A Novel Image Steganographic Method Using Tri-way Pixel-Value Differencing

Ko-Chin Chang[a], Chien-Ping Chang[a], Ping S. Huang[b], and Te-Ming Tu[a]

[a]Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University, Taoyuan 335, Taiwan, R.O.C.
[b]Department of Electronic Engineering, Ming Chuan University, Taoyuan 333, Taiwan, R.O.C.
Email: {g990301, cpchang}@ccit.edu.tw, pshuang@mcu.edu.tw, tutm@ccit.edu.tw

***Abstract*—To enlarge the capacity of the hidden secret information and to provide an imperceptible stego-image for human vision, a novel steganographic approach using tri-way pixel-value differencing (TPVD) is proposed in this paper. To upgrade the hiding capacity of original PVD method referring to only one direction, three different directional edges are considered and effectively adopted to design the scheme of tri-way pixel-value differencing. In addition, to reduce the quality distortion of stego-image brought from setting larger embedding capacity, an optimal approach of selecting the reference point and adaptive rules are presented. Theoretical estimation and experimental results demonstrate that the proposed scheme can provide superior embedding capacity and give secrecy protection from dual statistical stego-analysis. Besides, the embedded confidential information can be extracted from stego-images without the assistance of original images.**

***Index Terms*—Steganography, Pixel-value differencing, Data hiding.**

I. Introduction

To protect secret message from being stolen during transmission, there are two ways to solve this problem in general. One way is encryption, which refers to the process of encoding secret information in such a way that only the right person with a right key can decode and recover the original information successfully. Another way is steganography and this is a technique which hides secret information into a cover media or carrier so that it becomes unnoticed and less attractive. If the cover media is a digital image hidden with secret data, this image is called "stego-image". Moreover, the information hiding technique could be used extensively on applications of military, commercials, anti-criminal, and so on [1].

Watermarking, another way of data hiding, aims at different purposes from steganography. Copyright protection and authentication is one primary target of image watermarking and it is required that the embedded information can be prevented, resisted, or altered up to some degrees of distortion while the watermarked image is attacked or damaged. Because of this requirement, robustness becomes the main benchmark emphasized by the techniques of image watermarking. Unlike watermarking, capacity and invisibility are the benchmarks needed for data hiding techniques of steganography. Due to the benchmarks, fragile schemes of data hiding are often used on steganography and least-significant-bits (LSB) substitution is one of them [2-7].

The scheme of least-significant-bits (LSB) substitution is a common and well-known steganographic method [2-7]. Embedded data are converted to substitute the fixed length LSB of each pixel. However, since some pixels can not tolerate changes of substitution during the embedding process, then those pixels appear apparently different from their original values. This effect occurs seriously in the smooth area that those changes are noticeable for human eyes. Thus, improving the stego-image quality and adaptive adjusting hiding capacity are two major aims to expand related researches about LSB. Therefore, Wang et al. [4] proposed a method using the genetic algorithm to embed secret data into each host pixel and the transformed value is closer to the original host pixel. However, using the genetic algorithm consumes huge computational time and the solution of a bijective mapping function is not optimal. In 2002, Chang et al. [5] offered their dynamic programming strategy to pick out the best solution from all of possible conditions that can significantly reduce the computation time. Also, Chan and Cheng [6] proposed to hide data by simple LSB substitution with an optimal pixel adjustment process (OPAP). Using the OPAP algorithm can prove that the obtained worst-mean-square-error (WMSE) between the cover image and the stego-image is less than 1/2 of that obtained by the normal LSB. Those steganographic schemes aim to improve the stego-image quality. On the other hand, an adaptive method based on using variable amount of bits instead of fixed length is proposed [7] for adjusting the hiding capacity.

Recently, two benchmarks are adopted by steganographic techniques to evaluate the hiding performance. First one is the capacity of hiding data and another one is the imperceptibility of the stego-image, also called the quality of stego-image. The pixel-value differencing (PVD) method proposed by Wu and Tsai [8] can successfully provide both high embedding capacity and outstanding imperceptibility for the stego-image.

This work was supported by the National Science Council, R.O.C., under Grant NSC 96-2221-E-130 -022. Prof. Ping S. Huang is the corresponding author (e-mail: pshuang@mcu.edu.tw).

Therefore, based on PVD method, various approaches have also been proposed [9-10].

In this paper, a novel steganographic approach using tri-way pixel-value differencing (TPVD) is proposed. To increase the hiding capacity of original PVD method referring to only one direction, three different directional edges are considered and effectively adopted to design the tri-way differencing scheme. Also, to reduce the quality distortion of the stego-image brought from setting larger embedding capacity, an optimal approach of selecting the reference point and adaptive rules are presented. This can maintain the stego-image at an acceptable and satisfied quality.

The rest of this paper is organized as follows. Section 2 reviews the PVD method. In Section 3, the proposed construction scheme is presented. Experimental results are illustrated and discussed in Section 4, prior to Conclusions in Section 5.

