Adaptive Data Hiding in Palette Images by Color Ordering and Mapping With Security Protection

Chih-Hsuan Tzeng, Zhi-Fang Yang, and Wen-Hsiang Tsai, *Senior Member, IEEE*

***Abstract*—Palette images are widely used in multimedia and Internet applications. In this paper, a new method for data hiding in palette images with security protection by color ordering and mapping, as well as parameter randomization, is proposed. First, image pixels are classified as data embeddable or nonembeddable, and only the former ones are used to embed secret data. The proposed idea of data hiding is based on the use of a new type of color-ordering relationship, from which a color-mapping function is defined with binary values as output. When a secret data bit is to be embedded, a data-embeddable pixel is selected, and its color is adjusted to make the output of the color-mapping function equal to the secret bit value. The embedded secret data can be extracted correctly and quickly from the resulting stego-image by merely inspecting the outputs of the color-mapping function. Indetectability of the secret information embedded by the proposed method is also analyzed and confirmed. Furthermore, a number of possible security enhancement measures based on parameter randomization in the data-embedding process are proposed to protect the hidden data in the stego-image. The randomization effect is created by the use of a secret key and a number of random number-generating functions. The proposed data hiding method was tested with a variety of palette images. The experimental results show that secret data can be embedded and extracted successfully without producing visual artifacts in the cover image. A good balance between stego-image quality and data-embedding capacity can be achieved, which proves the efficiency and feasibility of the proposed method for practical applications.**

***Index Terms*—Color palette, color-mapping function, color-ordering relationship, cover image, data hiding, palette image, parameter randomization, random number-generating function, security protection, stego-image.**

I. Introduction

THE IDEA OF data hiding is to seamlessly modify a given image, called a *cover image* in this paper, in a prescribed manner so that secret information can be embedded in the resulting image, called the *stego-image*, without creating noticeable artifacts [1]. The recipient can correctly extract the embedded information from the stego-image, while other people are unaware of the existence of the secret behind the stego-image. Data-hiding techniques can be exploited for many applications, such as steganography, covert communication, data authentication, annotation association, etc., and so have been extensively investigated in recent years [2]. This paper is mainly on the topics of steganography and covert communication of palette-based images. Applications of such topics include intelligent information hiding or transmission for military purposes, secret keeping, or communication in business activities of companies, etc.

Palette-based images, or simply *palette images*, are popular in multimedia and Internet applications. Each palette image is composed of a color palette and a set of color indexes. The color palette is a list of entries of representative colors in the image, and the color indexes are some pointers to those palette entries that specify the red–green–blue (RGB) colors in the image. Use of this type of palette image format has the effect of image compression, which helps saving storage space and reducing transmission time. An example of palette images is that of the graphics interchange format (GIF). Popularity of palette images in multimedia and Internet applications makes them appropriate for use as cover images to conceal secret data without arousing suspicion when the resulting stego-images are inspected or transmitted over the Internet. Many studies have been devoted to data hiding [1]–[5]; however, little attention has been paid to such studies for palette images so far. The goal of this study is to design a new secure method for data hiding in palette images.

Since the indexes of the color palette used in a palette image are not color values themselves, hiding data in palette images is more challenging than in images of other formats. Either the palette or the image data can be used to hide secret data. Kwan [6] developed a program called "Gifshuffle" to embed data in GIF images. The idea is to permute the colors in the palette of an image in a specific order in accordance with the secret data. There are 256! possible permutations of the 256 entries of a color palette. So, at most $\log_2(256!)$ bits can be embedded into a GIF image. A merit of this approach is that the visual content of the cover image is not affected. However, a serious drawback is that palette image software will usually rearrange the order of the colors in the palette of a given image and so destroy the data hidden in the palette, when the image is reloaded and saved. Additionally, a palette with randomly ordered color entries might also be a hint of data hiding. In some steganographic applications [7]–[10], by using the dithering method, the palette size can be doubled and thus used to choose appropriate colors to replace those of the pixels where data are embedded. However, it was reported in [11] that if data hiding is achieved by manipulation of the color palette, hidden data is much more likely to be detected by steganalysis. As a result, data hiding by manipulation of the image content instead might be more favorable for practical applications, and this is the approach adopted in this paper.

Paper approved by K. Rose, the Editor for Source-Channel Coding of the IEEE Communications Society. Manuscript received November 15, 2001; revised November 28, 2002 and April 11. 2003. This work was supported in part by the MOE Program for Promoting Academic Excellency of Universities under Grant 89-1-FA04-1-4.

The authors are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: chtzeng@cis.nctu.edu.tw; zfyang@cc.nctu.edu.tw; whtsai@cis.nctu.edu.tw).

Digital Object Identifier 10.1109/TCOMM.2004.826379

0090-6778/04$20.00 © 2004 IEEE

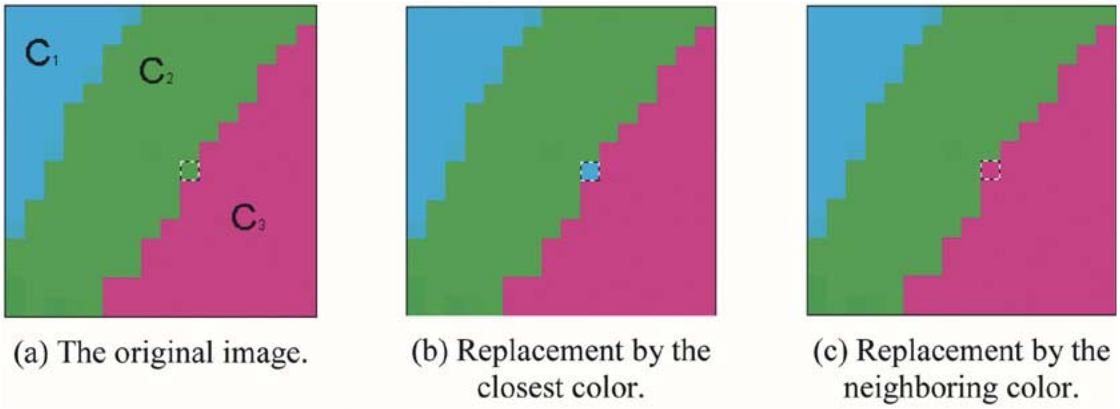

C1
C2
C3
(a) The original image.
(b) Replacement by the closest color.
(c) Replacement by the neighboring color.

