Large-Capacity Image Data Hiding based on Table Look-up

Wenjia Ding
Wuhan University
Huyin Zhang (huyinzhang@whu.edu.cn)
Wuhan University
Ralf Reulke
Humboldt-Universität zu Berlin
Yulin Wang
Wuhan University

Research Article

Keywords: Image data hiding, Digital watermarking, Image steganography, Information encoding.

Posted Date: September 30th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-929946/v1

Large-Capacity Image Data Hiding based on Table Look-up

Wenjia Ding[1], Huyin Zhang[1*], Ralf Reulke[2], Yulin Wang[1*]

1. School of Computer Science, Wuhan University, China

2. Institut für Informatik, Humboldt-Universität zu Berlin, Germany

* Huyin Zhang and Yulin Wang contribute the same to the article and are the corresponding authors

Corresponding Author's E-mail: huyinzhang@whu.edu.cn(Huyin Zhang), yulinwang@whu.edu.cn(Yulin Wang)

Abstract： In previous data hiding techniques, binary rules are usually used to guide the fine-tuning of the values of basic objects in the host media to hide bit 0 and bit 1. In this paper, we propose a new data hiding technique for gray images based on querying a 256x256 information table. The information table is constructed by cloning a 3x3 basic block, which we call seed block. Eight unsigned integer values between 0 and 7, *i.e.*, 3 bit binary data, are assigned to different elements of the seed block. Each time, a pair of pixels are chosen from a host image, and their pixel values are used as row and column numbers to look up the information table. If element value obtained by looking up the table is equal to the 3 bit binary data to be hidden, the values of the pixel pair will remain unchanged. Otherwise, take this element as the central point, we call it the focus element, to enclose a 3x3 window in the information table. Then in the window, find the element which is equal to the data to be hidden. Finally, update the pixel values of the pair with the row and column numbers of the found element in the window. Since the row and column numbers are in the range of 0-255, the updated pixel values will not overflow. In the proposed algorithm, a pair of pixels can hide 3 bits of information, so the embedding capacity is very high. Since the adjustment of pixel values is constrained in a 3x3 window, the modification amount of pixel values is small. The proposed technique belongs to fragile digital watermarking, so it can be used for image authentication and tamper localization. By the evaluation of data hiding capacity, security, imperceptibility, computational cost and extensibility, this algorithm is superior to existing information hiding techniques. The proposed technique can also be used in color image and audio data hiding.

Keywords: Image data hiding; Digital watermarking; Image steganography; Information encoding.

1. Introduction

The general model of most data hiding techniques can be described by Fig. 1. The information to be hidden is decomposed into a bit stream ***b***. The data of the host media is directly or indirectly fine tuned under the control of the bits to be hidden. Let ***X*** represent the original media, and let ***m*** represent its fine-tuning amount under the control of the information to be hidden. Then, the process of data hiding is equivalent to ***X*** superimposed with ***m***. The resulted ***X'***, ***X'*** =***X*** + ***m***, is stego media. Although ***X'*** is different from ***X*** in data value, it is possible to make it difficult for people to perceive the difference between the two in vision or hearing. This is the basic principle of data hiding.

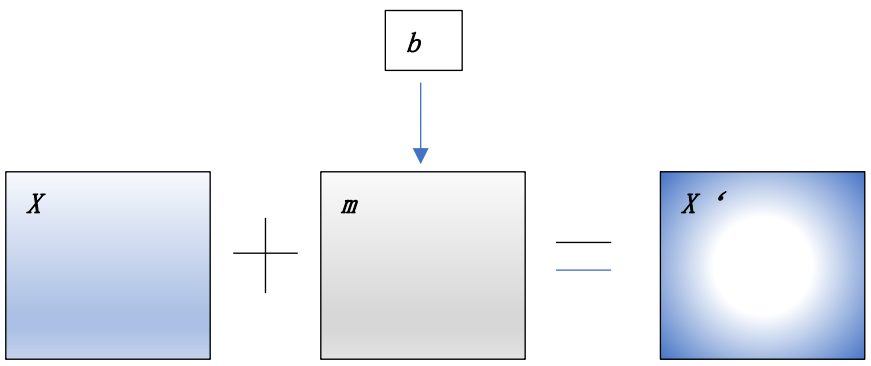


Fig. 1 A simple model of information hiding

The general model for data extraction is accordingly shown in Fig. 2. At the receiving end, the hidden information ***b*** can be extracted from the received ***X'***, (maybe ***X''*** if the stego media has encountered an attack during transmission). For reversible data hiding techniques, the original media ***X*** is further recovered from the received ***X'***.

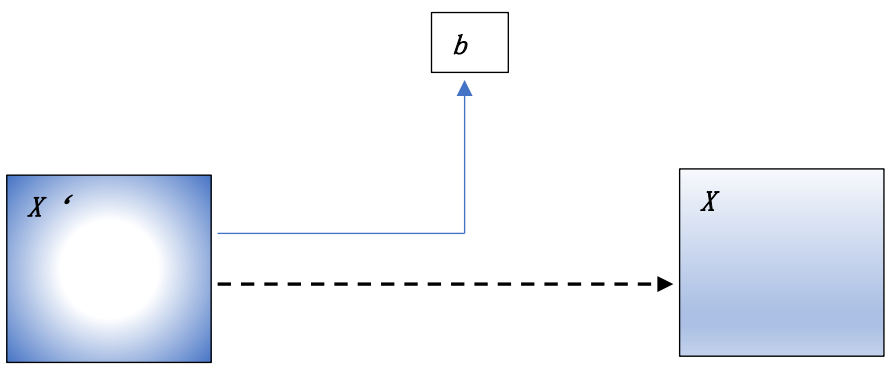


Fig. 2 General model of information extraction and original media recovery

Next, we discuss data hiding technology with image as host media. Image data hiding techniques are basically implemented along the similar steps as the above. In Fig. 3, the basic strategy is to choose some kind of object in the image, such as pixel in spatial domain or coefficient in transform domain, and appoint its two values with small difference correspond to the information bits "0" and "1" respectively. Next, the process of information hiding is to ensure that the value of the object matches the information bit to be hidden. If not, fine tune the object's value to match. At the receiving end, each hidden bit is extracted from the stego image by resolving the value of corresponding host object.

