Novel High-capacity Data Hiding Technique Using Summation and LSD Parity

Sheng-Kai Yang[1] and Ping-Sheng Huang[2*]

[1]Departement of Information Management, Chia Nan University of Pharmacy&Science
[2]Department of Electronic Engineering, Ming Chuan University

ABSTRACT

Theaim of designing algorithms for data hiding applications is to increase the hiding capacity and keep the original image quality. However, both goals are normally contradictive to each other. Owing to this, the proposed schemein this paper is to embed the secret bit sequence into the pixel groups by adopting the summation of neighboring pixel values and the achieved parity of Least Significant Digit (LSD). The summation method utilizes the concept of pixel groups from the method of pixel value differencing (PVD) but not using the difference values for data hiding. Instead, the decimal representation of LSD summation relationship from two neighboring pixels is adopted for adjusting pixel values for data hiding. At the same time, the parity of second pixel value is used for increasing one more hidden-bit while using the summation relationship for data hiding. Data extraction can be done in the reverse direction. Therefore, the overall hiding capacity can be greatly improved. Experimental results show that the proposed data hiding algorithm can have better performance than normal PVD and LSB methods. Also, the better image quality can be assured.

Keywords: capacity,data hiding, parity,summation.

利用總和法與最小位奇偶性之高容量影像藏密法

楊勝凱[1] 黃炳森[2*]

[1]嘉南藥理科技大學資訊管理系
[2]銘傳大學電子工程系

摘要

設計資料隱藏技術的主要目標在於能同時提升隱藏容量與保持原來的影像品質。然而，這兩項目標通常很難同時達成。有鑒於此，本論文提出的方法為，採用影像當中的相鄰像素值之和與其最小位數的奇偶數特性，並將秘密位元序列隱藏於其中。像素值和之方法所採用之概念來自於像素差值法(PVD)，但是我們並未採用差值來隱藏資料。相對的，我們利用的是像素值和個位數的十進位表示法關係，來調整像素值以隱藏資料。同時，第二個像素值的奇偶性被用來多增加一個位元的藏密量。資量萃取的方式只要逆向進行即可完成。因此，整體的資料隱藏容量可以獲得大幅度的提升。實驗結果顯示，本論文提出之方法可以比一般的 PVD 或 LSB 演算法得到更佳之效能，而且，影像品質也可獲得確保。

關鍵詞：容量，資料隱藏，極性，總和

文稿收件日期 104.04.10; 文稿修正後接受日期 105.01.13; *通訊作者
Manuscript received April 10, 2015; revised Januery 13, 2016; * Corresponding author

I. INTRODUCTION

During the recent twenty years, owing to the contribution of new electronic devices and application software, multimedia information are getting more and more easier to be retrieved and produced by worldwide customers. On the other hand, the innovative development of internet and wireless networks makes the transmission and sharing of multimedia information via those ubiquitous communication channels becoming more convenient and faster. However, events of illegal duplication and copyright invasion also appear consistently and frequently that turns into an essential issue. Therefore, many prevention methods from government units and technology experts for those illegal activities are investigated and proposed. Information hiding approaches are emerged under this environment. Since invisible marks or data can be embedded into digital images, audio data, and video files, information hiding is gradually becoming an important technique for protecting copyright or ownership. Furthermore, those hidden information can even be used for avoiding illegal duplication and copyright proof. As shown in Fig.1[1], the approaches of information hiding can be divided into two branches: steganography and watermarking. The main purpose of digital watermarking is for protecting the ownership of copyright. Therefore, robustness becomes the most important requirement. However, the major applications of steganography are secret information sharing and transmission in which image, audio, and video files are adopted as the transmission channels. As such, the goals of steganography to be achieved are high capacity and less distortion. The aim of cryptography is to protect the information content from revealing by encryption and the existence of information is hidden by steganography [2].The ultimate goals of steganography are undetectability, robustness and hiding capacity. The purpose of robustness aims to resist different operations of image preprocessing and compressionand those are also used for classifying the approaches of steganography. Apart from the analysis and classification for information hiding [2] and image steganography [1], essential issues, solutions [3], practical algorithms and applications [4] for data hiding into images and videos are also investigated.

Current steganography techniques can be divided into frequency domain and spatial domain. The method of frequency domain embeds the secret information into frequency coefficients such as the coefficients of Discrete Cosine Transform [5] or Discrete Wavelet Transform [6]. However, the secret information is hidden into images by changing the data of cover media in the spatial domain for the approach of spatial domain and those methods include predictive coding [7] and changing the values of Least Significant Bit (LSB) [8] or Pixel Value Difference (PVD) [9]. Since the secret information embedded by spatial domain methods are easily destroyed by image attacking operations, the technique of frequency domain is developed. However, the hidden data capacity of frequency domain methods is less than that of spatial domain techniques. This paper presents a novel data hiding technique in spatial domain with high data capacity. Therefore, related data hiding algorithms in the spatial domain are explained.

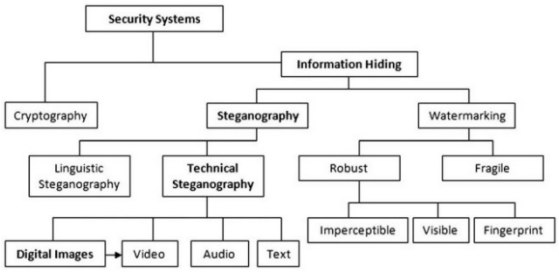

Fig.1. The Classification of Information Hiding approaches[1].