Fig. 1. Example of pixel color replacement. (a) Original image. (b) Replacement by the closest color. (c) Replacement by the neighboring color.

Fridrich and Du [12] proposed a method to embed data in palette images by first assigning a specific parity bit (0 or 1) to each color in the palette, and then adjusting the pixel index values in such a way that the parities of the new index values are equal to the message bits to be embedded. They proposed an optimal parity-assignment algorithm so that the resulting data-embedding process guarantees that an index is always replaced by the index of the closest color. Their adaptive method can be employed to conceal a moderate amount of data and has the least modification of pixel values. However, replacing a color with its closest one may not always be the best choice. Theoretically, minimization of the color difference after data embedding will result in the least distortion in the image. Practically, this might not be adequate for palette images, especially for those with low color depth. Fig. 1 illustrates an example of the observation about this phenomenon. For simplicity, only three colors c_1, c_2, and c_3 with RGB values (0, 174, 239), (57, 181, 74), and (236, 0, 140) are contained in the sample image. Fig. 1(a) shows the original image. The pixel to be adjusted is located near the center of the image and is marked by a bounding box. It can be noted that the original color of the pixel is c_2. Because the closest color of c_2 is c_1, c_2 is replaced by c_1 to minimize the color change of the pixel, as shown in Fig. 1(b). This results in a spark in the image. On the other hand, Fig. 1(c) shows the result of replacing c_2 with the neighboring color c_3. The image contains no visual artifact. As a result, in the development of data hiding in palette images, not only the color characteristics, but also the spatial properties of the image content should be considered.

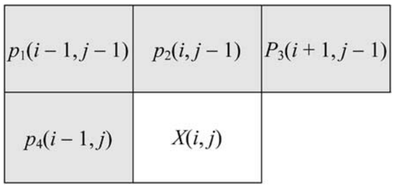
Though a few effective data-hiding methods for binary images have been proposed [4], [13]–[15], the embedding capacity of a binary image is essentially very limited. On the other hand, various palette images are available for secret data embedding, so we may choose images with larger numbers of colors rather than with two or only a few colors as cover images. Nevertheless, the proposed method also can be applied to palette images with small numbers of colors except that the embedding capacity of such images will be smaller. In case that a large amount of data is to be hidden, one way out is to use multiple images instead of just one for data embedding.

In this paper, a data-hiding scheme for palette images is proposed for steganographic applications. Secret information hidden in a stego-image is fragile to image manipulations. Also, such applications require that the secret information hidden in a stego-image is visually and statistically undetectable, and the hidden secret cannot be read even when its existence is detected. To achieve these two goals, the proposed scheme is based on the use of a certain color-ordering relationship. Such an ordering relationship is designed according to some color quality. The ordering relationship is used to choose an optimal set of colors to replace the original ones to achieve better hiding effects, based on the idea of minimizing color distortion. Besides, the proposed method also considers the necessity of reducing local image content changes as mentioned above, yielding secret embedding results with a good compromise between the resulting visual quality and data-hiding capacity. Furthermore, to ensure the security of the hidden data, it is usually desired to have some measures to protect the data behind the stego-image from being illegally discovered. For this purpose, we also propose a sequence of security protection measures, which are based on the concept of randomizing the parameters involved in the proposed data-embedding process. Parameter randomization is achieved by a set of random number generators, all of which are controlled conveniently by the use of a single secret key. The resulting data-embedding scheme becomes so dynamic and complicated that illicit discovery of the hidden data is expected to be impossible.

In the following, the proposed color-ordering relationship and color-mapping function will be introduced in Section II. In Sections III and IV, the proposed data embedding and extraction processes will be described, respectively. Analysis of indetectability of the secret information embedded by the method and the proposed security protection measures for the method will be described in Section V. In Section VI, several experiments results will be presented to show the feasibility of the proposed method. And finally, in Section VII, some conclusions will be given.

II. Proposed Color-Ordering Relationship and Color-Mapping Function

The proposed data-hiding method is based on the use of a color-ordering relationship and a binary color-mapping function, which we define in this section. The binary color-mapping function takes as input the color values of a group of image



$p_1(i-1,j-1)$	$p_2(i,j-1)$	$P_3(i+1,j-1)$
$p_4(i-1,j)$	$X(i,j)$	

Fig. 2. A pixel X and its four precedent neighbors (the shadowed ones) in the raster-scanning sequence.

pixels, and yields as output a binary value "0" or "1" in accordance with the color relationship among the pixels' colors. The basic idea of data embedding proposed in this paper is to modify the colors of some image pixels in such a way that the corresponding binary outputs of the color-mapping function with the colors of the resulting image pixels as input are equal to the data bits to be hidden.

First, we define the color-ordering relationship. Let c_1 and c_2 be two colors with RGB values (r_1, g_1, b_1) and (r_2, g_2, b_2), respectively. The color-ordering relationship is based on the luminance values v_1 and v_2 of c_1 and c_2, where v_1 and v_2 are computed as follows [16]:

$$\begin{aligned} v_1 &= 0.3 \times r_1 + 0.59 \times g_1 + 0.11 \times b_1 \\ v_2 &= 0.3 \times r_2 + 0.59 \times g_2 + 0.11 \times b_2. \end{aligned} \tag{1}$$

It is possible that c_1 and c_2 are different, but v_1 and v_2 are equal to each other. In this situation, the color-ordering relationship is defined by comparing the RGB values of c_1 and c_2 further to make the relationship unique. Three possible *color orders* between c_1 and c_2 are defined as follows, which we say to form a *color-ordering relationship* R_{co}:

$$R_{\mathrm{co}}: \begin{cases} c_1 > c_2, & \text{if } (v_1 > v_2) \text{ or} \\ & (v_1 = v_2 \text{ and } r_1 > r_2) \text{ or} \\ & (v_1 = v_2 \text{ and } r_1 = r_2 \text{ and } g_1 > g_2) \\ c_1 = c_2, & \text{if } (r_1 = r_2 \text{ and } g_1 = g_2 \text{ and } b_1 = b_2) \\ c_1 < c_2, & \text{otherwise.} \end{cases} \tag{2}$$

It is not difficult to see that the above relationship defines a unique ordering for all colors and serves the purpose of data hiding well. However, it is noted that by randomizing the ordering sequence determined by the above definition, more color-ordering relationships can be generated, as will be explained in detail in Section V.

