Development of a data hiding scheme based on combination theory for lowering the visual noise in binary images

Nan-I Wu

aDepartment of Animation and Game Design TOKO University
No.51, Sec. 2, Syuefu Rd., Puzih City, Chiayi County 61363, Taiwan (R.O.C.)
Email: 100001@mail.toko.edu.tw
Email-1: rocifik2000@yahoo.com.tw

Min-Shiang Hwang

b, cDepartment of Computer Science and Information Engineering, Asia University,
500 Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
and
cDepartment of Medical Research, China Medical University Hospital, China Medical University
91 Xueshi Rd, North District, Taichung 40402, Taiwan
Email: mshwang@asia.edu.tw

Aug. 30, 2016 (Submission)
Mar. 8, 2017 (Revision)
Apr. 11, 2017 (2nd Revision)
Apr. 16, 2017 (3rd Revision)
June 26, 2017 (4th Revision)
July 13, 2017 (5th Revision)
Abstract

In order to raise the embedding capacity and simultaneously reduce the artifact effect caused by embedding secret messages into binary images, a novel data hiding method based on the combination theory is proposed. In the proposed scheme, a secret position matrix is designed to improve the hiding capacity which is capable of preventing the least distortion based on the combination theory. Our new scheme enables users to conceal more than one bit of secret data by changing at most one pixel in one subimage. We have derived a formula for computing the payload and the possible modification pixels of a block. Compared with the existing schemes in terms of the hiding capacity and the visual artifacts, as our experimental results show, the proposed scheme is capable of providing a better image quality protector even with a more efficient secret data hider.

Keywords: Binary Image; Combination Theory; Data Hiding; High Capacity; Imperceptible.

1 Introduction

Nowadays, people can send messages everywhere around the world through computer networks with just a few clicks on the mouse to get the job done. In such an open world of information, exchanging messages in a safe and secure way is an important topic of research. So far, the most commonly accepted approach is to utilize a cryptographic system, such as DES or RSA to encrypt the messages in order to prevent the important secret data from leaking out [8]. Using modern cryptographic techniques is an approach to secure communication, but the result of cryptography is to encrypt data transmitted over open channels in the form of ciphertexts, which would easily draw the attention of potential attackers. Another popular way used to protect and secure secret data, called data hiding (i.e. steganography), is to transmit the secret data under the form of common and unimportant plaintexts so that the secret data will not be explicitly exposed and fall prey to malicious attackers [3, 6, 9, 10].

The data hiding technology is analogous to digital watermarking in hiding secret data of digital media by modifying its original content without considering the robustness; however, the application is different. Data hiding can be seen as a secret communication approach where secret messages are hidden into digital media in a way that it is hard to detect any peculiarity from the media’s
appearance. Therefore, to hide the secret messages into the cover media with a little modification done to the cover media is the key to develop data hiding because its success lies in imperceptibility. Unlike data hiding, watermarking relies on the robustness to protect the hidden significant data from various attacks [4, 18].

So far, many data hiding schemes have been proposed to embed secret messages into digital gray scale images [1, 4, 7, 14-16, 19]. Since the modification tolerant of gray scale images is more invisible to human eyes, the payload is higher than other image formats. One of the most common techniques is the least-significant-bit (LSB) method that modifies the least bit of pixels in the cover image to hide the secret message. Researchers Chan and Cheng [1] presented a low computational complexity method with high stego image quality by using a simple LSB substitution. Wu and Tsai [15] embedded more secret data in edge areas than in smooth areas by applying a pixel-value differencing (PVD) method, therefore preserving a better image quality. Wu et al. proposed a mixed scheme based on pixel-value differencing and LSB replacement to hide more secret data into smooth areas [16]. To reduce the distortion produced by a PVD scheme, Wang et al. proposed a high quality PVD scheme using a modulus function [14]. Lou et al. also proposed a novel adaptive HVS-based data hiding scheme [7] in which the performance is better than that of a PVD-based scheme in terms of payload and visual quality. In recent, Wu et al. presented the newest PVD-based scheme which is capable of reducing the image distortion at the lowest [19]. Najme et al. also presented a new adaptive and non-adaptive data hiding methods for grayscale images based on a modulus function and human vision sensitivity [4].