II. Review of the PVD method

In the original PVD method [8], a gray-valued cover image is partitioned into non-overlapping blocks composed with two consecutive pixels, p_i and p_{i+1}. From each block, a difference value d_i can be calculated by subtracting p_i from p_{i+1}. The set of all difference values ought to range from -255 to 255. Therefore, $|d_i|$ ranges from 0 to 255. Thus, the block with a small value $|d_i|$ locates in the smooth area, whereas a block with a large value $|d_i|$ is considered as a block with sharp edges. According to the properties of human vision, eyes can tolerate more changes in sharp-edge blocks than in smooth blocks. That is, more data can be embedded into the edge areas than into smooth areas. Therefore, in the PVD method, the first step is to design a range table with n contiguous ranges (R_k where $k = 1, 2, \ldots, n$) and the table range is from 0 to 255. The lower and upper boundary of R_k are denoted by l_k and u_k, respectively, then $R_k \in [l_k, u_k]$. The width w_k of R_k is calculated by $w_k = u_k - l_k + 1$ and w_k decides how many bits can be hidden in two consecutive pixels. Since R_k is designed as a variable, the original range table is required to extract the embedded secret data. Based on the consideration of security.

For the original PVD method [8], the secret data is assumed to be a long-bit stream and the cover image is a gray-level image. The embedding algorithm is described as follows:

1) Calculate the difference value d_i between two consecutive pixels p_i and p_{i+1} for each block in the cover image. The value is given by $d_i = p_{i+1} - p_i$.
2) Using $|d_i|$ to locate a suitable R_k in the designed range table, that is to compute $j = \min_k(u_k - |d_i|)$ where $u_k \geq |d_i|$ for all $1 \leq k \leq n$. Then R_j is the located range.
3) Compute the amount of secret data bits t that can be embedded in each pair of two consecutive pixels by R_j. The value t can be estimated from the width w_j of R_j, this can be defined by $t = \lfloor \log_2 w_j \rfloor$.
4) Read t bits from the binary secret data and transform the bit sequence into a decimal value b. For instance, if $\text{bit sequence} = 110$, then the converted value $b = 6$.
5) Calculate the new difference value d_i' given by $d_i' = l_j + b, \text{ if } d_i \geq 0$ or $d_i' = -(l_j + b), \text{ if } d_i < 0$ to replace the original difference d_i.
6) Modify the values of p_i and p_{i+1} by the following formula:
$(p_i', p_{i+1}') = (p_i - \lceil m/2 \rceil, p_{i+1} + \lfloor m/2 \rfloor)$
where $m = d_i' - d_i$. Until now, to embed the secret data into the pixel pair (p_i', p_{i+1}') is done by changing the values of p_i and p_{i+1}.

Repeat Step 1-6 until all secret data are embedded into the cover image, then the stego-image is obtained.

During the phase of secret extraction, the original designed range table is required. In the beginning, the same method in the embedding phase is used to partition the stego-image into pixel pairs (blocks). Then the difference value $\hat{d}_i$ for each pair of two consecutive pixels $\hat{p}_i$ and $\hat{p}_{i+1}$ in the stego-image is calculated. Next, $|\hat{d}_i|$ is used to locate the suitable R_j in Step 2 during the embedding phase. Therefore, $\hat{b}$ is obtained by subtracting l_j from $|\hat{d}_i|$. If the stego-image is not altered, $\hat{b}$ is equal to b. Finally, $\hat{b}$ is transformed from a decimal value into a binary sequence with t bits, where $t = \lfloor \log_2 w_j \rfloor$.

III. The Proposed Method

In the PVD method, two horizontal and consecutive pixels can only represent a vertical edge, but the edge can have different directions. This motivates us to improve the PVD method by considering three directions.

A. The Partition Pre-procedure

In general, the edges in an image are roughly classified into vertical, horizontal, and two kinds of diagonal directions. Motivated from the PVD method, using two-pixel pairs on one directional edge can work efficiently for information hiding. This should accomplish more efficiency while considering four directions from four two-pixel pairs. This can be implemented by dividing the image into 2×2 blocks and one example block is shown in Fig. 1. However, since the changing of pixel values for the fourth pixel pair affects the first and the second pairs, the fourth pair is useless and has to be discarded. Therefore, we propose that three pairs are used to embed the secret data. Before introducing the proposed algorithm, the pre-procedure is to partition the cover image into non-overlapping 2×2 blocks with 4 pixels.

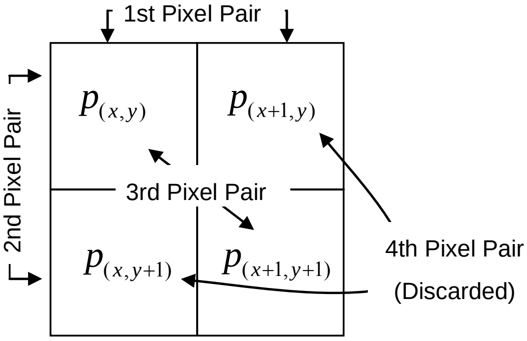

Figure 1. An example of four pixel pairs.

B. The Tri-way Differencing Scheme

As shown in Fig. 1, each 2×2 block includes four pixels of $p_{(x,y)}$ $p_{(x+1,y)}$ $p_{(x,y+1)}$ and $p_{(x+1,y+1)}$ where x and y are the pixel location in the image. Let $p_{(x,y)}$ be the starting point, then three pixel pairs can be found by grouping $p_{(x,y)}$ with the right, the lower, and the lower-right neighboring pixels. Those three pairs are named by P_0, P_1, and P_2 where $P_0=(p_{(x,y)},p_{(x+1,y)})$, $P_1=(p_{(x,y)},p_{(x,y+1)})$, and $P_2=(p_{(x,y)},p_{(x+1,y+1)})$, respectively.

