenic or a girl's picture) and making the unauthorized users unable to realize the difference between the counterfeit image and the embedding result. Many image hiding techniques have been proposed, such as the LSB (Least-Significant Bit) method [6], the patchwork technique and the texture block coding method [7], the VQ-based method [8], the table lookup method by Liaw and Chen [9], and so on. We briefly survey these methods and their limitations below.

The LSB method was proposed by Adelson [6], and an image is embedded into the counterfeit image by replacing the least-significant bit, i.e. the 8*th* bit of each pixel of the counterfeit image. The patchwork technique and the texture block coding method were both proposed by Bender, Gruhl, and Lu [7]. The patchwork technique is based on a pseudorandom and statistic process, and the texture block coding method embeds data into the continuous random texture patterns of the counterfeit image. Chen, Chang, and Hwang [8] proposed another image hiding method which is based on VQ technique. In this method, both of the counterfeit image and the main image are divided into non-overlapping blocks, and the size of these blocks are all the same. The blocks of the counterfeit image are regarded as a codebook, and those of the main image are regarded as the query vectors. The mapping result (i.e. the index sequence) and the counterfeit image are then transmitted to the

* To whom all correspondence should be sent.

receiver. The work proposed by Liaw and Chen [9] has two main phases: (1) preprocessing of the counterfeit image and the main image to provide enough space to the hidden data; and (2) embedding the main image into the counterfeit image by the use of a lookup table (LUT). In this method, the counterfeit image has been changed before actually doing the embedding process, and the quality of the embedding result is therefore degraded severely. There are several limitations inherent in these conventional methods: (1) the capability for image hiding is very small [6-7] or extremely depends on the actual variance of the histogram of the images (the main images or the counterfeit images) [7,9]; (2) preprocessing of the counterfeit image [9] or table lookup for embedding [7-9] is needed; (3) the main image cannot be losslessly retrieved [8]; and (4) computing time is long [8]. General speaking, these techniques also limit the size of the main image to be a quarter of the size of the counterfeit image.

In this paper, we propose a new, simple, and efficient technique for image hiding, which is based on the digit number transformation. The proposed method needs neither counterfeit image preprocessing nor table lookup, and the quality of the embedding results are high. The reconstruction result of the main image is lossless. Some embedding functions are also presented to increase the security of the main image.

The remainder part of this paper is organized as follows. In Section 2, the overview of the proposed system is presented. The proposed techniques, including the digit number transformation and the embedding functions, are described in Section 3, and several experimental results are illustrated in Section 4. Some concluding remarks as well as a few suggestions for future works are stated in Section 5.

2. System Overview

A general definition of image hiding is as follows: given a gray-valued image E, called the main image and another A, called the counterfeit image, we want to embed E into A, with the embedding result being named R (see Fig. 1). A flowchart of the proposed system is shown in Fig. 2. We divide A into non-overlapping fixed-sized subimages (the total number of the subimages should not be smaller than that of the pixels in E), and then embed E into A pixel by pixel using the proposed digit number transformation and embedding function. Of course, the smaller the difference between the images R and A is, the higher the quality of the embedding result is obtained.

3. Proposed Method

3.1 Digit number transformation

A simple property is first presented below.

Property 1. If a decimal integer d is to be represented as an n-digit number with $n \geq 2$, d has to be converted to a base-h number with $h = \lceil \log_n d \rceil$.

By Property 1, we can represent the gray value, say, $(g_E)_{10}$, of a pixel in the main image E as an n-digit number $(p_0 p_1 \cdots p_{n-1})_h$, then we can embed $(p_0 p_1 \cdots p_{n-1})_h$ into any n pixels (whose n gray values are g_{A_0}, g_{A_1}, $\cdots$, $g_{A_{n-1}}$) of the counterfeit image A using the following equation

$$g'_{A_i} = g_{A_i} + p_i, \quad i = 0, 1, \cdots, n\text{-}1, \qquad (1)$$

where g'_{A_i} is the gray value of the embedding result corresponding to g_{A_i}, $p_i \in \{0,1,\cdots,h-1\}$. We then define the difference value d_i by

$$d_i = \left| g'_{A_i} - g_{A_i} \right|, \quad i = 0, 1, \cdots, n\text{-}1. \qquad (2)$$