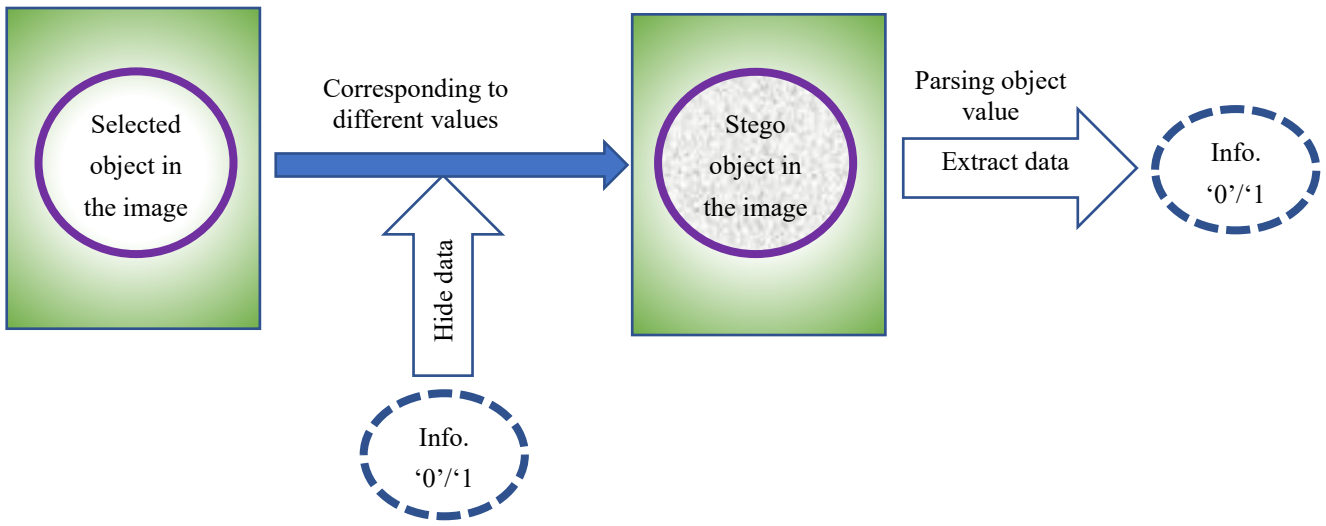


Fig. 3 Data hiding and data extraction

In some ways, data hiding can be analogous to digital modulation. The information bits to be hidden act as modulation signal, and the image acts as carrier signal, and the stego image corresponds to modulated signal. For images, the object that acts as basic carrier signal can be chosen from spatial domain or transform domain. For gray images in spatial domain, a single pixel, a pair of pixels, a pixel block, or a pair of pixel blocks can be used as basic carrier of information. In transform domain, a single coefficient, a pair of coefficients, a coefficient block, or a pair of coefficient blocks, can be used as basic carrier of information. There are many candidate transforms, such as Discrete Cosine Transform, Discreet Wavelet Transform, Discrete Fourier Transform, Haar Transform, Integer Wavelet Transform, and Contourlet Transform. For example, the high frequency coefficients of DCT are better suited for data hiding because they are visually more resistant to noise than low frequency coefficients. The DWT hides data in regions of high-resolution detail bands (HL, LH and HH) are less sensitive to human visual system. This is because that the higher sub band represents the finer details of the image, while the lowest sub band has the most important and relevant information of the image. Contourlet transform has wavelet features and the sub bands at each scale are decomposed into different directions. It provides multi-scale and multi directional representation of cover image using Laplacian Pyramid. Hence, so Contourlet transform is more effective than DWT to capture smooth contours and geometric structures.

To hide information, whether the information is a signature, voice or company logo, it is first encoded as a bit stream. Then, embed or parasitize the information bit by bit into the selected basic objects of the host image. The method of embedding or parasitizing information bits is very flexible. It can directly replace the LSB of the object value with information bits, or make rules to make information bits “0” and “1” correspond to different values of an object or different logical relations of a pair of objects, which is similar to digital modulation.

In least significant bit (LSB) based data hiding, the LSBs of pixel values [1-5] or high-frequency coefficients [6-7] are used as the holder of information bits. In [1-5], it works by replacing the LSBs of randomly selected pixels in the host image with secret message bits. There are many different data hiding methods based on LSB pixel or bit plane, and they have been developed in an effort to optimize payload while improving visual quality and undetectability. Some of them include adaptive LSB substitution based on edges, texture, intensity level, and the brightness of the host images [8]. Similarly, a different flavor such as optimized LSB substitution using cat swarm strategy [9], and LSB substitution with interpolation image [10] were also proposed.

In Even-Odd based data hiding, the parity of pixel value [11-13] or quantized high-frequency coefficient [14-18] is used to correspond to the bit to be hidden. Like LSB technique, a single pixel or quantized transform coefficient is used as the carrier of information, and its parity is directly used to represent the hidden bit. For example, if pixel value is odd, it means that it carries information bit 1; if it is even, it means bit 0. By adjusting the parity of the selected pixels or coefficients to meet the rules, information is hidden. At the receiver end, information can be extracted bit by bit by parsing the information hiding rule shared by the information embedder and extractor.

In Patchwork based image data hiding [19], the relative size of a certain characteristic value of a pair of blocks in an image is used to correspond to different bits (‘1’ or ‘0’). For example, the pixels in a region are divided into two blocks of the same size, *P* and *Q*, by sub sampling. Due to the proximity correlation of pixel values, the average values of pixels in *P* and *Q* are basically equal. Information hiding is realized according to the difference between the two blocks. To hide ‘1’, a small value is added to the pixel values in block *P,* and subtracted from the pixel values in block *Q*, making the average value of pixels in *P* slightly greater than in *Q*. (The average value of this region still remains unchanged). Since the difference can be well preserved under attacks such as image compression, and image de-noising, the algorithm has certain robustness.

In pixel value difference (PVD) based data hiding [20-25], the difference of a pair of pixels in the image is used as the basic carrier. To hide information bit, the difference value between two adjacent pixels is adjusted. The number of bits which can be hidden depends on the texture area of the image. Since the human vision system can tolerate more changes in sharp-edge blocks than in smooth blocks, so the larger the difference (higher texture), the more secret bits can be hidden in pixel pairs. To improve information hiding capacity, the PVD method is combined with adaptive LSB substitution and optimal pixel adjustment process [26]. To obtain good visual quality and high payload, Hsiao Shan et al. [27] proposed a multi-way PVD method by combining the tri-way PVD and mode selection process. Mandal et al. [28] proposed an adaptive PVD method with modulus function to resolve the fall-off-boundary conditions, while retaining payload and imperceptibility similar to original PVD.

In image watermarking techniques based on spread spectrum in spatial domain [29], whether the correlation value between host image and an equal-size matrix with small element values is greater than a threshold indicates whether the image is watermarked. In image watermarking techniques based on spread spectrum in transform domain [30], whether the correlation value between the coefficient matrix of the image in frequency domain and an equal-size matrix with small element values is greater than a threshold indicates the image is watermarked.

In self-reference based data hiding [31-32], a pixel and its estimated counterpart, or a high-frequency coefficient and its estimated counterpart, are used as the basic carrier of information. The relative size of their values represents information bit "0" or "1". For example, if the information bit to be hidden is consistent with the relative size of a pixel and its estimated counterpart, it is not necessary to modify pixel value. Otherwise, modify the pixel value to meet the relative size specified in the rule.

In Quantization Index Modulation (QIM) based data hiding [33-34], a single selected object, such as high frequency coefficient, of the host image is used to hide information bit. First of all, the QIM constructs a set of quantizers and assigns corresponding indexes. For binary data hiding, bit “0” and “1” correspond to two index values respectively. These two indexes correspond to two quantizers, which are used to quantize the selected objects of host image, so as to hide bit ‘1’ and ‘0’ respectively. In the aspect of information extraction, the minimum distance decoder or maximum likelihood decoder is used to complete the decoding.

Recently, some researchers have applied neural networks to steganography [35-45]. As a cooperative algorithm, convolutional neural network with deep supervision edge detector retains more edge pixels over conventional edge detection, so it can increase data hiding capacity [35]. In [36], the researchers combined the advantages of U-Net in image detail feature processing and the ability of wavelet transform to divide image details. To improve the capability of resisting steganalysis algorithms, an image steganography scheme based on neural network to transfer image is proposed [37]. In [38], support vector neural network is proposed to choose suitable pixels for data hiding. In steganography without embedding, based on GANs, the generated cover images themselves are stego images carrying secret information. But in this kind of steganography, there exists some weaknesses, such as low information recovery accuracy, low steganography capacity and poor natural showing. To improve the performance, some researchers

[39] proposed an attention-GAN model, where the generative model utilizes the attention method to improve the correlation among pixels and to correct errors such as image distortion and background abnormality. The soft margin discriminator is used to improve the compatibility of information recovery and fault tolerance of image generation. In [40], a steganography model—HIGAN is proposed. The encoding network composing of residual blocks hides color secret image into another color image of the same size, outputting a higher visual quality steganographic image. Moreover, it utilizes the adversarial training between the encoder-decoder network and the steganalysis model to improve the ability to resist the detection of steganalysis models based on deep learning. In coverless image steganography [41, 42], deep learning is applied to extract high-dimensional CNN features which are mapped into hash sequences. For the sender, a binary tree hash index is built to accelerate index speed of searching hidden information and DenseNet hash sequence. Then, all matched images are sent to the receiver. For the receiver, the secret information can be resolved successfully by calculating the DenseNet hash sequence of the cover image. In [43], they map the secret message to a polynomial and encode it into a set of points with different polarities, then generate a fingerprint image based on a piece of hologram phase constructed from the secret message. For the receiver, secret message can be extracted by detecting the encoded points in the constructed fingerprint. In [44], batch steganography using a generative network is proposed to information in multiple images. In order to counter steganalysis by trying to fool the machine learning classifiers, researchers in [45] proposed an adversarial embedding technique to generate adversarial stego images with minimum amount of adjustable elements.

In this paper, we present a data hiding technique based on looking-up an information table in which each element is an octal digit. The table is shared by both data hider and data extractor. Data hiding and data extraction are realized by querying the table with a pair of pixels of the host image. The proposed technique can achieve high embedding capacity, which makes it suitable for image steganography and image tampering authentication.

