A Novel Data Hiding Approach Using Addition and Multiplication on Groups of Three Pixels

Ping Sheng Huang[1], Sheng-Kai Yang[2]

[1] Department of Electronic Engineering, Ming Chuan University, Taiwan
[2] Departement of Information Management, Chia Nan University of Pharmacy & Science, Taiwan
pshuang@mail.mcu.edu.tw, patrick@mail.cnu.edu.tw

Abstract

Due to the development of quantum computers in recent years, the topic of quantum image processing (QIP) in data hiding has begun to draw increasing attention. Also, since more and more digital image are transmitted via the internet, data hiding becomes feasible. The aim of data hiding algorithms is to raise the hiding capacity and preserve the original image quality. However, both goals are contradictive to each other. This paper presents an irreversible approach of data hiding in the spatial domain. Different from the traditional LSB methods, the proposed technique firstly divides the image into groups of three consecutive pixels and disassembles the bit sequence with secret message. Then, based on the combinations of addition and multiplication for the dissembled sets, the odd-even relationship among those three pixels are calculated and used to adjust their pixel values and embed the secret message. Experimental results have shown that the hiding capacity can reach to 41.67% and the image quality can be preserved.

Keywords: Quantum image processing, Internet, Data hiding, Spatial domain, Bit sequence

1 Introduction

The applications of Internet of Things (IoT) have been proposed in recent years by embedding different elements into software and electronic devices that connect to the Internet [1]. With the rapid development of computer technology, mobile communication network, and the Internet, IoT is emerging at a historic moment. Massive data such as text documents and digital images are generated daily and shared through IoT. Since those data can be easily copied and redistributed, data security via internet transmission becomes a crucial issue. Data hiding and cryptography are the two main approaches for secure communication. Using cryptography, the plain data is changed into an unreadable form before transmission. However, the third party is always conscious about the communication of incomprehensible data. For data hiding, the data is hidden into a cover media and then transmitted over the Internet. Hiding the existence of secret information is the main advantage of data hiding techniques over cryptography [2]. This paper presents an irreversible approach of data hiding in the spatial domain. By embedding the secret information into an image, the information can be protected and transmitted via the internet.

In the past two decades, due to the innovation of new electronic devices and application software, multimedia data can be easily captured, processed, and produced. Digital images are the most popular data used among the multimedia information Moreover, the development of internet and wireless network technology facilitates people's use of these ubiquitous communication channels, making it faster to transmit and exchange multimedia data. As a result, piracy and infringement of intellectual property rights occur frequently, and become an important topic. In this regard, government units and technological experts have also begun to adopt methods and preventive measures to prevent such incidents. Therefore, the information hiding method came into being in such context and information hiding has gradually become a vital technology in many fields recently.

Obvious but invisible marks may be embedded into the digital media such as digital images, audio files and video files. These marks can be signs or serial numbers of intellectual property rights, and even can prevent illegal direct reproduction. The information is expected to be transmitted without being discovered. According to the definition [3], information hiding can be divided into two branches: steganography and watermarking. The purpose of digital watermarking is to protect the ownership of works, and in this aspect, robustness has become the key research direction. However, the main application of steganography is to share and transmit secret information through secret channels, including images, audio, and video files. Therefore, the goal of steganography is to achieve higher hiding capacity with less distortion. On the other hand, cryptography is used to protect the information content from being

[*] Corresponding Author: Sheng-Kai Yang; E-mail: patrick@mail.cnu.edu.tw

DOI: 10.3966/160792642021052203003

cracked by encryption, while steganography is to hide the information [4]. The ultimate goals of steganography are undetectability, robustness and capacity of hidden data, of which robustness is to be able to resist different image preprocessing and compression. These goals are also used to distinguish the classification of steganography. Apart from some researchers sorting out and analyzing the related topics on information hiding [4] and image steganography [3], some experts have also made in-depth research on the basic topics [5] and practical applications [6] of information hiding in images and videos.

For the current hiding technologies, they can be roughly divided into two types: frequency domain and spatial domain. The method in frequency domain is to hide secret information into frequency coefficients, such as Discrete Cosine Transform [7] or Discrete Wavelet Transform [8]. The approach of spatial domain is mainly to change the secret data and cover media in the spatial domain, such as encoding [9], changing the Least Signifier Bit (LSB) [10] or Pixel Value Difference (PVD) [11]. Considering the tolerance of attack, the data hidden by the spatial domain method can be easily destroyed, so the frequency domain technology is derived. However, considering the hiding capacity, the data hiding capacity of the frequency domain technology is usually less than that of spatial domain method.

With the rapid advances in 5G technologies and mobile devices, many users take photographs and transfer files by mobile devices. To prove the photograph copyright, devices usually embed user information by data hiding. This paper presents a novel data hiding approach using addition and multiplication on groups of three consecutive pixels. Using the relationship from those two operations, the secret information in a bit sequence is disassembled and embedded into those groups of the image. From the experimental results, the suggested approach can reach a hiding capacity up to 41.67% that helps users to embed more data in digital images while maintaining the image quality with a Peak Signal-to-Noise Ratio (PSNR) at around 35.75 dB.

2 Related Works

At present, many papers have published hiding methods in spatial domain and the Least Signifier Bit (LSB) method [10] is the most popular one used for data hiding in images. The main concept is to change the data in the secret bit sequence with the LSBs of each pixel value respectively to achieve the purpose of hiding. This method is fast, but the image quality will be seriously degraded when more than two bits are changed in each pixel. With the improvement to the LSB method, many algorithms have also appeared [12-16]. Most of the algorithms combine the LSB method with other methods to increase the hiding capacity or reduce the distortion of the cover image. Wang et al. [12] adopted the training and learning ability of genetic algorithm to decide the best LSB positions for data hiding. Yang et al. [13] proposed an adaptive method, in which pixels located in the edge area are embedded with more LSBs and fewer bits are hidden in those pixels located in the smooth area. Taking the same idea, Chen [14] further adopted a method of non-2 power modulus to modify pixel values. In addition, Wu et al. [15] put forward an algorithm that combines the PVD method and the LSB substitution method. When the pixel difference is less than a predefined threshold value, the LSB method is directly used to modify the pixel value to hide the data. Otherwise, the PVD method is used to. By using this approach, although the capacity of hidden data is increased, the quality of stego-images is also degraded. To increase the capacity of hidden data and maintain the image quality at the same time, we had proposed an approach by integrating the PVD method and LSB replacement technique [16]. Furthermore, to improve the hiding capacity, a data hiding technique using summation and LSD Parity is proposed [17].