Of course, the smaller the difference value d_i is, the higher quality the embedding result has. For example, if the main image E is a 256-level gray-valued image and n is set as 4, then $h = \lceil \log_4 256 \rceil = 4$ and $p_i \in \{0,1,2,3\}$. Therefore, $d_i = \left| g'_{A_i} - g_{A_i} \right| \leq 3$ and this value is small. Similarly, if n is set as 2, then $h = \lceil \log_2 256 \rceil = 16$, $p_i \in \{0,1,\cdots,15\}$, and $d_i \leq 15$. Owing to the description stated above, another property can therefore be obtained as follows.

Property 2. Any image whose size is not greater

than a half of the size of the counterfeit image A can be embedded into A by Equation (1).

The difference value d_i can even be smaller using an enhanced method presented below. An n-digit number $(p_0 p_1 \cdots p_{n-1})_h$ can be represented as a series of integer p'_0, p'_1, $\cdots$, p'_{n-1} by the following transformation:

$$\begin{cases} p'_i = p_i, & \text{if } p_i \le \left\lfloor \frac{h-1}{2} \right\rfloor \\ p'_i = \left\lfloor \frac{h-1}{2} \right\rfloor - p_i, & \text{if } p_i > \left\lfloor \frac{h-1}{2} \right\rfloor. \end{cases} \quad (3)$$

Here, $\lfloor \bullet \rfloor$ is the floor function to get a truncated integer (e.g., $\lfloor 3.7 \rfloor = 3$) and $p'_i \in \left\{ -\left\lfloor \frac{h-1}{2} \right\rfloor, \cdots, \left\lfloor \frac{h-1}{2} \right\rfloor \right\}$ for $i = 0, 1, \cdots, n\text{-}1$. The difference value d_i now is not greater than $\left| \left\lfloor \frac{h-1}{2} \right\rfloor - (h-1) \right| = \left\lceil \frac{h-1}{2} \right\rceil$ $(\le h-1)$. For example, if n is set as 4 (or 2), then $p'_i \in \{-2,-1,0,1\}$ (or $\{-8,-7,\cdots,7\}$) and $d_i \le 2$ (or 8). Obviously, the difference value d_i is smaller than that we described previously. Note that the more data we hide (i.e. the larger the size of the main image), the less secure we can obtain (i.e. the peak signal to noise ratio (PSNR) value of the embedding result is) [7]. Also it is easy to verify that the mean square error (MSE) values between the counterfeit image A and the embedding result R lie in the range of 0 ~ $\left(\left\lceil \frac{h-1}{2} \right\rceil \right)^2$ and that the corresponding PSNR values is smaller than $10 \log_{10} \frac{255^2}{\left(\left\lceil \frac{h-1}{2} \right\rceil \right)^2}$ (see Equations (3) and (4)).

Property 3. The PSNR (or MSE) values between the counterfeit image and the embedding results depends on the main image and is independent of the counterfeit image.

Proof of Property 3.

Suppose that the total number of subimages in image A is Z_S, and the total numbers of pixels in image E and image A are Z_E and Z_A, respectively, then

$$\begin{aligned} \text{PSNR} &= 10 \log_{10} \left(\frac{255^2}{MSE} \right) \\ &= 10 \log_{10} \left(\frac{255^2}{\frac{1}{Z_A} \sum_{r}^{Z_A} (g'_{A_r} - g_{A_r})^2} \right) \\ &= 10 \log_{10} \left(\frac{255^2}{\frac{1}{Z_A} \sum_{j}^{Z_S} \sum_{i}^{n} d_{ji}^2} \right) \\ &= 10 \log_{10} \left(\frac{255^2}{\frac{1}{Z_A} \sum_{j}^{Z_E} \sum_{i}^{n} p_{ji}^2} \right) \\ &\left[\ge 10 \log_{10} \left(\frac{255^2}{\frac{1}{Z_A} \sum_{j}^{Z_E} \sum_{i}^{n} \left(\left\lceil \frac{h-1}{2} \right\rceil \right)^2} \right) \right. \\ &\left. = 10 \log_{10} \frac{255^2}{\left(\left\lceil \frac{h-1}{2} \right\rceil \right)^2} . \right] \end{aligned} \quad (4)$$

Since $\{ p_{ji} \mid i = 0, 1, \cdots, n\text{-}1 \}$ is converted from the jth gray value of the main image E, which is independent of the counterfeit image, this property is obtained.