In order to facilitate reading and understanding the paper, some terms appearing in the paper are explained here. The “information table”, “code table”, or “table” means that 256x256 matrix in which the value of each element is an unsigned integer in [0, 7]. The “seed block”, or “block” means 3x3 matrix in which the value of each element is also an unsigned integer in [0, 7]. The information table is constructed by cloning a seed block. The “sliding window”, or “window” is a block which is temporarily enclosed in the table with an element in the table as the center (focus), and its shape and size are the same as the seed block. The “focus element” is an element located in the table with a pair of original pixel values of the host image.

The rest of the paper is organized as follows. Section 2 introduces the creation of an information table, information data hiding and data extraction; Section 3 presents the experimental results and performance analysis. Section 4 summarizes the paper, and analyzes the feasibility of the proposed method as a module applied to other data hiding methods.

2. Proposed Technique

Our proposed data hiding technique includes three parts, namely, creation of information table, data hiding by locating element in the table, and data extraction by looking-up the table. The table is shared by both data hiding and data extraction, and queried with a pair of pixels of host image.

2.1 Creation of an Information Table

Creation of Seed Block

The seed block is a 3x3 matrix. With a key, KEY_g, set the value of one element as any unsigned integer in [0, 7], say 0 as indicated with red in Fig. 4. Then set the other 8 elements in the block with 8 different unsigned integers in [0, 7]. Figure 4 shows an implementation of a seed block. There is total 8 x8! = 322560 implementations of seed blocks. This greatly enhances the security of data hiding, because information table is constructed with a seed block, which will be introduced next.

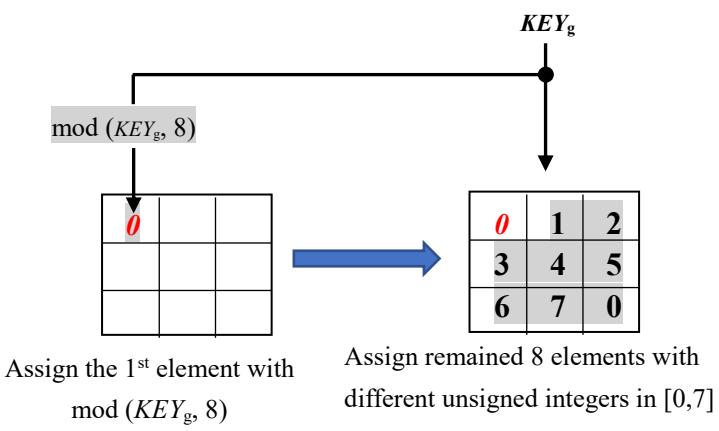


Fig. 4 An example of a seed block

Construction of Information Table

First, generate an empty information table, *i.e.*, a matrix with a size of 256x256. In the matrix, the columns and rows are represented by x and y respectively, where $\{x, y\} \in [0,255]$. Starting from the top left corner of the matrix, duplicate a 3x3 seed block horizontally and vertically without overlapping. Because the remainder of 256/3 is 1 instead of 0, the rightmost column of the table, *i.e.*, x=255, does not belong to a complete block. The bottom row of the table, y=255, is similar. As an isolated element, the element labeled purple at the lower right corner of the matrix can be assigned any value from 0 to 7, say 0. Finally, the information table is obtained, as shown in Fig. 5.

Let's observe the characteristics of the information table. Placing a 3x3 window (shown as a green dotted frame) anywhere in the table, eight different element values, 0 to 7, appear in the window. Therefore, if the element located in the table by a pair of pixel values (x, y) is not what we need, we can use the focus element to enclose a window. In the window, the element we need can be definitely found. Four examples of windows are shown in the Fig. 5, labeled as 1, 2(a), 2(b) & 2(c), representing the window located respectively inside the table, at the corner of the table, and on the edge of the table, respectively. In each window, the bold digit pointed by an arrow is the focus element, around which a window is formed.

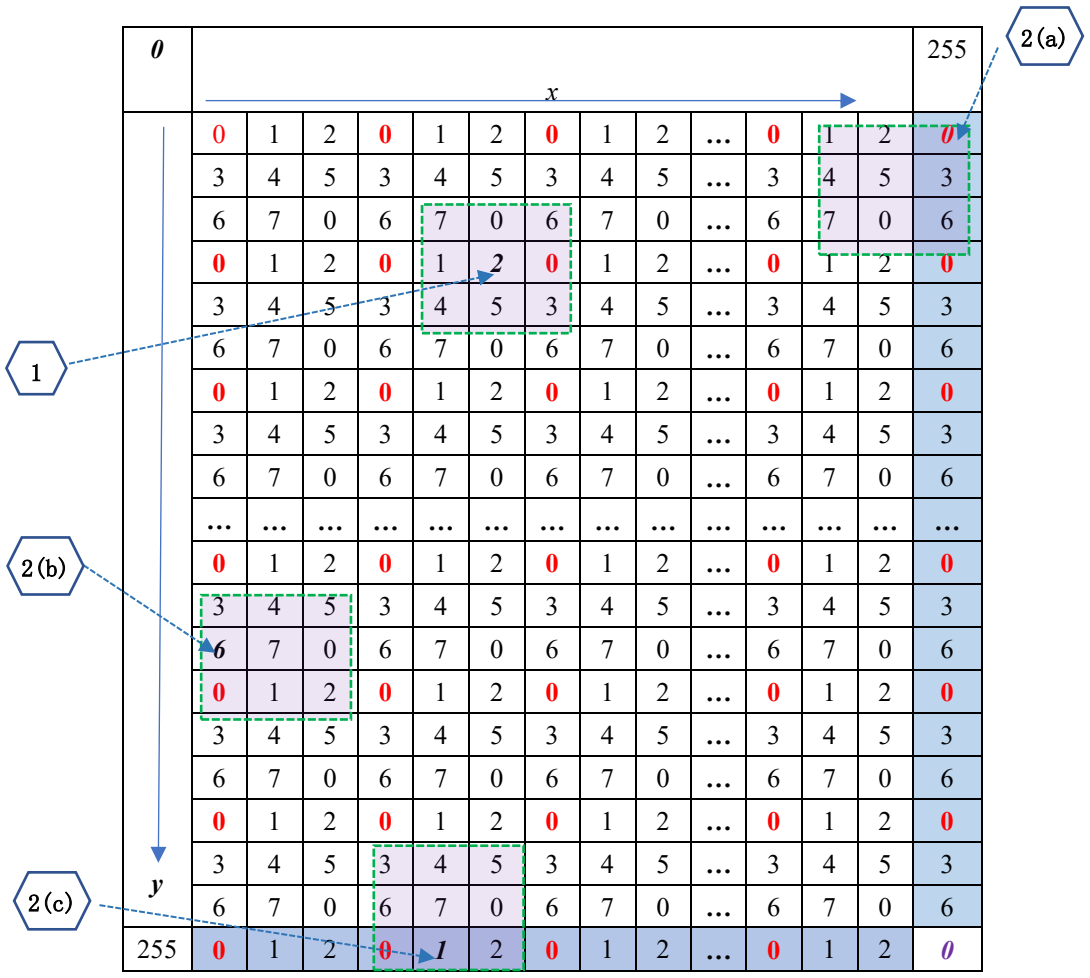


Fig. 5 Semi-randomly generated Coding Table

Next, we discuss the moving steps within the window from the focus element to the one we search for.

(1) The expression $(x, y) = ([1, 254], [1, 254])$ means the value of x is between 1 and 254, and y is between 1 and 254. Let (x, y) be the coordinates of the focus element. If $(x, y) = ([1, 254], [1, 254])$, the window is completely inside the table, and 1 in Fig. 5 is such an example. In such a window, moving from the focus element to any other element in the horizontal or vertical direction, the movement of x or y is only 1. When moving from the focus element to the element in the diagonal direction, the movement of both x and y is 1 (increasing or decreasing by 1). The result is summarized in Table 1, in which we assume that the focus element is 2 as shown in 1 in Fig. 5. From Table 1, we can see that from focus element to four elements (6, 7, 4, 3), the sum of changes in x and y is 2; while from focus element to other four elements (0, 1, 5, 0), the sum of changes in x and y is only 1.