When using the proposed tri-way PVD method to embed the secret data, each pair has it's modified P_i' and a new difference value d_i' for $i=0,1,2$. Here, the detailed embedding algorithm is left to be described in Section 3.F. Now, the new pixel values in each pair are different from their original ones. That is, we have three different values for the starting point $p_{(x,y)}$ named $p'_{0(x,y)}$, $p'_{1(x,y)}$, and $p'_{2(x,y)}$ from P_0, P_1, and P_2, respectively. However, only one value for $p'_{i(x,y)}$ can exist after finishing the embedding procedures. Therefore, one of $p'_{i(x,y)}$ is selected as the reference point to offset the other two pixel values. That is, two pixel values of one pair are used to adjust the other two pairs and construct a new 2×2 block. Suppose that the reference point is $p'_{1(x,y)}$, then the other two difference values, d_0' and d_2', can be proven unchanged after the adjustment given by

$$\begin{aligned}
d'_{0,(x,y)} &= p'_{0(x+1,y)} - p'_{0(x,y)}\\
&= p'_{0(x+1,y)} - p'_{0(x,y)} + (p'_{1(x,y)} - p'_{1(x,y)})\\
&= (p'_{0(x+1,y)} - p'_{1(x,y)}) - (p'_{0(x,y)} - p'_{1(x,y)})\\
d'_{2,(x,y)} &= p'_{2(x+1,y+1)} - p'_{2(x,y)}\\
&= p'_{2(x+1,y+1)} - p'_{2(x,y)} + (p'_{1(x,y)} - p'_{1(x,y)})\\
&= (p'_{2(x+1,y+1)} - p'_{1(x,y)}) - (p'_{2(x,y)} - p'_{1(x,y)})
\end{aligned} \tag{1}$$

Note that the embedded secret data are unaffected because of those three difference values are unaltered.

C. Optimal Selection Rules for the Reference Point

Selecting different reference points results in varied distortion to the stego-image. Here, we propose an optimal selection approach to achieve minimum Mean-Square-Error (MSE). Suppose that $m_i=d_i'-d_i$, d_i and d_i' are the difference values of pixel pair i before and after embedding procedures. The rules that can exactly determine one optimal reference pair without really estimating MSE are introduced as follows.

1) If all values of m_i are great than 1 or smaller than -1, the optimal pixel pair $i_{optimal}$ is the pair with the greatest $|m|$. For example, if $m_i=\{-8,-4,-3\}$, $i\in\{0,1,2\}$, then $i_{optimal}=0$.
2) If all m_i have the same sign and only one $m_i\in\{0,1,-1\}$, then the optimal pixel pair $i_{optimal}$ is selected from the other two pairs with the smallest $|m|$. For example, if $m_i=\{4,3,1\}, i\in\{0,1,2\}$, then $i_{optimal}=1$.
3) If only one m_i has a different sign from the other two pairs, the optimal pixel pair $i_{optimal}$ is selected from the other two pairs with the smallest $|m|$. For example, if $m_i=\{7,-4,3\}, i\in\{0,1,2\}$, then $i_{optimal}=2$.
4) If only one $m_i\in\{0,1,-1\}$ and the other two m_i has different signs, the optimal pixel pair $i_{optimal}$ is the pair with $m_i\in\{0,1,-1\}$. For example, if $m_i=\{0,-4,2\}, i\in\{0,1,2\}$, then $i_{optimal}=0$.
5) If there exists more than one pair with $m_i\in\{0,1,-1\}$, the optimal pixel pair $i_{optimal}$ can be selected as any one pair with $m_i\in\{0,1,-1\}$. For example, if $m_i=\{4,0,0\}$, $i\in\{0,1,2\}$ then $i_{optimal}=1$ or 2.

By following those selection rules described above, we can skip the calculation steps of MSE estimation to obtain the optimal reference pairs. Thus, the total computational complexity can be greatly reduced.

D. Adaptive Rules to Reduce Distortion

Although the proposed approach is feasible for embedding secret data, embedding large amount of bits can still cause serious image distortion easily. Since most distortion is generated from the offsetting process, the following two conditions are further designed to avoid too much offset described by

1) $embed\_bit(P_0)\ge 5$ and $embed\_bit(P_1)\ge 4$
2) $embed\_bit(P_0)< 5$ and $embed\_bit(P_2)\ge 6$

where $embed\_bit(P_i)$ represents the total embedding bits along the direction of P_i. If either one of above two conditions is satisfied, the current block being processed can probably result in higher distortion. Then we use two pixel pairs, P_0 and $P_3 = (p_{(x,y+1)}, p_{(x+1,y+1)})$, and adopt the original PVD method to individually process those two pairs along one direction. Here, we name those two conditions as “branch conditions”.

E. Theoretical Estimation

Suppose that $t_{\min}$ is the minimum amount of embedded bits, and $t_i \geq t_{\min}$. By considering the patterns and directional edges of a 2×2 pixel block, as shown in Table I, each block has 2 patterns for the smooth area and 14 patterns for the edge area. There are 2 patterns for the vertical edges, 2 patterns for the horizontal edges, and 10 patterns for two kinds of diagonal edges, respectively. To compare the capacity performance achieved by the PVD method and our approach, one sample block is adopted using $t_{block} = \sum t_i$ to analyze the capacity as shown in Table I.