By Property 3, the proposed method therefore requires neither counterfeit image preprocessing nor lookup table setup.

3.2 Proposed embedding function

After the stage of digit number transformation, we then embed image E into image A pixel by pixel. Given each pixel of image E, (whose gray value g_E has been converted into an n-digit number

$(p_0 p_1 \cdots p_{n-1})_h$), we want to embed it into one of subimages (say, S_A, whose size is not smaller than n pixels) in image A. We proposed here some one-to-one mapping functions, namely, embedding functions, which can be easily applied to create a mapping of the pixel location of g_E to that of S_A:

$$f(L(g_E)) \equiv (L(S_A) + k) \mod Z_S \quad (5)$$

$$f(L(g_E)) \equiv (k_1 \times L(S_A) + k_2) \mod Z_S \quad (6)$$

$$f(L(g_E)) \equiv \left(k_0 + k_1 \times L(S_A) + k_2 \times \left(L(S_A)^2\right) + \cdots + k_m \times (L(S_A))^m\right) \mod Z_S, \quad (7)$$

and so on, where $L(g_E)$ and $L(S_A)$ denote the locations of g_E and S_A, respectively, f is the so-called embedding function, Z_S is the total number of the subimages, and k_x are certain embedding keys. Equations (5), (6), and (7) can be regarded as a kind of Caesar cipher, affine transformation, and polynomial transformation, respectively [10].

We can even use the following approach to increase further the security of the main image: encrypt first the main image by some reported encryption methods [1-3], and then perform the embedding process described above (see Fig. 3).

In the receiver end, the main image can be losslessly retrieved from the counterfeit image A using Equations (1), (3), (5)-(7), the embedding key, and the encryption key (if the embedding function and the encryption method are used). This retrieving process is very similar to the embedding process and the description of its detail is omitted here.

4. Experimental results

In our experiments, two counterfeit images shown in Fig. 4 with sizes being both 512×512, and four main images shown in Fig. 5 with sizes being 256×256, 256×256, 256×256, and 256×512, respectively, are used. The embedding results are illustrated in Figs. 6 and 7. It is found the embedding results are very similar to the counterfeit images, and the PSNR values are high (see Table 1). Note that the quality of the embedding results is independent of the counterfeit images (e.g., the PSNR values of "Pepper" v.s. "Lena" and "Pepper" v.s. "House" are both 46.64), and all of the test images, including the counterfeit images and the main images, are 256-level gray-valued images.

5. Concluding Remarks and Future Works

In this paper, a new, simple, and efficient method for gray-valued image hiding by digit number transformation is proposed. Any image whose image size is not larger than a half size of the counterfeit image can be embedded. The quality of the embedding results is independent of the counterfeit image, and it is found the PSNR values are high, about 46 (or 34) if n is 4 (or 2). The proposed method needs neither counterfeit image preprocessing nor lookup table setup, and some embedding functions and encryption methods can be easily applied further to increase the security of the main image. In the future, the problem of embedding an image with size identical to that of the counterfeit image is worth study.

References

[1] N. Bourbakis and C. Alexopoulos, "Picture Data Encryption Using SCAN Patterns," *Pattern Recognition* 25(6), 567-581, 1992.

[2] C. Alexopoulos, N. G. Bourbakis, and N. Ioannou, "Image Encryption Method Using a Class of Fractals," *J. of Electronic Imaging* 4(3), 251-259, 1995.

[3] G. B. White, E. A. Fisch, and U. W. Pooch, *Computer System and Network Security*, *CRC Press, Inc., U.S.A.*, 1996.

[4] C. J. Kuo, "Novel Image Encryption Technique and Its Application in Progressive Transmission," *J. of Electronic imaging* 2(4), 345-351, 1993.

[5] T. J. Chuang and J. C. Lin, "A New Approach to Image Encryption," *J. of Electronic imaging* 7(2), 1998.