(2) If the focus element is at the edge of the table, *i.e.*, $(x, y) = \{(0, \text{any}), (255, \text{any}), (\text{any}, 0), \text{or } (\text{any}, 255)\}$, the window cannot be enclosed around the focus element in the table. This can be further divided into two cases:

(2.1) If the coordinates of the focus element, $(x, y) = \{(0, 0), (0, 255), (255, 0), (255, 255)\}$, the focus element is located at the corner of the table. With the focus element as a vertex, a 3x3 window is formed in the table, such as 2(a) in Fig. 5. Since the distance from the focus element to other elements in the window varies, the change of x and y of the focus element is very different when moving the focus element to other elements. We summarize the result in Table 2.

Table 1 Movement of x or y within a window completely inside the table

Destination elements (The focus element is 2 as shown in 1 in Fig. 5)	Sum of changes in x and y	Number of destination elements
0, 1, 5, 0	1	4
6, 7, 4, 3	2	4

Table 2 Movement of x or y within a window formed by the focus element located at the corner of the table

Destination elements (The focus element is 0 as shown in 2(a) in Fig. 5	Sum of changes in x and y	Number of destination elements
2, 3	1	2
1, 5, 6	2	3
4, 0	3	2
7	4	1

(2.2) If the coordinates of the focus element, $(x, y) = \{(0, [1, 255]), (255, [0, 254]), ([1, 255], 0), ([0, 254], 255)\}$, the focus element is located in the outermost column or row of the table, but not at the corner of the table. Take the focus element as the midpoint of the boundary side to form a 3x3 window in the table, such as 2(b) and 2(c) in Fig. 5. Moving from the focus element to other elements in the window, the change of the coordinate values x and y of the focus element is $\Delta x = \{0, 1, 2, -1, -2\}$ and/or $\Delta y = \{0, 1, 2, -1, -2\}$. Like in Table 1 and Table 2, we only consider the sum of absolute movement of x and y, and summarize the results in Table 3.

Table 3 Movement of x or y within a window formed by the focus element located in the outermost column or row of the table

Sum of changes in x and y	Destination elements (The focus element is 6 in 2(b) of Fig. 5)	Destination elements (The focus element is 1 in 2(c) in Fig. 5)	Number of destination elements
1	3, 7, 0	0, 7, 2	3
2	1, 0, 4	6, 4, 0	3
3	5, 2	3, 5	2

2.2 Data hiding

For an 8-bit gray image, each pixel value is an unsigned integer between [0, 255]. Correspondingly, the size of the information table in Fig. 5 is 256x256, so its row and column numbers are also assigned as unsigned integers between [0, 255]. The value of each element in the table is an unsigned integer between [0, 7], *i.e.*, 3 bit binary data. To hide information in the image, we first encode information into a binary bit stream, then divide the bit stream into 3-bit segments, each of which is between [000, 111], *i.e.*, unsigned integer between [0, 7]. Each time, 3 bits can be hidden in the image. To do so, a pair of pixels in the image are selected, then use their pixel values p_i and p_j as column and row (x and y), respectively, to locate an element e in the information table. If e is equal to the 3-bit value we will hide, nothing needs to be done; Otherwise, enclose a window in the table with e as the focus element. In the window, search for an element e' equal to the 3-bit value to be hidden. Finally, update the pixel values p_i and p_j with the column and row coordinates of e', respectively. Figure 6 depicts the entire data hiding process, and the steps are summarized as follows.

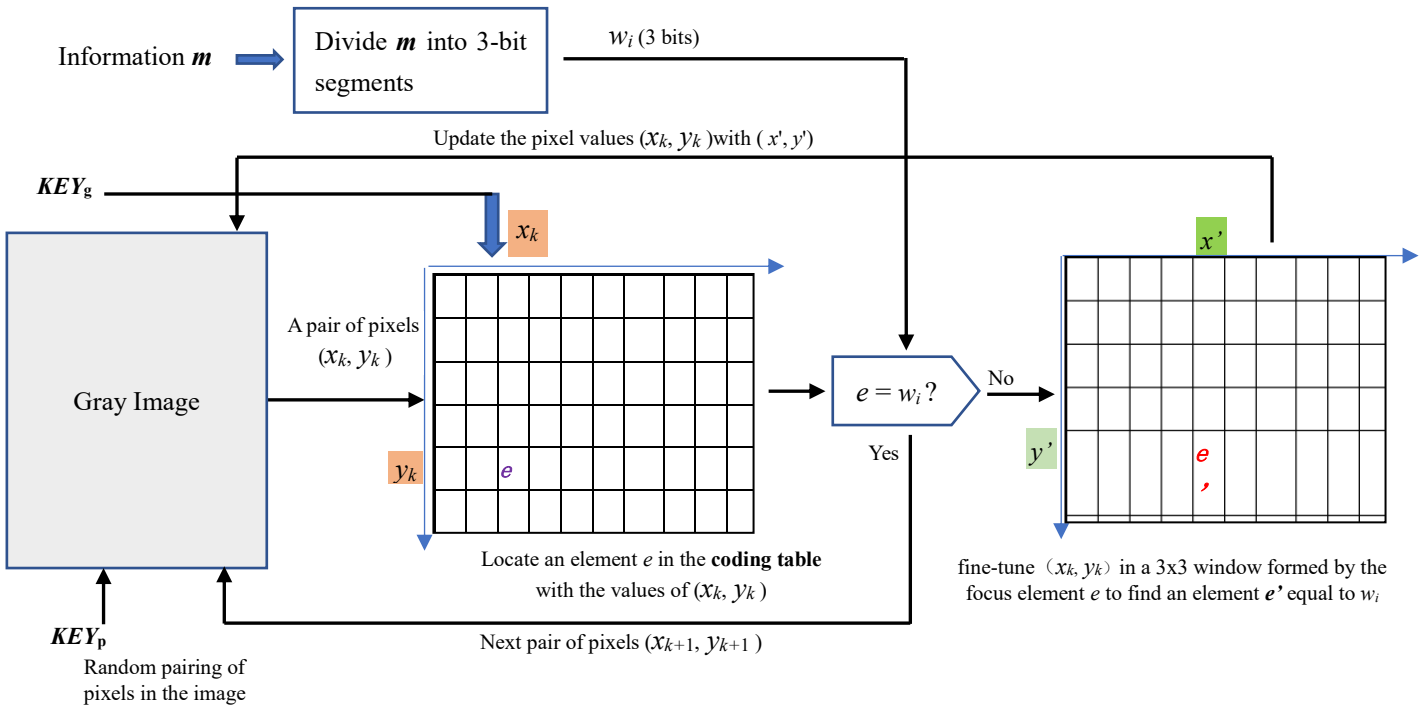


Fig. 6 Information hiding by looking up information table with a pair of pixel values of host image

Step 1: For simplicity, assume the length of the information bit stream m is multiple of 3. Divide m into 3-bit segments, so the value of each segment is unsigned integer between 0 and 7. Suppose there are T segments, then $m = 3 * T$ bits;

Step 2: Suppose that there are F pixels in the gray image, and use a key $\boldsymbol{KEY_p}$ to pair pixels, forming $N = \lfloor F/2 \rfloor$ pairs of pixels, where $\lfloor . \rfloor$ is the floor function. Assume $N \geq T$, *i.e.*, so the image can hold all information;

Step 3: Take 3 bit w_i, $i = 1, 2, ..., T,$ from the information m, and a pair of pixels $\{x_k, y_k\}$, $k = 1, 2, \ldots, N$ from the image. Use the pixel values x_k and y_k as row and column numbers to locate an element e in the information table;

Step 4: If $e = w_i$, the pixel values x_k and y_k will not change, and go to Step 5; If $e \neq w_i$, use e as the focus element to construct a 3x3 window in the table, as discussed in Section 2.1. In the window, find out the element e' which is equal to w_i. (If there is more than one element equal to w_i in the window, choose the one closest to e). Then, replace the pixel value of pair $\{x_k, y_k\}$ with the row number and column number of e';

Step 5: $i = i + 1$. If $i > T$, go to Step 6; otherwise, go to Step 3;

Step 6: End of information hiding.

Now, assuming that the number of pixels F is even, we derive the total number of pixel pairing strategies in the following:

For a given pixel at the beginning, there can be F-1 pixels paired with it. When the pixel is paired, the number of remaining pixels is F-2. In the remaining F-2 pixels, for a given pixel, there can be F-3 pixels paired with it. When the pixel is paired, the number of remaining pixels is F-4. In the remaining F-4 pixels, for a given pixel, there can be F-5 pixels paired with it. …, and so on. Near the end, when a pixel is paired, the number of remaining pixels is F-(F-4) =4. Then, in the remaining 4 pixels, for a given pixel, there can be 3 pixels paired with it. When the pixel is paired, the number of remaining pixels is F-(F-2) =2. In the remaining *2* pixels, for a given pixel, there can be 1 pixel paired with it. Through above analysis, we can derive that the total number of pixel pairing strategies = (F-1) + (F-3) + (F-5) +…+3 +1= (F^2)/4.

