IEICE TRANS. INF. & SYST., VOL.E103–D, NO.7 JULY 2020

PAPER Special Section on Information and Communication System Security

A Multilayer Steganography Method with High Embedding Efficiency for Palette Images

Han-Yan WU[†a)], ***Student Member***, **Ling-Hwei CHEN**[†b)], ***and*** **Yu-Tai CHING**[†c)], ***Nonmembers***

SUMMARY Embedding efficiency is an important issue in steganography methods. Matrix embedding $(1, n, h)$ steganography was proposed by Crandall to achieve high embedding efficiency for palette images. This paper proposes a steganography method based on multilayer matrix embedding for palette images. First, a parity assignment is provided to increase the image quality. Then, a multilayer matrix embedding $(k, 1, n, h)$ is presented to achieve high embedding efficiency and capacity. Without modifying the color palette, hk secret bits can be embedded into n pixels by changing at most k pixels. Under the same capacity, the embedding efficiency of the proposed method is compared with that of pixel-based steganography methods. The comparison indicates that the proposed method has higher embedding efficiency than pixel-based steganography methods. The experimental results also suggest that the proposed method provides higher image quality than some existing methods under the same embedding efficiency and capacity.

key words: *embedding efficiency, multilayer matrix embedding, palette images, parity assignment*

1. Introduction

Steganography is a technique used for secret communication. In steganography, a secret message is embedded into an innocent digital object and sent to the receiver over a public channel. The original innocent object is called the cover, and the object with the secret message embedded is called the stego.

Steganography can be categorized into two types: reversible steganography [1]–[8], in which the stego can be recovered after the secret message is extracted, and irreversible steganography [9]–[24], in which the stego cannot be recovered. In this study, we focus on irreversible steganography.

Digital files such as images, videos, and audios can be used as covers. Among them, images are popular because they are widely transmitted on the Internet. Many image-based steganography methods [1]–[24] have been proposed, most of which are applied in raw [1], [2], [9]–[13], jpeg [3]–[6], [14], VQ [7], [15], and absolute moment block truncation coding (AMBTC) [8], [16] images, only a few are applied in palette images [17]–[24].

Palette images [25] have recently gained popularity. GIF is a type of palette images frequently used by younger consumers to communicate and express themselves. GIF animations are also widely used on social network like Tumblr and Twitter, and message applications like Facebook and Line [25], [26]. A palette image includes a palette and an index table. The palette consists of a few (generally no more than 256) colors. Each pixel is represented by a color in the palette, and the index table records each pixel's corresponding color index in the palette. In general, the order of colors in the palette is random.

Because only limited colors are used in a palette image, modifying a pixel's color index may considerably distort the pixel color. In other words, embedding data into a palette image is more challenging than that in other image formats.

S-tools [17], with a capacity of 3 bits per pixel (bpp), is a well-known steganography program for palette images. It produces an unusual color pattern in a palette and enables easy detection [27]. Machado [18] proposed another well-known method, EzStego, with a capacity of 1 bpp. This method assumes that two similar colors have close luminance. EzStego sorts an image palette according to luminance and applies the LSB replacement in each pixel's color index to embed the secret message. However, two colors with close luminance may be very different. Fridrich [19] proposed a method in which the colors of a palette are partitioned into two sets with different parities ($R + G + B$ mod 2) to represent one secret bit. The capacity of this method is 1 bpp. However, this method cannot ensure that the embedding distortion is minimized. Under the same embedding capacity, Fridrich and Du [20] proposed an optimal parity assignment algorithm to improve the image quality. Tanaka et al. [21] proposed an algorithm to partition colors into 2^h sets. Each set represents one type of h-bit secret data. For each pixel, according to the secret data, the closet color in the corresponding set is found to replace the pixel's color. The embedding capacity is increased to h bpp.

All of the aforementioned methods take a pixel as a unit to embed secret data; such methods are called pixel-based methods (PBMs). Certain methods [22]–[24] take a block as a unit to embed secret data; such methods are called block-based methods (BBMs). Imaizumi and Ozawa [22] proposed a BBM to embed h bits into a block. First, the colors in a palette are sorted according to the color Euclidean distance. Next, for each 3×3 block, the total of all the color indices is calculated and divided by 2^h to obtain a remainder. Then, some pixels' color indices are adjusted to make the remainder equal to the value of h secret data bits

Manuscript received August 5, 2019.
Manuscript revised December 18, 2019.
Manuscript publicized April 7, 2020.

[†]The authors are with Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
a) E-mail: hywu.cs02g@nctu.edu.tw (Corresponding author)
b) E-mail: lhchen@cc.nctu.edu.tw
c) E-mail: ytc@cs.nctu.edu.tw

DOI: 10.1587/transinf.2019ICP0005

embedded with the least embedding distortion. The embedding capacity is $h/9$ bpp. Under the same capacity, the aforementioned method provides higher image quality than that provided by the method of Tanaka et al. [21]. Imaizumi et al. [23] adopted a block size of 2×2 to improve the embedding capacity of the method proposed in [22]. Aryal et al. [24] used a block size of 1×3 to further improve the embedding capacity of the method proposed in [23], where the LAB color space replaced the RGB color space to improve the image quality. The embedding capacities of the methods proposed in [23] and [24] were not higher than 1 bpp. Furthermore, the methods do not guarantee that two colors with adjacent color indices are similar, which degrades the image quality after data embedding.

In steganography, imperceptibility (image quality), capacity, and undetectability are three main requirements. Embedding data in an image has a trade-off between imperceptibility (or undetectability) and embedding capacity. The embedding efficiency proposed by Westfeld [28], [29] is one kind of measures for undetectability, and it is defined as the number of embedded random message bits per embedding change. Crandall [30] mentioned that the most obvious method to reduce the possibility of detecting hidden information is to reduce the change density in the cover message. Since decreasing the change density will raise the embedding efficiency, thus, higher embedding efficiency will lower the detectability.

Matrix embedding (t, n, h), which was proposed by Crandall [30], is a high-embedding-efficiency method for palette images. This method is based on blocks for embedding secret data. In the matrix embedding method, for a block of n pixels, at most t pixels are selected and their parity bits are changed to embed h bits through an error-correcting code [e.g., hamming code $(n, n-h)$] [31].

On the basis of matrix embedding, some steganography methods [11]–[13] have been proposed for raw images. Although Crandall mentioned that matrix embedding can be used in palette images, the embedding procedure remains an open research area. The embedding procedure includes two parts: color parity assignment for reducing embedding distortions and error correction code selection for increasing the embedding efficiency and capacity.

This paper proposes a steganography method based on multilayer matrix embedding $(k, 1, n, h)$ to obtain a high image quality, embedding efficiency, and capacity for palette images. Multilayer matrix embedding $(k, 1, n, h)$ is proposed to embed hk secret bits in a $1 \times n$ block by changing at most k pixels. First, a global parity assignment (GPA) is provided to improve the image quality, and then, the matrix embedding $(1, n, h)$ is applied k times to increase the embedding capacity and efficiency. Under the same embedding capacity, the proposed method can provide higher embedding efficiency and image quality than other methods can.