TABLE I
CAPACITY COMPARISON IN 2×2 PIXEL BLOCKS FOR PVD METHOD AND OUR METHOD

Pattern	Classification	PVD method	Our method
□□ ■■ □□ ■■	Smooth area	$2t_{\min}$	$3t_{\min}$
■□ □■ ■□ □■	Vertical edge	$2t_i$	$2t_i + t_{\min}$
■■ □□ □□ ■■	Horizontal edge	$2t_{\min}$	$2t_i + t_{\min}$
■□ □■	Top-right diagonal	$2t_i$	$2t_i + t_{\min}$
■■ □□ □■ ■□	Top-right diagonal	$t_i + t_{\min}$	$t_i + 2t_{\min}$
■□ □■ ■■ □□	Top-right diagonal	$t_i + t_{\min}$	$t_i + 2t_{\min}$
□■ ■□	Bottom-left diagonal	$2t_i$	$t_i + 2t_{\min}$
□■ ■□ ■■ □□	Bottom-left diagonal	$t_i + t_{\min}$	$3t_i$
□□ ■■ □■ ■□	Bottom-left diagonal	$t_i + t_{\min}$	$t_i + 2t_{\min}$

As shown in Table I, our approach can achieve better performance than the PVD method in capacity, only the pattern in the first case of bottom-left diagonal is an exception. Since one-pixel-wide edges have less chance to appear than the smooth areas for natural images, the image degradation resulted by this case can be neglected. In fact, using our proposed branch conditions can also decrease the appearance of one-pixel-wide edges. Although it is uncertain that the patterns fulfilling designed branch conditions are all one-pixel-wide edge patterns, branch conditions still play an important role at balancing between a larger capacity on 2×2 pixel blocks and the reduction of image distortion.

F. The Embedding Algorithm

The details of data hiding steps are described as follows.

1) Calculate four difference values $d_{i,(x,y)}$ for four pixel pairs in each block given by

$$\begin{aligned} d_{0,(x,y)} &= p_{(x+1,y)} - p_{(x,y)} \\ d_{1,(x,y)} &= p_{(x,y+1)} - p_{(x,y)} \\ d_{2,(x,y)} &= p_{(x+1,y+1)} - p_{(x,y)} \\ d_{3,(x,y)} &= p_{(x+1,y+1)} - p_{(x,y+1)} \end{aligned} \qquad (2)$$

2) Using $\left|d_{i,(x,y)}\right|$ ($i = 0,\ldots,3$) to locate a suitable $R_{k,i}$ in the designed range table, that is to compute $j = \min_k (u_{k,i} - \left|d_{i,(x,y)}\right|)$ where $u_{k,i} \geq \left|d_{i,(x,y)}\right|$ for all $1 \leq k \leq n$. Then the located range can be represented by $R_{j,i}$.

3) Compute the amount of secret data bits t_i that can be embedded in each pair by $R_{j,i}$. The value t_i can be estimated from the width $w_{j,i}$ of $R_{j,i}$, this can be defined by $t = \lfloor \log_2 w_{j,i} \rfloor$.

4) If t_i of P_i ($i = 0,1,2$) satisfies branch conditions, two pixel pairs of P_0 and P_3 are processed by the original PVD method. Otherwise, the proposed tri-way scheme is used to process P_i ($i = 0,1,2$).

5) Read t_i bits from the binary secret data and transform the bit sequence into a decimal value b_i.

6) Calculate the new difference value $d'_{i,(x,y)}$ given by $d'_i = l_{j,i} + b_i$, if $d_{i,(x,y)} \geq 0$ or $d'_i = -(l_{j,i} + b_i)$, if $d_{i,(x,y)} < 0$ to replace the original difference $d_{i,(x,y)}$.

7) Modify the values of p_n and p_{n+1} by the following formula:
$(p'_n, p'_{n+1}) = (p_n - \lceil m/2 \rceil, p_{n+1} + \lfloor m/2 \rfloor)$
where p_n and p_{n+1} represent two pixels in P_i and $m = d_n{}' - d_n$. Until now, to embed the secret data into the pixel pair (p'_n, p'_{n+1}) is done by changing the values of p_n and p_{n+1}. If t_i ($i = 0,1,2$) do not satisfy branch conditions, then the algorithm proceeds to Step 8. Otherwise, Step 7 only processes P'_0 and P'_3 using the original PVD method. Then compute t'_i ($i = 0,1,2$) and check that whether t'_i still satisfy the same branch condition before. If not, offsetting the pixel values in P'_3 to satisfy the previous conditions. Now, the new block is constructed and the algorithm proceeds to Step 9.

8) Using the selection rules to choose the optimal reference point $p'_{i(x,y)}$ with minimum MSE, then this selected point is used to offset the other two pixel pairs.

9) Now, the new block constructed from all pixel pairs and embedded with secret data is generated.