2.3 Data extraction

Because information hidden in the image is "recorded" in the information table, information can be extracted by looking up the table. First, the information extractor should build the same information table as in data hiding stage. Next, the process of information extraction is straight forward, as shown in Fig. 7. The steps of information extraction are as follows.

Step 1: To extract information from the stego image, use the same key $\boldsymbol{KEY_p}$ to pair pixels as done in data hiding stage, getting N pairs of pixels. Let m represent the container of information, which is initially empty.

Step 2: Sequentially select paired pixels $\{x_k, y_k\}$, $k = 1, 2, \ldots, N$ from the image. Use x_k and y_k as row and column numbers to locate an element e in the information table;

Step 3: Convert the value of e into 3-bit binary, then place it in m in sequence.

Step 4: $i = i + 1$. If $i > T$, go to Step 5; otherwise, go to Step 2;

Step 5: End of information extraction.

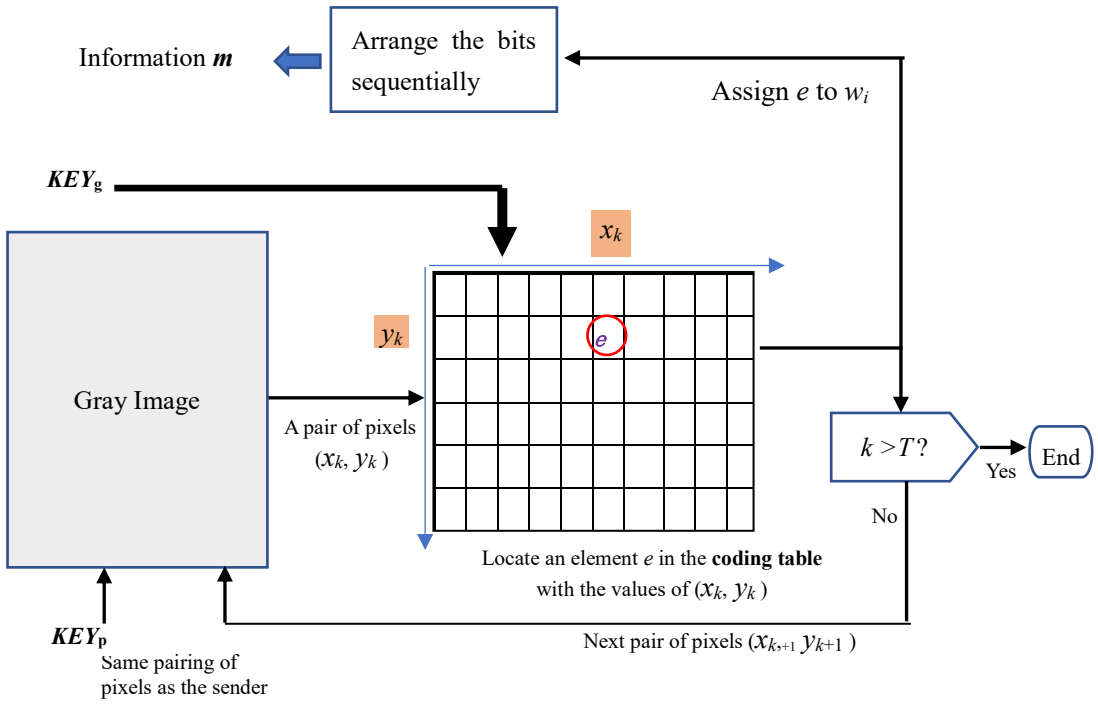


Fig. 7 Information extraction by looking up the code table with pair of pixels

3. Experiments and Performance analysis

Ten standard gray images with the size of 512x512 are used as test images in our experiments, including *Lake*, *Baboon*, *Peppers*, *Airplane*, *Couple*, *Milkdrop*, *Woman*, *Boats*, *Lena*, and *Tiffany,* shown in Fig. 8. In the following experiments, we will present and analyze the capacity of data hiding, imperceptibility of host image, security of data extraction, computational cost of the proposed algorithm, and the expansibility of the algorithm.

Fig. 8. Test images in the experiments

3.1 Capacity

Since a pair of pixels in a host image is used as the carrier of data hiding (or more accurately as the index of look-up table), the amount of data hiding is only related to the size of the image, nothing to do with the content of the image. Since each pair of pixels can carry 3 bits of information, for an image with a size of 512x512, data hiding capacity is $\lfloor(512x512)/2\rfloor*3$ bits = 393216 bits, or 1.5bits per pixel (bpp). The ten standard gray images in Fig. 8 are 512x512 in size, so data hiding capacity in each image is 393216 bits.

3.2 Imperceptibility

As we know, no matter what image data hiding technology is applied, pixel values of host image will be modified directly or indirectly. In our proposed technique, the exact number of pixels modified and the amount of modification of a single pixel are related to three factors:

(1) the amount of data actually hidden, and the specific composition of a bit stream of data to be hidden,
(2) the specific information table, and
(3) the content of image, and pixel pairing strategy.

In our experiment, we make the following settings:

(1) the amount of data to be hidden is 393216 bits, which is the maximum capacity that can be hidden in a 512x512 image, as described in Section 3.1. The 393216 bits consists of bit 1 and bit 0 alternately, that is, "10101010...".

(2) as mentioned in section 2.1, there are 322560 different seed blocks. Accordingly, 322560 different information tables can be constructed accordingly. Here, we choose a specific information table shown in Fig. 5 in our experiment.

(3) as derived in Section 2.2, the total number of pixel pairing strategies is $(F^2)/4$. For an image of size 512x512, F= 512x512 = 262144, so the total number of pixel pairing strategies = $(262144)^2/4 = 1.7180x10^{10}$. Here, we choose one of the pairing strategies -- two adjacent pixels in the same row as a pair. The pixel pairing does not overlap, that is, paired pixels are no longer paired with other pixels.

We perform data hiding in the10 test images respectively. The PSNR values of the stego images are listed in Table 4.

Table 4 PSNR of each stego image in which 393216 bits are hidden

Host images	PSNR (dB)
Lake	53.29
Baboon	52.34
Peppers	52.17
Airplane	53.13
Couple	51.38
Milkdrop	52.67
Woman	51.25
Boats	51.33
Lena	53.67
Tiffany	50.45

From Table 4, we can find that the PSNR values are all very high. This is because if the value of focus element positioned by a pixel pair is exactly equal to the 3-bit value to be hidden, the values of the pixel pair remain unchanged. The probability of this situation within a window is 1/9. If the value of the located element is not equal to 3-bit value to be hidden, we take the located element as focus to form a 3x3 window, then find the element equal to the 3-bit value by moving row and/or column in the window. Finally, the pixel values of the pairs in the host image are replaced with the row and column of the found element in the window. As described in Section 2.1, if the element initially located by the pixel pair falls on the table boundary, the final change of the pixel pair is often larger. Let's look at the probability of this situation in the following.

The statistical result of using adjacent pixels to look up the table is equivalent to the co-occurrence matrix of the host image. In our experiment, each pair of horizontally adjacent pixels is used to locate element in the information table. Equivalently, we work out the horizontal co-occurrence matrix of each host image, shown in Fig. 9. In Fig. 9, the size of the co-occurrence matrix is also 256x256, which is the same as the code table. The co-occurrence matrix can reflect the positon distribution of focus elements in the information table. In Fig. 9, the numbers marked in rows and columns, 0 -- 255, represent gray values of the host image. The brightness in the figure represents the relative values of elements --- black indicates small value, and white indicates large value. From Fig. 9, we can see that the proportion of white dots on the boundary is very small. This indirectly tells us that number of elements located by pixel pairs in the information table falls on the edge of the table is small. In other words, in most cases, the focus element is inside the information table, like 1 of Fig. 5. Therefore, if the value of focus element is not equal to the 3-bit information, the correct element can be found by moving only 1 horizontally or vertically, or 1 in the diagonal direction from the focus element in most cases. For the former, only one pixel value changes by1. For the latter, both pixel values change by 1. The probability of changing only one pixel value and the probability of changing two pixel values are all 4/9.