Next, we define some terms before giving the definition of the color-mapping function. In the proposed data-embedding process, an input cover image is processed in a raster-scanning manner. Given a pixel X in the cover image, we define its *precedent neighbors* to be those *four* neighboring pixels, among the eight neighboring ones in a 3×3 neighborhood of X, which are visited in sequence before the other four during the line-by-line raster scanning. More specifically, if X is located at coordinates (i, j) in the input image, then its precedent neighbors are the four pixels located at $(i-1, j-1)$, $(i, j-1)$, $(i+1, j-1)$, and $(i-1, j)$. See Fig. 2 for an illustration. Then, we define three related terms: 1) the *color difference* $|c_1 - c_2|$ *between two colors* c_1 and c_2 as the Euclidean distance between the RGB values (r_1, g_1, b_1) and (r_2, g_2, b_2) of c_1 and c_1, respectively, or more specifically, as $|c_1 - c_2| = \left[(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2\right]^{1/2}$; 2) the *color difference between two pixels* X_1 and X_2 with colors c_1 and c_2, respectively, as the color difference between c_1 and c_2; and 3) the *maximum color difference between a given pixel* X *and its four precedent neighbors* X_1–X_4 as the maximum of the four color differences between X and X_1–X_4, respectively.

We are now ready to define the color-mapping function. For a given pixel X with color c, let the colors of its four precedent neighbors X_1–X_4 be c_1 through c_4, respectively. Let the result of sorting the values of c_1–c_4 according to the color-ordering relationship be c'_1–c'_4, with c'_1 being the largest. We define a color-mapping function with binary outputs as follows, which we denote as f_{cm}:

$$f_{\mathrm{cm}}(c, c'_1, \ldots, c'_4) = \begin{cases} 0, & \text{if } c \geq c'_1 \\ 1, & \text{if } c'_1 > c \geq c'_2 \\ 0, & \text{if } c'_2 > c \geq c'_3 \\ 1, & \text{if } c'_3 > c \geq c'_4 \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

It can be seen that the function output depends on the ordering of the color c of X among those of the four precedent neighbors of X.

In addition, to reduce possible quality degradation in the resulting stego-image, the pixels in the cover image are classified into *data embeddable* and *nonembeddable* ones in this study during the raster-scanning process. Only data-embeddable pixels are used for data hiding; nonembeddable ones are skipped. Let c be the original color of a given pixel X and c' a possible replacement for c in the color palette P. When the color of X is c, assume that the corresponding output of the color-mapping function of X is b, and that the corresponding maximum color difference between X and its four precedent neighbors is β. When the color c of X is replaced by c', assume that the corresponding values of b and β are changed to be b' and β', respectively. Also assume that the number of distinct colors of X's four precedent neighbors is α. A pixel X is defined to be *data embeddable* if the following three conditions are satisfied: 1) α is larger than a threshold value T_c; 2) β is smaller than a threshold value T_d; and 3) there exists a color c' with the corresponding b' being the inverse of b, and the corresponding β' being smaller than the threshold value T_d. Or equivalently, we define the *data embeddability* of a pixel X as shown in (4) at the bottom of the page. In (4), the first condition $\alpha > T_c$ is used to ensure that X is located in a reasonably

$$\begin{array}{ll} X \text{ is data embeddable,} & \text{if } \alpha > T_c, \beta < T_d, \text{and there exists a } c' \text{ such that } b' \neq b \text{ and } \beta' < T_d \\ X \text{ is nonembeddable,} & \text{otherwise.} \end{array} \tag{4}$$

color-abundant region so that pixel color modification due to data embedding will arouse little suspicion. A reason here is that the change of a pixel's color among a region with a lot of colors presumably is less noticeable. The second condition $\beta < T_d$ is set mainly to avoid pixel color modification at high contrast regions with large β values, where sharp lines or edges will appear with higher probabilities. Modification of colors at such regions usually will cause more obvious image content changes, and so reduce the data-hiding effect. The third condition is set to ensure that X has the ability to embed a bit, and that X is still data embeddable after the data-embedding process. Note that, because the number of distinct colors of X's four precedent neighbors is not changed when the color of X is altered, the condition $\alpha > T_c$ will not be changed, and so is not included in the third condition.

III. Proposed Data-Embedding Process

Assume that the secret data S to be embedded in a cover palette image I is a bit stream, denoted as $S = b_1b_2, \ldots, b_n$. The basic idea of the proposed data-embedding process is to check each pixel of I in a raster-scanning manner for its data embeddability, and to embed each secret bit b_i of S sequentially into every data-embeddable pixel, until the bit stream of S is exhausted. During each secret bit-embedding step, if the binary output of the color-mapping function f_{cm} is the same as the secret bit value to be embedded, the color c of the currently checked data-embeddable pixel X is kept unchanged; otherwise, c is replaced with a color c_{opt}, called the *optimal replacement color* for X, selected from the color palette by the following algorithm.

Algorithm 1: Optimal replacement color selection for a given pixel

Input: The color c of the currently checked data-embeddable pixel X, the color palette P, and a secret bit value b to be embedded.

Output: The optimal replacement color c_{opt} for X.

Step 1) Let C denote a set of candidate optimal replacement colors for X, and set C empty initially.

Step 2) Put each color c_i in the color palette P into C if c_i satisfies the following two conditions: 1) c_i together with the colors of the four precedent neighbors of X as input to the color-mapping function f_{cm} yields a binary output value b_0 equal to the secret bit b; and 2) X is *still data embeddable* when its color is set to c_i.

Step 3) Find the color c' among those in C, whose color difference from c is the minimum, i.e., find the c' in C that satisfies the following condition:

$$|c - c'| = \min_{c_i \in C} |c - c_i|.$$

Step 4) Let N denote the subset of C that contains the colors of the four precedent neighbors of X. If the color difference $|c - c'|$ between c and c' is smaller than a preselected threshold T_v or N is empty, then take c' as the desired optimal replacement color c_{opt} for X and stop; otherwise, perform the next step.

Step 5) Find the color c' among those in N, whose color difference from c is the minimum, i.e., find the c' that satisfies the following condition:

$$|c - c'| = \min_{c_i \in N} |c - c_i|$$

take c' as the desired c_{opt} for X; and stop.