The remainder of this paper is organized as follows. The preliminary information is introduced in Sect. 2. The proposed method is presented in Sect. 3. The embedding efficiency is discussed in Sect. 4. The experimental results are presented in Sect. 5. Finally, the conclusions are presented in Sect. 6.

2. Preliminary Information

In this section, some terminologies and methods referenced for the proposed method, such as the hamming code [31], matrix embedding [30], and parity assignment, are described.

2.1 Hamming Code

The hamming code $(n, n-h)$ is a linear error-correcting code. Through even/odd parity checking, the hamming code can detect and correct one error bit. Given a message m with $n-h$ bits, the hamming code uses an $(n-h) \times n$ code generator matrix G to create n cover bits C, which include m and h redundant bits ($n = 2^h - 1$). The cover bits C are generated as follows:

$$C = m \times G. \tag{1}$$

Assuming that the receiver receives n cover bits C', he would use an $h \times n$ parity check matrix H to obtain an h-bit syndrome z through Eq. (2).

$$z = (H \times C'^{\mathrm{T}})^{\mathrm{T}}. \tag{2}$$

Then, z is converted to decimal z_d, which indicates the error bit location in C'. If no error occurs, $z_d = 0$.

For example, taking $h = 3$ and $n = 7$, the generator matrix G and check matrix H are expressed as follows:

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{3}$$

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{4}$$

If $m = 1011$, C is generated as follows:

$$\begin{aligned} C &= m \times G \\ &= \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \end{aligned}$$

Assuming that the receiver receives $C' = 0110111$ cover bits, they would use Eq. (2) to obtain $z = 101$. Then, z is converted to decimal $z_d = 5$, which indicates that the error occurs at the fifth bit in C'. Finally, the receiver can correct the error bit by flipping the fifth bit to obtain $C = 0110011$.

2.2 Matrix Embedding (t, n, h)

Matrix embedding (t, n, h) is a steganography method with

a high embedding efficiency. Let us assume that a parity assignment schema is adopted to assign a parity bit to each color in the palette of an image. For a $1 \times n$ block B, matrix embedding modifies the parity bits of at most t pixels to embed an h-bit secret message S through an error-correcting code. Because the hamming code $(n, n-h)$ is commonly used in some steganography methods, we use the hamming code in this study. The main idea is to let the h-bit syndrome represent S. In the embedding process, the parity bits from all the n pixels in B are concatenated as C' and Eq. (2) is used to obtain z. Then, Eq. (5) is used to get z'.

$$z' = z \oplus S, \tag{5}$$

where $\oplus$ is the exclusive-or operation. Finally, z' is converted to decimal z'_d. If $z'_d = 0$, no pixel is required to be changed; otherwise, assuming that the color of the z'_d-th pixel in B is c and its parity is p, we find the closest color c' from c with parity $1-p$ and replace c by c'.

In the extraction process, for a $1 \times n$ block B, first, the parity bits from all n pixels in B are concatenated as C. Then, Eq. (6) is used to obtain the secret message S.

$$S = (H \times C^{\mathrm{T}})^{\mathrm{T}}. \tag{6}$$

Note that $C = C' \oplus I_{z'_d}$, where $I_{z'_d} = (0, \ldots, 1, 0, \ldots, 0)$ with the z'_d-th bit = 1.

In the following text, we prove that Eq. (6) is true.

Proof:

$$\begin{aligned}
(H \times C^{\mathrm{T}})^{\mathrm{T}} &= (H \times (C' \oplus I_{z'_d})^{\mathrm{T}})^{\mathrm{T}} \\
&= (H \times (C'^{\mathrm{T}} \oplus I_{z'_d}{}^{\mathrm{T}})^{\mathrm{T}} \\
&= (H \times C'^{\mathrm{T}} \oplus (H \times I_{z'_d}{}^{\mathrm{T}}))^{\mathrm{T}} \\
&= (H \times C'^{\mathrm{T}})^{\mathrm{T}} \oplus (H \times I_{z'_d}{}^{\mathrm{T}})^{\mathrm{T}} \\
&= z \oplus z' \\
&= z \oplus z \oplus S \\
&= S
\end{aligned}$$

Note that the i-th column of H in Eq. (4) presents the binary representation of i, and $H \times I_i{}^{\mathrm{T}}$ is equal to the i-th column of H. Thus, $H \times I_{z'_d}{}^{\mathrm{T}}$ is equal to the binary representation of z'_d, that is, $H \times I_{z'_d}{}^{\mathrm{T}} = z'$.

An example of using the hamming code (7, 4) is provided here for a detailed explanation. On the sender side, assuming $S = 110$, the parity bits from all pixels are first concatenated in a 1×7 block B as C'. Assuming $C' = 0110111$, Eq. (2) is used to obtain $z = 101$ and Eq. (5) is used to obtain $z' = 011$ and $z'_d = 3$. According to z'_d, the color c of the third pixel in B is changed by the closest color c' of c with a parity different from that of c. At the receiver side, for block B, first, the parity bits from all pixels in B are concatenated as C; $C = C' \oplus I_3 = (0110111) \oplus (0010000) = (0100111)$. Then, Eq. (6) is used to obtain the secret message $S = 110$.

2.3 Parity Assignment

Parity assignment is used to reduce the embedding distortion. Some parity assignment schemas [19], [20] assign one parity bit to each color in the palette of a cover image; however, they can embed only one bit in the color of each pixel. Tanaka et al. provided an h-bit parity assignment algorithm [21], where each color is assigned h parity bits such that h bits can be embedded in the color of each pixel. The algorithm contains two parts. In the first part, from the palette of a cover image, the color c_f satisfying Eq. (7) is first assigned a parity $p = 0$.

$$c_f = \arg\min_{c_i \in A}(256^2 r_i + 256 g_i + b_i), \tag{7}$$

where A is the set of all colors in the palette and $c_i = (r_i, g_i, b_i)$ is the color with index i. Then, the color c with the minimal distance from the last assigned color with parity p is assigned the parity $p+1$. The procedure is repeated until 2^h colors are assigned parity. The colors are then grouped into a set E. Assuming that $R = A \backslash E$, $E_p = \{c \text{ with parity } p\}$. In the second part, first, from R, the color c with the minimal distance from the last assigned color is selected. Next, the minimal distance d_p of c from E_p is evaluated as follows:

$$d_p = \min_{c' \in E_p}(\mathtt{dis}(c, c')), \tag{8}$$

where $\mathrm{dis}(c, c')$ is the Euclidean distance between c and c'. Then, take $d_{p'}$ satisfying Eq. (9) and assign parity p' to color c.

$$d_{p'} = \max_{0 \le p < 2^h} d_p. \tag{9}$$

$E_{p'} = E_{p'} \cup \{c\}$ is updated and $R = R \backslash \{c\}$. The procedure is repeated until R is empty.