The data hiding method of PVD [11] is mainly based on changing the pixel difference between two adjacent and non-overlapping pixels. The method first determines the number of bits of data that can be hidden by using the quantization table corresponding to the difference value of two adjacent pixel values, then further adjusts that difference value according to the hidden data value, and further changes those two pixel-values to achieve the purpose of data hiding. Wang et al. [18], apart from using the PVD method of Wu et al. [11], further adopted modulus function to increase the hiding capacity in the remainder of pixel values. Li et al. [19] first divided the secret image and the cover image into blocks, and then hid the secret image into the cover image by using the difference and similarity of two blocks from individual images. Wang et al. [20] adopted the best block matching and K-means clustering to find the block number of each secret image block in the cover image and hid the block number into the LSB of the cover image. Small amount of distortion is allowed for the extracted secret images. However, as the number of LSB bits is gradually increased, the image quality will be greatly affected at the same time. Lou et al. [21] proposed an adaptive data hiding method using human visual sensitivity and image region complexity to hide data. To protect from pixel difference histogram (PDH) analysis and RS analysis, Pradhan et al. [22] designed two hybrid techniques of data hiding by combining the approaches of LSB substitution, PVD, and exploiting modification directions (EMD). Also, an adaptive version is proposed [23] by using the methods of LSB and PVD.

Although the previous methods of hiding secret can have a large capacity, however, after the secret data is extracted, the original cover image is also irrecoverable.

Therefore, in recent years, many scholars have developed reversible data hiding methods [24-28]. These methods are mainly used to protect the cover image from being damaged and restore such cover image to the original state after the hidden data is extracted. Among them, the most typical method is the histogram modification proposed by Ni et al. [24], which shifts the group with peak pixel values and then hides the bit sequence into the empty positions. By modifying the histogram, Tai et al. [25] and Lin et al. [26] have successively published two reversible data hiding methods by using the methods of pixel difference and difference image. Luo et al. [27] first calculated the difference histogram and the block mean value of the image blocks, and then used the multi-level histogram shifting mechanism to hide the secrets with reference to the block mean value. Based on the method of difference expansion, Wang et al. [28] adopted the 2D vector map, which is suitable for reversible watermark, to restore image quality after the watermark is extracted. Since the mechanism of how to restore the original image needs to be considered, the hiding capacity of the general reversible data hiding methods is inferior to that of the spatial domain hiding method.

Due to the development of quantum computers in recent years, the excellent performance of quantum computation such as entanglement and parallelism are adopted and applied for quantum image processing (QIP) [29-31]. Especially, QIP-based security topics including quantum watermarking, quantum image encryption, and quantum image steganography are investigated [29]. By using quantum computation, the aim of QIP applications is to extend the approaches of conventional image processing to the quantum computing framework and improve the computation performance [30]. Therefore, the traditional algorithms of image processing which is normally computation-intensive can be modified into a quantum computing framework for appropriate applications in the future. Quantum watermarking is the process that embeds the invisible quantum information into the quantum multimedia signal for copyright safety. After dividing the image into non-overlapping 2×2 blocks and using the approach of tri-way pixel value differencing, a novel quantum watermarking method [31] based on the modified least significant bit (LSB) substitution is proposed and applied at the quantum circuit.

In the spatial domain, although the LSB replacement technique [10] and the PVD method [11] are easily implemented and excellent performance can be accomplished, the hiding capacity and the robustness of protecting from the attack of PDH analysis and RS analysis still need to be improved. Motivated by maximizing the hiding capacity and improving our previous approach [17], this paper develops a data hiding method in the spatial domain using simple arithmetic operations. Instead of hiding the data into the LSBs of pixel values, the innovation factor and the contribution of this paper focus at embedding the data into the LSDs of each group with three pixels. To sacrifice a small amount of image quality, the hiding capacity can be greatly increased.

3 The Proposed Method

The ideal case of information hiding algorithms is to increase the hiding capacity and maintain the original image quality as much as possible. However, these two goals usually contradict each other. The algorithm we proposed in this paper mainly divides the image in units of three consecutive pixels and disassembles the bits of the secret sequence into a possible addition and multiplication function composed of three values. Meanwhile, after combining the single-digit and ten-digit odd-even characteristics of the three pixels, the best combination of new pixel values for secret hiding is found, and finally secret bit sequence data is hidden into each pixel group. Based on to the proposed method, all three-pixel groups of the whole image can be processed and changed sequentially to achieve the purpose of hiding secrets. Since the maximum hiding capacity of each group of pixels (3x8=24 bits) is 10 bits, the hiding capacity ratio can reach 10/24=41.67%. The method of addition and multiplication function proposed in this paper adopts the concept of pixel value grouping from the PVD method [11]. However, instead of using the difference value to hide the data, the numerical value of the hidden bit sequence is disassembled into possible combination of addition and multiplication functions, and then the single-digit and ten-digit odd-even characteristics of the three pixels were matched to select the best new combination of hidden pixel values. Finally, the data was hidden by adjusting the pixel values. At first, we introduced the LSB method [10], PVD method [11] and Wu's method [15] to facilitate understanding the implementation steps of the method proposed in this paper.

3.1 LSB Method

The LSB data hiding method [10] is the simplest and most common data hiding technology in the spatial domain. It mainly uses the principle that human vision is not easy to feel the slight changes of pixels, so the purpose of hiding data is achieved by changing the smallest bits of pixels. We took LSB data hiding with 1 bit as an example. When each pixel value of the image is converted to binary, the smallest bit is either 0 or 1. At this time, that value that can be altered and hidden in each pixel value is either 0 or 1. Therefore, theoretically, the probability of randomly transforming each minimum bit is 0.5. Replacing 0 or 1 for the minimum bit does not actually affect the whole image quality, but if the hidden data of each pixel is changed to more than 2 bits, the image quality will be greatly affected, so the LSB method is not suitable for data hiding with more than 2 bits. Without using the binary pixel values, this paper proposed to use the odd-even characteristics of single bits and ten bits with continuous three pixel-values to further increase the hiding capacity of four bits so as to improve the overall hiding capacity.

3.2 PVD Method

Wu et al. [11] proposed a PVD data hiding method. It is mainly based on changing the pixel difference between two adjacent and non-overlapping pixels. The method first determines the number of bits of data that can be hidden by using the quantization table corresponding to the difference value of two pixel-values, then further adjusts the difference value of the two pixels to achieve the purpose of data hiding. Assume that there is a gray-scale image with a height and a width of M pixels and N pixels respectively, the image contains a total of $M \times N$ pixels. Furthermore, the positions of two adjacent pixels are represented by (i, j) and $(i, j+1)$ respectively, the corresponding pixel values are $f(i, j)$ and $f(i, j+1)$. If we define the difference between two pixels is $d = \left| f(i, j) - f(i, j+1) \right|$, then the d value should be between 0 and 255. Table 1 shows an example of the PVD difference range table by dividing d into 6 areas. The value k indicates the number of bits that can be hidden. After k bits are taken out from the hidden data, they are converted into a decimal value. Then, the minimum value g of the area will be added and the pixel values of $f(i, j)$ and $f(i, j+1)$ are adjusted to achieve the purpose of information hiding. Based on this scheme, this paper proposed a new approach by using the relationship between least significant digits (LSDs) and ten digits of three pixels. Also, by combining with possible addition and multiplication functions disassembled by the numerical value of hidden bit sequences, all sets of new pixel values for data hiding can be found. Finally, the least squares method was used to select the best combination of pixel values and hide the data by adjusting the pixel values.