For data hiding algorithms in spatial domain, LSB method [8] is most frequently used. The main idea is to achieve data hiding by switching the least significant bits of image pixels with those bits in the secret bit stream and this method is fast and straightforward. However, the image quality is greatly reduced when more than two bits of each pixel are altered. Owing to that, many algorithms improving LSB methods are proposed [10-14] and they are mainly combined with other algorithms to increase data capacity or reduce the distortion of stego-images. For example, the LSB method and the genetic algorithm (GA) are integrated [10] and the learning ability is adopted to embed the secret information into the best positions. Yang et al. [11] proposed an adaptive scheme of data hiding byhiding more LSBs into pixels of edge areas and less bits into those of smooth areas.

Motivated by this, Chen [12] also used the smooth a reas to hide less bits and a module-based scheme is developed to modify pixel values. Furthermore, by combining pixel-value differencing and LSB replacement methods an image steganographic scheme [13] is proposed. Based on LSB replacement methods, the LSB method is directly adopted to embed information when the difference value of adjacent pixels is smaller than a threshold. On the contrary, the difference values are used for hiding information.

Although this can increase the data capacity of hidden information, the stego-image quality is also degraded. To meliorate this drawback, we have developed an algorithm [14] to increase data capacity and preserve image quality simultaneously by using compensation concept.

The method of pixel-value differencing (PVD) [9] is based on the difference value from two adjacent and non-overlapped pixels. At first, the quantization table of difference values from adjacent pixels is created and the number of bits embedded is decided. Then the difference values are individually adjusted according to the data values to be hidden. Motivated by PVD, we have proposed a tri-way PVD algorithm [15] to greatly increase the hidden data capacity. However, the image quality is unavoidably also degraded. Apart from using the PVD method, C. M. Wang et al. [16] adopted the modulus function and embedded more data bits into the remainders. Instead of using pixel difference values, Li et al. [17] separately divided the secret image and cover image into small blocks and then the secret image is embedded into the cover image using difference values and similarities between blocks. Best-block matching and K-means clustering are combined by R.-Z. Wang et al. [18] to search the number of each secret image block inside the cover image and the block number is then embedded into the lowest bit-plane of the cover image. For those techniques, when the number of hidden LSB bits is gradually increased, the image quality is also affected. Therefore, an adaptive scheme based on local complexity and human vision sensitivity is proposed by D. C. Lou et al. [19] for data hiding. Based on a self-defined modulus function, Lee and Chen [20] arranged the pixels and neighboring pixels of each cover image into groups and the secret data is embedded into or extracted from the cover image using this modulus function and the mapping process of multiplication combination from pixel groups.

Although large data capacity can be achieved by image steganographic schemes, the original cover image is also changed and distorted. Therefore, reversible data hiding approaches [21-25] are presented recently in which the cover image can be recovered to the original status after the secret data is extracted. Histogram-shifting is the most representative algorithm [21]. At first, the pixel group of the highest peak is shifted and the vacancy remained is used to embed the secret sequence. Based on the Histogram-shifting algorithm, pixel difference values [22] and the multilevel histogram [23] are further proposed to modify the difference histogram for data hiding. Also, Luo et al. [24] firstly calculated the average values and the difference histogram of image blocks and then a multilevel histogram shifting mechanism is presented for data hiding by referring to the block average values. Furthermore, difference expansion and 2D vector map are adopted by Yang et al. [25] to embed and extract the watermark from the image and the original image can be recovered. Since recovering to the original image quality has to be achieved, the data capacity accomplished by reversible data hiding algorithms is normally less than that of data hiding methods in spatial domain.

Based on the data hiding algorithms [14, 15] we have developed before and motivated by those approaches of using modulus functions [16, 20], this paper presents a novel technique by adopting the combination relationship of summation values from pixel groups. Also, the parity of second pixel value in each group is further used to embed one more bit and achieve the image quality compensation simultaneously. This paper is organized as follows. Section II describes the related algorithms and the details of the proposed method. Experimental results are demonstrated in Section III, before the conclusions in Section IV.

Ⅱ. RELATED WORKS

The main purpose of designing an information hiding algorithm is to increase the hiding capacity as large as possible and preserve the stego-image quality to the original image at the same time. However, those two goals are contradicted each other all the time. Therefore,

this paper presents a novel approach by combining the characteristics from the summation operation and the parity of least significant digit (LSD) to embed a secret bit sequence into pixel groups. The summation operations use the concept of pixel grouping but not the difference values for information hiding from the method of pixel value differencing [9]. Alternatively, the summation relationship of LSDs from adjacent two pixels is adopted to adjust two pixel values for hiding information. Furthermore, the LSD parity of the second pixel is used for embedding one more bit for increasing the data capacity. At first, in this section, the methods of Lease Significant Bit (LSB)[8], Pixel Value Differencing (PVD)[9], and H.-C. Wu's approach[13] are explained for understanding the proposed algorithm in this paper.

2.1 The Method of Using Least Significant Bits (LSBs)

The LSB data hiding approach [8] is the simplest and well-known technique in spatial domain. Based on miniature changes not easily sensed by human visual systems, the LSB approach embeds the secret data bits by changing the least significant bits of each pixel. Taking one bit of LSB hiding for an example, the least significant bit of each pixel value is 0 or 1 after this value is converted into a binary number. Therefore, the value can be changed and embedded is 0 or 1 for each pixel. Theoretically, the chance to randomly alter the least significant bit is 0.5 and the image quality will not be affected seriously. However, when more than two bits are changed in each pixel, the image quality will be distorted greatly. This means the LSB approach is unsuitable to be used for data hiding more than two bits. Motivated by that, in this paper, we adopt the parity of the second pixel value in each group to embed one more bit. This can increase the data capacity and preserve the image quality by compensation.