The first three steps of the above algorithm aim to select from the color palette a color c' which is "most similar" in color to that of the currently checked pixel X. However, if this color c' differs too much from that of X, use of it as the desired optimal replacement color will cause an obvious visual artifact at X, as illustrated by Fig. 1(b). Therefore, Step 4) is performed with the preselected threshold T_v used to avoid such a case. If this case is found to exist, then in Step 5), the "most similar" color found from those of the four precedent pixels is taken to be the desired optimal replacement color for the currently checked pixel X, as illustrated by Fig. 1(c). An important idea behind the proposed algorithm above is that we require the pixel X to be *still data embeddable* after its color is replaced, as depicted by the second condition in Step 2). This measure facilitates the work of identifying those pixels into which secret bits have been embedded during the secret data-extraction process.

We are now ready to describe the proposed secret data-embedding process as an algorithm in the following.

Algorithm 2: Secret data-embedding process

Input: A cover palette image I, and a secret bit stream $S = b_1b_2, \ldots, b_n$ to be embedded.

Output: An image I' with the secret S being embedded.

Step 1) For each secret bit b_j in S, perform the following steps until all secret bits in S are embedded.

Step 2) Perform a raster scan of I and check the data embeddability of each scanned pixel, until a data-embeddable pixel X is found.

Step 3) Take the color c of X and the sorted colors c'_1–c'_4 of the four precedent neighbors of X as input to the color-mapping function f_{cm} to yield a binary output bit b_0.

Step 4) Check whether the secret bit b_j is equal to b_0. If so, regard the secret bit b_j to be already existing at X, and go to Step 1) to embed the next bit; otherwise, perform the next step.

Step 5) Find the optimal replacement color c_{opt} for X by Algorithm 1; substitute the color c of X with c_{opt}; and go to Step 1) to embed the next bit.

IV. Proposed Data-Extraction Process

During the data-extraction process, a given stego-image is also processed in a raster-scanning manner. The definition given in (4) is used to check the pixels' data embeddability. Only data-embeddable pixels are processed further; all nonembeddable ones are skipped. For each data-embeddable pixel, we take its color and those of its four precedent neighbors as input to the color-mapping function f_{cm}, and compute the binary output value. If the output is "0," then the extracted secret bit is taken to be "0;" otherwise, "1." As can be seen, the extraction process is simple and so can be performed very fast. The detailed data-extraction process is described in the following as an algorithm.

```
Algorithm 3: Secret data-extraction process
  Input: An input stego-image $I'$ in which a secret bit stream $S = b_1 b_2, \ldots, b_n$ was embedded.
  Output: The secret bit stream $S$.
  Step 1) Set $S$ as an empty bit stream initially.
  Step 2) Perform a raster scan of $I'$ and execute the following steps until all pixels of $I'$ are processed.
  Step 3) For each pixel $X$ of $I'$, check whether it is data embeddable. If not, regard no secret bit to exist at $X$, and go to Step 2); otherwise, perform the next step.
  Step 4) Take the color $c$ of $X$ and the sorted colors $c'_1$–$c'_4$ of the four precedent neighbors of $X$ as input to the color-mapping function $f_{\rm cm}$ to yield a binary output bit $b$.
  Step 5) Append $b$ to the end of $S$, and go to Step 2).
```

V. Security Protection for Proposed Method

A. Indetectability of Hidden Information

In steganographic applications, it is required that the existence of hidden information is visually and statistically undetectable. The imperceptibility of hidden information is demonstrated by some experimental results in Section VI. In this section, we show that the hidden information in the stego-images generated by the proposed method cannot be detected by some methods of stego-image analysis.

Intuitively, a general statistical model of palette images can be used to determine whether an image is "normal" or possibly embedded with certain extra information. However, because of the variety of palette images, it is difficult to define a general statistical model for analyzing palette images [2]. Without a general model, detection of hidden information in palette images can only be performed in certain prescribed manners. In [2], some hidden information detection techniques are described and can be categorized as follows:

1) investigation of abnormal palette structures;
2) detection of outstanding pixels;
3) detection of special patterns in palette images.

Because the proposed method does not manipulate image palettes, no abnormal palette structure, such as a special palette color ordering, will be generated. Techniques belonging to Category 1) above therefore have no effect on the proposed method. In addition, the use of the data embeddability proposed in this paper prevents the resulting stego-images from having outstanding pixels which are visually or statistically detectable. Thus, techniques belonging to Category 2) have no effect on the proposed method, either. Finally, one reported method of special pattern detection in watermarked palette images is Maes [17], which checks the histogram of a watermarked image to detect whether it includes special patterns resulting from modification of the palette content. Such patterns, called twin peaks, are produced by some artificially created palette colors that are similar to the original colors in the cover image. But since our method does not alter the palette, such an approach again does not work on the stego-images yielded by our method.

B. Confidentiality of Hidden Information

When data hiding becomes popular and the public gets acquainted with it, people might try illegally to inspect or extract the secret data behind an image. Hence, it is necessary to take appropriate measures to ensure the security of the hidden data, hoping that an unauthorized user will never extract the hidden data successfully or correctly, even if he or she knows in advance that there are secret data behind the stego-image. To protect the embedded data securely, a common way is to use a cryptographic method additionally before data hiding. That is, the secret data are first encrypted with a key, followed by the process of hiding the encryption result in a cover image. This way, though offering an additional level of security protection, does not provide the hidden data with direct protection. It is desired to add security protection directly into the data-hiding process, aiming to increase the difficulty for an unauthorized user to extract, alter, and forge hidden data by manipulating the stego-image. The details of the proposed measures to achieve this goal are described in the following.

Recall that during the proposed secret data-extraction process, the original cover image is not required. Instead, to extract the hidden data from a stego-image correctly, the following three types of information are required:

1) the positions of the data-embeddable pixels determined by the two threshold values T_c and T_d involved in the definition of data embeddability in (4);
2) the order of colors used in defining the color-ordering relationship by (2);
3) the binary outputs of the color-mapping function defined by (3).

If the above three types of information are all discovered illegally, the possibility for the embedded secret data to be extracted out will be increased. The essence of the proposed secret protection measures to secure these three types of critical information is to randomize the involved parameters (the threshold values, orders, and function outputs) through the use of a single secret key and several random number-generating functions. The secret key may be chosen freely by a user to control the random number-generating functions, and, in turn, to yield a corresponding secret data-embedding process. Subsequently, when data extraction is to be conducted, the same secret key is used to yield the corresponding data-extraction process that is basically a reverse of the data-embedding process. We now describe below the detail of the security enhancement measure for protecting each of the three types of information mentioned above.