In contrast to a gray-level image, the binary image is more difficult to raise the payload, and the hiding artifact is usually evident. Therefore, some researchers focus on how to raise hiding capacity in binary images while others concern about how to prevent the visual artifact after hiding data with the low payload [2, 5, 10, 12, 17, 20, 21]. For example, Tseng et al. [11] proposed a secure data hiding scheme that can embed a great deal of secret data into binary images. In their scheme, a secret key and a weight matrix are used and designed to protect secret data and raise the hiding capacity. Wu and Liu [17] embedded secret data into binary images for authentication and annotation by using a shuffle technique for
reducing noticeable artifacts. Based on Wu and Liu’s scheme, Gou and Wu presented an improving scheme by super-pixels to increase the embedding capacity [2]. To reduce the detectable visual, Yang and Kot presented a pattern-based data hiding scheme for binary images by connectivity-preserving [20]. After that, Yang et al. proposed a high capacity approach in a morphological transform domain for binary images [21]. To improve the distortion of the previous schemes [2, 17, 20, 21], Wang et al. presented a high capacity scheme based on block patterns [12]. In addition, a data hiding method in binary images for key authentication using block masking was presented by Jung and Yoo [5].

Basically, how to raise the payload and reduce the visual noise after embedding the secret data into the cover image is the major concerns when it comes to data hiding. In general, we expected that the cover image is able to maintain good image quality after embedding a large quantity of secret data. Nevertheless, visible noises often come as a result from a large quantity of secret data; especially, when the cover image is a binary image, and since each pixel of a binary image is either pure white or pure black. Therefore, it is extremely difficult to develop a data hiding technique for binary images that can offer the high capacity with good stego image quality.

In this paper, a quality data hiding scheme with high capacity for binary images is proposed. We designed a secret position matrix based on the combination theory for raising the capacity of hiding while preserving a good stego image quality. The proposed scheme can embed as many as \( \left\lceil \log_2(m \times n + 1) \right\rceil \) bits of secret data into a binary block of size \( m \times n \), and only one bit in each block will be changed. Furthermore, we will introduce the extending application of the proposed scheme so that more ideal visual quality can be achieved by selecting more ideal cover positions as the modification pixels.

The rest of this paper is organized as follows. In Section 2, the related works and the fundamental concept of the combination theory will be introduced. In Section 3, the framework design of the secret position matrix and the data hiding process will be presented. Experimental results and comparisons will be given in Section 4. Finally, a brief conclusion will be presented in Section 5.

2 Related Works

2.1 Odd-Even Mapping Scheme
The data hiding scheme for binary images usually takes a block of sized $m \times n$ as a unit to convey a bit of secret data. The total number of white pixels or black pixels in a block is an important characteristic to record the bit '1' or '0' of secret data. For example, we enable the total number of white pixels is odd while the secret data is bit '1', and vice versa. A higher security depends on the random odd-even mapping produced by the hider’s private key. The major disadvantage of odd-even mapping schemes shows up when the embedding capacity is very low due to only one bit hidden in a block of sized $m \times n$. Selecting pixels for alteration with as little visual noises as possible is the major concern in an odd-even mapping scheme. Therefore, it is important to take the human perceptual factor into account by studying each pixel and its immediate neighbors to estimate whether they are flappable or not [2, 17, 20, 21]. Before hiding the data, the complexity of each pixel should be estimated whether it is flappable for hiding data or not. We avoid selecting pixels with all neighboring pixels as black or white for hiding data since the pixel modification in a pure block is easily detected.

The well-known estimation mechanism was presented by Wu and Liu [17]. Given a binary image pixel, they determined the flippability of pixels by observing the smoothness and connectivity. The smoothness is measured by the horizontal, vertical, and diagonal transitions while the connectivity is measured by the number of black and white clusters. The high flippability score indicates that the texture of smoothness and connectivity are more difficult to be altered for conveying secrets. Based on Wu and Liu’s scheme, Yang and Kot [2, 20, 21] also presented a connectivity-preserving criterion to estimate pixel flippability by using $3 \times 3$ neighborhood patterns. Their method preserves the original connectivity after hiding data in which the artifact noise is very unnoticeable. This flippable pixel does not destroy the connectivity among pixels before and after hiding the data with the visual distortion of stego blocks ensured.

### 2.2 Weight Matrix Scheme

In order to increase the embedding capacity and maintain an acceptable stego
image quality, Tseng et al. [11] proposed a new scheme with a weight matrix. In their scheme, the number of possible pixels modified in each subimage does not go beyond two, but the payload of each subimage can be increased up to the size of each subimage. Assume the size of each subimage is $m \times n$, each subimage can cover $\left\lfloor \log_2(m \times n + 1) \right\rfloor$ bits of secret data by changing at most two bits in the subimage. The embedding algorithm of Tseng et al.’s scheme is shown as follows:

**Step 1.** Given a binary image $I$ with $m \times n$ pixels, partition $I$ into a series of non-overlapping subimages of a fixed size, $F_i$ is the $i$-th subimage of size $m \times n$, and $F_i(x,y)$ is the pixel at position $(x,y)$ in the $i$-th subimage $F_i$.