In the algorithm of Tanaka et al., the color selected to be assigned parity only locally considers the color distances between the last assigned color and those remaining unassigned. Consequently, the image quality is degraded. To address this issue, we provide a GPA algorithm by globally determining all distances between all assigned and unassigned colors. The algorithm is described in the following text.

GPA algorithm

1. Set $A = \{0, \ldots, 255\}$ to be the indices of all the colors in a palette. Set $E = \varnothing$ and $E_p = \varnothing$. Assign parity p to E_p $(p = 0, \ldots, 2^h - 1)$.
2. Use Eq. (10) to calculate the distances $d_{i,j}$ between each pair of colors (c_i, c_j) in the palette for RGB color space.

$$d_{i,j} = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}, \tag{10}$$

   where $c_i = (r_i, g_i, b_i)$ and $c_j = (r_j, g_j, b_j)$.
3. Let D be the set of distances between all color pairs. Find the minimum d_{i^*,j^*} in D. $E_0 = E_0 \cup \{i^*\}$; $E_1 = E_1 \cup \{j^*\}$; $A = A \backslash \{i^*, j^*\}$; and $E = E \cup \{i^*, j^*\}$. Then, set $q = 1$.
4. Let $q = q + 1$. Find the first entry I_q from A for set E_q using Eq. (11).

$$I_q = \arg\min_{j \in A}(\min_{i \in E}\{d_{i,j}\}). \tag{11}$$

$E_q = E_q \cup \{I_q\}$; $A = A \backslash \{I_q\}$; and $E = E \cup \{I_q\}$.

5. Repeat Step 4 until $q = 2^h - 1$.
6. Find an entry I_f in A with minimum distance from E by using the following equation:

$$I_f = \arg \min_{j \in A} (\min_{i \in E} \{d_{i,j}\}). \tag{12}$$

7. In each E_p, find the minimum color distance md_p from I_f by using the following equation:

$$md_p = \min_{i \in E_p} d_{i,I_f}. \tag{13}$$

8. Use Eq. (14) to find parity p', and assign p' to I_f.

$$p' = \arg \max_{p \in [0,\ldots,2^h-1]} md_p. \tag{14}$$

$E_{p'} = E_{p'} \cup \{I_f\}$; $A = A \backslash \{I_f\}$; and $E = E \cup \{I_f\}$.

9. Repeat Steps 6–8 until A is empty

In the aforementioned algorithm, GPA is used to assign h parity bits to each color in the palette of an image, where two most similar colors are assigned different parity bits. In the data embedding, if the parity bits p_b of one color c are changed to p'_b, the possibility of selecting the closest color c' of c with parity p'_b increases. Thus, the image distortion can be decreased.

3. Proposed Methods

In this section, a steganography method based on multilayer matrix embedding $(k, 1, n, h)$ is proposed for palette images. The proposed method contains three processes: parity assignment, embedding, and extraction. In parity assignment, GPA is used to assign k parity bits to each color in an image palette to decrease embedding distortion. In the embedding process, the cover image is divided into several $1 \times n$ blocks. Multilayer matrix embedding $(k, 1, n, h)$ is applied to each block to embed hk secret bits. In the extraction process, for a stego image, each color in the palette is first assigned a k-bit parity through the GPA used in the embedding process. Then, the stego image is divided into several $1 \times n$ blocks. In each block, according to the color parity bits of all pixels, hk secret bits are extracted by applying the multilayer matrix extraction process. The extraction process is described in Sect. 3.3, and the embedding process is described in Sect. 3.2. Multilayer matrix embedding is introduced in the following subsection.

3.1 Multilayer Matrix Embedding $(k, 1, n, h)$ for a Block in a Palette Image

Multilayer matrix embedding $(k, 1, n, h)$ is used to embed hk bits in each block for increasing the embedding capacity and embedding efficiency. For this purpose, first, GPA is used to assign k-bit parity to each color in a palette. Then, matrix embedding $(1, n, h)$ is applied k times to each block according to a certain rule. At the j-th time, h secret bits S_j are embedded by changing the j-th parity bit of at most one pixel with its previous parity bits fixed. The details of the process are described as in the following text.

Multilayer matrix embedding procedure in a block

1. Let $j = 1$ and p^i_j be the lower j-th parity bit of the i-th pixel in the block.
2. Concatenate the j-th parity bit of all pixels in the block to form $C' = (p^1_j, p^2_j, \ldots, p^n_j)$. Apply Eq. (2) to obtain z. Then, use S_j and Eq. (5) to obtain z'.
3. Convert z' to z'_d. If $z'_d = 0$, then go to Step 5.
4. Find the color c' with the minimum color distance from the color of the z'_d-th pixel and with the first j parity bits equal to $(p_1^{z'_d}, p_2^{z'_d}, \ldots, p_{j-1}^{z'_d}, 1 - p_j^{z'_d})$. Then, c' replaces the color of the z'_d-th pixel.
5. If $j < k$, $j = j + 1$. In this case, go to Step 2.

3.2 Embedding Process

In the embedding process, to increase the security, first, the secret message S is encrypted into S' by a stego key K_e. Second, for a cover image I, GPA is used to assign k-bit parity to each color in the palette of I. Third, all pixels of I are permuted to form I' by K_e, and then, I' is divided into several $1 \times n$ blocks and S' is divided into several segments of hk bits. Fourth, for each block B, the multilayer matrix embedding is used for embedding one secret segment of hk bits to form block B'. Finally, all blocks B' are grouped into an image I'_e, and then, an inverse permutation is applied to all pixels of I'_e by using the stego key K_e to obtain a stego image I_s.

3.3 Extraction Process

In the extraction process, for a stego image I_s, first, the GPA used in the embedding process is adopted to assign k-bit parity to each color in the palette of I_s. Second, I_s is permuted to form I'_e by K_e, and then, I'_e is divided into several $1 \times n$ blocks. Third, for each block B', the k-bit parity p^i of the i-th pixel in the block is taken, $i = 1, 2, \ldots, n$, and $p^i = (p^i_1, p^i_2, \ldots, p^i_{k-1}, p^i_k)$. Then, the multilayer matrix extraction is applied to B' to extract hk secret bits $S_{B'}$. The details of the process are described in the following text.

Multilayer matrix extraction in a block

1. Let $j = 1$.
2. Let $C = (p^1_j, p^2_j, \ldots, p^n_j)$ and p^i_j be the lower j-th parity bit of the i-th pixel in the block.
3. Apply Eq. (6) to obtain S_j of h bits.
4. If $j < k$, $j = j + 1$. In this case, go to Step 2.
5. Concatenate all S_j, $j = 1, 2, \ldots, k$, to form the encrypted secret message $S_{B'}$.