1) Randomization of Data-Embeddable Pixel Positions: From (4), we see that T_c and T_d are two thresholds that determine pixel embeddability during secret bit embedding. People cannot know whether a pixel is *data embeddable* or not by inspecting the stego-image only; the only way is to try all possible combinations of (T_c, T_d) to see if any combination can result in the extraction of reasonable data. To increase the difficulty of finding the correct combination of (T_c, T_d), when the embeddability of each pixel is checked during the image raster-scanning process, a random combination of (T_c, T_d) is generated by two random number-generating functions f_{rg1}

and f_{rg2}, with the initial seeds of both functions set to a secret key value, say k. In this paper, f_{rg1} is used to generate a value of T_c randomly in the range 1–3, and f_{rg2} in the range 5–50. In this way, the embeddability of each pixel will vary so that the positions of the data-embeddable pixels will become unpredictable. This, in turn, means that only when the secret key is used in the data-extraction process for data-embeddability determination can the embedded data be recovered unambiguously. In addition, it is noted that a random number generator will always yield the same output number or the same number sequence, as long as the input seed key is identical.

Let P_a be the probability for an unauthorized user to discover a single setting of (T_c, T_d) by enumerating all possible combinations of them. It is easy to see that P_a is equal to $[1/(3 \times 46)] \approx 7.2 \times 10^{-3}$ in our case.

2) Randomization of Color-Ordering Relationships: The color-ordering relationship R_{co} defined by (2) is based on an order yielded from comparing the magnitudes of the luminance values of all colors. This order, though unique, is fixed. Making it dynamically defined for each pixel will increase the security protection effect. The method we adopt to achieve this goal is described as follows.

At the beginning of the data embedding or extraction process, all colors, say m ones, in the palette are sorted according to their luminance values based on the color-ordering relationship R_{co} defined by (2). Let the resulting color sequence be $Q_c = c_1, c_2, \ldots, c_m$. We say that color c_j in Q_c has the *order number* j. Whenever a pixel X with color c and order number r is encountered, we use the secret key k to generate a random number sequence $Q' = o_1 o_2, \ldots, o_m$ using a random number-generating function f_{rg3} with each number in Q' being in the range from 1 to m. Then, we assign the new order number of o_j to each color c_j in Q_c to replace its old one j. Accordingly, the new order number for the color c of X with the old order number r is now o_r. Furthermore, the color of each of the four precedent neighbors of X is also assigned a new order number in the meantime.

Now, a new color-ordering relationship R'_{co} between two colors c_i and c_j with new order numbers o_i and o_j, respectively, can be defined as follows:

$$R'_{\mathrm{co}}: \begin{cases} c_i > c_j, & \text{if } o_i > o_j \\ c_i = c_j, & \text{if } o_i = o_j \\ c_i < c_j, & \text{otherwise.} \end{cases} \tag{5}$$

In short, the above definition gives a color-order randomization effect at each image pixel, leading to more assurance of security protection of the resulting data embedding.

Let P_b be the probability for an unauthorized user to discover a single color-ordering relationship by enumerating all possible ones. It can be seen that if there are K distinct colors in the image, P_b is equal to $1/(K!)$, which decreases with the increase of K. Furthermore, even when the image is just with 32 distinct colors, P_b is still quite small, which is about 3.8×10^{-36}.

3) Randomization of Color-Mapping Functions: The color-mapping function f_{cm} specified in (3) is randomized in the following way to increase the security protection of the embedded data. First, the secret key k is used to generate a binary random sequence $W = w_1, w_2, \ldots, w_5$, using a random number-generating function f_{rg4}. Based on the dynamically defined color-ordering relationship described previously, the color-mapping function f_{cm} in (3) is redefined in accordance with W as follows:

$$f'_{\mathrm{cm}}(c, c'_1, \ldots, c'_4) = \begin{cases} w_1, & \text{if } c \geq c'_1 \\ w_2, & \text{if } c'_1 > c \geq c'_2 \\ w_3, & \text{if } c'_2 > c \geq c'_3 \\ w_4, & \text{if } c'_3 > c \geq c'_4 \\ w_5, & \text{otherwise.} \end{cases} \tag{6}$$

Hence, f'_{cm} is different each time when a secret bit is embedded into a data-embeddable pixel. Because f_{rg4} always generates the same random number sequence as long as an identical seed (which is the secret key k) is used, the function f'_{cm}, which is dynamically defined for each pixel during the embedding and the extraction processes, is identical. As a result, the hidden data can be extracted correctly. This way of randomizing the color-mapping function obviously will strengthen the security of the hidden data.