2.2 The Method of Pixel-value Differencing (PVD)

Based on changing the difference value between two adjacent pixels, the PVD data hiding algorithm [9] uses the quantization table obtained from the difference values to decide the allowable data bits embedded. Then the difference values are adjusted according to the secret data value that is desired to embed. Finally, the secret data is embedded into the new difference values. TABLEI lists an example of quantizationtabledividing the pixel values of 0 to 255 into 6 regions and showing the minimum values of each region. Also, the secret data bits can be embedded into the difference values are decided.

TABLE I
THE RANGES OF PIXEL VALUE DIFFERENCES (PVD)

region	difference range	bits to hide(k)	range minimum value(g)
1	0to 7	3	0
2	8to 15	3	8
3	16to 31	4	16
4	32to 63	5	32
5	64 to127	6	64
6	128to 255	7	128

Assume that the height and the width of a gray-level image are M and N, respectively. Therefore, there are $M \times N$pixels in this image in which each pixel can be considered as an intensity component with 8 bits. Furthermore, define that the positions of two neighboring pixels are $(i,j)$and $(i,j+1)$, corresponding to two pixel values of $f(i,j)$ and $f(i,j+1)$, respectively. When the difference value of two pixels is defined as $d = |f(i,j) - f(i,j+1)|$, then the value d is between 0 to 255. After dividing the range of the value d into 6 regions, one possible combination is shown in TABLEI in which kis the number of bits to be embedded. During the data hiding stage, after kbits are extracted from the secret data, they are further converted into a decimal number and added with the minimum value gin that region. Then the pixel values of $f(i,j)$ and $f(i,j+1)$ are individually adjusted to embed those k bits. This paper proposes a data hiding scheme by using the summation relationship from least significant digits of two neighboring pixel values. Then the table look-up method is used to adjust two pixel values and the secret data is finally embedded.

2.3 The Method of Combining PVD and LSB Replacement

The data hiding approach of integrating PVD and LSB replacement [13] mainly calculates the difference value between two neighboring and non-overlapped pixels and then

the number of data hiding bits is decided according to this value. The LSB method [8] is adopted for data hiding when the difference value is less than the preset threshold value and the PVD method [9] is used for the other cases. For example, assume the threshold value is 7 and the values of two neighboring pixels are $f(i,j)$and $f(i,j+1)$.

When the difference value is larger than 7, then the PVD method [9] is used for data hiding. Otherwise, 4 bits are extracted from the bit sequence of data hiding (replacement bits = $2 \times 2 = 4$) and they are individually placed into the lowest 2 bits of two neighboring and non-overlapped pixel values. Now the new pixel values are changed to $f'(i,j)$ and $f'(i,j+1)$.Furthermore, the absolute difference value of those two new pixel values is checked. The process is over and the next pixel group is calculated when the difference value is smaller than or equal to 7. When the difference value is greater than 7, the adjustment to each pixel value is performed by deducting 4 from the pixel with a larger value is and adding 4 to another pixel. Now the new pixel values are represented by $f''(i,j)$ and $f''(i,j+1)$. Until now, the process is over and the next pixel group is calculated. According to the experimental results [13], the image quality is greatly degraded when the data capacity is increased. Owing to that, in this paper, we are planning to increase the data capacity by using the summation operation of two pixels for data hiding and the parity of the second pixel value is further adopted to embed one more bit into the least significant digit of the second pixel. This will increase the total hiding capacity for the image.

Ⅲ. PROPOSED METHOD

The proposed approach is based on adjusting the values of two neighboring pixels to embed the secret bit sequence. At first, the secret bit sequence to be embedded is randomly permuted by the prepared key value. Then the cover image is divided into groups of two neighboring pixels ready for data hiding using the summation operations.

The first step of the algorithm sequentially extracts 5 bits from the secret bit sequence. Then the summation value of two least significant digits (LSDs) from every group of two neighboring pixels is adjusted to be the value of adding the decimal value of the first 4 bits with 2.This can accomplish the data hiding for 4 bits.

TABLE II lists the possible combination sets for 4-bit secret data (added by 2) and the summation of 2 LSDs from the pixel group. The generation of TABLE II is described as follows. At first, the first column lists the binary values of 4 hidden bits. Then, the decimal values of 4 hidden bits are individually added by 2 and then listed in the second column. Therefore, the new range of decimal values is from (0, 15) to (2, 17). Next, the sets of LSD addition combination for those decimal values are listed in the third column. Finally, the fourth column demonstrates the number of sets listed in the third column. The reason for adding 2 into the LSD summation of two pixels is to increase the number of combination sets and then improve the performance of data hiding.