Therefore, $I = \{F_i(x,y) | i = 1, 2, ..., M \times N \over m \times n, x = 1, 2, ..., m, y = 1, 2, ..., n, F_i(x,y) \in \{0,1\} \}$. Define $k$ as the maximum number of secret data bits that can be embedded into each subimage. Obtain the $k$ value by applying Equation (1). For example, if the size of each $F_i$ is $3 \times 3$, then $k = 3$; and 3 bits of secret data can be embedded in each subimage.

$$k = \left\lfloor \log_2(m \times n + 1) \right\rfloor$$

**Step 2.** Design a secret integer weight matrix $W$ whose size is $m \times n$, and each element $W_{x,y}$ in $W$ is an integer whose value ranges go from 1 to $2^k - 1$. $W$ is an important encryption and decryption secret key shared by the sender and the receiver. For instance, assuming the payload of a $3 \times 3$-subimage is 3 bits. its weight matrix $W$ can be designed as follows:

$$W = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 1 & 2 \end{bmatrix}$$

**Step 3.** Let $\otimes$ be the component-wise multiplication operator. The new weight element $F_i(x,y)$ of $F_i$ can be obtained by $F_i(x,y) \otimes W_{x,y}$, and then we can sum up all the weights in the new weight matrix. An example of computing $F_i(x,y) \otimes W_{x,y}$ is shown below, and the total weight can be found by $SUM(F_i \otimes W) = 1 + 2 + 3 + 6 + 1 + 2 = 15$. 


Step 4. Read \( k \) bits of binary data from the secret data stream and transform them into decimal value \( d \).

Step 5. Hide \( k \) bits into \( F_i \) by using the following process:

If \( SUM(F_i \otimes W) \mod 2^k = d \)

Embedding process is completed without altering any pixels of \( F_i \).

Else

Alter one or two pixels in \( F_i \) such that \( SUM(F_i \otimes W) \mod 2^k = d \).

End

Here is a simple example to show how Step 5 works. Assume that the secret data is \( 000_2 \), then the decimal value is \( d = 0_{10} \). Since \( SUM(F_i \otimes W) \mod 2^3 = 15 \mod 8 = 7 \) and \( 7 \neq 0 \), two elements of \( F_i \) must be altered such that \( SUM(F_i \otimes W) \mod 2^3 = 0 \). Hence, we can obtain a stego image \( F'_i \) after flipping the pixels in the positions \( F_i(3,2) \) and \( F_i(2,3) \) as follows:

\[
F'_i = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}
\]

When the receiver needs to decrypt the secret data from the stego image \( F'_i \), he can use the same weight matrix to extract the secret data by executing \( SUM(F_i \otimes W) \mod 2^k \).

2.3 The Combination Theory

In this section, we shall briefly go through some fundamental concepts of the combination theory. Let \( A \) be a set of \( k \) distinct objects. The combination of \( A \) is simply a subset of \( A \). More precisely, for \( 0 \leq r \leq k \), an \( r \)-combination of \( A \) is an \( r \)-element subset of \( A \). For example, if \( A = \{ S_1, S_2, S_3, S_4 \} \), then the 1-combination, 2-combination, 3-combination, and 4-combination of \( A \) are as follows:
We denote $C_r^k$ as the number of $r$-combination of a $k$-element set $A$. Consider the above example, namely $k = 4$, and we can get $C_1^4 = 4, C_2^4 = 6, C_3^4 = 4$, and $C_4^4 = 1$.

Finally, we calculate the total number of combinations of $k$ by the following formula and define $T$ representing the total number of combinations.

$$T = \sum_{r=1}^{k} C_r^k = 2^k - 1.$$  \hspace{1cm} (2)

For example, when $k = 4$, then $T$ is 15, and if $k = 3$, $T$ is 7.

3 The Proposed Method

In order to further reduce the number of possible pixels modified in secret data embedding processing, a secret position matrix will be presented for the purpose of improving the stego image quality. Our new scheme is capable of producing more embedding capacity with the existing schemes by changing one pixel in each subimage at most. With the proposed secret position matrix, we can obtain a feature value for each subimage, and the feature value can stand for the secret data. When the secret data is unequal to the feature value, only one pixel of the subimage will be altered.