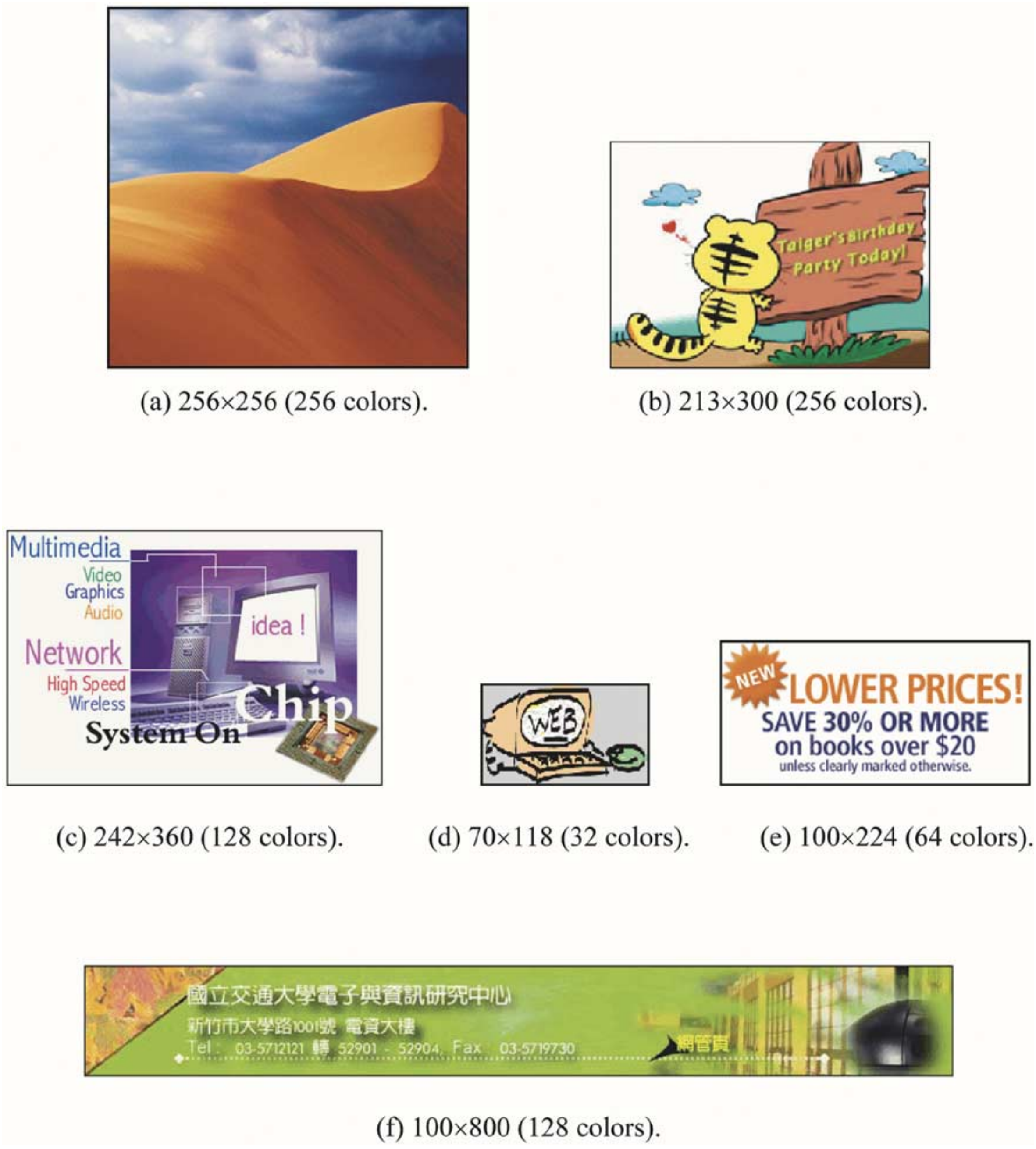
Let P_c be the probability for an unauthorized user to discover a single color-mapping function by enumerating all possible ones. It can be seen that P_c is equal to $2^{-5} \approx 3.1 \times 10^{-2}$.

4) Security Control by Parameter Randomization: The three aforementioned security enhancement measures are based on the use of random number generators with an input secret key k to randomize the three types of information mentioned previously. The random number generators will always yield the same output numbers or the same number sequences, as long as k is identical. This ensures that, in the secret data-extraction process, if the secret key k used in the secret data-embedding process is provided, the three types of information can be correctly discovered without causing synchronization errors. That is, the randomization of the three types of information will not affect the correctness of the proposed method.

In addition, the enhancement measures are designed to protect the three types of information by randomizing them. Assume that all of the above security measures are employed, and that the three types of information all together are randomized L times. This means that for an $H \times W$ image, a randomization operation must be performed for every $(H \times W)/L$ pixels. Accordingly, the probability P_s for an unauthorized user to discover the L sets of the three types of information is $(P_a \times P_b \times P_c)^L$, which is equal to $[(7.2 \times 10^{-3}) \times (3.8 \times 10^{-36}) \times (3.1 \times 10^{-2})]^L \approx (8.5 \times 10^{-40})^L$ for an image with 32 distinct colors.

Now, by selecting an appropriate L to randomize the three types of information, P_s may be decreased to be as small as possible. For example, assume that in a high-security application with a 256 × 256 input image which has 32 distinct colors, P_s is required to be smaller than 10^{-100}. Then, since $(8.5 \times 10^{-40})^3 \approx 6.1 \times 10^{-118} < 10^{-100}$, just three times of randomization will be sufficient. This, in turn, means that the randomization need be conducted once every $\lceil (256 \times 256)/3 \rceil$ pixels in the secret data-embedding process, where $\lceil\rceil$ is the integer ceiling function.

In summary, if all of the above security enhancement measures are implemented into the secret data-embedding process, it


Taiger's Birthday Party Today!
Multimedia
Video
Graphics
Audio
Network
High Speed
Wireless
System On
Chip
idea !
WEB
NEW
LOWER PRICES!
SAVE 30% OR MORE
on books over $20
unless clearly marked otherwise.
國立交通大學電子與資訊研究中心
新竹市大學路1001號 電資大樓
Tel: 03-5712121 轉 52901 - 52904, Fax: 03-5719730
網管頁

(a) 256×256 (256 colors).

(b) 213×300 (256 colors).

(c) 242×360 (128 colors).

(d) 70×118 (32 colors).

(e) 100×224 (64 colors).