Finally, group all $S_{B'}$ to form S' and then use K_e to decrypt S' into S.

3.4 Example

An example is presented to provide a detailed explanation

of the proposed method. Without loss of generality, we skip the secret message encryption and image permutation and consider only one block of an image. Let $k = 2$, $n = 7$, $h = 3$, the embedding bits $S_1 = 101$, $S_2 = 111$ and $c_1, c_2, \ldots, c_7$ be the pixel colors in a block. In the embedding process, first, GPA is used to assign two parity bits to each color in a palette and the parity bits of c_i are (p_1^i, p_2^i). Suppose $(p_1^1, p_2^1) = (1,0)$, $(p_1^2, p_2^2) = (0,0)$, $(p_1^3, p_2^3) = (1,1)$, $(p_1^4, p_2^4) = (1,1)$, $(p_1^5, p_2^5) = (0,0)$, $(p_1^6, p_2^6) = (0,1)$, and $(p_1^7, p_2^7) = (1,1)$. In the first time, $C' = (p_1^1, p_1^2, p_1^3, p_1^4, p_1^5, p_1^6, p_1^7) = (1,0,1,1,0,0,1)$. Obtain $z = 001$ by using Eq. (2), and then, obtain $z' = 100$ by using Eq. (5) and S_1. Since the value of $z' = 100$ and $z'_d = 4$, flip p_1^4 to 0 and find the closest color c'_4 with the first parity bit = 0 for c_4. The color c'_4 replaces c_4. Suppose the parity bits of c'_4 are (0, 0); then, $(p_1^4, p_2^4) = (0,0)$. In the second time, note that, c_4 is replaced by c'_4, so (p_1^4, p_2^4) is updated by (0, 0), then $C' = (p_2^1, p_2^2, p_2^3, p_2^4, p_2^5, p_2^6, p_2^7) = (0,0,1,0,0,1,1)$. Obtain $z = 010$ by using Eq. (2), and then, obtain $z' = 101$ and $z'_d = 5$ by using Eq. (5) and S_2. According to z'_d, flip p_2^5 to 1 and find the closest color c'_5 with parity bits (0, 1) for c_5. The color c'_5 replaces c_5. Finally, the stego block has the pixel colors $c_1, c_2, c_3, c'_4, c'_5, c_6, c_7$.

In the extraction process, let $c_1, c_2, c_3, c'_4, c'_5, c_6, c_7$ be the pixel colors in a stego block. First, GPA is used to assign two parity bits to each color in a palette. Then, the parity bits (p_1^i, p_2^i) of c_i are extracted to obtain $(p_1^1, p_2^1) = (1,0)$, $(p_1^2, p_2^2) = (0,0)$, $(p_1^3, p_2^3) = (1,1)$, $(p_1^4, p_2^4) = (0,0)$, $(p_1^5, p_2^5) = (0,1)$, $(p_1^6, p_2^6) = (0,1)$, and $(p_1^7, p_2^7) = (1,1)$. In the first time, extract $C = (p_1^1, p_1^2, p_1^3, p_1^4, p_1^5, p_1^6, p_1^7) = (1,0,1,0,0,0,1)$ and apply Eq. (6) to obtain $S_1 = 101$. In the second time, extract $C = (p_2^1, p_2^2, p_2^3, p_2^4, p_2^5, p_2^6, p_2^7) = (0,0,1,0,1,1,1)$ and apply Eq. (6) to obtain $S_2 = 111$.

4. Discussion on the Embedding Efficiency

Fridrich et al. [29] defined embedding efficiency as the number of embedded random message bits per embedding change. According to this definition, embedding efficiency (EF) can be expressed as follows:

$$EF = \frac{\texttt{number of embedding bits}}{\texttt{number of embedding pixel changes}}. \tag{15}$$

Some PBMs [17]–[21] use one pixel as an embedding unit to embed secret data. If the first LSB of the color index of a pixel [18] or one parity bit of the color in a pixel [19], [20] is used as the embedding bit, EF can be evaluated as follows:

$$EF = \frac{1}{0 \times \frac{1}{2} + 1 \times \frac{1}{2}} = 2, \tag{16}$$

where 1/2 in the denominator indicates the probability of the embedding pixel being changed or unchanged.

If the first and second parity bits of the color in a pixel are used as the embedding bits [21], EF can be evaluated as follows:

$$EF = \frac{2}{0 \times \frac{1}{4} + 1 \times \frac{3}{4}} = 2.667, \tag{17}$$

where 1/4 and 3/4 in the denominator indicate the probabilities of the embedding pixel being unchanged and changed, respectively.

If the first, second, and third parity bits of the color in a pixel are used as the embedding bits, EF can be evaluated as follows:

$$EF = \frac{3}{0 \times \frac{1}{8} + 1 \times \frac{7}{8}} = 3.428, \tag{18}$$

where 1/8 and 7/8 in the denominator indicate the probabilities of the embedding pixel being unchanged and changed, respectively.

To compare PBMs with the proposed multilayer matrix embedding $(k, 1, n, h)$, the embedding capacity is fixed to be the same as that in our method. In the following text, three cases are discussed.

Case 1: $k = 1$, $n = 7$, and $h = 3$, that is, three bits are embedded in a 1×7 block.

When a PBM is used to embed three bits in a 1×7 block, there exist three subcases: (a) three pixels are selected, where each selected pixel embeds one bit. Since there are eight changing possibilities for the embedding bits of these three pixels: (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), and (1, 1, 1), 0 stands for bit unchanged, 1 stands for bit changed, $EF = \frac{3}{(0\times(1/8)+1\times(3/8)+2\times(3/8)+3\times(1/8))} = 2$; (b) two pixels are selected, where one pixel X_1 embeds one bit and the other X_2 embeds two bits. Since there are eight changing possibilities $(x^1, (x_1^2, x_2^2))$ for the embedding bit of X_1 and the two embedding bits of X_2: $(0, (0,0))$ stands for 0 pixel changed; $(0, (0,1))$, $(0, (1,0))$, $(0, (1,1))$, and $(1, (0,0))$ stand for 1 pixel changed; $(1, (0,1))$, $(1, (1,0))$, and $(1, (1,1))$ stand for 2 pixels changed, $EF = \frac{3}{0\times(1/8)+1\times(4/8)+2\times(3/8)} = 2.4$; and (c) one pixel is selected to embed three bits, $EF = \frac{3}{0\times(1/8)+1\times(7/8)} = 3.428$.

The proposed multilayer matrix embedding $(1, 1, 7, 3)$ can embed three bits per 1×7 block with at most one pixel changed, $EF = 3.428$.

Case 2: $k = 2$, $n = 7$, and $h = 3$, that is, six bits are embedded in a 1×7 block.