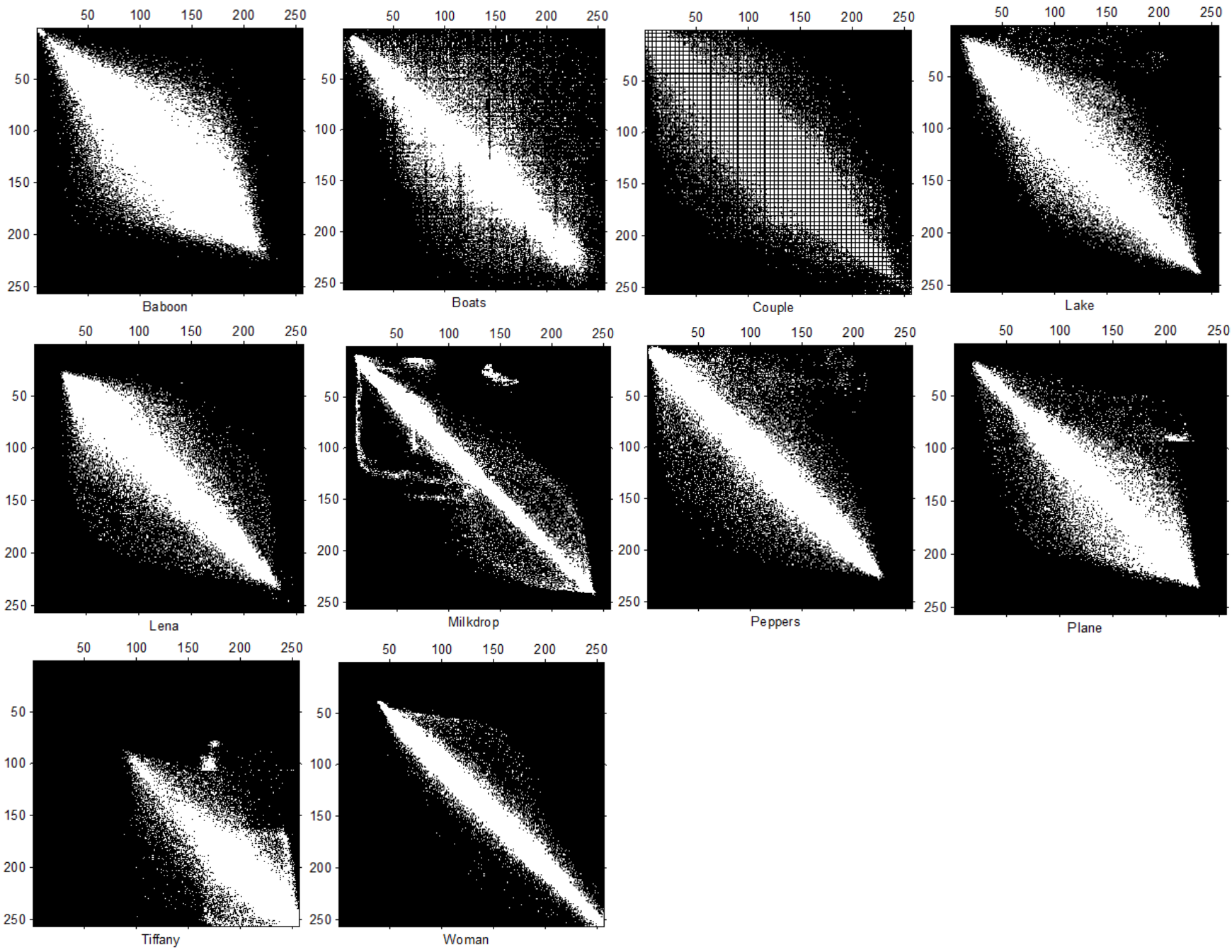


Fig. 9 Co-occurrence matrix of each host image

3.3 Security

The security of the proposed algorithm is reflected in two aspects. One aspect is the construction of information table. The information table is generated by the duplication of a specific 3x3 seed block. Since there are a total of 8 x8! = 322560 implementations of seed blocks, accordingly there are 322560 possible information tables in total. The

selection of a seed block is based on a key KEY_g. Another aspect is the selection of pixel pairs from the host image. Pixel pairing in an image is based on a key KEY_p. For an image with F pixels, the total number of pixel pairing strategies is $(F^2)/4$, which is calculated in Section 2.2. Without knowing the KEY_g, a third party cannot construct the correct information table. Without knowing the KEY_p, a third party cannot extract hidden information correctly even if s/he knows the information table.

3.4 Computational cost

The time complexity of this algorithm is $O(n)$, where n is the number of times to execute the embedding algorithm. Each time, three bits can be hidden in the image. Both information hiding and extraction are based on looking up an information table, and element searching is limited to a range of 3x3 window, so the computational cost is very low. For data hiding, the modification of some pixels is done by simple data replacement. Besides, since element matching is within the information table, there is no overflow of pixel values.

3.5 Comprehensive performance comparison

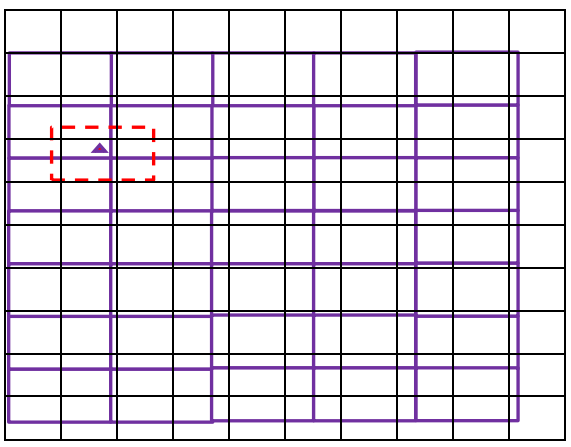
The quantitative test results of data hiding methods are related to parameters of the algorithms, bit stream composition of data to be hidden, and the content of host image. According to the test results of data hiding in a large number of images, we get the qualitative comparison results shown in Table 5. From Table 5, it can be seen that our method has advantages in capacity, PSNR, security and computational cost. Our technique is not robust against attacks, so it belongs to fragile watermarking. It is suitable for applications such as image active authentication and image tampering detection.

Table 5 Comprehensive performance comparison

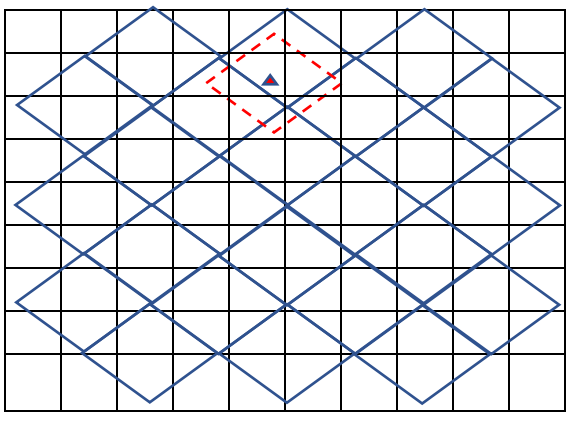
	Capacity	**PSNR**	**Security**	**Computational cost**	**Robustness**
LSB[1]	High	High	Low	Low	Low
Patchwork[19]	Low	High	Low	High	Limited
SS[29]	Low	High	Low	High	Limited
PVD[20]	High	High	Moderate	Moderate	Low
QIM[33]	Low	High	Low	High	Limited
Ours	Very high	Very High	Very High	Low	Low

3.6 Extensibility of Information Table

Before information is hidden in an image, it is usually encoded into a binary bit stream. Next, as a generalization of this paper, the bit string is then cut into m-bit segments of equal length as basic hidden unit, and each m-bit segment is converted into decimal number as element value in seed block. Therefore, the values of elements in the block should include 0, 1, 2, ..., 2^m-1 completely. In this paper, m is set to 3. If the number of elements in the seed block is N, $N \geq 2^m$, there will be N-2^m redundant elements, *i.e.*, elements with duplicate values. In this paper, the information table is generated by cloning a 3x3 seed block, so the number of elements N in the block is 9. The information bit string is cut into segments of m = 3, so the values of elements in the block are 0, 1, 2, ..., 7. There exists N-2^m =9-8=1 redundant element. Theoretically, the table can be built by cloning seed block of other size or shape, such as rectangle or diamond, like Fig. 10. In Fig. 10, the small red triangle represents the focus element. The red dashed box is the window formed with the focus element. The window is the same shape and size as the seed block. Within the window, the element equal to the data to be hidden can be found.