An illustration of the data embedding process is shown in Fig. 2. In Fig. 2, suppose that the sample block is comprised by $(p_{(x,y)}, p_{(x+1,y)}, p_{(x,y+1)}, p_{(x+1,y+1)})$ and the gray values are $(100,126,115,107)$. At first, we set the pixel values of four pairs to be $P_0 = (100,126)$, $P_1 = (100,115)$, $P_2 = (100,107)$, and $P_3 = (115,107)$, respectively. Then, difference values calculated from four pairs are 26, 15, 7, and -8, respectively. After using their absolute values to locate suitable $R_{k,i}$ from the designed range table separately, we can obtain that $R_{2,0} = [16,31]$, $R_{1,1} = [8,15]$, $R_{0,2} = [0,7]$, and $R_{1,3} = [8,15]$. That is, we can compute that $w_{2,0} = 16$, $w_{1,1} = 8$, $w_{0,2} = 8$, and $w_{1,3} = 8$. The amount of bits can be embedded into each of four pairs are $t_0 = 4$, $t_1 = 3$, $t_2 = 3$, and $t_3 = 3$, respectively. By considering the branch conditions in this block, pixel pair P_3 is discarded. Suppose that the binary secret data to be embedded is 11010101001101. Therefore, the individual binary bit streams of P_i, $i \in \{0,1,2\}$ are 1101, 010, and 100. The corresponding decimal values of those bit streams are 13, 2, and 4, respectively. By following the definition of $d_i' = l_{j,i} + b_i$, the new difference values of those three pairs are computed as $d_0' = 16 + 13 = 29$, $d_1' = 8 + 2 = 10$, and $d_2' = 0 + 4 = 4$, respectively. Based on Step 7, we can obtain the new gray values of three pixel pairs, where $P_0' = (98,127)$, $P_1' = (103,113)$, and $P_2' = (102,106)$. Then $m_0 = 3$, $m_1 = -5$, and $m_2 = -3$ can be computed. Furthermore, since this situation satisfies rule 3 of the optimal selection approach for the reference point, the optimal reference pair is 2. Based on $p'_{2(x,y)}$ of P_2', the pixel values in another two pixel pairs are offset, until $p'_{i(x,y)}$ is equal to $p'_{2,(x,y)}$. Now, the embedded block is obtained and given by $(102,131,112,106)$.

G. The Extraction Algorithm

To retrieve the embedded secret data from the stego-image, the extraction algorithm is described in the following steps.

1) Partition the stego-image into 2×2 pixel blocks, and the partition order is the same as that in the embedding stage.
2) Calculate the difference values $\hat{d}_{i,(x,y)}$ separately for each block in the stego-image given by

$$\begin{aligned}\hat{d}_{0,(x,y)} &= p_{(x+1,y)} - p_{(x,y)}\\ \hat{d}_{1,(x,y)} &= p_{(x,y+1)} - p_{(x,y)}\\ \hat{d}_{2,(x,y)} &= p_{(x+1,y+1)} - p_{(x,y)}\\ \hat{d}_{3,(x,y)} &= p_{(x+1,y+1)} - p_{(x,y+1)}\end{aligned} \tag{3}$$

3) $\left|\hat{d}_{i,(x,y)}\right|$ is used to locate the suitable $R_{k,i}$ as introduced in Step 2 of the embedding phase. At the same time, the amount of embedding bits t_i, where $t_i = \lfloor \log_2 w_{j,i} \rfloor$ is obtained. If t_i satisfies the branch conditions, two independent pixel pairs are selected; otherwise, three pixel pairs are used for further processing.
4) After $R_{k,i}$ is located, $l_{j,i}$ is subtracted from the selected $\left|\hat{d}_{i,(x,y)}\right|$ and $\hat{b}_i$ is obtained. If the stego-

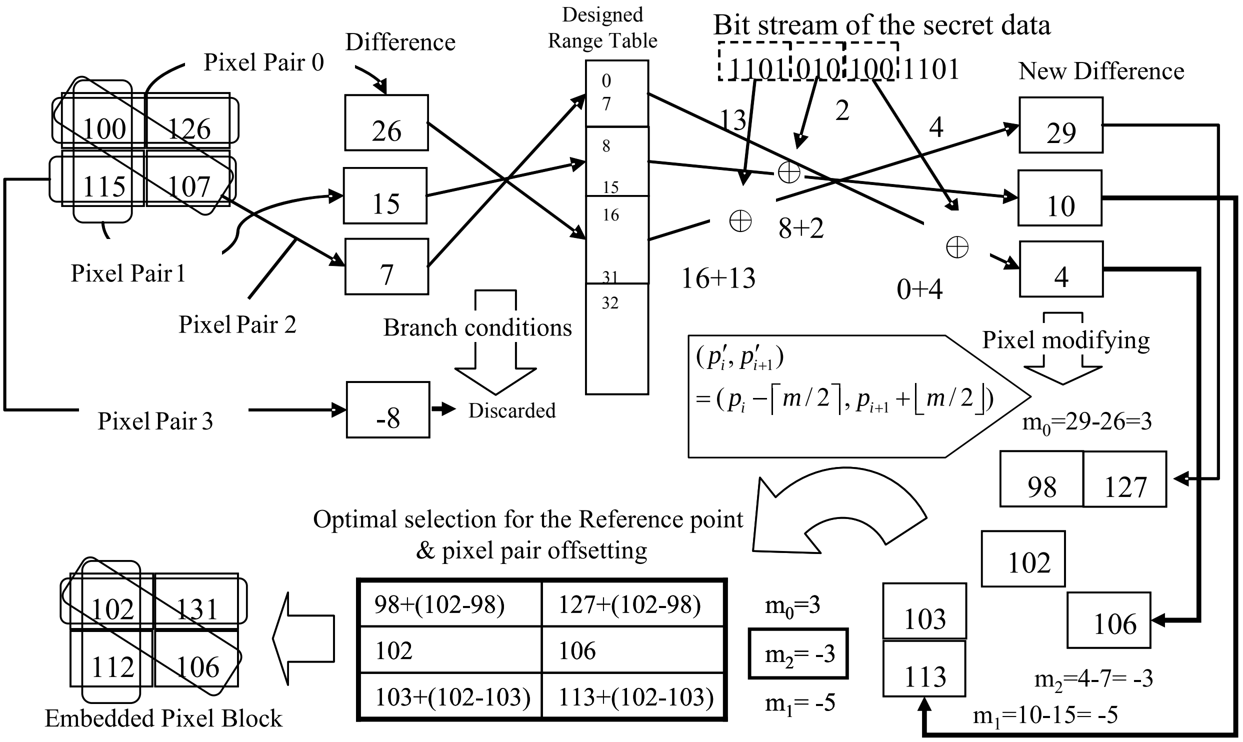

Figure 2. An illustration of the data embedding process.

