Journal of Information Hiding and Multimedia Signal Processing ©2010 ISSN 2073-4212
Ubiquitous International Volume **1**, Number **1**, January **2010**

A Novel Information Concealing Method Based on Exploiting Modification Direction

Zhi-Hui Wang

School of Software
Dalian University of Technology
Dalian, Liaoning, China
wangzhihui1017@yahoo.cn

The Duc Kieu

Department of Information Engineering and Computer Science
Feng Chia University
Taichung 401724, Taiwan, R.O.C.
ktduc0323@yahoo.com.au

Chin-Chen Chang

Department of Information Engineering and Computer Science
Feng Chia University
Taichung 401724, Taiwan, R.O.C.
Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan, R.O.C.
ccc@cs.ccu.edu.tw

Ming-Chu Li

School of Software
Dalian University of Technology
Dalian, Liaoning, China
limingchu@yahoo.com

Received February 2009; revised June 2009

Abstract. *A steganographic scheme with a very good visual quality of stego images was proposed by Zhang and Wang. However, the maximum hiding capacity of this method is 1 bit per pixel (bpp). To improve the hiding capacity of Zhang and Wang's method, we propose a steganographic scheme. The proposed method embeds 2K secret digits in the 5-ary notational systems into each group of (2K + 1) cover pixels, where K is a positive integer. Thus, the maximum hiding capacity of the proposed method can approach 2 bpp. The experimental results show that the PSNR values of the proposed method are always higher than 45 dB at the hiding capacity of 1.99 bpp for all test images.*
Keywords: Steganography, Information hiding, Data embedding.

1. **Introduction.** The development of network technologies and digital devices make digital data distributions easy and fast. However, distributing digital data over public networks such as the Internet is unsafe due to copy violation, counterfeiting, forgery, and fraud. Therefore, the protection of digital data, especially for confidential data, becomes a critical issue. Traditionally, secret data can be protected by cryptographic methods such as DES [1]. The drawback of cryptography is that cryptography can protect the

secret data in transit, but once they have been decrypted, the content of the secret data has no further protection [2]. Alternatively, confidential data can be protected by using information hiding techniques. An information hiding system (also called data hiding or data embedding) embeds secret data into a cover object (e.g., an image, audio, video, or written text) to obtain an embedded object (also called a watermarked object in watermarking applications or a stego object in steganographic applications). For more secure, a cryptographic technique can be utilized to encrypt the secret data prior to embedding.

In general, information hiding consists of digital watermarking and steganography [3]. Watermarking is used for copyright protection, broadcast monitoring, transaction tracking, etc. A watermarking scheme imperceptibly alters a cover object to conceal a message related to the cover object (e.g., owner's identifier) [2]. The robustness (i.e. the ability to resist certain malicious attacks such as common signal processing operations) of digital watermarking schemes plays an important role. In contrast, steganography is used for convert communications. A steganographic method undetectably modifies a cover object to conceal a secret message [2]. Thus, steganographic methods can hide the very existence of secret communications.

In the literature, secret data can be embedded into three domains, namely spatial domain [4], compressed domain [5], and transformed domain [6]. In spatial domain, more redundant spaces are available to secret data embedding so high hiding capacity (also called data embedding capacity) can be obtained, and less time is needed for embedding and extracting processes. Nevertheless, data hiding schemes in spatial domain are vulnerable to common attacks such as statistical steganalysis. As for compressed domain, a cover image is first compressed by using some compression method such as vector quantization (VQ) compression method to get VQ indices (also called index table). Then, secret data are hidden into the VQ indices. In general, data hiding schemes in compressed domain are robust against some attacks such as common image processing operations and these schemes are suitable for low bandwidth transmission channels because less data are transmitted. However, the embedding capacity of these schemes is low and additional time is needed for compression and decompression procedures. With respect to transformed domain, a transformation (e.g. discrete cosine transform [6], discrete wavelet transform [7], or discrete Fourier transform [8]) is first performed on a cover image to obtain frequency coefficients. Then, secret data are embedded into significant coefficients. Generally speaking, information hiding schemes in transformed domains are robust against common image processing operations such as JPEG compression, rotation, and cropping. However, embedding capacity of these schemes is low because only a few significant coefficients can be employed to embed secret data. In addition, additional time is needed for transformation and inverse transformation procedures.