TABLE II
THE POSSIBLE COMBINATION SETS FOR 4-BIT SECRET DATA (ADDED BY 2) AND THE SUMMATION OF 2 LSDS FROM THE PIXEL GROUP

Binary number of 4 bits t	$r=(t)_{10}+2$	combination sets of decimal addition for the number r	Comb. number $n(r)$
$(0000)_2$	2	(0,2),(1,1),(2,0)	3
$(0001)_2$	3	(0,3),(3,0),(1,2),(2,1)	4
$(0010)_2$	4	(0,4),(4,0),(1,3),(3,1),(2,2)	5
$(0011)_2$	5	(0,5),(5,0),(1,4),(4,1),(2,3),(3,2)	6
$(0100)_2$	6	(0,6),(6,0),(1,5),(5,1),(2,4),(4,2),(3,3)	7
$(0101)_2$	7	(0,7),(7,0),(1,6),(6,1),(2,5),(5,2),(3,4),(4,3)	8
$(0110)_2$	8	(0,8),(8,0),(1,7),(7,1),(2,6),(6,2),(3,5),(5,3),(4,4)	9
$(0111)_2$	9	(0,9),(9,0),(1,8),(8,1),(2,7),(7,2),(3,6),(6,3),(4,5),(5,4)	10
$(1000)_2$	10	(1,9),(9,1),(2,8),(8,2),(3,7),(7,3),(4,6),(6,4),(5,5)	9
$(1001)_2$	11	(2,9),(9,2),(3,8),(8,3),(4,7),(7,4),(5,6),(6,5)	8
$(1010)_2$	12	(3,9),(9,3),(4,8),(8,4),(5,7),(7,5),(6,6)	7
$(1011)_2$	13	(4,9),(9,4),(5,8),(8,5),(6,7),(7,6)	6
$(1100)_2$	14	(5,9),(9,5),(6,8),(8,6),(7,7)	5
$(1101)_2$	15	(6,9),(9,6),(7,8),(8,7)	4
$(1110)_2$	16	(7,9),(9,7),(8,8)	3
$(1111)_2$	17	(8,9),(9,8)	2

When selecting the matching pair to the summation value of two LSDs, we propose hiding one more bit (0 or 1) by choosing the parity of the second pixel value. Therefore, every 5 bits from the secret bit sequence can be sequentially embedded into two pixel values from each pixel group. That is, after using the summation operations for data hiding, the parity

of the second pixel value can be further adopted for hiding more bits into each group. Since only the parity is used without changing any pixel value, the image quality can be preserved.

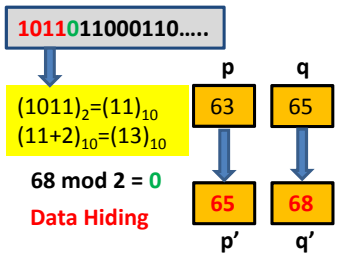

Fig.2. Data hiding for the number 13.

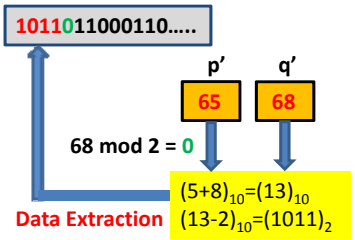

Fig.3. Data extraction for the number 13

(1) Possible 6 groups:
(64, 69),(69, 64),(68, 65)
(65, 68),(66, 67),(67, 66)

(2) Possible 9 groups of (64, 69):
(64, 69),(64, 59),(64, 79)
(54, 69),(54, 59),(54, 79)
(74, 69),(74, 59),(74, 79)

(3) Total possible groups = 6x9 = 54

Fig.4. Combination sets for the number 13.

To clearly demonstrate this algorithm, one example is taken and described beforehand. Then detailed steps of the proposed algorithm are listed and explained.By referring to Fig. 2, assume that the 5 bits extracted from the secret bit sequence are (10110)2. Since the first 4 bits are (1011)2=(11)10, the decimal value becomes (11+2)10=(13)10 after added by 2. The fifth bit in (10110)2 is 0 which is an even number. The pixel group used for demonstration is (63, 65), that is, $f(i,j) = 63$and $f(i,j+1) = 65$. At first, the summation of the two LSDs from two pixels is calculated and the result is $3 + 5 = 8$. Since $8 \neq 13$, the summation value needs to be adjusted to13. According to the information listed along the row (1011)2 in TABLEII, there are 6 combination sets : (4,9), (9,4), (5,8), (8,5), (6,7), and (7,6). Furthermore, for each set of those 6 combinations (for example, the set (64, 69) in Fig. 4(2)), two more values of ± 10 are added to two numbers of this set and the original pixel set is preserved. Now there are 9 combination sets shown in Fig. 4(2). As shown in Fig. 4(3), there are $6 \times 9 = 54$ possible sets obtained. Finally, for each of those 54 sets, the square error to the original pixel set (63, 65) is calculated. Then the set with the minimum square error and an evennumber of second new pixel value (corresponding to 0) is selected as the candidate for the new pixel set. The new set is (65, 68) in this case. That is, $f'(i,j) = 65$ and $f'(i,j+1) = 68$ as shown in Fig. 2. For those combination sets with numbers over 255 or less than 0 when doing ± 10 operations, they are directly removed and the calculation of square errors is not needed.

The demonstration example of data extraction is shown in Fig. 3 and in the first step, the summation of two LSDs of two pixel values in the pixel set is calculated and the parity of the second pixel value is checked. Then the summation of two LSDs is deducted by 2 and converted into a binary number of 4 bits. Furthermore, the bit value decided from the checked parity is combined with those 4 bits to become a 5-bit value. By following this procedure, the data embedding and extraction of each 5-bit secret data can be achieved. To fully understand the proposed algorithm, the detailed explanation is described by a flowchart and sequential steps.