3.1 The Design Rules of the Secret Position Matrix

In this section, we will demonstrate how to generate the secret position matrix based on the combination theory. Before the secret data can be embedded into a binary image, a secret position matrix must be produced. The secret position matrix is an important encryption and decryption key in our scheme, and its design rules are listed as follows:

(1) Let $P$ denotes an empty matrix whose size is identical to that of the subimage. Assume the size of the subimage is $m \times n$, then the size of $P$ is $m \times n$, and $P_{x,y}$ is the element of $P$ at position $(x, y)$.

(2) The maximum hiding capacity $k$ of a subimage can be obtained by utilizing
Equation (1) with $S$ standing for the $k$-bit secret data $S = \{S_r | r = 1, 2, ..., k, S_r \in \{0,1\}\}$. For instance, assume 4 bits of secret data are 1101, so $S = S_1S_2S_3S_4$; in other words, $S_1 = 1, S_2 = 1, S_3 = 0, \text{and } S_4 = 1$.

(3) Then we generate the whole combination of the $k$-bit secret data using the combination theory as mentioned in Section 2.3. For example, suppose that the 4-bit secret data $S_1S_2S_3S_4$ is to be embedded, we can obtain all the combinations of $S_1S_2S_3S_4$ as follows:

$$S_{c1} = \{S_1\},$$
$$S_{c2} = \{S_2\},$$
$$S_{c3} = \{S_3\},$$
$$S_{c4} = \{S_4\},$$
$$S_{c5} = \{S_1S_2\},$$
$$S_{c6} = \{S_1S_3\},$$
$$S_{c7} = \{S_1S_4\},$$
$$S_{c8} = \{S_2S_3\},$$
$$S_{c9} = \{S_2S_4\},$$
$$S_{c10} = \{S_3S_4\},$$
$$S_{c11} = \{S_1S_2S_3\},$$
$$S_{c12} = \{S_1S_2S_4\},$$
$$S_{c13} = \{S_1S_3S_4\},$$
$$S_{c14} = \{S_2S_3S_4\},$$
$$S_{c15} = \{S_1S_2S_3S_4\}.$$

where $S_{c_i}, i = 1, 2, ..., T,$ is the code name of each combination, and $T$ can be computed by Equation (2). In our proposed scheme, each subimage will generate $k$-bit of original image feature using the secret position matrix. After comparing $k$-bit of the original secret binary data, there are possible $2^k - 1$ differences, and each combination of this step represents one status of comparison results.
(4) The major purpose of this step is to ensure each possible difference between the original and secret binary secret data, at least one position to be altered for hiding data. Therefore, each combination $S_{ci}, i=1,2,...,T$; must be assigned to at least one position in $P$. If $P$ has extra positions, then we would re-assign one of the combinations to the extra positions until all the positions in $P$ are filled. After that, a secret position matrix $P$ is generated.

$$P = \{P_{x,y} | x=1,2,...,m, y=1,2,...,n, P_{x,y} \in \{S_{ci}, i=1, 2, ..., T\} \}.$$  

It is obvious that each combination $S_{ci}, i=1,2,...,T$, can be assigned to at least one position in $P$, since $T = 2^k - 1$ is only natural when $T < 2^k$. In the meanwhile, $k = \lceil \log_2 (m \times n + 1) \rceil$; that is, $2^k \leq m \times n + 1$. Therefore, we have $T < m \times n + 1$, which means $T < m \times n$, and there are $(m \times n)!$ choices of $P$ in which the position of combination is different. More importantly, we usually enhance the security by using random positions of matrix. Three examples of how a secret position matrix $P$ of sized $4 \times 4$ can be created are shown below with different positions of content among three matrices $P$, $P_{random1}$, and $P_{random2}$. Basically, they can all be used to hide 4 bits of secret data into a subimage of sized $4 \times 4$. Notice that there are $(m \times n)!$ kinds of random position $P_{random}(i)$ where $1 \leq i \leq (m \times n)!$ and its generation rule of random position matrix in the following algorithm.