Data embedding schemes can be classified into two types, namely irreversible schemes [9, 13] and reversible schemes [10]. With regard to irreversible schemes, the original cover image can not be restored after secret data have been extracted. In contrast, in reversible schemes, the original cover image can be completely recovered after secret data have been extracted. Reversible data hiding schemes are very useful for healthcare industry and online content distribution systems.

The basic requirements of a data hiding scheme are the visual quality of embedded images (also called visual quality for short), hiding capacity (also called payload), and robustness [11]. A data hiding scheme with low image distortion is more secure than that with high distortion because it does not raise any suspicions of adversaries. A data hiding system with high payload (i.e., the number of bits which can be embedded into a cover

pixel) is expected because more secret data can be exchanged. The robustness is particularly important to robust watermarking applications but it is technically challenging to achieve the robustness requirement for high payload data hiding systems. Visual quality, hiding capacity, and robustness are the conflicting factors (i.e., inversely proportional). Thus, the tradeoff between the three factors above varies from application to application, depending on users' requirements and application domains. Consequently, different techniques are utilized for different applications. Therefore, a class of data hiding schemes is needed to span the range of possible applications.

A simple and well-known information hiding scheme in spatial domain is the least significant bit (LSB) replacement method proposed by Turner in 1989 [12]. In the LSB replacement scheme, binary secret bits are embedded into a cover image by replacing the LSBs of cover pixels with secret bits to obtain a stego image. This method either increases even-valued pixels by one or keeps them unchanged and odd-valued pixels are either kept intact or decreased by one. This asymmetric modification makes the method vulnerable to common attacks. The LSB replacement method was recently improved by the LSB matching revisited method proposed by Mielikainen in 2006 [4]. In the LSB matching revisited scheme, the binary function and four embedding rules are used to embed two secret bits into a cover pixel pair. Specifically, Mielikainen's method has the hiding capacity of 1 bit per pixel (bpp) and good visual quality measured by peak signal-to-noise ratio (PSNR) by using a binary function g. The embedding process of Mielikainen's scheme is described as follows. The secret data in binary form $S = b_1 b_2 ... b_{LS}$, where LS is the length of S, is partitioned into secret bit pairs (b_i, b_{i+1})'s, where $i = 1, 3, ..., LS - 1$. Let us assume that a grayscale cover image X sized $H \times W$ is $X = \{x_1, x_2, ..., x_{H \times W} \mid x_i \in [0, 255], 1 \leq i \leq H \times W\}$. The grayscale cover image X is partitioned into cover pixel pairs (x_i, x_{i+1})'s. During the embedding process, each secret bit pair (b_i, b_{i+1}) is hidden into one cover pixel pair (x_i, x_{i+1}) at a time by adding and subtracting x_i or x_{i+1} by one, or keeping x_i or x_{i+1} unmodified. These embedding rules are determined by the secret bit pair (b_i, b_{i+1}), least significant bit of x_i (i.e., LSB(x_i)), the returned value of the binary function g (i.e., $g(x_i, x_{i+1})$ or $g(x_i - 1, x_{i+1})$), and whether the x_{i+1}'s grayscale value is even or odd. The aforementioned binary function g is defined as $g(x_i, x_{i+1}) = \text{LSB}(\lfloor \frac{x_i}{2} \rfloor + x_{i+1})$.

However, the modification directions of Mielikainen's scheme are not fully explored. To fully exploit the modification directions of Mielikainen's scheme, Zhang and Wang proposed the steganographic scheme [13]. Zhang and Wang's method embeds one secret digit d in the base-$(2n + 1)$ notational system into a group of n cover pixels. Therefore, theoretically, the hiding capacity of Zhang and Wang's method is $\log_2(2n + 1)/n$ bpp. When n equals 2, this method achieves its maximum hiding capacity of 1 bpp. To improve the hiding capacity of Zhang and Wang's method, we propose a steganographic scheme with the maximum hiding capacity of about 2 bpp.

The remaining of the paper is organized as follows. The review of Zhang and Wang's scheme is presented in Section 2. Section 3 describes the proposed scheme in detail. The experimental results and discussions are shown in Section 4. Finally, some conclusions are made in Section 5.