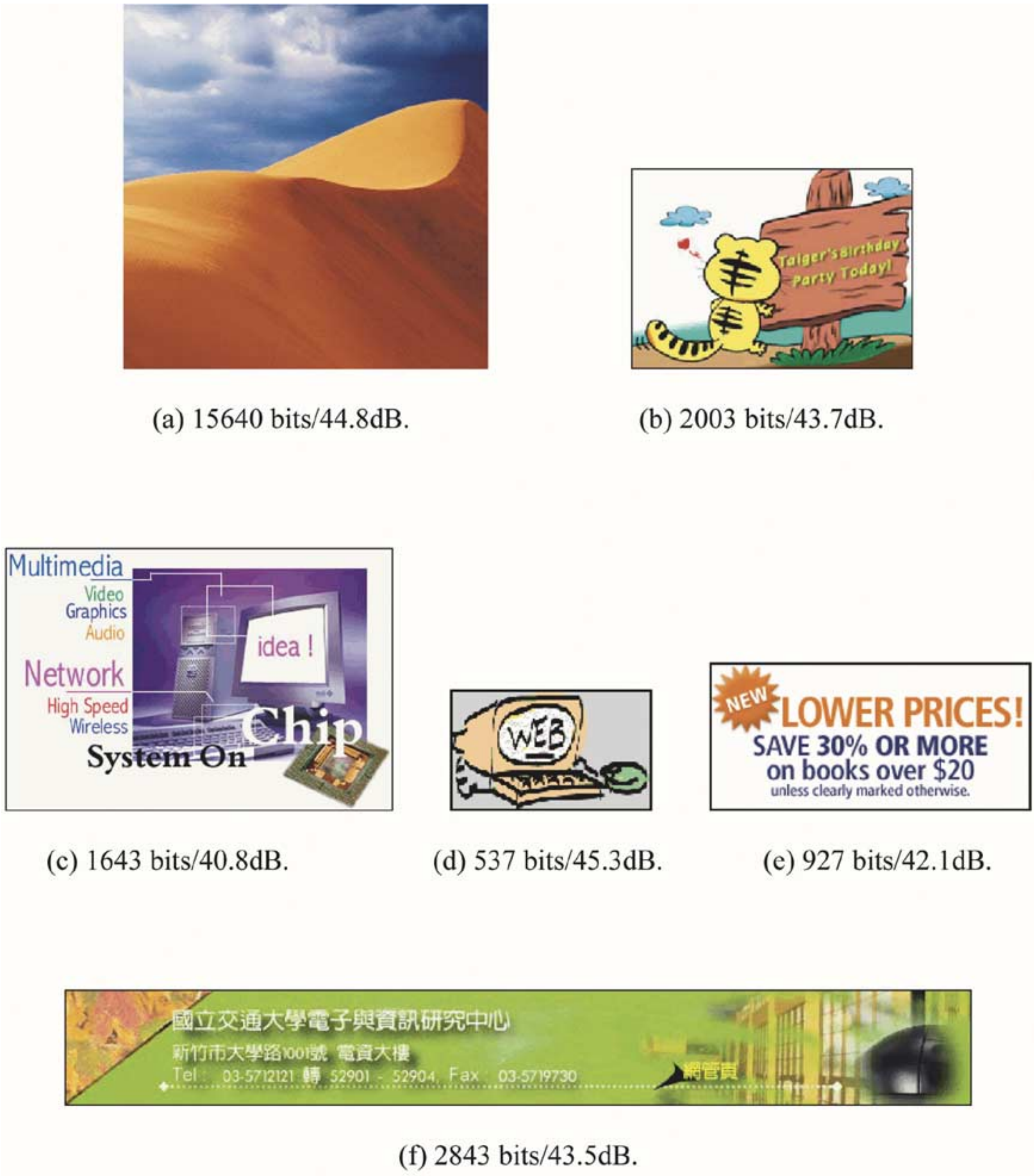
(f) 100×800 (128 colors).

Fig. 3. Typical palette images. (a) 256 × 256 (256 colors). (b) 213 × 300 (256 colors). (c) 242 × 360 (128 colors). (d) 70 × 118 (32 colors). (e) 100 × 224 (64 colors). (f) 100 × 800 (128 colors).

is expected that an unauthorized attempt of extracting the hidden data will be extremely difficult. And this makes the proposed method very feasible for practical data-hiding applications.

VI. Experimental Results

To test the performance of the proposed method, a series of experiments were conducted on a collection of 100 palette images. These images were selected to simulate the use of palette images in real-world applications. According to the definition of the public palette image format, there are at most 256 different colors in a palette image, so that the use of palette images is restricted to applications which do not need a great number of colors. Fig. 3 includes a set of typical palette images in our collections. Fig. 3(a) and (b) are a photograph and a comic picture, respectively. In addition to the purpose of reducing storage space, these kinds of images can be used as thumbnails to serve as previews of real or huge images during browsing. Because more colors are usually required to make photographs and comic pictures visually acceptable, the palettes of these pictures contain more than 128 colors. Fig. 3(c) includes an image containing both graphical and text regions. This image is used as an “image map” in web pages. Users can enter another page by clicking the topics located in the left part of the image. The number of colors used in this kind of image is dependent on the image content. Fig. 3(d) shows a clip art that is usually stored as a palette image. It requires fewer colors than a photograph. Generally, the number


Taiger's Birthday Party Today!
Multimedia
Video
Graphics
Audio
Network
High Speed
Wireless
System On Chip
idea !
WEB
NEW
LOWER PRICES!
SAVE 30% OR MORE
on books over $20
unless clearly marked otherwise.
國立交通大學電子與資訊研究中心
新竹市大學路1001號 電資大樓
Tel : 03-5712121 轉 52901 - 52904 Fax : 03-5719730
網管員

(a) 15640 bits/44.8dB.

(b) 2003 bits/43.7dB.

(c) 1643 bits/40.8dB.

(d) 537 bits/45.3dB.

(e) 927 bits/42.1dB.

(f) 2843 bits/43.5dB.

Fig. 4. Embedding results of the proposed method applied to the images in Fig. 3. (a) 15 640 bits/44.8 dB. (b) 2003 bits/43.7 dB. (c) 1643 bits/40.8 dB. (d) 537 bits/45.3 dB. (e) 927 bits/42.1 dB. (f) 2843 bits/43.5 dB.

of colors used in a clip art is between 32 and 128. Fig. 3(e) and (f) are two images that are mainly composed of text with simple or complex backgrounds. These kinds of images are often used as banners or for advertisements.

The corresponding stego-images, created by the proposed method, are shown in Fig. 4. In our experiment, the threshold values T_c, T_d, and T_v were set to be 2, 20, and 10, respectively. Accordingly, a pixel is data embeddable only when there are at least three distinct colors among its four precedent neighbors, and the maximum color difference between a pixel X and the four precedent neighbors is smaller than 20. The number of bits embedded and the resulting peak signal-to-noise ratio (PSNR) value are shown below each stego-image in the figures. The experimental results show that secret data can be embedded without introducing visual artifacts and extracted correctly by the proposed method.

Furthermore, we also investigated the performance of the proposed method with different threshold values. In our experiment, T_v is set to a half of T_d. Table I illustrates the embedding performance of the proposed method. As expected, the number of bits that can be embedded is increased, when T_c is decreased and T_d is increased. The image quality after data embedding, though, is still very good, with the resulting PSNR higher than 35 dB. On the other hand, when the thresholds T_c and T_d are set in the reverse way, we see that even better image quality was obtained without sacrificing the embedding capacity. This

TABLE I
EMBEDDING PERFORMANCE FOR DIFFERENT VALUES OF T_c AND T_d

Test image in Figure. 3	Maximum number of bits embedded			PSNR (dB)		
	$T_c = 1$ $T_d = 30$	$T_c = 2$ $T_d = 20$	$T_c = 3$ $T_d = 10$	$T_c = 1$ $T_d = 30$	$T_c = 2$ $T_d = 20$	$T_c = 3$ $T_d = 10$
(a) 256×256 (256 colors)	21475	15640	1387	36.9	44.8	58.6
(b) 213×300 (256 colors)	10372	2003	951	35.5	43.7	50.8
(c) 242×360 (128 colors)	11157	1643	485	35.7	40.8	52.3
(d) 70×118 (32 colors)	732	537	101	37.1	45.3	60.6
(e) 100×224 (64 colors)	1072	927	93	36.3	42.1	59.7
(f) 100×800 (128 colors)	9392	2843	1277	36.4	43.5	49.2

shows that the proposed method has a good tradeoff between embedding capacity and image quality, and so is quite flexible and practical for various applications.