Table 1. PVD difference range table

Area	Difference range	Hidden digits (k)	Area minimum (g)
1	0 to 7	3	0
2	8 to 15	3	8
3	16 to 31	4	16
4	32 to 63	5	32
5	64 to 127	6	64
6	128 to 255	7	128

3.3 Wu's Method

Wu et al. [15] proposed the pixel difference and the LSB substitution method, which mainly calculates the pixel difference between two adjacent and non-overlapping pixels first, and then determines the hiding method according to the difference. If the difference value is less than the predefined boundary value, the LSB substitution method is used to change the pixel value; otherwise, the PVD method is used for data hiding. For example, assume that the predetermined boundary value is 7. When the pixel difference is greater than the boundary value, the information is hidden according to PVD method [11]; if it is less than that boundary value, the 4-bit value is taken out from the hidden data bit sequence (substitution bit $2 \times 2 = 4$), and the lowest 2 bits of two adjacent and non-overlapping pixel values are respectively placed. The new pixel values obtained at this time are $f'(i, j)$ and $f'(i, j+1)$. Finally, it is judged whether the absolute value of the difference between the two pixel-values is less than or equal to 7, and if it is less than or equal to 7, it ends and returns to the calculation of the next set of two adjacent and non-overlapping pixel values. If it is greater than 7, the pixel values are adjusted respectively, and the larger pixel value minus 4 and the other pixel value adds 4 to obtain the final pixel value $f''(i, j)$ and $f''(i, j+1)$. Then it ends and returns to the calculation of the next set of two adjacent and non-overlapping pixel values. From the experimental data in Wu's paper [15], as the amount of hiding information increases, the image quality will be greatly reduced. In this paper, we mainly considered how to further increase the hiding capacity of four bits by using the odd and even number characteristics of each group of three pixel-values while using the combination method of addition and multiplication functions to hide data, so as to improve the overall hiding capacity.

To improve our previous paper [17], this paper groups the image into three consecutive pixels and disassembles the bit sequence to be hidden in this pixel group. Then, by using all the possible addition and multiplication function compositions after disassembly and matching the odd-even relationship between the LSD of the first pixel value and the decimal number of the three pixel-values, a new set of pixel groups that can be hidden is found. Then, the values of three pixels in each group in the new set are changed by ± 10 and ± 20 respectively to expand the number of sets. Finally, the least squares method is used to select the best combination of pixel values, and this combination is used to adjust and change the three old pixel values. According to this method, all three-pixel groups of the whole image can be processed and changed in sequence to achieve the purpose of hiding secrets. The maximum hiding capacity in each group of three pixels (3×8=24 bits) is 10 bits, and the hiding capacity ratio can reach 10/24=41.67%.

The method [17] proposed previously employed the odd and even characteristics of the LSDs and the sum of two adjacent pixel values Then the values of those two pixels are changed to hide the secret sequence with five bits. Different from this algorithm, the advanced method proposed in this paper adopted the addition and multiplication operations from the values of three pixels to achieve the purpose of hiding. The main step is to group the image into three consecutive pixels in units of three adjacent and non-overlapping pixels, and then adjust the single-digit and ten-digit values of each grouped pixel value according to the odd and even numbers of bits to be hidden, so as to achieve the purpose of hiding secret bit sequence. The length of bit sequence to be hidden each time can be selected as five to ten bits according to different requirements. As the three integers (***p***, ***q***, and ***r***) making up the least significant digits of three pixels are between 0 and 9, $0 \le \boldsymbol{A} = \boldsymbol{p} \times \boldsymbol{q} + \boldsymbol{r} \le 90$. Since the quadratic number between this range is between 0 and 6, the maximum number of bits that can be hidden is 6 (2^6=64). Furthermore, based on our test results, the odd and even numbers of ***p*** that meet the requirement of $0 \le \boldsymbol{A} \le 63$ all exist. Therefore, the longest sequence that the advanced method can hide into the three-pixel group is 10 bits (6+4=10).

Assume that the number of bits to hidden in each group is ***N***, ***N*** = 5,…,10 and the bit sequence is represented as $(b_1 b_2 ... b_{N-1} b_N)_2$. The first ***N***-4 bits of the secret sequence are used to determine the initial set of values for the new pixel group and the last four values, $(b_{N-3} b_{N-2} b_{N-1} b_N)_2$, are used to filter single-bit and ten-bit values of new pixel values of each group. At first, the secret bit sequence to be hidden is randomly permuted by the prepared key. Then, the cover image is grouped in units of non-overlapping and consecutive three adjacent pixels. In the decryption part, after all the data are extracted, the secret bit sequence can be fully recovered by reverse permutation using the prepared key.

3.4 Data Hiding Algorithm

Step 1: All pixels of the gray-scale cover image are grouped in units of non-overlapping three consecutive adjacent pixels and the last group is discarded when the number is less than three pixels. Meanwhile, the number of bits ***N***, ***N***=5,…,10, to be hidden in each group is determined.

Step 2: When all the hidden pixel groups have been used up and all the data bits to be hidden are embedded, the algorithm ends; otherwise, a pixel group is sequentially taken out from the cover image for processing, assume that the three pixel values of this group are respectively $f(i,j)$, $f(i,j+1)$, and $f(i,j+2)$. Meanwhile, an ***N***-bit sequence is taken out from the binary secret sequence sequentially to be processed, assume that the sequence is $(b_1 b_2 ... b_{N-1} b_N)_2$.

Step 3: The first ***N***-4 bits of the secret sequence are converted to the decimal number ***A***, which can be expressed by the following formula:

$$(\boldsymbol{A})_{10} = (\boldsymbol{b}_1 \boldsymbol{b}_2 ... \boldsymbol{b}_{N-4})_2$$

Step 4: Assume that there are three integer values ***p***, ***q***, and ***r***, in which $0 \le \boldsymbol{p},\boldsymbol{q},\boldsymbol{r} \le 9$. The sets of $(\boldsymbol{p},\boldsymbol{q},\boldsymbol{r})$ that meets both the following conditions are found:

$$\boldsymbol{A} = \boldsymbol{p} \times \boldsymbol{q} + \boldsymbol{r} \text{ and } \boldsymbol{b}_{N-3} = \boldsymbol{p} \bmod 2 \quad \textbf{(1)}$$