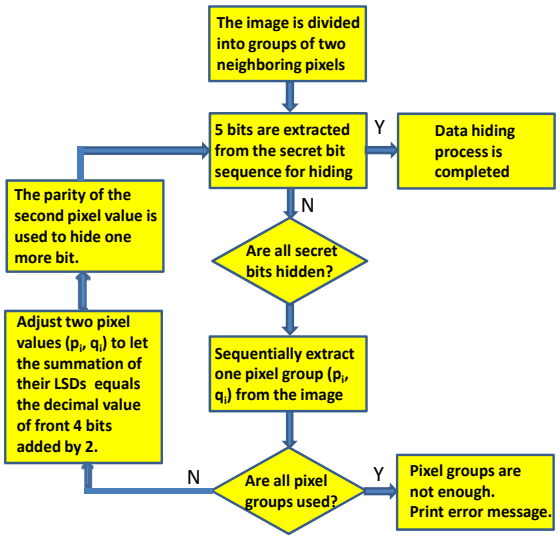

Fig.5. Data hiding flowchart.

Fig. 5 and Fig. 6 depict two flowcharts

corresponding to data hiding and data extraction algorithms proposed by this paper, respectively. As shown in Fig. 5, the flowchart of data hiding, the cover image is divided into groups of two pixels in the beginning. Also, before performing data hiding, to protect the data safety, the secret data sequence is randomly permuted and encrypted by a key value. The order of pixel positions embedded with secret data can also be randomly permuted by another key value to increase the security level. When the length of the secret data sequence is not a multiple of 5, the tail of this bit sequence is padded with 0. As shown in Fig. 6, the flowchart of data extraction, after the secret data sequence is extracted and combined, the bit sequence is reversely permuted into the original order by the previous key value.

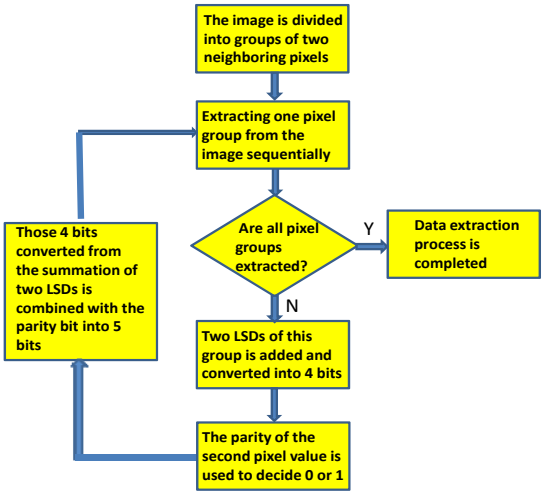

Fig.6. Data extraction flowchart.

Assume that we have a secret bit sequence and the detailed data hiding steps are described as follows.

Data Hiding

Step 1 Input one gray-level image and divide the pixels of thisimage into groups of two non-overlapped and neighboring pixels.

Step 2 Take 5 consecutive bits $(b_4b_3b_2b_1b_0)_2$ from the secret bit sequence. When the secret data bits are all hidden into the cover image, the data hiding algorithm is over. Otherwise, the next pixel group $(f(i,j), f(i,j+1))$ is taken from the cover image. When there is no pixel group left for hiding the remaining secret data bits, the data hiding error is generated and the hiding algorithm is over.

Step 3 Assume $t = (b_4b_3b_2b_1)_{10}$ and the sum of two LSDsfrom two pixel values of one group is s. This equation is given by

$$s = (f(i,j) \bmod 10) + (f(i,j+1) \bmod 10) \quad (1).$$

When $t \neq s$, the algorithm is going to Step 4, otherwise, the algorithm is processed by the following two cases depended on the value of b_0.

Case 1: $b_0 = 0$

If $f(i,j+1)$ is an odd number, then $(f(i,j), f(i,j+1))$ is adjusted to$(f'(i,j),,f'(i,j+1))$ by

$$\begin{cases} f'(i,j) = f(i,j) + 1 \\ f'(i,j+1) = f(i,j+1) - 1 \end{cases} \quad (2)$$

Then the algorithm goes to **Step 2**.

If $f(i,j+1)$ is an even number, then the adjustment of $(f(i,j), f(i,j+1))$ is not needed. The algorithm goes to **Step 2**.

Case 2:$b_0 = 1$

If $f(i,j+1)$ is an even number, then $(f(i,j), f(i,j+1))$ is adjusted to $(f'(i,j), f'(i,j+1))$ by

$$\begin{cases} f'(i,j) = f(i,j) - 1 \\ f'(i,j+1) = f(i,j+1) + 1 \end{cases} \quad (3)$$

The algorithm goes to **Step 2**.
If $f(i,j+1)$ is an odd number, then the adjustment of $(f(i,j), f(i,j+1))$ is not needed. The algorithm goes to **Step 2**.

Step 4 By referring to TABLEII, the summation value $r = t + 2$ is used to decide the number$n(r)$ of combination sets. Then, for those two pixel values in each combination set, $(g_i, h_i), i = 1$ to $n(r)$, two more values of ± 10 are calculated. Therefore, those two values have three possible changes: $(g_i + 10, g_i, g_i - 10)$ and $(h_i + 10, h_i, h_i - 10)$. Now, 9 matching combination sets can be generated and there are $m = n(r) \times 9$ matching pairs obtained after considering the previous $n(r)$

combination sets. During adding and subtracting operations, the matching pair is removed when the new pixel value is greater than 255 or less than 0.