[6] E. Adelson, "Digital Signal Encoding and Decoding Apparatus," *U.S. Patent*, No. 4939515, 1990.

[7] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for Data Hiding," *IBM Systems Journal* 35(3&4), 313-336, 1996.

[8] T. S. Chen, C. C. Chang, and M.S. Hwang, "A Virtual Image Cryptosystem Using Vector Quantization," *Proceedings of National Information Security Conference, Republic of China*, 10-16, 1996.

[9] M. S. Liaw and L. H. Chen, "An Effective Data Hiding Method," *IPPR Conference on Computer Vision, Graphics, and Image Processing, Taichung, Taiwan, R.O.C.*, 146-153, 1997.

[10] M. Y. Rhee, *Cryptography and Secure Communications*, McGran-Hill Book Co., Singapore, 1994.

Table 1. The PSNR values of the embedding results.

PSNR (Main images / Counterfeit images)	Pepper (256x256)	Bridge (256x256)	Baboon (256x256)	F-16 (256x512)
Lena	46.64	46.69	46.52	34.30
House	46.64	46.69	46.52	34.30

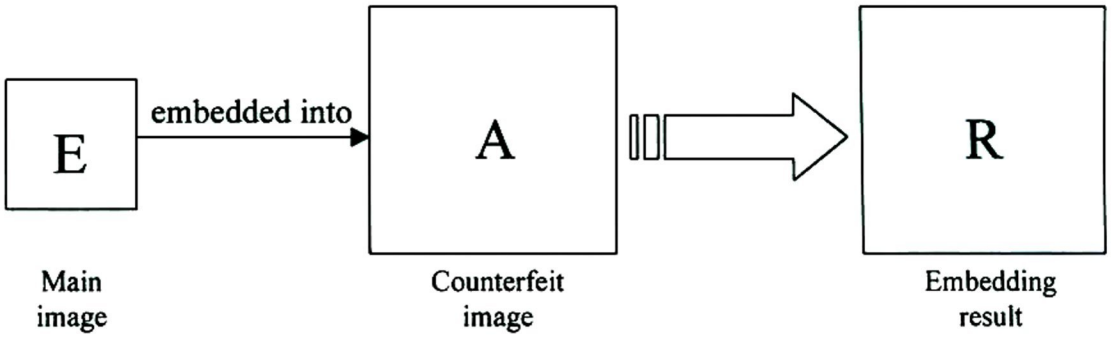


Fig. 1 The embedding process.