Here, (***p*** mod 2) is calculated as the remainder of ***p***divided by 2. The set formed by these conditions can be calculated as follows:

$$\boldsymbol{S}_A = \left\{ (\boldsymbol{p},\boldsymbol{q},\boldsymbol{r}) \middle| \begin{array}{c} \boldsymbol{A} = \boldsymbol{p} \times \boldsymbol{q} + \boldsymbol{r} \text{ and} \\ \boldsymbol{b}_{N-3} = \boldsymbol{p} \bmod 2 \\ 0 \le \boldsymbol{p},\boldsymbol{q},\boldsymbol{r} \le 9 \end{array} \right. \quad \textbf{(2)}$$

Step 5: Taking one element $(\boldsymbol{p},\boldsymbol{q},\boldsymbol{r})$ out of $\boldsymbol{S}_A$ in sequence, the three LSDs in each pixel group are respectively modified in the following three forms:

$$f'(i,j) = (f(i,j) \div 10) \times 10 + \boldsymbol{p} \quad \textbf{(3)}$$

$$f'(i,j+1) = (f(i,j+1) \div 10) \times 10 + \boldsymbol{q} \quad \textbf{(4)}$$

$$f'(i,j+2) = (f(i,j+2) \div 10) \times 10 + \boldsymbol{r} \quad \textbf{(5)}$$

Here, $(f(i,j) \div 10)$ is the quotient of $f(i,j)$ divided by 10. Assume that all three pixel values of these modified LSDs form a set $\boldsymbol{S}_{LSD}$ expressed by

$$\boldsymbol{S}_{LSD} = \left\{ (\boldsymbol{P},\boldsymbol{Q},\boldsymbol{R}) \middle| \begin{array}{c} \boldsymbol{P} = f'(i,j), \\ \boldsymbol{Q} = f'(i,j+1) \text{ and} \\ \boldsymbol{R} = f'(i,j+2), \\ (\boldsymbol{p},\boldsymbol{q},\boldsymbol{r}) \in \boldsymbol{S}_A \end{array} \right. \quad \textbf{(6)}$$

Step 6: Taking one element $(\boldsymbol{P},\boldsymbol{Q},\boldsymbol{R})$ out of $\boldsymbol{S}_{LSD}$ in sequence and those three values are changed in the way of ± 10 and ± 20. After adding the original value, each element $(\boldsymbol{P},\boldsymbol{Q},\boldsymbol{R})$ will be extended to 125 groups $(5 \times 5 \times 5 = 125)$. As the range of gray scale pixel value must be between 0 and 255, the group in which any one of the three values exceeds the range is removed. Assume that all the remaining elements form the following set

$$\boldsymbol{S}_{AS} = \{(\boldsymbol{P}',\boldsymbol{Q}',\boldsymbol{R}')\} \quad \textbf{(7)}$$

Step 7: Taking one element $(\boldsymbol{P}',\boldsymbol{Q}',\boldsymbol{R}')$ out of $\boldsymbol{S}_{AS}$ in sequence and checking out the following three conditions, only elements that meet all three conditions at the same time can be kept; otherwise, they will be eliminated from the set $\boldsymbol{S}_{AS}$.

(a) $b_{N\text{-}2} = (\boldsymbol{P}' \div 10) \bmod 2$

(b) $b_{N\text{-}1} = (\boldsymbol{Q}' \div 10) \bmod 2$

(c) $b_N = (\boldsymbol{R}' \div 10) \bmod 2$

This is mainly to confine the last three bits of the secret sequence to fulfil the odd-even property of three pixel-values. Assume that the set formed by all the remaining elements after elimination is

$$\boldsymbol{S}_{ASn} = \{(\boldsymbol{P}_n, \boldsymbol{Q}_n, \boldsymbol{R}_n)\}\,. \tag{8}$$

Step 8: Taking one element $(\boldsymbol{P}_n, \boldsymbol{Q}_n, \boldsymbol{R}_n)$ out of $\boldsymbol{S}_{ASn}$ in sequence and the least squares method is used to calculate the difference value between $(\boldsymbol{P}_n, \boldsymbol{Q}_n, \boldsymbol{R}_n)$ and the original three-pixel values $f(i,j)$, $f(i,j+1)$, and $f(i,j+2)$. Finally, among all the difference values, the one that results in the smallest value is selected as $(\boldsymbol{P}_{nb}, \boldsymbol{Q}_{nb}, \boldsymbol{R}_{nb})$, which is the best combination of pixel values that can be used to replace the original three-pixel values. Assume that the final modified and replaced three-pixel values are expressed as

$$f''(i,j) = \boldsymbol{P}_{nb} \tag{9}$$

$$f''(i,j+1) = \boldsymbol{Q}_{nb} \tag{10}$$

$$f''(i,j+2) = \boldsymbol{R}_{nb} \tag{11}$$

Step 9: Back to Step 2.

Next, the method of data extraction is explained. The data extraction is basically carried out in the opposite direction corresponding to the data hiding.

3.5 Data Extraction Algorithm

Step 1: All pixels of the stego-image are grouped in units of three consecutive non-overlapping adjacent pixels, and the number of bits hidden in each grouping is known to be ***N***, in which ***N***=5,…,10.

Step 2: If all the hidden pixel groups have been used up and all the hidden data bits have been extracted, the algorithm is finished. Otherwise, a pixel group is taken out of the stego-image in sequence for processing. Assume that the values of three pixels in this group are $f(i,j)$, $f(i,j+1)$ and $f(i,j+2)$.

Step 3: The LSD values of three pixels (***p, q***, and ***r***) can be calculated as follows:

$$\boldsymbol{p} = f\,(i,j) \bmod 10 \tag{12}$$

$$\boldsymbol{q} = f\,(i,j{+}1) \bmod 10 \tag{13}$$

$$\boldsymbol{r} = f\,(i,j{+}2) \bmod 10 \tag{14}$$

Step 4: The decimal value ***A*** of the binary sequence hidden in this pixel group can be calculated by the following formula:

$$\boldsymbol{A} = \boldsymbol{p} \times \boldsymbol{q} + \boldsymbol{r} \tag{15}$$

Next, the decimal value ***A*** is converted into the binary value $(b_1 \ldots b_{N-4})_2 = (\boldsymbol{A})_{10}$ with N–4 bits. This binary sequence is the first ***N***–4 bits among the ***N*** hiding bits.

Step 5: The determination of $(b_{N-3} b_{N-2} b_{N-1} b_N)_2$:

First, the tens digits of three pixel-values $((f(i,j)$, $f(i,j+1)$, and $f(i,j+2))$, ***s***, ***t*** and ***u*** can be calculated by the following three equations respectively.

$$\boldsymbol{s} = (\,f\,(i,j) \div 10) \bmod 10 \tag{16}$$

$$\boldsymbol{t} = (\,f\,(i,j{+}1) \div 10) \bmod 10 \tag{17}$$

$$\boldsymbol{u} = (\,f\,(i,j{+}2) \div 10) \bmod 10 \tag{18}$$

Next, the value of $(b_{N-3} b_{N-2} b_{N-1} b_N)_2$ is determined according to the following four conditions:

(a) If the LSD ***p*** of $f(i,j)$ is an odd number, that is $(\boldsymbol{p} \bmod 2) = 1$, then b_{N-3}=1; otherwise b_{N-3}=0.