(a) seed block is rectangle



(b) seed block is diamond

Fig. 10 Information table generated by cloning a rectangle or diamond block

Theoretically, any size of seed block can be employed. The larger the block, the higher the embedding rate, but the movement in the window to find the matching element increases. This will decrease the PSNR of the stego image. Data hiding efficiency is affected by the number of redundant elements in the seed block, and the moving distance in the window. Besides, irregularity of seed block splicing at table boundary, as shown in Fig. 10, also affects data hiding efficiency. Figure 11 shows the relationship between the number of bits m embedded at one time and PSNR value. Figure 12 shows the relationship between the number of bits m embedded at one time and the number of hidden bits per pixel BPP (bpp).

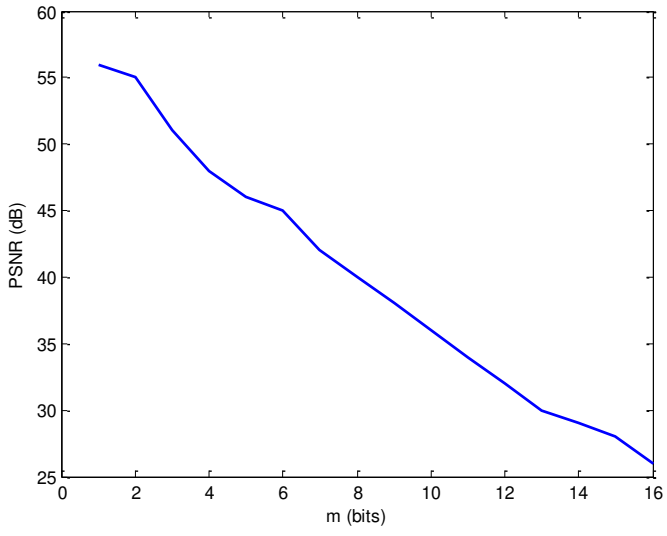


Fig. 11 Number of bits *m* embedded at one time *vs.* average PSNR of images

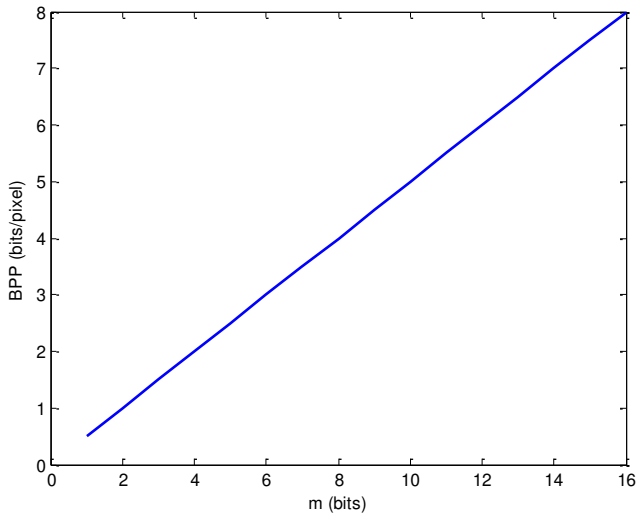


Fig. 12 Number of bits *m* embedded at one time *vs.* bpp

4. Conclusions

This paper presents an image data hiding technique based on an information table in which each element is a 3-bit binary data. To hide information in a gray image, the gray values of a pair of pixels are used as coordinates to locate an element in the table. If the element value is equal to the 3-bit information value, the pixel values of the pair remain unchanged. Otherwise, search for the element equal to 3-bit information value in a small area of the table, then replace the pixel values with the coordinate values of the found element. To extract information at the receiver end, the gray values of a pair of pixels in the host image are used as coordinates in the table to locate an element. The value of the located element is the extracted 3-bit information value.

As the basic carrier of information hiding, a pair of pixels can carry 3 bits of information, so the efficiency of data hiding is high. If the selected pair of pixels are adjacent, the probability of the located focus element falling on the edge including at the vertex of the information table is very low, so the moving step required to find the correct element is small, thus the modification of the pixel values is small. Therefore, the stego image has high fidelity.

The row and column values of the information table are [0, 255]. The pixel values are updated with the values of the rows and columns of the matched element in the information table, so pixel values will not overflow after data hiding.

By applying the technique to RGB planes respectively, the proposed method can be applied to data hiding in color images. The embedding capacity of color image will be three times the same size gray image. This algorithm can also be used as a common module in other information hiding techniques, such as audio data hiding. For example, if the value range of the basic carrier is [a, b], and the minimum step that can be changed is $\triangle$, the seed block of 3x3 is constructed in the same way as in this paper. By duplicating the seed block, an information table with the size of $\left\lfloor\frac{b-a+1}{\Delta}\right\rfloor$ x$\left\lfloor\frac{b-a+1}{\Delta}\right\rfloor$is constructed, then information hiding can be realized by looking up the table as well.

Conflict of interest

All authors declare that they have no conflict of interest.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Data Availability

All data generated or analyzed during this study are included in this article.

Acknowledgement

This work was supported by 2018 international academic cooperation cultivation project of Wuhan University, China.

References

[1] C.-K. Chan, L.-M. Cheng, Hiding data in images by simple LSB substitution, Pattern Recognition, 37(1), 2004, pp. 469-474.

[2] Lakhdar Laimeche, Abdallah Meraoumia, Hakim Bendjenna, Enhancing LSB embedding schemes using chaotic maps systems, Neural Computing and Applications, 2020, Vol. 32, pp. 16605–16623, https://doi.org/10.1007/s00521-019-04523-z

[3] Serdar Solak, High Embedding Capacity Data Hiding Technique Based on EMSD and LSB Substitution Algorithms, IEEE Access, 2020, Vol. 8, pp. 166513 – 166524

[4] Pratik D. Shah, Rajankumar S. Bichkar, Secret data modification based image steganography technique using genetic algorithm having a flexible chromosome structure, Engineering Science and Technology, an International Journal, 2021, Vol. 24, Issue 3, pp. 782-794, https://doi.org/10.1016/j.jestch.2020.11.008

[5] V. Kalaichelvi, P. Meenakshi, P. Vimala Devi, H. Manikandan, P. Venkateswari, S. Swaminathan, A stable image steganography: a novel approach based on modified RSA algorithm and 2–4 least significant bit (LSB) technique, Journal of Ambient Intelligence and Humanized Computing, 2021, Vol. 12, pp. 7235–7243, https://doi.org/10.1007/s12652-020-02398-w

[6] Zihan Yuan, Decheng Liu, Xueting Zhang, Huanying Wang, Qingtang Su, DCT-based color digital image blind watermarking method with variable steps, Multimedia Tools and Applications, 2020, Vol. 79, pp. 30557–30581, https://doi.org/10.1007/s11042-020-09499-w

[7] Liyan Zhu, Xiangyang Luo, Chunfang Yang, etc., Invariances of JPEG-quantized DCT coefficients and their application in robust image steganography, Signal Processing, 2021, Vol. 183, https://doi.org/10.1016/j.sigpro.2021.108015

[8] Supriadi Rustad, De Rosal Ignatius Moses Setiadi, Pulung Nurtantio Andono, Inverted LSB image steganography using adaptive pattern to improve imperceptibility,Journal of King Saud University - Computer and Information Sciences, 2021, online, https://doi.org/10.1016/j.jksuci.2020.12.017

[9] Ashraf AbdelRaouf, A new data hiding approach for image steganography based on visual color sensitivity, Multimedia Tools and Applications, 2021, Vol. 80, pp. 23393–23417, https://doi.org/10.1007/s11042-020-10224-w

[10] Lili Tang, Dongrui Wu, Honghui Wang, Mingzhi Chen, Jialiang Xie, An adaptive fuzzy inference approach for color image steganography, Soft Computing, 2021, Vol. 25, pp. 10987–11004, https://doi.org/10.1007/s00500-021-05825-y

[11] Aruna Malik, Samayveer Singh, Rajeev Kumar, Recovery based high capacity reversible data hiding scheme using even-odd embedding, Multimedia Tools and Applications, Vol. 77, 2018. pp. 15803–15827