2. **Zhang and Wang's Method.**

To fully exploit the modification directions of Mielikainen's scheme, Zhang and Wang proposed a new exploiting modification direction (EMD) method [13]. The embedding process of the EMD scheme is summarized as follows. Assume that a grayscale cover image X sized $H \times W$ is $X = \{x_1, x_2, ..., x_{H \times W} \mid x_i \in [0, 255], 1 \leq i \leq H \times W\}$. Firstly, the binary secret data S is first partitioned into the segments of λ bits. Secondly, each λ-bit segment is converted into secret digits in the

base-$(2n+1)$ numeral system, where $n \in \{2, 3, ..., H \times W\}$. The value of λ is computed by $\lambda = \lfloor \gamma \times \log_2(2n+1) \rfloor$. For instance, if $n = 2$ and $\gamma = 2$, then the number of bits of one segment is $\lambda = 4$. Thirdly, the grayscale cover image X is divided into groups of n cover pixels. Next, each secret digit d in the base-$(2n+1)$ numeral system (also called a secret digit for short) is embedded into one group of n cover pixels by increasing or decreasing only one cover pixel in the group by 1. More specifically, let us denote the grayscale values of pixels in an n-pixel group as an n-dimensional vector $(x_1, x_2, ..., x_n)$. Then, calculate the value y of the extraction function f as a weighted sum modulo $(2n+1)$. That is,

$$y = f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} (x_i \times i) \mod (2n+1). \tag{1}$$

The several values of the extraction function f when $n = 2$ are demonstrated in Figure 1.

x_1 \ x_2	0	...	11	12	13	14	15	16	17	18	...
0	0	...	2	4	1	3	0	2	4	1	...
:	:	:	:	:	:	:	:	:	:	:	
11	1	...	3	0	2	4	1	3	0	2	...
12	2	...	4	1	3	0	2	4	1	3	...
13	3	...	0	2	4	1	3	0	2	4	...
14	4	...	1	3	0	2	4	1	3	0	...
15	0	...	2	4	1	3	0	2	4	1	...
16	1	...	3	0	2	4	1	3	0	2	...
17	2	...	4	1	3	0	2	4	1	3	...
18	3	...	0	2	4	1	3	0	2	4	...
:	:	...	:	:	:	:	:	:	:	:	...

FIGURE 1. The values of f in 2-dimensional hyper-cubes

According to the secret digit d and the value y of the extraction function f, the embedding rules can be divided into three cases as follows. Case 1: if $d = y$, then d is embedded by no modification on the cover pixel group. Case 2: if $d \neq y$, calculate $t = (d - y) \mod (2n+1)$. Next, if t is less than or equal to n, then increase the value of x_t by 1. Case 3: if $d \neq y$, calculate $t = (d - y) \mod (2n+1)$. Then, if t is greater than n, then decrease the value of x_{2n+1-t} by 1.

3. The Proposed Method.

This section presents our proposed scheme. The embedding phase is described in Subsection 3.1 followed by a demonstrative example given in Subsection 3.2. The extracting phase is provided in Subsection 3.3 and an illustrative example is given in Subsection 3.4.

3.1. The embedding phase.

Given a grayscale cover image X sized $H \times W$ and a binary secret message S with the length LS. Suppose that $X = \{x_1, x_2, ..., x_{H \times W} \mid x_i \in [0, 255], 1 \leq i \leq H \times W\}$. Analogous to the EMD method, the binary secret message S is first converted into the secret digits in the 5-ary notational system (also called the secret digits for short). The proposed method embeds $2K$ secret digits into a group of $(2K+1)$ cover pixels in X at a time, where $K \in \{1, 2, 3, ..., (H \times W - 1)/2\}$. That

is, the proposed method hides 2 secret digits into 3 cover pixels, 4 secret digits into 5 cover pixels, or 6 secret digits into 7 cover pixels, and so on. Firstly, the cover image X is partitioned into $(H \times W)/(2K+1)$ groups of $(2K+1)$ cover pixels, where the notation $\lfloor . \rfloor$ denotes the floor function. Let us denote a cover pixel group as $G = (x_1, x_2, ..., x_{2K+1})$. Secondly, $2K$ cover pixel pairs are generated from the cover pixel group G as $(x_1, x_{K+1}), (x_2, x_{K+1}), ..., (x_K, x_{K+1}), (x_{K+1}, x_{K+2}), (x_{K+1}, x_{K+3}), ..., (x_{K+1}, x_{2K+1})$. Thirdly, $2K$ secret digits $d_1, d_2, ..., d_K, d_{K+1}, d_{K+2}, ..., d_{2K}$ are concealed into the $2K$ generated cover pixel pairs $(x_1, x_{K+1}), (x_2, x_{K+1}), ..., (x_K, x_{K+1}), (x_{K+1}, x_{K+2}), (x_{K+1}, x_{K+3}), ...,$ (x_{K+1}, x_{2K+1}), respectively. Specifically, the first K secret digits $d_1, d_2, ..., d_K$ are embedded into the first K generated cover pixel pairs $(x_1, x_{K+1}), (x_2, x_{K+1}), ..., (x_K, x_{K+1})$ by using the proposed embedding rule **RV**. The remaining K secret digits $d_{K+1}, d_{K+2}, ..., d_{2K}$ are hidden into the remaining K generated cover pixel pairs $(x_{K+1}, x_{K+2}), (x_{K+1}, x_{K+3}), ...,$ (x_{K+1}, x_{2K+1}) by using the proposed embedding rule **RH**.