Step 5 For each pixel matching pair, $(g'_k, h'_k), k = 1$ to m, from **Step 4,** the square error d_k between the matching pair and the pixel group $(f(i,j), f(i,j+1))$ is calculated by

$$d_k = \sqrt{(f(i,j) - g'_k)^2 + (f(i,j+1) - h'_k)^2} \quad (4)$$

in which $k = 1$ to m.

Step 6 According to the square error d_k calculated from **Step 5** and the value of b_0, two cases are processed by the followingtwo cases depended on the value of b_0.

Case 1: $b_0 = 0$

From m matching pairs of (g'_k, h'_k), the pair (g'_{min}, h'_{min}) with a minimum square error d_k and an even h'_k is selected. Then the pixel group $(f(i,j), f(i,j+1))$ is modified to

$$\begin{cases} f'(i,j) = g'_{min} \\ f'(i,j+1) = h'_{min} \end{cases} \quad (5)$$

Then the algorithm goes back to **Step 2**.

Case 2:$b_0 = 1$

From m matching pairs of (g'_k, h'_k), the pair (g'_{min}, h'_{min}) with a minimum square error d_k and an odd h'_k is selected. Then the pixel group $(f(i,j), f(i,j+1))$ is modified to $(f'(i,j), f'(i,j+1))$ by Equation (5) and the algorithm goes back to **Step 2**.

After the data hiding steps are described, the data extraction steps are explained as follows. Note that data extraction for the hidden data is in a reverse direction to data hiding.

Data Extraction

Step 1 All the pixels in the cover image are divided into groups of two non-overlapped and neighboring pixels.

Step 2 The data extraction algorithm is over when all hidden secret data bits are extracted. Otherwise, one pixel group is taken from the remaining group sets and we assume two pixel values in this group are $f(i,j)$ and $f(i,j+1)$, respectively.

Step 3 The summation *s*of two LSDs from $f(i,j)$ and $f(i,j+1)$ is given by

$$s = (f(i,j) \bmod 10) + (f(i,j+1) \bmod 10) \quad (6)$$

Step 4 The decimal value ***s*** can be converted into a binary value with 4 bits by $(b_4 b_3 b_2 b_1)_2 = (s)_2$.

Step 5 The parity of $f(i,j+1)$ is checked to recover the value of b_0 and the algorithm is processed by the following two cases.

Case 1:

When $(f(i,j+1) \bmod 2) = 0$, we assume $b_0 = 0$ and b_0 iscombined with $(b_4 b_3 b_2 b_1)_2$ into a 5-bit value $(b_4 b_3 b_2 b_1 b_0)_2$. Those 5 bits are then placed into the secret data bit sequence. After that, the algorithm goes back to **Step 2**.

Case2:

When $(f(i,j+1) \bmod 2) = 1$, we assume $b_0 = 1$ and b_0 is combined with $(b_4 b_3 b_2 b_1)_2$ into a 5-bit value $(b_4 b_3 b_2 b_1 b_0)_2$. Those 5 bits are then placed into the secret data bit sequence. After that, the algorithm goes back to **Step 2**.

IV. EXPERIMENTAL RESULTS

To verify the performance of the proposed algorithm, we have performed three experiments. MATLAB R2009a has been adopted for software development and the computer specifications are Intel(R) Core(TM)2 Quad CPU Q8400 with 1.75GB RAM. As shown in Fig. 7, the images used in the experiments are 5 gray-level images with the size of 512×512(Lena, Baboon, Peppers, Jet, and sailboat). The goals of the proposed algorithm are to increase the data capacity and preserve the image quality simultaneously. The secret data sequence used is a randomly permuted bit sequence and all experimental results are averaged values afterrunning 1000 times. When

				
Lena	Baboon	Peppers	Jet	Sailboat

Fig.7.Five test gray-level images with the size of 512x512

Algorithm	*PVD method*	*H.-C. Wu's method(2bits)*	*Proposed method*
Stego-image			
PSNR value(dB)	*41.09*	*38.8*	*41.55*
capacity(bytes)	*51219*	*66064*	*81920*

Fig.8.The comparison of data capacity under similar image quality.

the PVD method [18] is used, the range table is divided into 8, 8, 16, 32, 64 and 128. The evaluation measure adopted for the stego-image quality is the PSNR(Peak Signal to Noise Ratio) value and the unit is dB.The PSNR value greater than 30 is used as the admitted range and there is almost no visual difference to the original image when the PSNR value is greater than 40.

In the first experiment, the Lena image is used and the corresponding experimental results to three approaches are shown in Fig. 8. To verify that the proposed method can embed more data bits and preserve the image quality simultaneously, we have compared the performance of data hiding for three approaches: PVD method [9], H.-C. Wu's method [13], and the proposed method. Based on the requirement of maintaining similar image quality (40<PSNR<41), Fig. 8 demonstrates that the proposed method can embed more bytes than the PVD method [9] and the number of extra data is 81920-51219=30701 bytes. On the other hand, by taking the maximum data capacity can be achieved by the H.-C. Wu method [13], the extra data can be embedded by the proposed method is 81920-66064=15856 bytes. Note that two LSB bits are used for the H.-C. Wu's method [19] and the image quality have been degraded to 38.8 dB. Furthermore, the proposed method can still preserve the image quality to 41.55 dB and embed more bytes than the H.-C. Wu's method [19]. Therefore, the proposed method can indeed embed more data bytes and preserve the image quality simultaneously than the PVD method [9] and the H.-C. Wu's method [13].