**Initial** $P_{1,1} = S_{ci}$, where $i = \text{rand(number)} \mod (m \times n - 1)$;

**For** $x=1:m, y=2:n$

Temp=0;

$i = \text{rand(number)} \mod (m \times n - 1)$;

**For** $j=1:m, k=1:n$

If $S_{ci} == P_{(j,k)}$

Temp=Temp+1;

**End**

**End**

If Temp=0
\[ P_{x,y} = S_{c_i} ; \]

End

End

\[ P_{m,n} = S_{c_i} , \text{where } i = \text{rand(number)} \mod (m \times n - 1) ; \]

In above algorithm, the code name of combination \( S_{c_i} \) must be obtained before producing the random position matrix. We also utilize one secret key to control the code name of combination and its real corresponding combination to enhance the security. Hence, there are \((m \times n - 1)!\) kinds of corresponding between the code name and the real combination. The following are three examples to show the secret position matrix that can to be used in the data embedding phase.

\[
P = \begin{bmatrix}
S_1 & S_2 & S_3 & S_4 \\
S_2 S_2 & S_3 S_4 & S_4 S_3 & S_3 S_2 \\
S_4 S_3 & S_2 S_4 & S_1 S_3 & S_1 S_2 \\
S_3 S_3 S_4 & S_2 S_3 S_4 & S_1 S_3 S_4 & S_1 S_2 S_4
\end{bmatrix}
= \begin{bmatrix}
S_{c_1} & S_{c_2} & S_{c_3} & S_{c_4} \\
S_{c_5} & S_{c_6} & S_{c_7} & S_{c_8} \\
S_{c_9} & S_{c_{10}} & S_{c_{11}} & S_{c_{12}} \\
S_{c_{13}} & S_{c_{14}} & S_{c_{15}} & S_{c_{16}}
\end{bmatrix}
\]

\[
P_{\text{random}} = \begin{bmatrix}
S_2 & S_1 & S_4 & S_3 \\
S_3 S_3 & S_4 S_2 & S_2 S_3 & S_1 S_4 \\
S_1 S_2 S_4 & S_2 S_4 & S_4 S_3 & S_2 S_3 S_3 \\
S_1 S_2 S_3 S_4 & S_2 S_3 S_4 & S_1 S_3 S_4 & S_1 S_2 S_4
\end{bmatrix}
= \begin{bmatrix}
S_{c_2} & S_{c_1} & S_{c_4} & S_{c_3} \\
S_{c_6} & S_{c_5} & S_{c_8} & S_{c_7} \\
S_{c_{12}} & S_{c_9} & S_{c_{10}} & S_{c_{11}} \\
S_{c_{16}} & S_{c_{15}} & S_{c_{14}} & S_{c_{13}}
\end{bmatrix}
\]

\[
P_{\text{random}} = \begin{bmatrix}
S_1 & S_2 & S_3 & S_4 \\
S_1 S_2 & S_1 S_3 & S_1 S_4 & S_2 S_3 \\
S_2 S_4 & S_3 S_4 & S_1 S_2 S_3 & S_1 S_2 S_4 \\
S_1 S_3 S_4 & S_2 S_3 S_4 & S_1 S_2 S_3 S_4 & S_1
\end{bmatrix}
= \begin{bmatrix}
S_{c_1} & S_{c_2} & S_{c_3} & S_{c_4} \\
S_{c_5} & S_{c_6} & S_{c_7} & S_{c_8} \\
S_{c_{10}} & S_{c_{11}} & S_{c_{12}} & S_{c_{13}} \\
S_{c_{14}} & S_{c_{15}} & S_{c_{16}} & S_{c_{16}}
\end{bmatrix}
\]

3.2 The Embedding Algorithm

In this section, we will introduce the process of embedding secret data into the cover image by using the secret position matrix. In the following, our proposed data to hide embedding algorithm is described in details.

**Step 1.** Given an \( m \times n \) subimage \( F_i \), calculate its maximum payload \( k \) by
Step 2. Design a secret position matrix $P$ according to Section 3.1. Generally, the secret position matrix should be shared between two parties before hiding data.

Step 3. Define the new secret position matrix $P'$ that can be calculated by

$$P \otimes F_i$$

where $P'_{x,y} F_{i(x,y)}$ for $x=1,2,\ldots,m$ and $y=1,2,\ldots,n$. Then compute the total number $T(S_r,r=1,2,\ldots,k)$ of each piece of secret data which still survive in $P'$ according to the following algorithm:

FOR $x = 1 : m$; $y = 1 : n$

IF $P'_{x,y}$ has $S_r$

$$T(S_r) = T(S_r) + 1$$

END

END

We illustrate the Step3 as follows

$$P' = P \otimes F_i = \begin{bmatrix} S_1 & S_2 & S_3 & S_4 \\ S_1S_2 & S_3S_1 & S_1S_4 & S_2S_3 \\ S_2S_4 & S_3S_4 & S_1S_2S_3 & S_1S_2S_4 \\ S_1S_3S_4 & S_2S_3S_4 & S_1S_2S_3S_4 & S_1 \\ \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ \end{bmatrix}$$