The proposed embedding rule **RV**: Each secret digit $d_j \in \{d_1, d_2, ..., d_K\}$ is embedded into the generated cover pixel pair (x_j, x_{K+1}), where $j \in \{1, 2, ..., K\}$, as follows. Let us call the matrix shown in Figure 1 the reference matrix M. According to Eq. (1), when $n = 2$ we have $M(x_1, x_2) = f(x_1, x_2) = x_1 + 2 \times x_2 \mod 5$. This means that each cover pixel pair (x_j, x_{K+1}) can be mapped to an element $M(x_j, x_{K+1})$ in the reference matrix M, where x_j and x_{K+1} play the roles of x_1 and x_2, respectively, in Eq. (1) for $n = 2$. From the located element $M(x_j, x_{K+1})$, vertically identify the set of candidate elements VCE as follows:

If $x_j > 1$ and $x_j < 254$, then
$VCE = \{M(x_j - 2, x_{K+1}), M(x_j - 1, x_{K+1}), M(x_j, x_{K+1}), M(x_j + 1, x_{K+1}),$
$M(x_j + 2, x_{K+1})\}$
Else if $x_j \leq 1$, then
$VCE = \{M(0, x_{K+1}), M(1, x_{K+1}), M(2, x_{K+1}), M(3, x_{K+1}), M(4, x_{K+1})\}$
Else (i.e., $x_j \geq 254$)
$VCE = \{M(251, x_{K+1}), M(252, x_{K+1}), M(253, x_{K+1}), M(254, x_{K+1}), M(255, x_{K+1})\}$

Next, search from the set VCE to find out the element $M(x_v, x_{K+1})$ that is equal to the secret digit d_j. Then, the stego pixel pair is computed by $(y_j, y_{K+1}) = (x_v, x_{K+1})$.

The proposed embedding rule **RH**: Each secret digit $d_k \in \{d_{K+1}, d_{K+2}, ..., d_{2K}\}$ is hidden into the generated cover pixel pair (x_{K+1}, x_k), where $k \in \{K+2, K+3, ..., 2K\}$, as follows. Each cover pixel pair (x_{K+1}, x_k) can be corresponded to an element $M(x_{K+1}, x_k)$ in the reference matrix M, where x_{K+1} and x_k play the roles of x_1 and x_2, respectively, in Eq. (1) for $n = 2$. From the located element $M(x_{K+1}, x_k)$, horizontally specify the set of candidate elements HCE as follows:

If $x_k > 1$ and $x_k < 254$, then
$HCE = \{M(x_{K+1}, x_k - 2), M(x_{K+1}, x_k - 1), M(x_{K+1}, x_k), M(x_{K+1}, x_k + 1),$
$M(x_{K+1}, x_k + 2)\}$
Else if $x_k \leq 1$, then
$HCE = \{M(x_{K+1}, 0), M(x_{K+1}, 1), M(x_{K+1}, 2), M(x_{K+1}, 3), M(x_{K+1}, 4)\}$
Else (i.e., $x_k \geq 254$)
$HCE = \{M(x_{K+1}, 251), M(x_{K+1}, 253), M(x_{K+1}, 253), M(x_{K+1}, 254), M(x_{K+1}, 255)\}$

Next, look up from the set HCE to identify the element $M(x_{K+1}, x_h)$ that is identical to

the secret digit d_k. Then, the stego pixel pair is calculated by $(y_{K+1}, y_k) = (x_{K+1}, x_h)$. That is, after $2K$ secret digits $d_1, d_2, ..., d_K, d_{K+1}, d_{K+2}, ..., d_{2K}$ are concealed, the cover pixel group $(x_1, x_2, ..., x_{2K+1})$ becomes the stego pixel group $(y_1, y_2, ..., y_{2K+1})$.