When a PBM is used to embed six bits in a 1×7 block, there exist eight subcases. The embedding efficiency of each subcase is listed in Table 1, in which, N_1, N_2, and N_3 stand for the pixel numbers with one, two, and three embedding bits, respectively.

The proposed multilayer matrix embedding $(2, 1, 7, 3)$ can embed six bits per 1×7 block with at most two pixels changed. The matrix embedding method is applied two times. In the first application, three bits are embedded in a 1×7 block with at most one pixel selected as the change pixel. In the second application, another three bits are embedded in the block, with at most one pixel selected as the

change pixel. Let x_1 and x_2 indicate the two pixels selected to be changed, and let x_i indicate the pixel selected in the i-th time ($x_i = 0, 1, \ldots, 7$). There exist three possible cases: (a) no pixel selected to be changed, (b) one pixel selected to be changed, and (c) two pixels selected to be changed. Table 2 lists the probability of each case.

According to Table 2, EF can be evaluated as follows:

$$EF = \frac{6}{0 \times \frac{1}{64} + 1 \times \frac{3\times 7}{64} + 2 \times \frac{42}{64}} = 3.657. \quad (19)$$

Case 3: $k = 3$, $n = 7$, and $h = 3$, that is, nine bits are embedded in a 1×7 block.

When a PBM is used to embed nine bits in a 1×7 block, there exist 10 subcases. The embedding efficiency of each subcase is listed in Table 3.

The proposed multilayer matrix embedding $(3, 1, 7, 3)$ can embed nine bits per 1×7 block with at most three pixels changed. The matrix embedding method is applied three times. In each application, three bits are embedded in a 1×7 block with at most one pixel selected as the change pixel. Let x_1, x_2, and x_3 indicate the three pixels selected to be changed. There exist four possible cases: (a) no pixel is selected to be changed, (b) one pixel is selected to be changed, (c) two pixels are selected to be changed, and (d) three pixels are selected to be changed. Table 4 presents the probability of each case.

According to Table 4, EF can be evaluated as follows:

$$\begin{aligned} EF &= \frac{9}{0 \times \frac{1}{512} + 1 \times \frac{7\times 7}{512} + 2 \times \frac{6\times 42}{512} + 3 \times \frac{210}{512}} \\ &= 3.895. \end{aligned} \quad (20)$$

Under the same capacity, Table 5 summarizes the EF values for the multilayer matrix embedding method and a

Table 1 EF obtained using a PBM for case 2.

N_1	N_2	N_3	EF
6	0	0	2
0	3	0	2.667
0	0	2	3.428
4	1	0	2.181
2	2	0	2.4
3	0	1	2.526
1	1	1	2.824

Table 2 Probabilities of the three cases of possible pixel change for multilayer matrix embedding $(2, 1, 7, 3)$.

Change cases	Changed position		Probability
No. of pixel changed	2nd time	1st time	
0	U/A	U/A	1/64
1	U/A	x	7/64
	x	U/A	7/64
	x	x	7/64
2	y	x	42/64

Table 3 EF obtained using a PBM for case 3.

N_1	N_2	N_3	EF
6	0	1	2.323
3	0	2	2.769
0	0	3	3.428
5	2	0	2.25
3	3	0	2.4
1	4	0	2.571
0	3	1	2.88
2	2	1	2.667
1	1	2	3
4	1	1	2.482

Table 4 Probabilities of the four cases of possible pixel change for multilayer matrix embedding $(3, 1, 7, 3)$.

Change cases	Changed position			Probability
No. of pixel changed	3rd time	2nd time	1st time	
0	U/A	U/A	U/A	1/512
1	U/A	U/A	x	7/512
	U/A	x	U/A	7/512
	x	U/A	U/A	7/512
	U/A	x	x	7/512
	x	U/A	x	7/512
	x	x	U/A	7/512
	x	x	x	7/512
2	x	y	U/A	42/512
	x	U/A	y	42/512
	U/A	x	y	42/512
	x	x	y	42/512
	x	y	x	42/512
	y	x	x	42/512
3	x	y	z	210/512

Table 5 EF values for multilayer matrix embedding $(k, 1, 7, 3)$ and the PBM under the same embedding capacity.

	Multilayer matrix embedding			PBM		
k	1	2	3			
Embedding bits	3	6	9	3	6	9
EF	3.428	3.657	3.895	2 ~ 3.428	2 ~ 3.428	2.25 ~ 3.428

PBM. In all the cases, the EF values for multilayer matrix embedding are higher than those for PBMs. Later, in the experimental results, we will see that under the same capacity, multilayer matrix embedding will have higher image quality than those of PBMs, thus the multilayer matrix embedding is less detectable.

5. Experimental Results

In general, under a given target scenario/application, to show the actual effectiveness of the developed method is to do comparisons with other existing ones. The embedding capacity is usually fixed, the higher image quality and embedding efficiency for the proposed steganographic method are desired. Under this consideration, in this paper, we provided some comparisons among the proposed method, Tanaka et al.'s [21] and others [23], [24] to show the effectiveness of our method. Four experiments were conducted using five images from the University of Southern California Signal and Image Processing database [32] (Fig. 1). The embedded secret data were produced using a pseudorandom number generator. The image quality was measured using the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [33], and color-based structural similarity index (CSSIM) [34].

(a) Airplane (b) Baboon (c) Lake (d) Lena (e) Peppers

Fig. 1 Test images.

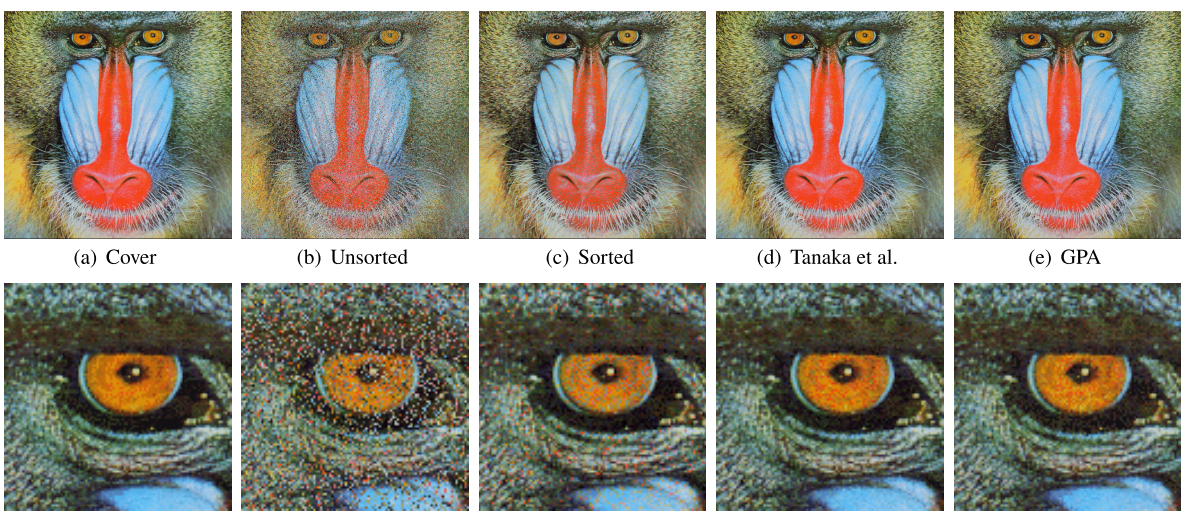
In the first experiment, the embedding process of the proposed multilayer matrix embedding $(k, 1, 7, 3)$ was used and four different parity assignments were conducted for image quality comparison with embedding capacity $3k/7$ bpp. The first assignment (called unsorted) uses the LSBs of a color index in the original palette as parity bits. The second assignment (called sorted) uses the LSBs of a color index in

Table 6 Comparison of the PSNR values with embedding capacity $3k/7$ bpp.