image is not altered, $\hat{b}_i$ is equal to b_i. Finally, $\hat{b}_i$ is converted from a decimal value into a binary sequence with t_i bits where $t_i = \lfloor \log_2 w_{j,i} \rfloor$. Note that the t_i-bit stream is only one part of the secret data before embedding.

IV. Experimental Results

To demonstrate the accomplished performance of our proposed approach in capacity and security for hiding secret data in the stego-image, apart from the results of theoretical estimation listed in Table I, we have also conducted different experiments using four images to compare the proposed approach with the PVD method. According to the invisibility benchmark for the watermarked images [11], a minimum peak signal-to-noise ratio (PSNR) value of 38 dB is adopted as the quality requirement for the stego-images in the experiments.

A. Capacity and PSNR

The secret binary data sequence S is generated by pseudo-random numbers. We set the designed range table with the width in the set of $w_k \in \{8,8,16,32,64,128\}$. The size of all cover images is 512×512. Here, PSNR value is utilized to evaluate the invisibility of the stego-images. Some examples of the cover image and its stego-image are shown in Fig. 3. To compare the proposed approach with the PVD method, Table II lists the experimental results after the secret data is embedded using those two approaches. The hiding capacity (in bytes) and PSNR values achieved by the proposed scheme and the PVD method for four images are shown. The listed values are the average results after embedding 100 randomly generated bit-sequences into the cover images. From Table II, the accomplished capacity ratio of our method to the PVD approach is near 1.5 in average. Although the PSNR value is smaller than 38 for both schemes after embedding the data into the Baboon image, two stego-images are still hardly observed that the secret data is hidden inside. This is because of the high variance existed in the pixel values of the Baboon image. Therefore, this demonstrates that the proposed approach can provide a promising performance in increasing the capacity of the stego-images and maintaining the imperceptible quality simultaneously.

TABLE II
COMPARISON OF RESULTS FOR THE PROPOSED METHOD AND PVD METHOD

	PVD Method		Our Method	
Cover images	Capacity	PSNR	Capacity	PSNR
(512×512)	(bytes)	(dB)	(bytes)	(dB)
Lena	50960	41.79	75836	38.89
Baboon	56291	37.90	82407	33.93
Jet	51243	40.97	76352	38.70
Peppers	50685	40.97	75579	38.50

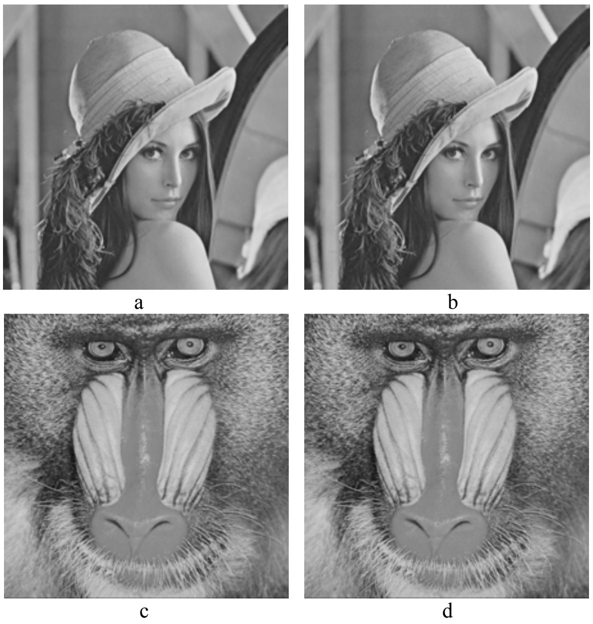

Figure 3. Cover image and stego-image. (a) Cover image: Lena (b) Stego-image after embedding secret data with 75836 btyes; PSNR is 38.8 (c) Cover image: Baboon (d) Stego-image after embedding secret data with 82407 btyes; PSNR is 33.9

B. Security Verification using RS Stego-analysis

After the process of data hiding into cover images, the main aim is to make stego-images perceptually similar to the original cover image as much as possible. The techniques, named stego-analysis, are normally developed to check and detect whether an image is a stego-image or not. The dual statistical stego-analytic technique proposed by Fridrich *et al.* [12] can successfully detect stego-images using LSB substitution. Therefore, in this paper, this newly announced approach of statistical steg-analysis, RS-diagram, is used to test the stego-images embedded with the secret data by our method. Using this method, the test image is divided into groups of n consecutive or disjoint pixels first. Suppose that one pixel group is $G = (x_1, x_2, \ldots, x_n)$. Through a discrimination function f, the smoothness or regularity of each group can be computed and the function f is defined as $f(x_1, x_2, \ldots, x_n) = \sum_{u=1}^{n-1} |x_{i+1} - x_i|$ in general.