The above embedding process is applied to the next cover pixel group until all the secret digits are embedded into the cover image X to obtain the stego image $Y = \{y_1, y_2, ..., x_{H\times W} \mid y_i \in [0, 255], 1 \le i \le H \times W\}$. It is noted that the last $H \times W - \lfloor (H \times W)/(2K + 1) \rfloor \times (2K + 1)$ pixels in X are not used for embedding secret digits. That is, these pixels are kept intact because they can not be grouped into a group of $(2K + 1)$ pixels.

Example 3.1. *An example is now given to demonstrate the embedding phase of the proposed method. Suppose that the cover pixel group is $G = (x_1, x_2, x_3) = (13, 14, 15)$ (i.e., $K = 1$) and two secret digits are $d_1 = 2$ and $d_2 = 3$. The two generated cover pixel pairs are $(x_1, x_2) = (13, 14)$ and $(x_2, x_3) = (14, 15)$. The two secret digits d_1 and d_2 are hidden into the two generated cover pixel pairs (x_1, x_2) and (x_2, x_3), respectively.*

For the secret digit $d_1 = 2$, the set of candidate elements is $VCE = \{M(x_1 - 2, x_2), M(x_1-1, x_2), M(x_1, x_2), M(x_1+1, x_2), M(x_1+2, x_2)\} = \{M(11, 14), M(12, 14), M(13, 14), M(14, 14), M(15, 14)\} = \{4, 0, 1, 2, 3\}$. The element in the set VCE that is equal to the secret digit $d_1 = 2$ is $M(x_v, x_2) = M(x_1 + 1, x_2) = M(14, 14)$. Thus, the stego pixel pair is computed by $(y_1, y_2) = (14, 14)$.

Similarly, for the secret digit $d_2 = 3$, the set of candidate elements is $HCE = \{M(x_2, x_3-2), M(x_2, x_3 - 1), M(x_2, x_3), M(x_2, x_3 + 1), M(x_2, x_3 + 2)\} = \{M(14, 13), M(14, 14), M(14, 15), M(14, 16), M(14, 17)\} = \{0, 2, 4, 1, 3\}$. The element in the set HCE that equals the secret digit $d_2 = 3$ is $M(x_2, x_h) = M(x_2, x_3 + 2) = M(14, 17)$. Therefore, the stego pixel pair is calculated by $(y_2, y_3) = (14, 17)$. That is, after two secret digits $d_1 = 2$ and $d_2 = 3$ are concealed, the cover pixel group $(x_1, x_2, x_3) = (13, 14, 15)$ becomes the stego pixel group $(y_1, y_2, y_3) = (14, 14, 17)$.

3.2. **The extracting phase.** At the receiving side, a receiver can easily extract the embedded secret digits from the received stego image $Y = \{y_1, y_2, ..., x_{H\times W} \mid y_i \in [0, 255], 1 \le i \le H\times W\}$. Firstly, the stego image Y is segmented into $(H\times W)/(2K+1)$ groups of $(2K + 1)$ stego pixels. Let us denote a stego pixel group as $G' = (y_1, y_2, ..., y_{2K+1})$. Secondly, $2K$ stego pixel pairs are constructed from the stego pixel group G' as $(y_1, y_{K+1}), (y_2, y_{K+1}), ..., (y_K, y_{K+1}), (y_{K+1}, y_{K+2}), (y_{K+1}, y_{K+3}), ..., (y_{K+1}, y_{2K+1})$. Thirdly, $2K$ embedded secret digits $d_1, d_2, ..., d_K, d_{K+1}, d_{K+2}, ..., d_{2K}$ are extracted from the $2K$ generated stego pixel pairs $(y_1, y_{K+1}), (y_2, y_{K+1}), ..., (y_K, y_{K+1}), (y_{K+1}, y_{K+2}), (y_{K+1}, y_{K+3}), ..., (y_{K+1}, y_{2K+1})$, respectively. Specifically, the embedded secret digit d_j is extracted from the generated stego pixel pair (y_j, y_{K+1}), where $j \in \{1, 2, ..., K\}$, by $d_j = f(y_j, y_{K+1}) = y_j + 2 \times y_{K+1} \mod 5$ (i.e., $d_j = M(y_j, y_{K+1)})$. Similarly, the embedded secret digit d_k is extracted from the generated stego pixel pair (y_{K+1}, y_k), where $k \in \{K + 2, K + 3, ..., 2K\}$, by $d_k = f(y_{K+1}, y_k) = y_{K+1} + 2 \times y_k \mod 5$ (i.e., $d_k = M(y_{K+1}, y_k)$).