	Unsorted			**Sorted (luminance)**			**Tanaka et al.**			**Proposed method (GPA)**		
k	1	2	3	1	2	3	1	2	3	1	2	3
Airplane	19.54	17.17	15.41	32.43	29.72	28.34	44.65	40.32	37.41	44.47	40.53	37.40
Baboon	19.42	16.82	15.32	23.57	20.97	19.59	37.66	33.83	31.14	37.69	33.93	31.22
Lake	17.94	15.37	13.95	29.68	27.18	25.78	40.41	36.44	33.65	40.42	36.56	33.77
Lena	21.32	18.32	16.66	31.23	28.53	27.27	42.88	39.13	36.12	42.95	39.10	36.21
Peppers	19.85	17.25	15.28	27.27	24.03	22.37	40.42	36.25	33.18	40.47	36.28	33.10

Table 7 Comparison of the CSSIM values with embedding capacity $3k/7$ bpp.

	Unsorted			**Sorted (luminance)**			**Tanaka et al.**			**Proposed method (GPA)**		
k	1	2	3	1	2	3	1	2	3	1	2	3
Airplane	0.280	0.174	0.117	0.980	0.966	0.947	0.986	0.974	0.959	0.986	0.975	0.956
Baboon	0.554	0.380	0.271	0.893	0.824	0.762	0.969	0.942	0.907	0.969	0.943	0.908
Lake	0.308	0.173	0.117	0.949	0.917	0.881	0.980	0.957	0.919	0.980	0.956	0.924
Lena	0.348	0.189	0.118	0.949	0.909	0.878	0.980	0.959	0.927	0.980	0.959	0.928
Peppers	0.306	0.165	0.090	0.921	0.849	0.785	0.973	0.945	0.904	0.974	0.944	0.900



(a) Cover (b) Unsorted (c) Sorted (d) Tanaka et al. (e) GPA

(f) Enlarged (a) (g) Enlarged (b) (h) Enlarged (c) (i) Enlarged (d) (j) Enlarged (e)

Fig. 2 Stego images obtained using the four parity assignments for baboon with $k = 3$.

a sorted palette (according to the color luminance) as parity bits [18]. The third assignment uses Tanaka et al.'s method. And the fourth assignment uses GPA. Note that each color is assigned k parity bits in Tanaka et al.'s and GPA assignment.

Under the same embedding capacity and efficiency, Tables 6 and 7 indicate the PSNRs and CSSIMs of stego images of using the four parity assignments. These tables show that GPA provides the highest image quality than unsorted and sorted assignment, and in most cases, it is better than Tanaka et al.

Figure 2 displays the stego images obtained when using the four parity assignments for Baboon with $k = 3$. The stego image (Figs. 2 (e) and 2 (j)) obtained using GPA is less noisy than the other images and is also the most similar to the cover image displayed in Figs. 2 (a) and 2 (f).

In the second experiment, under the same embedding capacity 9/7 bpp, the image qualities obtained with the proposed method $(3, 1, 7, 3)$ and Tanaka et al. [21] were compared. In the experiment, for each 1×7 block, three pixels are randomly picked, and Tanaka et al.'s method is applied to each picked pixel to embed h bits with $h = 3$. The details are described in Sect. 4. Table 8 lists the image qualities with nine bits embedded in each 1×7 block (i.e. capacity 9/7 bpp). This table indicates that the proposed method provides higher image quality than Tanaka et al.'s method under the same embedding capacity.

Table 8 PSNR and CSSIM values for the proposed and Tanaka et al.'s methods under 9/7 bpp.

	Multilayer matrix embedding (3, 1, 7, 3)		**Tanaka et al.'s method**	
	PSNR	CSSIM	PSNR	CSSIM
Airplane	37.40	0.956	35.13	0.945
Baboon	31.22	0.908	29.77	0.887
Lake	33.77	0.924	32.33	0.898
Lena	36.21	0.928	34.57	0.904
Peppers	33.10	0.900	31.56	0.873

In the third experiment, the image qualities obtained with BBMs were analyzed. Imaizumi et al.'s method [23], which embeds three bits into a 2×2 block by at most changing four pixels, and Aryal et al.'s method [24], which embeds three bits into a 1×3 block by at most changing three pixels, were adopted to perform a comparison. In [24], the PSNRs and SSIMs for capacities of 10800 and 21600 bits are provided. Therefore, we consider these two capacities in the experiment. Ten test images from [35] were used, and the comparison results are presented in Tables 9 and 10. The results indicate that the proposed method provides better image quality than the methods of Imaizumi et al. and Aryal et al. Furthermore, our proposed method using $k = 1$ or $k = 2$, with embedding capacity 21600 bits always keeps higher image quality than Imaizumi et al. and Aryal et al. with embedding capacity 10800 bits. On the other hand, we also find that under the same capacity, our proposed method will have higher image quality using smaller k.

In the fourth experiment, the proposed method was applied to GIF animations. Two GIF animations (Rabbit and Cat) obtained from Giphy [25] were used. The Rabbit and Cat animations had frame numbers of 17 and 4, respectively.

Table 9 PSNR comparisons with capacities of 10800 ($n1$) and 21600 ($n2$) bits.