(b) If the tens digit ***s*** of $f(i,j)$ is an odd number, that is $(\boldsymbol{s} \bmod 2) = 1$, then b_{N-3}=1; otherwise b_{N-2}=0.

(c) If the tens digit ***t*** of $f(i,j+1)$ is an odd number, that is $(\boldsymbol{t} \bmod 2) = 1$, then b_{N-1}=1; otherwise b_{N-1}=0.

(d) If the tens digit ***u*** of $f(i,j+2)$ is an odd number, that is $(\boldsymbol{u} \bmod 2) = 1$, then b_N=1; otherwise b_N=0.

Step 6: After the binary sequence obtained in **Step 4** and **Step 5** is combined to form $(b_1 b_2 \ldots b_{N-1} b_N)_2$, the data extraction of a pixel group is completed.

Back to **Step 2**.

3.6 Example Demonstration

To quickly understand the algorithm operations proposed in this paper, two examples are adopted for demonstration.

[Example 1] Hide five bits of $(01000)_2$ into the group of three pixels (89, 58, 62).

(a) Data hiding

Since the first bit of this sequence is $(0)_2$, this is converted into a decimal value of $(0)_{10}$, represented by ***A***. Furthermore, the second bit is $(1)_2$, which is regarded as an odd number. Assume that ***p***, ***q***, and ***r*** are all integers between 0 and 9. At first, the combinations that meet $\boldsymbol{A} = \boldsymbol{p} \times \boldsymbol{q} + \boldsymbol{r}$ need to be found.

As $\boldsymbol{A} = 0$, the combination of the three values should meet the condition of $\boldsymbol{p} \times \boldsymbol{q} + \boldsymbol{r} = 0$ The $(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r})$ combinations that meets this condition include the following 19 possible set selections including $(0,0,0), (1,0,0), (0,1,0), (2,0,0), \ldots,$ and $(0,2,0)$. As the second bit $(1)_2$ to be hidden is regarded as odd, the LSD of the first pixel value needs to be odd. Therefore, the 14 sets including $(0,0,0), (2,0,0), (4,0,0),$ $(6,0,0), \ldots,$ and $(8,0,0)$ need to be eliminated, only five combinations including $(1,0,0), (3,0,0), (5,0,0),$

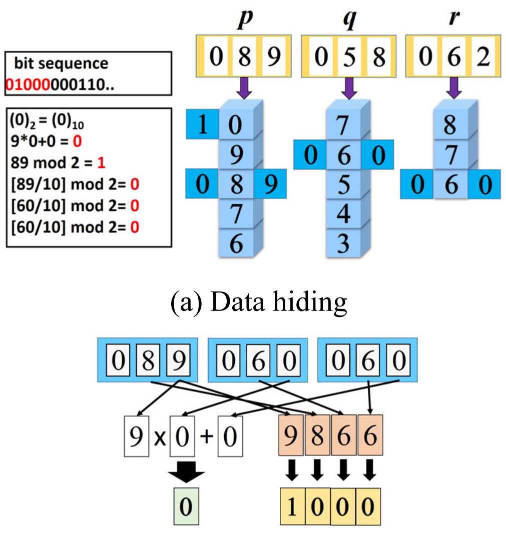
(7,0,0), and (9,0,0) are left. Then, we substitute the LSDs of the original pixel group (89,58,62) with those five combinations and the results include (81,50,60), (83,50,60), (85,50,60), (87,50,60), and (89,50,60).

Then, the values of three pixels in each of the five groups are added and subtracted by 20, 10 and 0, respectively. In this way, nine combinations can be derived from each group, and those pixel values exceeding the range of 0 and 255 are discarded.

Next, we used the last three bits of the hidden secret sequence to further filter the corresponding values of three pixels in each group, leaving only the combination of ten-digit values conforming to the odd-even characteristics of hidden bits. The corresponding ten-digit values with bit $(0)_2$ must be even and the corresponding ten-digit values of bit $(1)_2$ must be odd. Based on such condition, the third bit $(0)_2$ is regarded as an even number, the fourth bit $(0)_2$ is regarded as an even number, and the fifth bit $(0)_2$ is also regarded as an even number. After applying this condition, the remaining combinations are (81, 60, 60), (61, 60, 60), (101, 60, 60), (83, 60, 50), (85, 60, 60), (87, 60, 60),…, and (89, 60, 60). Finally, the least squares method was used to calculate the difference between each of the above combinations and the original pixel groups (89, 58, 62), and the combination with the minimum difference is selected as the best secret hiding group. In this example, the minimum difference is 8 and the corresponding optimal substitution group is (89, 60, 60), thus completing a five-bit secret hiding. Figure 1(a) shows the schematic diagram of data hiding.

(b) Data extraction

Referring to Figure 1(b) and data extraction algorithm, the five-bit information $(01000)_2$ can be easily and quickly recovered to the original hidden information in the three-pixel group (89, 58, 62).



(a) Data hiding

(b) Data extraction

Figure 1. Data hiding and extraction of five bits in (89, 58, 62)

[Example 2] Hide ten bits of information $(0101101010)_2$ into the group of three pixels (224, 88, 9).

(a) Data hiding

Since the first six-bit values of this sequence is $(010110)_2$, this is converted into a decimal value of $(22)_{10}$, represented by $\boldsymbol{A}$. Furthermore, the seventh bit is $(1)_2$, which is regarded as an odd number. Assume that $\boldsymbol{p}$, $\boldsymbol{q}$ and $\boldsymbol{r}$ are all integers between 0 and 9.

As $\boldsymbol{A} = 22$, the combination of the three values should meet the condition of $\boldsymbol{p} \times \boldsymbol{q} + \boldsymbol{r} = 22$. The $(\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r})$ combinations that meet this condition include the following 15 possible set selections including (2,7,8), (2,8,6), (2,9,4), (3,5,7), …, and (3,6,4). As the seventh bit $(1)_2$ to be hidden is regarded as odd, the LSD of the first pixel value needs to be odd. Therefore, the 7 sets including (2,7,8), (2,8,6), (2,9,4), (4,4,6), (4,5,2), (6,3,4), and (8,2,6) are to be eliminated, only 8 combinations including (3,5,7), (3,6,4), (3,7,1), (5,3,7), (5,4,2), (7,2,8), (7,3,1), and (9,2,4) are left. Then, we substitute the LSDs of the original pixel group (224,88,9) in accordance with the conditions, and the results are the combinations including (223,85,7), (223,86,4), (223,87,1), …, and (225,83,7). Then, the values of three pixels in each group are added and subtracted by 20, 10, and 0, respectively. In this way, nine combinations can be derived from each group, and those pixel values exceeding the range of 0 and 255 are discarded.