Obviously, $S_1$ occurs in six positions where the coordinate are (1,1), (2,1), (3,3), (3,4), (4,1), and (4,3) in $P'$. Similarly, $S_2$ occurs in five positions where the coordinates are (1,2), (3,1), (3,3), (3,4), and (4,3) in $P'$. $S_3$ occurs in five positions where the coordinate are (2,2), (3,2), (3,3), (4,1), and (4,3). Finally, $S_4$ also occurs in five positions where the coordinate are (3,1), (3,2), (3,4), (4,1), and (4,3) in $P'$. Therefore, we can obtain $T(S_1) = 6$, $T(S_2) = 5$, $T(S_3) = 5$, and $T(S_4) = 5$, respectively.
**Step 4.** Compute the remainder value $S'_r$ by modulo calculation for the total number of occurrences of each bit $S_r$.

$$S'_r = T(S_r) \mod 2, \text{where } r = 1, 2, ..., k. \quad (3)$$

When all the obtained remainder values $S'_r, r = 1, 2, ..., k$, are put together, it can be regarded as the original feature value of the subimage.

**Step 5.** Compare the original secret data $S_r$ with the feature value $S'_r$, where $r = 1, 2, ..., k$. After the comparison, we can obtain a new combination $S''$ using Equation (4) which consists of $S_r$ such that $S_r \neq S'_r$, where $r = 1, 2, ..., k$.

$$S'' = S_r \odot (S_r \oplus S'_r) \quad (4)$$

For example, if secret data $S_1S_2S_3S_4 = 1101$ and the feature value $S'_1S'_2S'_3S'_4 = 0111$, then a new combination $S'' = (S_1S_2S_3S_4) \odot ((1101) \oplus (0111)) = (S_1S_2S_3S_4) \odot (1010) = (S_1S_4)$ can be obtained such that $S_1 \neq S'_1$ and $S_4 \neq S'_4$.

**Step 6.** In Step 5, if all the original feature values are equal to their secret data bit counterparts, then the embedding algorithm is completed without modifying any pixels of the subimage. Otherwise, hide the secret data into subimage $F_i$ by modifying only one pixel of $F_i$ according to the following algorithm.

FOR $x = 1 : m; \ y = 1 : n$

IF $P_{x,y} = S''$

IF $F_i(x, y) = 1$

$F_i(x, y) = 0$

ELSE

$F_i(x, y) = 1$

END
An illustration of Step 6 is shown below. Assume that $S' = S_1S_3$ can be found at position (2,2) in $P$. We then flipped the pixel at position (2,2) in $F_i$ to hide the secret data into $F_i'$.

$$
\therefore S' = S_1S_3 = P_{2,2} \quad \therefore F_i' = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0
\end{bmatrix}
$$

### 3.3 The Extracting Algorithm

In this section, we will demonstrate how to retrieve the embedded messages from the stego block $F_i'$. Basically, the receiver must use the same secret position matrix $P$ to extract the secret message. According to Step 3 and Step 4 as described in Section 3.2, we will briefly introduce the data extraction algorithm as follows:

**Step 1.** Compute the new matrix $P' = P \otimes F_i'$ where $P'_{x,y} = P_{x,y} \times F'_{i(x,y)}$ for all $x = 1,2,...,m$, and $y = 1,2,...,n$.

**Step 2.** Compute the total number of each secret data bit $T(S_r), r = 1,2,...,k$ in the new matrix $P'$ by

FOR $x = 1 : m$; $y = 1 : n$

IF $P'_{x,y}$ has $S_r$

$T(S_r) = T(S_r) + 1$

END

END

**Step 3.** Reconstruct the binary secret data $S_r \mid S_r \in 0,1, r = 1,2,...,k$, by calculating

$$
S_r = T(S_r) \mod 2 \quad (5)
$$

### 4 Experimental Results and Comparisons

#### 4.1 Experimental Results
In order to demonstrate the effectiveness of our new method, we tested the proposed scheme on the three test binary images, as shown in Figure 1(a), Figure 2(a), and Figure 3(a), respectively. The secret data is generated by a pseudo-random number. The sizes of all cover images are 512×512 pixels. With the inspection of the content of Chinese and English character images as shown in Figure 1(a) and Figure 2(a), we have discovered that the white pixels distribute over on the horizontal direction. Therefore, we adopted the horizontal scanning method to divide the cover images in Figure 1(a) and Figure 2(a) into a set of non-overlapping 1×256 subimages. The payload of each block is to be hidden with 8 bits of secret data. This scanning method for a character image is efficient enough to avoid selecting a block with the pure white pixels to be the hidden data. In addition, we only chose those subimages whose number of black pixels is between 52 and 205 to prevent salient artifacts. In our proposed scheme, at most one pixel which is different between a cover image and a stego image caused by the proposed embedding scheme. If the number of black pixels was less than 52 or greater than 205 after secret data hiding, then this subimage would not be used to hold any secret data. Such unqualified subimages would be skipped until a suitable subimage was met.