In the second experiment, four more test images are used to evaluate the performance of the proposed approach and those images are Baboon, Peppers, F-16, and Sailboat. As shown in TABLE III, the proposed method can embed the same amount of data individually into those four test images and the data capacity hidden are more than that by the other two methods. Meanwhile, high image quality over 40 dB can be maintained by the proposed approach at the same time.

TABLE III
THE PERFORMANCE COMPARISON TO PVD METHOD,H-C WU'S METHOD, AND THE PROPOSED METHOD USING 4 TEST IMAGES

Cover image (512x512)	Da-Chun Wu's PVDmethod		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	81920	41.55
Baboon	56291	37.90	68007	33.33	81920	41.59
Peppers	50685	41.73	66032	37.50	81920	41.59
F-16	51243	40.97	66256	37.63	81920	41.47
Sailboat	52779	39.32	66622	35.01	81920	41.59

After performing 1000 times of data hiding experiments by randomly permuting the bit sequence and calculating the averaged values, we can conclude that same amount of data can be embedded into those five test images listed in Fig. 7 for the image size of 512×512. Since the test data used are randomly permuted bit sequences, the PSNR values are varied for different images. Note that the set of test bit sequences used are the same for different approaches in the experiments.

TABLE IV
THE PERFORMANCE COMPARISON TO THE PURE SUMMATION METHOD AND THE PROPOSED METHOD USING 5 TEST IMAGES

Cover image (512x512)	Pure Summation Method		Proposed Method	
	Capacity (bytes)	PSNR (dB)	Capacity (bytes)	PSNR (dB)
Lena	65536	42.24	81920	41.55
Baboon	65536	42.29	81920	41.59
Peppers	65536	42.28	81920	41.59
F-16	65536	42.16	81920	41.47
Sailboat	65536	42.29	81920	41.59

In the third experiment, five test images listed in Fig. 7 are used again to evaluate the performance of the proposed approach. However, the method using the pure summation operations only is compared this time. As shown in TABLE IV, the proposed method can embed more data than the pure summation method and the extra data bytes can be embedded are 81902-65536=16384 bytes. Only a little image qualityis sacrificed by the proposed method. By comparing TABLE III with TABLE IV, even the pure summation method can still achieve better performance than the PVD method [9] and the H.-C. Wu's method [13]. Therefore, the pure summation method can be used in those cases that higher image quality and data capacity are needed at the same time.

Since the main goal of the proposed approach is to improve the PVD method [9], also, more hiding data and high image quality can be simultaneously achieved after the data hiding, the PVD method [9] is adopted to be compared with the proposed approach. As shown in TABLE III, the averaged hiding data capacity of the proposed method is increased 50% more than the PVD method [9]. Furthermore, the data can be individually embedded into five test images with the size of 512x512 are 81920 bytes. For the image quality after data hiding, the PSNR value accomplished by the proposed approach is higher than that of the PVD method [9] and the value is around 41.5 dB with a high image quality. To compare the H.-C. Wu method [13] (replacing 2 LSB bits) with the proposed approach, Fig. 8 and TABLE III have demonstrated that the proposed method can achieve higher data capacity and better image quality (PSNR) than the H.-C. Wu method [13].

The proposed approach takes 5 bits from the secret data sequence for data hiding each time. Therefore, the theoretical performance for the hiding capacity of an $N \times N$ image is fixed. Assume that the image size is $N \times N$, the data hiding capacity is $((N \times N)/2) \times 5$ bits or $((N \times N)/2 \times 5)/8$ bytes. On the other hand, the image quality of the stego-image depends on the cover image that is hard to be analyzed theoretically. In general, the stego-images can have an average PSNR value over 41 dB and the difference to the cover images is hard to be perceived.

Since the main goal of data hiding algorithms is to increase the data hiding capacity and image quality simultaneously, the robustness is not considered in this paper. To protect the security of the secret data, the method of random permutation to the secret data by a security key can be adopted before data hiding process and the key can be used to recover to the original sequence of secret data after data extraction.

In the proposed approach, five fixed bits are used to be embedded into each pixel group. This might not be fully adaptive to all images. Therefore, this needs to be investigated and

improved in the future. Also, the goals of maintaining high data capacity and image quality simultaneously should be satisfied.

V. CONCLUSION

The capacity of data hiding and the quality of the cover image are two major factors to be improved for data hiding techniques. However, since they are normally contradictive to each other, those two goals are unable to be satisfied at the same time. For data hiding techniques in the spatial domain, the pixel values in the cover image are modified to accomplish the goal of hiding secret information. This paper is motivated by the data hiding techniques of PVD method and LSB substitution in the spatial domain. The ultimate goal of the proposed approach is to maintain the image quality after hiding the maximum secret data.

The difference value between two neighboring pixels is adopted by the PVD method [9] to embed secret information. The approach [13] combining the PVD method [9] and LSB substitution can increase the data capacity more than the PVD method. However, the image quality is also sacrificed. This paper presents the data hiding algorithm by combining summation operations of two neighboring pixel vales and the parity of their LSDs. The secret bit sequence is embedded into the modified pixel values of each pixel group. Instead of using the difference value of two neighboring pixels for data hiding, the proposed method only uses the concept of pixel groups. Furthermore, the summation characteristics of decimal pixel values are adopted and the LSD is used to adjust the pixel values for data hiding. This algorithm can not only embed more secret information but also the image qualitycan be maintained. Those two advantages have been verified by the experimental results. Although promising performance has been achieved, the test image database needs to be extended in the future for thorough evaluation. Also, other basic mathematic operations can be designed for data hiding applications.