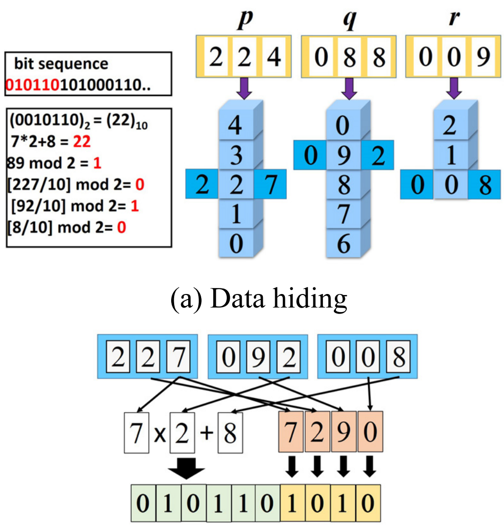
Next, we used the last three bits of the hidden secret sequence to further filter the corresponding values of three pixels in each group, leaving only the combination of ten- digit values conforming to the odd-even characteristics of hidden bits. As the eighth bit $(0)_2$ is regarded as even, the ninth bit $(1)_2$ is regarded as odd, and the tenth bit $(0)_2$ is regarded as even. After applying this condition, the remaining combinations are (203,75,7), (213,95,7), (213,75,7), …, and (213,95,7). Finally, the least squares method was used to calculate the difference between the each of above combinations and the original pixel groups (224,88,9), and the combination with the minimum difference is selected as the best secret hiding group. In this example, the minimum difference is 26, and the corresponding optimal substitution group is (227,92,8), thus completing a ten-bit secret hiding. Figure 2(a) shows the schematic diagram of data hiding.

(b) Data extraction

Referring to Figure 2(b) and data extraction algorithm, the ten-bit information $(0101101010)_2$ can be easily and quickly recovered to the original hidden information in the three-pixel group (224,88,9).

4 Experimental Results

In this paper, we have carried out three experiments on this algorithm, using five test gray-scale images, i.e.



(a) Data hiding

(b) Data extraction

Figure 2. Data hiding and extraction of ten bits in (224, 88, 9)

Lena, Baboon, Peppers, F-16, and Sailboat, with the size of 512×512, as shown in Figure 3. The goal of our proposed method is to increase the data hiding capacity and maintain the quality of the image as much as possible. The hidden data used are binary bit sequences generated by random permutation. The number of bits is subject to the size of the test image. The experimental results generated are obtained from the average values taken after 1000 times of execution.

The first experiment is to compare the proposed method with our previous algorithm [17] using the Lena image. As shown in Figure 4, using the test image of Lena, the proposed method can hide 34.37% more capacity (109226-81290)/81290) than the previous method [17]. Also, the performance comparison between this paper and the other two approaches are also listed. To sacrifice a small amount of image quality, this method can achieve a much more capacity than the other two methods. By using five test images, Table 2 lists the comparison results of this paper to the methods of PVD method [11] and Wu's method [15]. Also, this paper has achieved highest hiding capacity among three method.

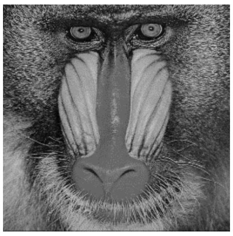
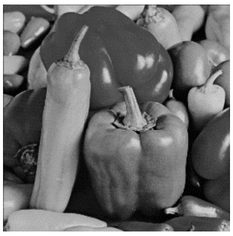
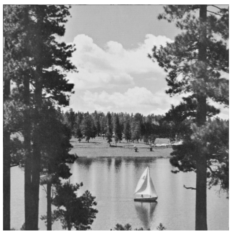
				
Lena	Baboon	Peppers	F-16	Sailboat

Figure 3. Gray scale 512×512 test images

 Lena (gray scale) original image	 PVD method [11]	 Wu's method (2 bits) [15]	 Previous method [17]	 The proposed method
PSNR value (dB)	41.09	40.94	41.55	35.08
Hiding capacity (bytes)	51219	63654	81290	109226

Figure 4. Comparison of hiding capacity and quality for Lena image

Table 2. The performance comparison of three methods using 5 test images

Cover image	Da-Chun Wu's PVD method		H.-C. Wu method (2 LSB bits)		The proposed method	
	Capacity (bytes)	PSNR (dB)	Capacity (bytes)	PSNR (dB)	Capacity (bytes)	PSNR (dB)
Lena	50960	41.79	66064	38.80	109226	35.08
Baboon	56291	37.90	68007	33.33	109226	36.07
Peppers	50685	41.73	66032	37.50	109226	35.02
F-16	51243	40.97	66256	37.63	109226	35.07
Sailboat	52779	39.32	66622	35.01	109226	35.08

The second experiment is to verify the maximum hiding capacity and the image quality of the five test images using the proposed method. ten bits are hidden into each three-pixel group. The parameters of evaluating the image quality are Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). As shown in Table 3, for each of the test images with the size of 512×512, the maximum capacity achieved is 109226 bytes. Although PSNR values are only about 35dB, SSIM values and visual inspection show that the image quality can still maintain a high similarity with the original image. Especially, the Baboon image have demonstrated a high SSIM value up to around 95%. By comparing the image content with that of the other four test images, the Baboon image provides more complicated objects with high-frequency information. and this is worth further investigated in the future.

The third experiment is to verify the maximum hiding capacity and image quality that can be achieved when the number of bits that can be hidden in each group of pixels is five to ten bits. Table 4 shows the experimental results when the Lena image is used, of which the image quality of hiding eight bits is the best. The results show that the proposed method may be especially suitable for the case of using an image as the secret data and this is worth further expanding the test image database and experimental analysis in the future. In addition, Table 4 also shows that the selected hidden secret bits of 5 to 10 are similarly affected at the image quality. This situation is unchanged due to the increase or decrease of hidden secret bits. From the experimental results, the information hiding method proposed in this paper has achieved better efficiency than the previous method [17] in terms of hiding capacity and the number selectivity of hidden bits. Also, the image quality of the stego-image is acceptable.

Table 3. Comparison of PSNR and SSIM with 10 hidden bits

Cover image	The proposed method		
	Capacity(bytes)	PSNR(dB)	SSIM
Lena	109226	35.08	0.869348
Baboon	109226	35.07	0.951968
Peppers	109226	35.02	0.869912
F-16	109226	35.07	0.868339
Sailboat	109226	35.08	0.899980

Table 4. Capacity and image quality of Lena image hidden in 5 to 10 bits

Hidden bits (*N*)	5	6	7	8	9	10
Capacity (bytes)	54613	65535	76458	87381	98303	109226
PSNR	34.78480	34.92637	35.40508	35.75601	35.71939	35.06523
SSIM	0.861956	0.865574	0.877070	0.885236	0.884617	0.869032

To further verify the performance of our approach to images that are constructed by complex objects in the real world, we have randomly selected 12 test images from the internet listed in Figure 5. The experimental data is listed in Table 5. As described in the algorithm, the efficacy of embedding the data into the real images is the same as that in the benchmarks. The hiding capacity is staying stable at 41.67% for each cover image by hiding 10 bits into each group with three pixels. The minimum PSNR is 33.9494 and the maximum PSNR is 35.1203. This proves that the proposed approach is also stable for those images that are constructed by complex objects in the real world.