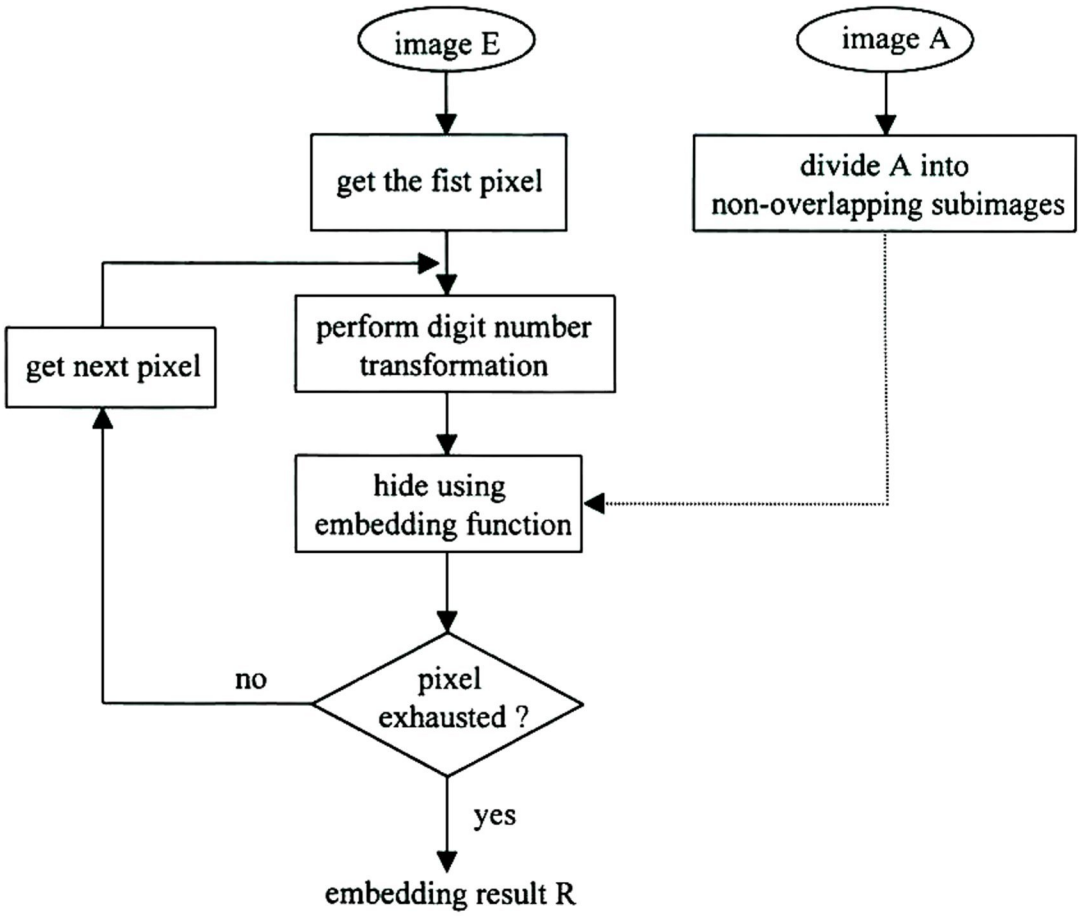


Fig. 2 The system overview.

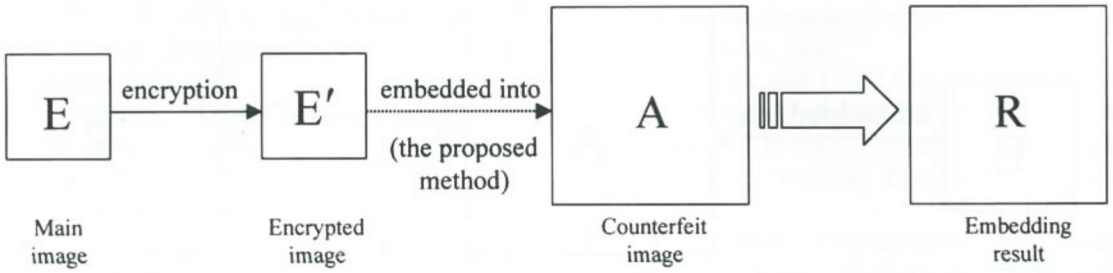


Fig. 3 The embedding process combined with the encryption method.

(a) (b)

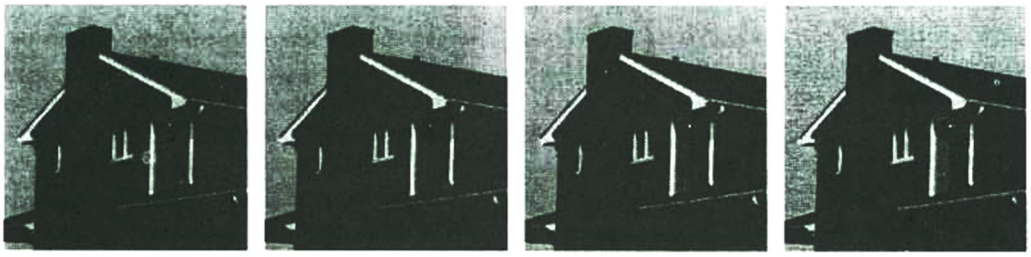
Fig. 4 The counterfeit images. (a) "Lena" (512×512) and (b) "House" (512×512).

(a) (b) (c) (d)

Fig. 5 The main images. (a) "Pepper" (256×256); (b) "Baboon" (256×256); (c) "Bridge" (256×256); and (d) "F-16" (256×512).

(a) (b) (c) (d)

Fig. 6 The embedding results of embedding Figs. 5 (a)-(d) into Fig. 4 (a), respectively.



(a) (b) (c) (d)

Fig. 7 The embedding results of embedding Figs. 5 (a)-(d) into Fig. 4 (b), respectively.