[12] Sandeep Singh, Jaspreet Kaur, Odd-Even Message Bit Sequence Based Image Steganography, International Journal of Computer Science and Information Technologies, 6 (4), 2015, pp. 3930-3932

[13] Ramandeep Kaur Brar, Ankit Sharma, Improved Steganography Using Odd Even Substitution, Cognitive Computing in Human Cognition. Learning and Analytics in Intelligent Systems, Vol 17. Springer, 2020, pp. 1-8

[14] Inas Jawad Kadhim, Prashan Premaratne, Peter James Vial, High capacity adaptive image steganography with cover region selection using dual-tree complex wavelet transform, Cognitive Systems Research, 2019, Vol. 60, pp. 20-32, https://doi.org/10.1016/j.cogsys.2019.11.002

[15] Rajwinder Kaur, Butta Singh, A hybrid algorithm for robust image steganography, Multidimensional Systems and Signal Processing, 2021, Vol. 32, pp. 1-23, https://doi.org/10.1007/s11045-020-00725-0

[16] Nabanita Mukherjee (Ganguly), Goutam Paul, Sanjoy Kumar Saha, Two-point FFT-based high capacity image steganography using calendar based message encoding, Information Sciences, 2020, Vol. 552, pp. 278-290, https://doi.org/10.1016/j.ins.2020.11.044.

[17] Sudipta Kr Ghosal, Souradeep Mukhopadhyay, Sabbir Hossain, Ram Sarkar, Exploiting Laguerre transform in image steganography, Computers & Electrical Engineering, 2021, Vol. 89, https://doi.org/10.1016/j.compeleceng.2020.106964

[18] Adnan Gutub，Faiza Al-Shaarani，Efficient Implementation of Multi-image Secret Hiding Based on LSB and DWT Steganography Comparisons，Arabian Journal for Science and Engineering, Vol. 45, 2020, pp. 2631-2644

[19] W. Bender, D. Gruhl, N. Morimoto and A. Lu. 1996. Techniques for Data Hiding, IBM Systems Journal, MIT Media Lab, 1996, 35(3), pp. 313-336.

[20] D.-C. Wu, W.-H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognition Letter, 24 (2003) 1613--1626.

[21] Guang Hua, Lifan Zhao, Haijian Zhang, Guoan Bi, Yong Xiang, Random Matching Pursuit for Image Watermarking, IEEE Transactions on Circuits and Systems for Video Technology, 2019, Vol. 29, Issue 3, pp. 625 – 639

[22] G. Swain, Very high capacity image steganography technique using quotient value differencing and LSB substitution, Arabian Journal of Science Engineering, 2019, Vol. 44, No. 4, pp. 2995–3004

[23] Aditya Kumar Sahu, Gandharba Swain, Monalisa Sahu, J. Hemalatha, Multi-directional block based PVD and modulus function image steganography to avoid FOBP and IEP, Journal of Information Security and Applications, 2021, Vol. 58, https://doi.org/10.1016/j.jisa.2021.102808

[24] C.-H. Yang, S.-J. Wang, C.-Y. Weng, Capacity-raising steganography using multi pixel differencing and pixel-value shifting operations, Fund. Inform., 2020, Vol. 98, pp. 321-336.

[25] F. Pan, J. Li, X. Yang, Image steganography method based on PVD and modulus function, in: Electronics, Communications and Control, ICECC, 2011 International Conference on, IEEE, 2011, pp. 282-284.

[26] M. Khodaei, K. Faez, New adaptive steganographic method using least significant bit substitution and pixel-value differencing, IET Image Processing, 2012, Vol. 6, pp. 677-686.

[27] H.-S. Huang, A combined image steganographic method using multi-way pixel value differencing, in: Sixth International Conference on Graphic and Image Pro-cessing, ICGIP 2014, International Society for Optics and Photonics, 2015 pp. 944319-944319-944315

[28] J. Mandal, D. Das, Steganography using adaptive pixel value differencing (APVD) of gray images through exclusion of overflow/underflow, Signal Processing, 2017, Vol. 122, pp. 234-244

[29] Santi Maity, Malay Kumar Kundu, Spatial image watermarking using spread spectrum modulation, 2nd International Conference on Computers and Devices for Communication System (CODEC-04), January 2004

[30] Hamidreza Sadreazami, Marzieh Amini, A robust spread spectrum based image watermarking in ridgelet domain, AEU - International Journal of Electronics and Communications, 2012, Vol. 66, Issue 5, pp. 364-371

[31] Yulin Wang, A. Pearmain.2004. Blind image data hiding based on self reference, Pattern Recognition Letters, 2004, Vol. 25, Issue 15, pp. 1681-1689

[32] Chin-Chen Chang, Pei-Yu Lin, Jung-San Lee, A Self-Reference Watermarking Scheme Based on Wet Paper Coding, 2009 IEEE Ninth International Conference on Hybrid Intelligent Systems, 12-14 Aug. 2009,Shenyang, China

[33] B. Chen and G. W. Wornell. 1999. An Information-Theoretic Approach to the Design of Robust Digital Watermarking Systems, Proceedings of international conference on Acoustics, Speech and Signal Processing (ICASSP), Phoneix, AZ, March 1999

[34] Zhaoxia Yin, Longfei Ke, Robust Adaptive Steganography Based on Dither Modulation and Modification With Re-Compression, IEEE Transactions on Signal and Information Processing over Networks, 2021, Vol. 7, pp. 336 – 345

[35] Biswarup Ray, Souradeep Mukhopadhyay, Sabbir Hossain, Sudipta Kr Ghosal, Ram Sarkar, Image steganography using deep learning based edge detection, Multimedia Tools and Applications, 2021, https://doi.org/10.1007/s11042-021-11177-4

[36] Lianshan Liu, Lingzhuang Meng, Yanjun Peng, Xiaoli Wang, A data hiding scheme based on U-Net and wavelet transform, Knowledge-Based Systems, 2021, Vol. 223, https://doi.org/10.1016/j.knosys.2021.107022

[37] Qi Li, Xingyuan Wang, Bin Ma, etc., Image steganography based on style transfer and quaternion exponent moments, Applied Soft Computing, 2021, Vol. 110, https://doi.org/10.1016/j.asoc.2021.107618

[38] V. K. Reshma, R. S. Vinod Kumar, D. Shahi, M. B. Shyjith, Optimized support vector neural network and contourlet transform for image steganography, Evolutionary Intelligence, 2020, https://doi.org/10.1007/s12065-020-00387-8

[39] Cong Yu, Donghui Hu, Shuli Zheng, Wenjie Jiang, Meng Li, Zhong-qiu Zhao, An improved steganography without embedding based on attention GAN, Peer-to-Peer Networking and Applications, 2021, Vol 14, pp. 1446–1457, https://doi.org/10.1007/s12083-020-01033-x

[40] Zhangjie Fu, Fan Wang, Xu Cheng, The secure steganography for hiding images via GAN, EURASIP Journal on Image and Video Processing, 2020, Vol. 46, https://doi.org/10.1186/s13640-020-00534-2

[41] Qiang Liu, Xuyu Xiang, Jiaohua Qin, Yun Tan, Yao Qiu, Coverless image steganography based on DenseNet feature mapping, EURASIP Journal on Image and Video Processing, 2020, Vol. 39, https://doi.org/10.1186/s13640-020-00521-7

[42] Inas Jawad Kadhim, Prashan Premaratne, Peter James Vial, Improved image steganography based on super-pixel and coefficient-plane-selection, Signal Processing, 2020, Vol. 171, https://doi.org/10.1016/j.sigpro.2020.107481

[43] Sheng Li, Xinpeng Zhang, Toward Construction-Based Data Hiding: From Secrets to Fingerprint Images, IEEE Transactions on Image Processing, 2019, Vol. 28, Issue 3, pp. 1482 – 1497

[44] Nan Zhong, Zhenxing Qian, Zichi Wang, Xinpeng Zhang, Xiaolong Li, Batch Steganography via Generative Network, IEEE Transactions on Circuits and Systems for Video Technology, 2021, Vol. 31, Issue 1, pp. 88 – 97

[45] Weixuan Tang, Bin Li, Shunquan Tan, Mauro Barni, Jiwu Huang, CNN-Based Adversarial Embedding for Image Steganography, IEEE Transactions on Information Forensics and Security, 2019, Vol. 14, Issue 8, pp. 2074 – 2087