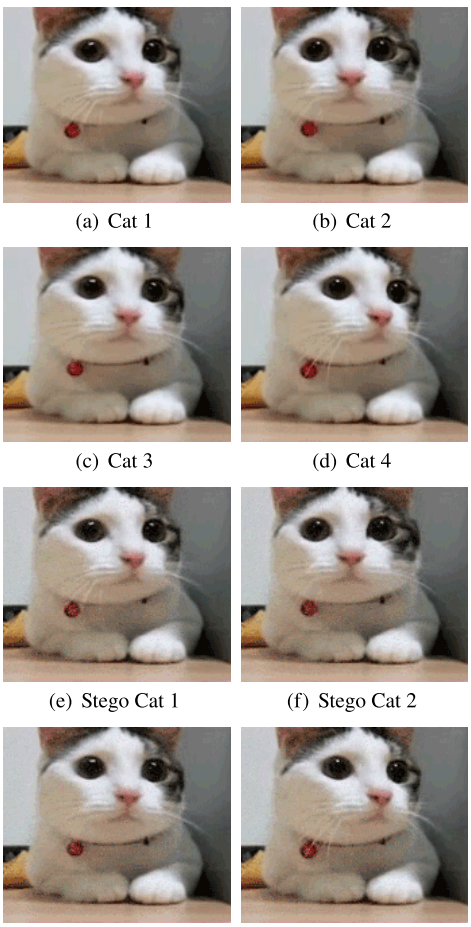
	Imaizumi et al.		**Aryal et al.**		**Multilayer matrix embedding (k, 1, 7, 3)**					
					$k = 1$		$k = 2$		$k = 3$	
Capacity	$n1$	$n2$	$n1$	$n2$	$n1$	$n2$	$n1$	$n2$	$n1$	$n2$
Aerial	39.61	36.81	41.96	38.87	46.51	43.46	45.25	42.18	43.87	40.91
Airplane	40.54	37.76	43.13	40.15	49.19	46.40	47.31	44.71	46.33	43.21
Balloon	40.73	37.68	43.49	40.62	49.58	46.54	48.27	45.62	46.96	44.13
Couple	39.18	35.98	41.87	38.77	48.29	45.19	47.32	44.18	46.46	42.98
Earth	43.38	40.54	45.59	42.54	52.09	49.09	51.49	48.34	49.89	46.74
Girl	36.23	32.88	40.34	37.30	46.08	43.00	45.12	42.04	43.91	40.84
Splash	40.30	37.24	42.93	39.83	48.81	45.70	47.55	44.46	45.89	42.69
Parrots	35.77	32.48	39.33	36.30	43.98	40.97	43.10	39.91	41.57	38.53
Pepper	36.58	33.67	40.34	37.19	45.09	42.29	43.68	40.58	42.02	39.33
Lake	38.57	35.68	41.32	38.41	46.42	43.33	45.49	42.43	44.29	41.33

Table 10 SSIM comparisons with capacities of 10800 ($n1$) and 21600 ($n2$) bits.

	Imaizumi et al.		**Aryal et al.**		**Multilayer matrix embedding (k, 1, 7, 3)**					
					$k = 1$		$k = 2$		$k = 3$	
Capacity	$n1$	$n2$	$n1$	$n2$	$n1$	$n2$	$n1$	$n2$	$n1$	$n2$
Aerial	0.862	0.729	0.902	0.801	0.998	0.996	0.997	0.994	0.996	0.993
Airplane	0.825	0.670	0.846	0.709	0.999	0.997	0.998	0.997	0.997	0.995
Balloon	0.822	0.674	0.867	0.751	0.997	0.994	0.996	0.993	0.995	0.990
Couple	0.849	0.710	0.896	0.799	0.995	0.990	0.994	0.988	0.993	0.985
Earth	0.828	0.665	0.882	0.765	0.999	0.997	0.999	0.998	0.997	0.994
Girl	0.722	0.585	0.840	0.706	0.996	0.992	0.995	0.990	0.993	0.986
Splash	0.835	0.683	0.871	0.753	0.997	0.994	0.996	0.992	0.994	0.988
Parrots	0.825	0.681	0.854	0.723	0.992	0.985	0.990	0.982	0.989	0.978
Pepper	0.831	0.699	0.871	0.754	0.995	0.991	0.995	0.990	0.993	0.986
Lake	0.827	0.685	0.863	0.755	0.997	0.995	0.997	0.993	0.996	0.990

(a) Rabbit (b) Cat

Fig. 3 First frames of two GIF animations.



(a) Cat 1 (b) Cat 2

(c) Cat 3 (d) Cat 4

(e) Stego Cat 1 (f) Stego Cat 2

(g) Stego Cat 3 (h) Stego Cat 4

Fig. 4 The cat animation and its stego images.

Table 11 PSNR and CSSIM values for the two animations with 9/7 bpp.

	Rabbit	Cat
PSNR	45.04	37.93
CSSIM	0.950	0.809

The first frame of each animation is displayed in Fig. 3. Figures 4 (a)–4 (d) illustrate all the frames of the Cat animation. The stego animations with multilayer matrix embedding $(3, 1, 7, 3)$ are depicted and the capacity is 9/7 bpp in Figs. 4 (e)–4 (h). Table 11 presents the PSNRs and CSSIMs of each stego animation. Figure 4 and Table 11 indicate that the proposed method maintains good image quality for each animation.

6. Conclusions

This paper provides a novel steganography method that uses the proposed multilayer matrix embedding technique and GPA. Under the same embedding capacity, the embedding efficiency of the proposed method was higher than that of PBMs, this makes the proposed method less detectable. Moreover, the experimental results indicated that the proposed method can provide higher image quality than PBM [21] and BBMs [23], [24] under the same embedding capacity. On the other hand, the experimental results show that under the same embedding efficiency, the proposed method using GPA can provide higher image quality than that using other parity assignments. Finally, the proposed method exhibited good performance for GIF animations.

Acknowledgments

This work was supported in part by National Science Council, Republic of China, under the grant number MOST 105-2221-E-009-121.

References

[1] K.-H. Jung, "A high-capacity reversible data hiding scheme based on sorting and prediction in digital images," Multimed. Tools Appl., vol.76, no.11, pp.13127–13137, 2017.
[2] H. Nyeem, "Reversible data hiding with image bit-plane slicing," 20th Int. Conf. Comp. and Inf. Technol. (ICCIT), Dhaka, Bangladesh, Dec. 2017. DOI: 10.1109/ICCITECHN.2017.8281763
[3] C.-N. Yang, C. Kim, and Y.-H. Lo, "Adaptive real-time reversible data hiding for JPEG images," Journal of Real-Time Image Process., vol.14, no.1, pp.147–157, 2018.
[4] F.T. Wedaj, S. Kim, H.J. Kim, and F. Huang, "Improved reversible data hiding in JPEG images based on new coefficient selection strategy," EURASIP Journal on Image and Video Process., vol.2017, no.63, 2017. DOI: 10.1186/s13640-017-0206-1
[5] C. Lu, B. Ou, and X. Quan, "Reversible data hiding for JPEG images based on channel selection," Proc. 3rd Int. Conf. Multimed. Syst. and Signal Process., pp.56–63, April 2018.
DOI: 10.1145/3220162.3220189
[6] Z. Qian, S. Dai, and B. Chen, "Reversible data hiding in JPEG images using ordered embedding," KSII Trans. Internet and Inf. Syst., vol.11, no.2, pp.945–958, 2017.
[7] P.-H. Cheng, K.-C. Chang, and C.-L. Liu, "A reversible data hiding scheme for VQ indices using histogram shifting of prediction errors," Multimed. Tools Appl., vol.76, no.4, pp.6031–6050, 2017.
[8] W. Hong, Y.-B. Ma, H.-C. Wu, and T.-S. Chen, "An efficient reversible data hiding method for AMBTC compressed images," Multimed. Tools Appl., vol.76, no.4, pp.5441–5460, 2017.
[9] X. Zhang, Z. Sun, Z. Tang, C. Yu, and X. Wang, "High capacity data hiding based on interpolated image," Multimed. Tools Appl., vol.76, no.7, pp.9195–9218, 2017.
[10] M. Hussain, A.W.A. Wahab, A.T.S. Ho, N. Javed, and K.-H. Jung, "A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement," Signal Process.: Image Commun., vol.50, pp.44–57, Feb. 2017.
[11] W. Zhang, S. Wang, and X. Zhang, "Improving embedding efficiency of covering codes for applications in steganography," IEEE Commun. Lett., vol.11, no.8, pp.680–682, 2007.
[12] C.-C. Chang, T.D. Kieu, and Y.-C. Chou, "Using nearest covering codes to embed secret information in gray scale images," Proc.