5 Conclusions

There are two important factors to be considered for developing data hiding algorithms. They are data hiding capacity and the integrity and quality of stego-images. These two are usually contradictory and cannot be achieved at the same time. This paper presents a novel data hiding approach in the spatial domain. The operations of addition and multiplication are applied on groups of three consecutive pixels. Using the relationship from those two operations, the secret bit sequence is divided and embedded into those groups of the image. From the experimental results, the proposed approach can reach a hiding capacity up to 41.67% that helps users to embed more data into digital images while maintaining the image quality with a Peak Signal-to-Noise Ratio (PSNR) at around 35.75 dB.

The method proposed in this paper can not only hide more data, but preserve the image quality, which has been verified by the experimental results in this paper. Based on the experimental results, the future research will be to develop an adaptive algorithm to select the best number of hidden bits for each three-pixel group. Also, how to hide the small gray-scale image into the cover image is worth investigated.

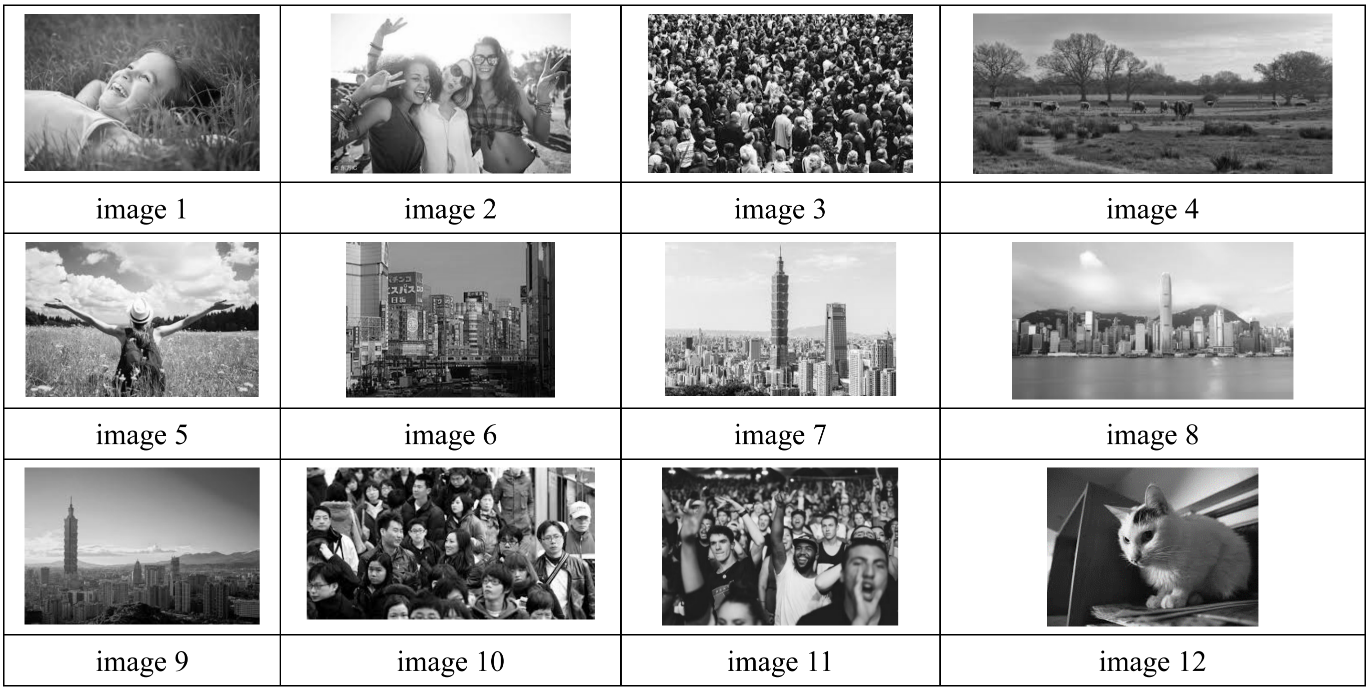


Figure 5. Twelve test images with complex objects in the real world

Table 5. Experimental results for twelve test images in Figure 5

image name	image size	PSNR value	Hiding capacity (bytes)	Hiding capacity (ratio)
image 1	275× 183	35.0795	20968	41.67%
image 2	275× 183	33.9494	20968	41.67%
image 3	290× 174	34.6627	21025	41.67%
image 4	337× 150	35.0703	21062.5	41.67%
image 5	275× 183	34.9955	20968	41.67%
image 6	259× 194	34.9169	20936	41.67%
image 7	275× 183	34.9861	20968	41.67%
image 8	300× 168	34.9742	21000	41.67%
image 9	275× 183	34.8518	20968	41.67%
image 10	310× 163	34.3550	21055	41.67%
image 11	640× 427	35.1203	113867	41.67%
image 12	259× 194	35.0402	20936	41.67%

References

[1] U. Khadam, M. M. Iqbal, M. Alruily, M. A. Al Ghamdi, M. Ramzan, S. H. Almotiri, Text Data Security and Privacy in the Internet of Things: Threats, Challenges, and Future Directions, *Wireless Communications and Mobile Computing*, Vol. 2020, Article ID 7105625, February, 2020. https://doi.org/10.1155/2020/7105625.

[2] A. Shaik, V. Thanikaiselvan, R. Amitharajan, Data Security Through Data Hiding in Images: A Review, *Journal of Artificial Intelligence*, Vol. 10, No. 1, pp. 1-21, 2017.

[3] A. Cheddad, J. Condell, K. Curran, P. M. Kevitt, Digital image steganography: Survey and analysis of current methods, *Signal Processing*, Vol. 90, No. 3, pp. 727-752, March, 2010.

[4] J. Shukla, M. Shandilya, A Recent Survey on Information-Hiding Techniques, In: R. K. Shukla, J. Agrawal, S. Sharma, G. Singh Tomer (eds), *Data, Engineering and Applications*, Springer, Singapore, 2019, pp. 57-70.

[5] M. Wu, B. Liu, Data Hiding in Image and Video: Part I - Fundamental Issues and Solutions, *IEEE Transactions on Image Processing*, Vol. 12, No. 6, pp. 685-695, June, 2003.