Three types of pixel groups are defined by calculating from the function and an invertible function F. The function F, also called flipping function, has to match the property that $F^2(x)$ is identity or $F(F(x)) = x$. The definitions of three pixel types are described as follows:

Regular groups: $G \in R \Leftrightarrow f(F(G)) > f(G)$,

Singular groups: $G \in S \Leftrightarrow f(F(G)) < f(G)$,

Unusable groups: $G \in U \Leftrightarrow f(F(G)) = f(G)$.

Consequently, using two complement masks can simulate the joining of different noises and the masks are n-tuples with values of -1, 0, and 1, for instance $M = [0\ 1\ 1\ 0]$ and $-M = [0\ \text{-1}\ \text{-1}\ 0]$. The results from accumulating amounts of Regular and Singular groups in the entire image and from computing the percent of both two pixel groups are drawn to the diagrams. The diagrams with relation to image pixels in which secret data are embedded and relative numbers of regular and singular pixel groups are referred to as RS-diagrams. For natural images, the percent of regular groups and singular groups generally follow the relation that $R_M + S_M \le 1$, $R_{-M} + S_{-M} \le 1$, and $(R_M \cong R_{-M}) > (S_M \cong S_{-M})$, where R_M and S_M are the percentages with the mask M of the two pixel group, regular and singular, respectively. When there are more LSBs replaced with random data, the percentages of R_M and S_M are going to gradually approximate to a same value. On the other hand, the difference of percentages R_{-M} and S_{-M} will be increased in the statistical analysis process. After analyzing the experimental results, the variation trends of the percentages R_M and S_M are approximately parabolic, and the curves of R_{-M} and S_{-M} in the RS-diagrams are nearly linear. From Fridrich et al. [12], the following quadratic equation is acquired with the initial conjecture point p based on the assumption that the curves of R_M and S_M intersect at 50 percent with the relationship of $R_M(1/2) = S_M(1/2)$:

$$2(d_1 + d_0)x^2 + (d_{-0} - d_{-1} - d_1 - 3d_0)x + d_0 - d_{-0} = 0,$$

where

$$d_0 = R_M(p/2) - S_M(p/2),$$
$$d_{-0} = R_{-M}(p/2) - S_{-M}(p/2),$$
$$d_1 = R_M(1 - p/2) - S_M(1 - p/2),$$
$$d_{-1} = R_{-M}(1 - p/2) - S_{-M}(1 - p/2).$$

From the root x whose absolute value is the smaller one of the quadratic equation, predictive secret message length P can be estimated by

$$P = x/(x - 1/2)$$

where the value range of P is 0~100%. If $P \le 0$, it means no embedded message exists in the image.

Due to the influence that the initial conjecture point p affects the accuracy of the secret message length estimation, we use continuous and different initial conjecture points to calculate the length estimation P and try to get a fittest result to use on later experiments. As shown in Table III, the fittest result is 38% on initial conjecture point selection. Therefore, this fittest result is adopted to estimate the secret message length for various images with various embedding ratios. The estimated results displayed in Table IV are approximately close to the real message length of each image, except the baboon image. This is because of that more details information exists in the baboon image, the initial bias of RS stego-analysis is dissimilar from other natural images. Thus, for the baboon image, the accuracy of secret message length estimation exhibits larger variation than the real message length. Nevertheless, the usability of fittest result has been confirmed by the experimental data of Table IV. Based on the selection of fittest initial conjecture point, the estimation results for the message length of various images using our proposed method with constant secret message capacity are shown in Table V. Although the estimation of message length for the Peppers image shows that tiny percent of secret message is embedded in the image, the value is affected by the initial bias locating in the range of standard deviation of inaccuracy. Based on the experimental results described above, it is proved that our proposed method can avoid the detection of stego-images and survive from dual statistics stego-analysis.

TABLE III
RESULTS OF THE REAL LSB MESSAGE LENGTH AND LENGTH ESTIMATION BASED ON RS STEGO-ANALYSIS USING CONTINUOUS DIFFERENT INITIAL CONJECTURE POINTS FOR LENA IMAGE.

	Real LSB message length		
Initial conjecture point (%)	25.98%	45.08%	74.42%
20	29.9	47.6	94.2
22	31.1	47.3	91.2
24	31.7	48.3	88.0
26	30.8	47.9	85.6
28	31.2	49.8	83.8
30	30.1	49.3	83.1
32	29.4	46.2	80.3
34	28.5	46.0	77.7
36	26.3	44.1	76.6
38	26.0	44.0	74.2
40	25.5	42.3	72.6

TABLE IV
RESULTS OF THE REAL LSB SECRET MESSAGE LENGTH AND LENGTH ESTIMATION BASED ON RS STEGO-ANALYSIS USING FITTEST INITIAL CONJECTURE POINTS .

Real LSB Message length	RS estimation			
	Lena	F-16	Peppers	Baboon
0.00%	<0%	<0%	2.24%	2.32%
10.35%	8.95%	14.18%	27.91%	29.01%
30.87%	28.12%	33.81%	31.92%	54.65%
62.16%	54.61%	66.11%	70.80%	89.66%
93.01%	100.00%	100.00%	100.00%	100.00%

TABLE V
RESULTS OF THE REAL SECRET MESSAGE LENGTH EMBEDDED BY THE PROPOSED METHOD AND LENGTH ESTIMATION BASED ON RS STEGO-ANALYSIS USING FITTEST INITIAL CONJECTURE POINTS