2nd Int. Conf. Ubiquitous Inf. Management and Commun., Suwon, Korea, pp.315–320, Jan. 2008. DOI: 10.1145/1352793.1352860
[13] Z. Cao, Z. Yin, H. Hu, X. Gao, and L. Wang, "High capacity data hiding scheme based on (7, 4) Hamming code," SpringerPlus 5, no.175, pp.1–15, 2016. DOI: 10.1186/s40064-016-1818-0
[14] Y. Guo, X. Cao, R. Wang, and C. Jin, "A new data embedding method with a new data embedding domain for JPEG images," 2018 IEEE 4th Int. Conf. Multimed. Big Data (BigMM), Xi'an, China, pp.1–5, Sept. 2018.
[15] W. Hong, X. Zhou, D.-C. Lou, T.S. Chen, and Y. Li, "Joint image coding and lossless data hiding in VQ indices using adaptive coding techniques," Inf. Sci., vol.463–464, pp.245–260, Oct. 2018.
[16] W. Hong, T.S. Chen, Z. Yin, B. Luo, and Y. Ma, "Data hiding in AMBTC images using quantization level modification and perturbation technique," Multimed. Tools Appl., vol.76, no.3, pp.3761–3782, Feb. 2017.
[17] S-Tools, https://www.compzets.com/view-upload.php?id=61&action=view, accessed April 6 2018.
[18] R. Machado, EZ Stego, http://www.stego.com
[19] J. Fridrich, "A new steganographic method for palette-based image," Proc. IS&T PICS Conf., Savannah, GA, pp.285–289, 1998.
[20] J. Fridrich and R. Du, "Secure steganographic methods for palette images," Lecture Notes in Comp. Sci., vol.1768, pp.47–60, 2000.
[21] G. Tanaka, N. Suetake, and E. Uchino, "A steganographic method realizing high capacity data embedding for palette-based images," Proc. Int. Workshop on Smart Info-Media Syst. in Asia, Osaka, Japan, pp.92–95, Oct. 2009.
[22] S. Imaizumi and K. Ozawa, "Multibit embedding algorithm for steganography of palette-based images," Lecture Notes in Comp. Sci., vol.8333, pp.99–110, 2013.
[23] S. Imaizumi and K. Ozawa, "Palette-based image steganography for high-capacity embedding," Bull. Soc. Photogr. Imag. Japan, vol.25, no.1, pp.7–11, 2015.
[24] A. Aryal, K. Motegi, S. Imaizumi, and N. Aoki, "Improvement of multi-bit information embedding algorithm for palette-based images," ISC 2015, Trondheim, Norway, vol.9290, pp.511–523, 2015.
[25] Giphy, https://giphy.com/, accessed May 29 2018.
[26] K.M. Miltner and T. Highfield, "Never gonna GIF you up: analyzing the cultural significance of the animated GIF," Social Media + Society, vol.3, no.3, pp.1–11, 2017.
[27] N.F. Johnson and S. Jajodia, "Steganalysis of images created using current steganography software," Lecture Notes in Comp. Sci., vol.1525, pp.273–289, 1998.
[28] A. Westfeld, "F5—a steganographic algorithm," Lecture Notes in Comp. Sci., vol.2137, pp.289–302, Springer, Heidelberg, 2001.
[29] J. Fridrich, P. Lisoněk, and D. Soukal, "On steganographic embedding efficiency," Proc. 8th Int. Workshop Inf. Hiding, New York, vol.4437, pp.282–296, July 2006
[30] R. Crandall, "Some notes on steganography," posted on steganography mailing list, 1998, http://dde.binghamton.edu/download/Crandall_matrix.pdf, accessed May 29 2018.
[31] R.W. Hamming, "Error detecting and error correcting codes," The Bell Syst. Technical Journal, vol.29, no.2, pp.147–160, April 1950.
[32] Image database, http://sipi.usc.edu/database/, accessed April 6 2018.
[33] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Trans. Image Process., vol.13, no.4, pp.600–612, 2004.
[34] M.A. Hassan and M.S. Bashraheel, "Color-based structural similarity image quality assessment," 8th Int. Conf. Inf. Technol. (ICIT), Jordan, pp.691–696, May 2017. DOI: 10.1109/ICITECH.2017.8079929
[35] SIDBA, http://www.ess.ic.kanagawa-it.ac.jp/app_images_j.html, accessed Sept. 7 2018.

Han-Yan Wu received the B.S. in Department of Multimedia and Game Design and M.S. degrees in Department and Graduate Institute of Information Management from Yu Da University of Science and Technology, Taiwan, in 2011 and 2013. He is currently pursuing the Ph.D. degree at the Institute of computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan. His research interests include image processing and information security.

Ling-Hwei Chen received M.S. degree in Applied Mathematics from National Tsing Hua University, Taiwan in 1977 and Ph.D. degree in Computer Engineering from National Chiao Tung University, Taiwan in 1987. From 1977 to 1979 she worked in Chung-Shan Institute of Science and Technology. From 1979 to 1981 she worked in Electronic Research and Service Organization, Industry Technology Research Institute. From 1981 to 1983 she worked in Institute of Information Industry. She is now a Professor of the Department of Computer Science at National Chiao Tung University. Her current research interests include pattern recognition, Multimedia compression, content-based retrieval and Multimedia Steganography.

Yu-Tai Ching received the B.S. degree in Industrial Engineering from Tsing Hua University, Taiwan, R.O.C., in 1980, and the M.S. and Ph.D. degrees in Computer Science from Northwestern University, Evanston, IL, in 1983 and 1987. He is currently a Professor in the Department of Computer Science at National Chiao Tung University, Hsinchu, Taiwan. His research interests are design and analysis of computer algorithms, medical image analysis, and computer graphics.