[6] M. Wu, H. Yu, B. Liu, Data Hiding in Image and Video: Part II - Designs and Applications, *IEEE Transactions on Image Processing*, Vol. 12, No. 6, pp. 696-705, June, 2003.

[7] Y.-K. Lin, High Capacity Reversible Data Hiding Scheme based upon Discrete Cosine Transformation *Journal of Systems and Software*, Vol. 85, No. 10, pp. 2395-2404, October, 2012.

[8] G. Xuan, Y. Q. Shi, Z. C. Ni, J. Chen, C. Yang, Y. Zhen, J. Zheng, High Capacity Lossless Data Hiding Based on Integer Wavelet Transform, in *Proceedings of the 2004 International Symposium on Circuits and Systems*, Vol. 2, Vancouver, BC, Canada, May 2004, pp. 29-32.

[9] Y.-H. Yu, C.-C. Chang, Y.-C. Hu, Hiding Secret Data in Images via Predictive Coding, *Pattern Recognition*, Vol. 38, No. 5, pp. 691-705, May, 2005.

[10] C.-K. Chan, L. M. Cheng, Hiding Data in Images by Simple LSB Substitution, *Pattern Recognition*, Vol. 37, No. 3, pp. 469-474, March, 2004.

[11] D.-C. Wu, W.-H. Tsai, A Steganographic Method for Images by Pixel-value Differencing, *Pattern Recognition Letters*, Vol. 24, No. 9-10, pp. 1613-1626, June, 2003.

[12] R. Z. Wang, C. F. Lin, J. C. Lin, Image Hiding by Optimal LSB Substitution and Genetic Algorithm, *Pattern Recognition*, Vol. 34, No. 3, pp. 671-683, March, 2001.

[13] C. H. Yang, C. Y. Weng, S. J. Wang, H. M. Sun, Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems, *IEEE Transactions on Information Forensics and Security*, Vol. 3, No. 3, pp. 488-497, September, 2008.

[14] S.-K. Chen, A Module-based LSB Substitution Method with Lossless Secret Data Compression, *Computer Standards & Interfaces*, Vol. 33, No. 4, pp. 367-371, June, 2011.

[15] 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 Proceedings on Vision, Image and Signal Processing*, Vol. 152, No. 5, pp. 611-615, October, 2005.

[16] S.-K. Yang, P. S. Huang, Image Steganographic Approach by Integrating Pixel-value Differencing and LSB Replacement Schemes, *Journal of Chung Cheng Institute of Technology*, Vol. 41, No. 2, pp. 89-98, November, 2012.

[17] S.-K. Yang, P.-S. Huang, Novel High-capacity Data Hiding Technique Using Summation and LSD Parity, *Journal of Chung Cheng Institute of Technology*, Vol. 45, No. 1, pp. 52-63, May, 2016.

[18] C. M. Wang, N. I. Wu, C. S. Tsai, M. S. Hwang, A High Quality Steganographic Method with Pixel-value Differencing and Modulus Function, *Journal of Systems and Software*, Vol. 81, No. 1, pp. 150-158, January, 2008.

[19] S.-L. Li, K.-C. Leung, L. M. Cheng, C.-K. Chan, A Novel Image-hiding Scheme Based on Block Difference, *Pattern Recognition*, Vol. 39, No. 6, pp. 1168-1176, June, 2006.

[20] R.-Z. Wang, Y.-D. Tsai, An Image-hiding Method with High Hiding Capacity Based on Best-Block Matching and K-means Clustering, *Pattern Recognition*, Vol. 40, No. 2, pp. 398-409, February, 2007.

[21] D. C. Lou, N. I. Wu, C. M. Wang, Z. H. Lin, C. S. Tsai, A Novel Adaptive Steganography Based on Local Complexity and Human Vision Sensitivity, *Journal of Systems and Software*, Vol. 83, No. 7, pp. 1236-1248, July, 2010.

[22] A. Pradhan, K. R. Sekhar, G. Swain, Digital Image Steganography Using LSB Substitution, PVD, and EMD, *Mathematical Problems in Engineering*, Vol. 2018, Article ID 1804953, September, 2018. https://doi.org/10.1155/2018/1804953.

[23] A. K. Sahu, G. Swain, Data Hiding Using adaptive LSB and PVD Technique Resisting PDH and RS analysis, *International Journal of Electronic Security and Digital Forensics*, Vol. 11, No. 4, pp. 458-476, 2019.

[24] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, Reversible Data Hiding, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 3, pp. 354-362, March, 2006.

[25] W.-L. Tai, C.-M. Yeh, C.-C. Chang, Reversible Data Hiding Based on Histogram Modification of Pixel Differences, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 6, pp. 906-910, June, 2009.

[26] C.-C. Lin, W.-L. Tai, C.-C. Chang, Multilevel Reversible Data Hiding Based on Histogram Modification of Difference images, *Pattern Recognition*, Vol. 41, No. 12, pp. 3582-3591, December, 2008.

[27] H. Luo, F.-X. Yu, H. Chen, Z.-L. Huang, H. Li, P.-H. Wang, Reversible Data Hiding Based on Block Median Preservation, *Information Sciences*, Vol. 181, No. 2, pp. 308-328, January, 2011.

[28] X. Wang, C. Shao, X. Xu, X. Niu, Reversible Data-Hiding Scheme for 2-D Vector Maps Based on Difference Expansion, *IEEE Transactions on Information Forensics and Security*, Vol. 2, No. 3, pp. 311-320, September, 2007.

[29] F. Yan, A. M. Iliyasu, P. Q. Le, Quantum Image Processing: A Review of Advances in Its Security Technologies, *International Journal of Quantum Information*, Vol. 15, No. 3, Article No. 1730001, April, 2017.

[30] W. Hu, R. Zhou, A. El-Rafei, S. Jiang, Quantum Image Watermarking Algorithm Based on Haar Wavelet Transform, *IEEE Access*, Vol. 7, pp. 121303-121320, August, 2019.

[31] G. Luo, R.-G. Zhou, J. Luo, W. Hu, Y. Zhou, H. Ian, Adaptive LSB Quantum Watermarking Method Using Tri-way Pixel Value Differencing, *Quantum Information Processing*, Vol. 18, No. 2, Article No. 49, February, 2019. https://doi. org/10.1007/s11128-018-2165-6.

Biographies

Ping Sheng Huang received his Ph.D. degree in Electronics and Computer Science from University of Southampton, UK, in 1999. He was with the Department of Electrical Engineering, Chung Cheng Institute of Technology as a Professor from 1999 to 2007. He has been served as a Professor in the Department of Electronic Engineering, Ming Chuan University, Taiwan from 2007 until now.

Sheng-Kai Yang was born in Taiwan in 1965. He received the master degree of Information System from Griffith University, Queensland Australia, in 1995. Since 1996, he has been with the Chia-Nan University of Pharmacy and Science, Tainan, Taiwan, where he is currently an Assistant Professor in Department of Information Management.