VII. CONCLUSIONS

In this paper, a new method for data hiding in palette images has been proposed. Data embeddability of cover image pixels, as well as color-mapping functions based on color-ordering relationships, are defined in this paper. The major idea of the proposed data-embedding process is to modify the colors of data-embeddable image pixels so that the binary outputs of the color-mapping function with the colors of these image pixels as input may be taken as the data to be hidden. The color of a data-embeddable pixel is modified to be an optimal one, which is selected from the color palette and has the least distortion under the color conditions around the pixel. Different from other data-hiding methods that consider only color difference in secret embedding, our method in addition takes spatial properties of the image content into account. Hence, when cover images contain limited colors that are visually uncorrelated, the proposed method can yield embedding results with better visual quality. Indetectability of the secret information embedded by the proposed method has also been analyzed and ensured. Furthermore, several possible security enhancement measures for the proposed method have been proposed. These measures essentially randomize the parameters involved in the proposed data-embedding process, and provide the proposed method with good security protection. Good experimental results were obtained, which show the feasibility of the proposed method.

REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," *Proc. IEEE*, vol. 87, pp. 1062–1078, July 1999.

[2] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*. Boston, MA: Artech House, 2000.

[3] L. M. Marvel, J. C. G. Boncelet, and C. T. Retter, "Spread spectrum image steganography," *IEEE Trans. Image Processing*, vol. 8, pp. 1075–1083, Aug. 1999.

[4] K. Matsui and K. Tanaka, "Video-steganography: How to secretly embed a signature in a picture," in *Proc. IMA Intellectual Property Project*, vol. 1, 1994, pp. 187–205.

[5] M. Wu, H. Yu, and A. Gelman, "Multi-level data hiding for digital image and video," in *Proc. SPIE Photonics East*, vol. 3845, Boston, MA, 1999, pp. 10–21.

[6] M. Kwan. (1998) GIF Colormap Steganography. [Online]. Available: http://www.darkside.com.au/gifshuffle/

[7] R. Machado. (1997) EzStego, Stego Online, Stego. [Online]. Available: http://www.stego.com

[8] H. Repp. (1996) Hide4PGP. [Online]. Available: http://www.rugeley.demon.co.uk/security/hide4pgp.zip

[9] A. Brown. (1996) S-Tools for Windows. [Online]. Available: ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/ s-tools4.zip

[10] C. Maroney. (1994–1997) Hide and Seek. [Online]. Available: ftp://ftp.csua.berkeley.edu/pub/cypherpunks/steganography/hdsk41b.zip

[11] N. F. Johnson and S. Jajodia, "Steganalysis of images created using current steganography software," . New York: Springer-Verlag, 1998, vol. 1525, Lecture Notes in Computer Science, pp. 273–289.

[12] J. Fridrich and R. Du, "Secure steganographic methods for palette images," in *Proc. 3rd Int. Workshop Information Hiding*, Dresden, Germany, 1999, pp. 47–60.

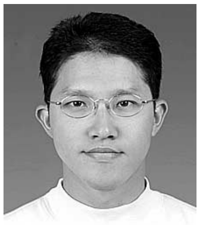
[13] N. F. Maxemchuk and S. Low, "Marking text documents," in *Proc. Int. Conf. Image Processing*, vol. 3, Santa Barbara, CA, 1997, p. 13.

[14] M. Wu, E. Tang, and B. Liu, "Data hiding in digital binary images," in *Proc. Int. Conf. Multimedia and Expo*, vol. 1, New York, NY, 2000, pp. 393–396.

[15] Y.-C. Tseng, Y.-Y. Chen, and H.-K. Pan, "A secure data hiding scheme for binary images," *IEEE Trans. Commun.*, vol. 50, pp. 1227–1231, Aug. 2002.

[16] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Reading, MA: Addison-Wesley, 2002.

[17] M. Maes, *Twin Peaks: The Histogram Attack on Fixed Depth Image Watermarks*. New York: Springer-Verlag, 1998, vol. 1525, Lecture Notes in Computer Science, pp. 290–305.



Chih-Hsuan Tzeng received the B.S. and Ph.D. degrees from the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, in 1995 and 2003, respectively.

During his Ph.D. study, he was with the Computer Vision Laboratory, Department of Computer and Information Science, National Chiao Tung University, as a Research Assistant. He is currently a Postdoctoral Fellow with National Chiao Tung University. His research interests include multimedia data hiding, image compression, pattern recognition, and document image processing.

Zhi-Fang Yang received the Ph.D. degree in computer and information science from National Chiao Tung University, Hsinchu, Taiwan in 1999.

She was a Postdoctoral Fellow with the Institute of Information Science, Academia Sinica, Taiwan from December 1999 to December 2000. Since January 2001, she has been a contracted Assistant Professor with the Department of Computer and Information Science, National Chiao Tung University. Her current research interests include multimedia security and communication, multimedia information processing, and watermarking theory.

Wen-Hsiang Tsai (S'78–M'85–SM'91) received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1979.

He joined the faculty of National Chiao Tung University, Hsinchu, Taiwan in 1979, and is currently a Professor in the Department of Computer and Information Science and the Vice President of the University. He has served as the Head of the Department, the Dean of Academic Affairs of the University, the Chairman of the Chinese Image Processing and Pattern Recognition Society at Taiwan, the Editor of several international journals, and the Editor-in-Chief of the *Journal of Information Science and Engineering*. He has published 293 academic papers, including 120 journal papers and 173 conference papers. His major research interests include image processing, pattern recognition, computer vision, virtual reality, and information hiding. He has supervised the thesis studies of 29 Ph.D. students and 113 master students.

Dr. Tsai has received many awards, including four Outstanding Research Awards, two Special Researcher Awards, and one Distinguished Researcher Award, all from the National Science Council, R. O. C. He was also the recipient of the Academic Award of the Ministry of Education, the 13th Annual Best Paper Award of the Pattern Recognition Society of the USA, and many academic paper awards made by several academic societies. He is a member of the Chinese Image Processing and Pattern Recognition Society, and the Chairman of the Computer Society of IEEE Taipei Section.