Capacity	Lena		F-16	
(Bytes)	Real length	RS estimation	Real length	RS estimation
10,175	13.86%	<0%	13.71%	<0%
30,346	40.54%	<0%	40.83%	<0%
61,108	80.50%	<0%	79.18%	1.89%
73,258	96.49%	0.40%	94.98%	<0%

Capacity	Peppers		Baboon	
(Bytes)	Real length	RS estimation	Real length	RS estimation
10,175	13.34%	<0%	10.32%	<0%
30,346	39.87%	0.76%	31.46%	4.00%
61,108	80.54%	6.62%	67.84%	<0%
73,258	96.68%	1.45%	82.37%	<0%

V. CONCLUSIONS

Using three different directional edges can hide more secret data into the cover image than the PVD method. Also, we have presented an optimal selection approach for the reference point with adaptive rules to reduce the quality distortion of the stego-image. Experiment results demonstrate that the secret data embedded in the stego-image is imperceptible for human vision while compared with the cover image. Furthermore, the proposed approach can achieve superior embedding capacity than the PVD method, not only from theoretical estimation, but also from the experimental results. After the detection test for the embedded data by the proposed approach, the approach demonstrates the robustness to avoid the data detection and survive from dual statistical stego-analysis. Also, the extraction of the embedded secret data can work correctly from stego-images without the participation of original cover images. This has shown multiple merits of the proposed technique for data hiding.

REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, “Information Hiding - a Survey,” Proceedings of the IEEE, Vol. 87, pp. 1062–1078, 1999.

[2] W. Bender, D. Gruhl, N. Morimoto, A. Lu, “Techniques for data hiding,” IBM Systems Journal Vol. 35 (3–4), pp. 313–336, 1996.

[3] Y. K. Lee, L. H. Chen, “High capacity image steganographic model,” IEE Proceedings on Vision, Image and Signal Processing, Vol. 147, No.3, pp. 288-294, 2000.

[4] R.-Z. Wang, C.-F. Lin, and J.-C. Lin, “Image hiding by optimal LSB substitution and genetic algorithm,” Pattern Recognition Vol. 34, pp. 671–683, 2001.

[5] C.-C. Chang, J.-Y. Hsiao, C.-S. Chan, “Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy,” Pattern Recognition Vol. 36, Issue 7, pp. 1583-1595, 2003.

[6] C.-K. Chan, L. M. Cheng, “Hiding data in images by simple LSB substitution,” Pattern Recognition Vol. 37, Issue 3, pp. 469-474, 2004.

[7] W.-N. Lie, and L.-C. Chang, “Data hiding in images with adaptive numbers of least significant bits based on the human visual system,” IEEE International Conference on Image Processing, Vol. 1, pp. 286–290, 1999.

[8] D.-C. Wu, and W.-H. Tsai, “A steganographic method for images by pixel-value differencing,” Pattern Recognition Letters, Vol. 24, pp. 1613–1626, 2003.

[9] H.-C. Wu, N.-I. Wu, C.-S. Tsai, and 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, 2005.

[10] S.-L. Li, K.-C. Leung, L.-M. Cheng, and C.-K. Chan, “Data Hiding in Images by Adaptive LSB Substitution Based on the Pixel-Value Differencing,” First International Conference on Innovative Computing, Information and Control (ICICIC'06), Vol. 3, pp. 58-61, 2006.

[11] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su, “Attacks on digital watermarks: classification, estimation based attacks, and benchmarks,” IEEE Communications Magazine, Vol. 39, Issue 8, 118-126, 2001.

[12] J. Fridrich, M. Goljan, and R. Du, “Detecting LSB steganography in color, and gray-scale images,” IEEE multimedia, Vol. 8, Issue 4, pp. 22-28, 2001.

Ko-Chin Chang received the M.S. degrees from Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University, Taiwan, in 2003, and is currently pursuing the Ph.D. degree in the Department of Electrical and Electronic Engineering, National Defense University, Taiwan. His research interests include image and signal processing, data hiding, and image compression. His recent research has focused on high capacity image steganography.

Dr Chien-Ping Chang received the BS degree in Electrical Engineering from Chung Cheng Institute of Technology in 1986, and the PhD degree in Computer and Information Science from National Chiao Tung University, Taiwan, Republic of China, in 1998. He is currently an associate professor in the Department of Electrical and Electronic, Chung Cheng Institute of Technology, Taiwan, Republic of China. His research interests include parallel computing, interconnection networks, graph theory, image processing, and data hiding.

Prof. Ping Sheng Huang received his BSEE degree from Chung Cheng Institute of Technology in 1985, the MS degree in Computer Science from University of Southern California in 1990, and the PhD degree in Electronics and Computer Science from University of Southampton, United Kingdom, in 1999. After that, he was with the Department of Electrical Engineering, Chung Cheng Institute of Technology as an associate professor, then a professor until July 2007. Currently he is a professor in the Department of Electronic Engineering, Ming Chuan University, Taiwan. His research interests include digital image processing, biometric identification, and statistical pattern recognition.

Prof. Te-Ming Tu received the BSEE degree from Chung Cheng Institute of Technology in 1986, the MSEE degree from National Sun Yat-Sen University in 1991, and the PhD degree in electrical engineering from the National Cheng Kung University, in 1996. Since 1981, he has served in the R.O.C. Army. He was a teaching assistant from 1986 to 1989, an instructor from 1991 to 1993, and is currently a professor in the Department of Electrical and Electronic, Chung Cheng Institute of Technology. His current research interests include multispectral/hyperspectral remote sensing, medical imaging, independent component analysis, and statistical pattern recognition.