ACKNOWLEDGEMENTS

This research is supported in part by the National Science Council, Taiwan, under the grant NSC 102-2221-E-130-010.

REFERENCES

[1] Cheddad, A., Condell, J., Curran, K., and Kevitt, P. M., “Digital image steganography: Survey and analysis of current methods,” Signal Processing, 90(3), pp. 727–752, March 2010.

[2] Petitcolas, F. A. P., Anderson, R. J., and Kuhn, M. G., “Information Hiding – A Survey,” Proceedings of the IEEE, Vol. 87, No. 7, pp. 1062-1078, July 1999.

[3] Wu, M., and Liu, B., "Data hiding in image and video: Part I - Fundamental issues and solutions," IEEE Transactions on Image Processing, 12(6), pp. 685-695, June 2003.

[4] Wu, M., Yu, H., and Liu, B., "Data hiding in image and video: Part II - Designs and Applications," IEEE Transactions on Image Processing, 12(6), pp. 696-705, June 2003.

[5] Lin, Y.-K., “High capacity reversible data hiding scheme based upon discrete cosine transformation,” Journal of Systems and Software, 85(10), pp. 2395–2404, October 2012.

[6] Xuan, G., Shi, Y.Q., Ni, Z. C., Chen, J., Yang, C., Zhen, Y., and Zheng, J., "High capacity lossless data hiding based on integer wavelet transform," Proceedings of the 2004 International Symposium on Circuits and Systems, Vol. 2, pp. 29-32, 23-26 May 2004.

[7] Yu, Y.-H., Chang, C.-C., and Hu, Y.-C., “Hiding secret data in images via predictive coding,” Pattern Recognition, 38(5), pp. 691–705, May 2005.

[8] Chan, C.-K., and Cheng, L. M., “Hiding Data in Images by Simple LSB Substitution,” Pattern Recognition, 37(3), pp. 469-474, 2004.

[9] Wu, D.-C., and Tsai, W.-H., “A Steganographic Method for Images by Pixel-value Differencing,” Pattern Recognition Letters, 24(9-10), pp. 1613-1626, June 2003.

[10] Wang, R. Z., Lin, C. F., and Lin, J. C., “Image hiding by optimal LSB substitution and genetic algorithm,” Pattern Recognition, 34(3), 671–683, 2001.

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

[12] Chen, S.-K., “A module-based LSB substitution method with lossless secret data compression,” Computer Standards & Interfaces, 33(4), pp. 367–371, June 2011.
[13] Wu, H.-C., Wu, N.-I., Tsai, C.-S., and Hwang, M.-S., "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, 2005.
[14] Yang, S.-K., and Huang, P. S., “An 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, Nov. 2012.
[15] Chang, K.-C., Chang, C.-P., Huang, P. S., and Tu, T.-M., “A Novel Image Steganographic Method Using Tri-way Pixel-value Differencing,” Journal of Multimedia, Vol. 3, No. 2, pp. 37-44, June 2008.
[16] Wang, C. M., Wu, N. I., Tsai, C. S., and Hwang, M. S., “A High Quality Steganographic Method with Pixel-value Differencing and Modulus Function,” Journal of Systems and Software, Vol. 81, No. 1, pp.150-158, 2008.
[17] Li, S.-L., Leung, K.-C., Cheng, L. M., and Chan, C.-K., “A novel image-hiding scheme based on block difference,” Pattern Recognition, 39(6), pp. 1168–1176, 2006.
[18] Wang, R.-Z., and Tsai, Y.-D., “An Image-hiding Method with High Hiding Capacity Based on Best-Block Matching and K-means Clustering,” Pattern Recognition Letters, 40(2), pp. 398-409, February 2007.
[19] Lou, D. C., Wu, N. I., Wang, C. M., Lin, Z. H., and Tsai, C. S., “A Novel Adaptive Steganography Based on Local Complexity and Human Vision Sensitivity,” Journal of Systems and Software, Vol. 83, No. 7, pp.1236-1248, 2010.
[20] Lee, C.-F., and Chen, H.-L., “A novel data hiding scheme based on modulus function,” Journal of Systems and Software, 83(5), pp. 832–843, May 2010.
[21] Ni, Z., Shi, Y.-Q., Ansari, N., and Su, W., “Reversible data hiding,” IEEE Transactions on Circuits and Systems for Video Technology, 16(3), pp. 354-362, March 2006.
[22] Tai, W.-L., Yeh, C.-M., and Chang, C.-C., “Reversible Data Hiding Based on Histogram Modification of Pixel Differences,” IEEE Transactions on Circuits and Systems for Video Technology, 19(6), pp. 906-910, June 2009.
[23] Lin, C.-C., Tai, W.-L., and Chang, C.-C., “Multilevel reversible data hiding based on histogram modification of difference images,” Pattern Recognition, 39(6), pp. 3582–3591, 2008.
[24] Luo, H., Yu, F.-X., Chen, H., Huang, Z.-L., Li, H., and Wang, P.-H., “Reversible data hiding based on block median preservation,” Information Sciences, 181(2), pp. 308–328, January 2011.
[25] Wang, X., Shao, C. Y., Xu, X., Niu, X., “Reversible Data-Hiding Scheme for 2-D Vector Maps Based on Difference Expansion,” IEEE Transactions on Information Forensics and Security, Vol. 2(3), pp. 311-320, Sept. 2007.