Example 3.2. *An example is now presented to illustrate the extracting phase of the proposed method. From the received stego pixel group $(y_1, y_2, y_3) = (14, 14, 17)$, the two generated stego pixel pairs are $(y_1, y_2) = (14, 14)$ and $(y_2, y_3) = (14, 17)$. The two embedded secret digits are extracted by $d_1 = f(y_1, y_2) = y_1 + 2 \times y_2 \mod 5 = 14 + 28 \mod 5 = 2$ (i.e., $d_1 = M(14, 14)$) and $d_2 = f(y_2, y_3) = y_2 + 2 \times y_3 \mod 5 = 14 + 34 \mod 5 = 3$ (i.e., $d_2 = M(14, 17)$).*

4. **Experimental Results and Discussions.** To evaluate the performance of the proposed method, we implemented Zhang and Wang's scheme and the proposed scheme by using Borland C++ Builder 6.0 software running on the Pentium IV, 3.6GHz CPU, and 1.49GB RAM hardware platform. The secret message S was randomly generated by using the library function random(). Six commonly used grayscale images sized 512×512, as shown in Figure 2, were used as the cover images in our simulations to test the performance of the proposed method in terms of hiding capacity and visual quality of stego images.

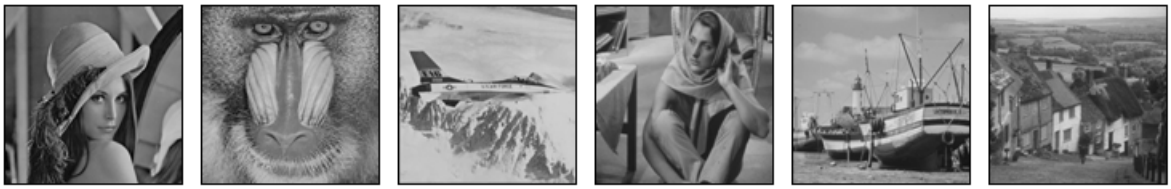

FIGURE 2. Six grayscale test images sized 512×512

The visual quality of stego images (called visual quality for short) was evaluated by using peak signal-to-noise ratio (PSNR) which is defined as $PSNR = 10 \times \log_{10}(255^2/MSE)$ (dB), where MSE is the mean square error representing the distortion between the original cover image X sized $H \times W$ and the stego image Y sized $H \times W$. Hiding capacity C (also called capacity for short) was measured by bit per pixel (bpp). That is, it is computed by the ratio of the total number of embedded secret bits and the total number of pixels in the cover image X.

Theoretically, the hiding capacity of Zhang and Wang's method is $\log_2(2n+1)/n$ bpp. When n equals 2, this method achieves its maximum hiding capacity (i.e., 1 bpp). Thus, Zhang and Wang's method was implemented with $n = 2$. That is, one secret digit in the base-5 notational system was embedded into one cover pixel pair. Equivalently, two secret bits were hidden into one cover pixel pair. This means that the largest hiding capacity of Zhang and Wang's method with $n = 2$ is 1 bpp. The performance results of Zhang and Wang's method in terms of visual quality and hiding capacity are shown in Table 1. It can be seen from Table 1 that the EMD method achieves very good visual quality of stego images. That is, the PSNR values of the EMD method are always greater than 52 dB for all test images. However, the maximum hiding capacity of Zhang and Wang's method is only 1 bpp.

TABLE 1. The performance results of Zhang and Wang's method

Images / Factors	Lena	Baboon	F16	Barbara	Boat	Goldhill
Capacity	1	1	1	1	1	1
PSNR	52.09	52.10	52.12	52.12	52.10	52.11