The stego images produced in the first experiment (i.e. 1×256-block-basis) are shown in Figures 1(b) and 2(b). The visual artifact noises of the stego images produced by the proposed scheme as shown in Figures 1(b) and 2(b) are imperceptible to human eyes. In our second experiment, we took the popular image “Baboon” (Figure 3) of sized 512×512 as the test image. The image characteristic of “Baboon” is more complex than that of character images (Figures 1 and 2). Thus, we used 16×16 pixels as a unit to partition the cover image ”Baboon”. Figure 3(b) shows the visual artifact produced by the proposed scheme is difficult to be observed by human eyes. The reason is that the number of modified bits for each subimage is limited by only one pixel at most.

4.2 Payload and Visual Distortion Comparison

First, we make a comparison on payload with other existing schemes [2, 12, 17, 20, 21] under the same visual noise after hiding data. As mentioned in Section 2.1, the payload and visual noise for each 16×16 blocks is one bit [2, 12, 17,
20, 21]. However, the payload of the proposed scheme is 8 bits of secret data by changing at most one bit for a binary block of $16 \times 16$ size. Clearly, given a $16 \times 16$-block, our scheme is capable of increasing 8 times the payload of those existing schemes [2, 12, 17, 20, 21]. Also from Table 1, our scheme outperforms [2, 12, 17, 20, 21] in terms of the total payload for three tested images.

<table>
<thead>
<tr>
<th>Cover image</th>
<th>Schemes [2, 12, 17, 20, 21]</th>
<th>Our new scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese character image</td>
<td>1024</td>
<td>4664</td>
</tr>
<tr>
<td>English character image</td>
<td>1024</td>
<td>4528</td>
</tr>
<tr>
<td>Baboon</td>
<td>1024</td>
<td>4016</td>
</tr>
</tbody>
</table>

Secondly, the performance of the proposed scheme is compared with that of Tseng et al.’s scheme [11] in two aspects: (1) Visual artifact noise/stego image quality at the same payload level, and (2) hiding capacity under the same visual noise.

- **Visual artifact noises:** Basically, the better policy to maintain better stego image quality is to reduce the noise in a binary image. To prevent visual artifact noises, the secret position matrix of the proposed scheme has shown more promise than the weight matrix of Tseng et al.’s scheme. Figure 4 is the visual comparison results between the proposed scheme and Tseng et al.’s scheme. The amount of artifacts caused by the proposed scheme is less than that of Tseng et al.’s scheme by visual inspection. The proposed scheme preserves better stego image quality than Tseng et al.’s scheme since our method only changes one bit in each subimage at most, while Tseng et al.’s method possibly changes two bits. As expected, our method is capable of effectively reducing the number of pixels being modified and, therefore, improves the visual artifacts of a stego image.
Figure 1: (a) Cover image: Chinese characters, (b) the stego image after embedding 4664 bits.

Figure 2: (a) Cover image: English characters, (b) the stego image after embedding 4528 bits.

Figure 3: (a) Cover image: The size of “Baboon” is 512×512 pixels, (b) the stego image after embedding 4016 bits.
Hiding capacity: As a matter of fact, the proposed scheme can further divide one subimage into two subimages. If so, just like Tseng et al.’s method, at most two bits are changed in each subimage. However, the difference is that the proposed scheme can embed more than \(\lceil \log_2(m \times n + 1) \rceil\) bits at the same subimage size while Tseng et al.’s method cannot. As shown in Table 2, all the payloads of the proposed scheme at various subimage sizes are unexceptionally larger than those of Tseng et al.’s scheme. Therefore, the proposed scheme can indeed provide more payload than Tseng et al.’s scheme. Herein, we will demonstrate how to increase the payload using our new scheme. Assume that an \(m \times n\) block has 2 subimages, and the payloads