The proposed method was implemented with various values of K. Generally, the hiding capacity of the proposed method is $(2 \times K)/(2K+1)$ digit per pixel. Equivalently, the hiding capacity of the proposed method is $(4 \times K)/(2 \times K+1)$ bpp. This means that the hiding capacity of the proposed method can approach 2 bpp. Specifically, when K

equals 1, 2, and 3, the proposed method conceals 2, 4, and 6 secret digits in the base-5 notational system into the groups of 3, 5, and 7 cover pixels, respectively. Equivalently, the proposed method embeds 4, 8, and 12 secret bits into the groups of 3, 5, and 7 cover pixels, respectively. Therefore, the corresponding hiding capacities of the proposed method are 1.33, 1.6, and 1.71 bpp. Thus, the hiding capacity of the proposed method is significantly greater than that of the EMD method. The hiding capacities and the PSNR values of the proposed method with various values of K for test images are shown in Table 2. It can be seen from Table 2 that the PSNR values of the proposed method are less than those of the EMD method about 6 or 7 dB. However, the visual quality of stego images produced by the proposed method is still very good. That is, the PSNR values of the proposed method are always greater than 45 dB for all test images.

TABLE 2. The performance results of the proposed method

Values of K	Images / Factors	Lena	Baboon	F16	Barbara	Boat	Goldhill
$K = 1$	Capacity	1.13	1.13	1.13	1.13	1.13	1.13
	PSNR	46.90	46.89	46.89	46.88	46.88	46.89
$K = 2$	Capacity	1.6	1.6	1.6	1.6	1.6	1.6
	PSNR	46.09	46.10	46.08	46.08	46.11	46.09
$K = 3$	Capacity	1.71	1.71	1.71	1.71	1.71	1.71
	PSNR	45.80	45.78	45.79	45.78	45.79	45.79
$K = 6$	Capacity	1.85	1.85	1.85	1.85	1.85	1.85
	PSNR	45.47	45.47	45.46	45.47	45.46	45.46
$K = 10$	Capacity	1.9	1.9	1.9	1.9	1.9	1.9
	PSNR	45.33	45.33	45.34	45.32	45.33	45.33
$K = 18$	Capacity	1.95	1.95	1.95	1.95	1.95	1.95
	PSNR	45.23	45.24	45.25	45.24	45.24	45.23
$K = 70$	Capacity	1.99	1.99	1.99	1.99	1.99	1.99
	PSNR	45.16	45.15	45.16	45.15	45.16	45.15

5. **Conclusions.** In this paper, we propose a steganographic scheme to improve the hiding capacity of Zhang and Wang's method. The maximum hiding capacity of the EMD method is 1 bpp whereas that of the proposed method approaches 2 bpp. Although the PSNR value of the proposed method is less than that of the EMD method, the PSNR value of the proposed method is always greater than 45 dB at the hiding capacity of 1.99 bpp for all test images. Thus, we can conclude that the proposed method is applicable to steganographic applications where a high hiding capacity is desired.

REFERENCES

[1] R. M. Davis, The data encryption standard in perspective, *IEEE Communications Magazine*, vol. 16, no. 6, pp. 5–9, 1978.

[2] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, Morgan Kauffman, 2007, ISBN: 978-0-12-372585-1.

[3] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, Information hiding-a survey, *Proceeding of IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[4] J. Mielikainen, LSB matching revisited, *Proc. of IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285–287, 2006.

[5] C. C. Chang, W. L. Tai, and C. C. Lin, A reversible data hiding scheme based on side match vector quantization, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1301–1308, 2006.

[6] Y. T. Wu and Y. S. Frank, Digital watermarking based on chaotic map and reference register, *Pattern Recognition*, vol. 40, no. 12, pp. 3753–3763, 2007.

[7] S. Lee, C. D. Yoo, and T. Kalker, Reversible image watermarking based on integer-to-integer wavelet transform, *IEEE Trans. Information Forensics and Security*, vol. 2, no. 3, pp. 321–330, 2007.

[8] X. Wu and Z. H. Guan, A novel digital watermark algorithm based on chaotic maps, *Physics Letters A*, vol. 365, no. 5-6, pp. 403–406, 2007.

[9] T. H. Lan and A. H. Tewfik, A novel high-capacity data-embedding system, *IEEE Trans. Image Processing*, vol. 15, no. 8, pp. 2431–2440, 2006.

[10] Y. Hu, H. K. Lee, and J. Li, DE-based reversible data hiding with improved overflow location map, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 250–260, 2009.

[11] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, Watermarking digital image and video data: a state-of-the-art overview, *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20–46, 2000.

[12] L. F. Turner, Digital Data Security System, *Patent IPN*, WO 89/08915, 1989.

[13] X. Zhang and S. Wang, Efficient steganographic embedding by exploiting modification direction, *IEEE Communications Letters*, vol. 10, no. 11, pp. 1–3, 2006.