<table>
<thead>
<tr>
<th>Figure 4: (a), (a1) The cover image, (b), (b1) the stego image produced by Tseng et al.’s scheme [11]. (c) (c1) the stego image produced by the proposed scheme.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data hiding/Steganography technology is analogous to watermarking without considering the robustness secret data in digital media by modifying its original content; however, the application is different from watermarking. Data hiding can be treated as a secret communication to hide secret messages into digital media, and the process of delivering the digital media no one can detect.</td>
</tr>
</tbody>
</table>

- **Hiding capacity:** As a matter of fact, the proposed scheme can further divide one subimage into two subimages. If so, just like Tseng et al.’s method, at most two bits are changed in each subimage. However, the difference is that the proposed scheme can embed more than \(\lceil \log_2(m \times n + 1) \rceil\) bits at the same subimage size while Tseng et al.’s method cannot. As shown in Table 2, all the payloads of the proposed scheme at various subimage sizes are unexceptionally larger than those of Tseng et al.’s scheme. Therefore, the proposed scheme can indeed provide more payload than Tseng et al.’s scheme. Herein, we will demonstrate how to increase the payload using our new scheme. Assume that an \(m \times n\) block has 2 subimages, and the payloads
of the two subimages are \( t_1 \)-bit(s) and \( t_2 \)-bit(s), respectively. In order to ensure the number of possible pixels modified in each subimage not to exceed one, the following inequality must be satisfied:

\[
\sum_{i=1}^{[\log_2 m]} C^i_{n_i} + \sum_{i=1}^{[\log_2 n]} C^i_{m_i} \leq (2^{[\log_2 m]}) - 1) + (2^{[\log_2 n]}) - 1) \leq m \times n,
\]

(6)

and the maximum hiding capacity \( k \) can be figured out by the following inequality:

\[
k = \max(t_1 + t_2), \text{ and } (2^{t_1} + 2^{t_2}) - 2 \leq m \times n
\]

(7)

where \( 0 \leq t_1, t_2 \leq \left\lfloor \log_2 (m \times n + 1) \right\rfloor \). For example, given a 8×8 subimage, it can be divided into two even smaller subimages. In order to obtain the maximum hiding capacity, we must divide the 8×8 blocks into two 4×8 subimages by applying Equations (6) and (7), namely, \( t_1 = 5 \) and \( t_2 = 5 \). That is to say, the payload of each 4×8 subimage is 5 bits, and the number of possible pixels to be altered is 1. As a result, the total payload is 10 bits, and the number of possible pixels to be modified does not go beyond 2 in a 8×8 subimage. By inspecting Table 2, the proposed scheme can achieve a higher embedding capacity while keeping the same stego image of Tseng et al.’s scheme.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3×3</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4×4</td>
<td>6</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5×5</td>
<td>7</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>6×6</td>
<td>8</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>7×7</td>
<td>9</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>8×8</td>
<td>10</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>9×9</td>
<td>10</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>10×10</td>
<td>11</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>12×12</td>
<td>12</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>16×16</td>
<td>14</td>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2: The comparison results of the proposed scheme and the existing methods in terms of the hidden capacity by changing no more than two bits in each block.
5 Conclusions

In this paper, we proposed a novel data hiding technique for binary images based on the combination theory to reduce visual distortion. To raise embedding capacity and simultaneously reduce the visual artifacts, a secret position matrix is utilized to hide \( \left\lfloor \log_2(m \times n + 1) \right\rfloor \) bits into a block of \( m \times n \) size by changing one pixel for each block at most. Compared with the existing schemes [2, 11, 12, 17, 20, 21], our new scheme can achieve a very high hiding capacity while keeping the imperceptibility after hiding the data. Under the same payload, the proposed scheme is capable of producing a better image quality than that of Tseng et al.’s scheme [11]. Moreover, the experimental results also demonstrated that the proposed scheme is capable of yielding a higher payload at the same visual noise than other existing schemes [2, 12, 17, 20, 21].
References

93 (2014) 152-162.

steganographic method with pixel-value differencing and modulus function,


[16] H.C. Wu, N.I. Wu, C.S. Tsai, M.S. Hwang, Image steganographic scheme
based on pixel-value differencing and LSB replacement methods, IEE

[17] M. Wu, B. Liu, Data hiding in binary image for authentication and


[20] H. Yang, A.C. Kot, Pattern-based data hiding for binary image
authentication by connectivity-preserving, IEEE Trans. Multimed. 9

[21] H. Yang, A.C. Kot, S. Rahardja, Orthogonal data embedding for binary
images in morphological transform domain - A high-capacity approach,