Improved Steganographic Technique for the Image Quality of PVD

Min-Yen Chiu[#1], Yu-Sheng Liao[#2], Jiun-Jian Liaw[#3],

[#]*Department of Information and Communication Engineering,*
Chaoyang University of Technology,
No.168, Jifeng E. Rd., Wufeng Township, Taichung County 413, Taiwan

[1]s9530614@cyut.edu.tw

[3]jjliaw@cyut.edu.tw

***Abstract*— Hiding a lot of data into cover-image causes serious distortion. Therefore, hiding capacity is limited by maintaining image quality. A method is proposed to solve this problem in this paper. Firstly, we divide cover-image into many non-overlapping blocks. Each block consists of two contiguous pixels. The hiding bits are transformed into decimal format and are separated into two parts. Then, modulus operation is used to hide two parts into the first pixel and the second pixel of a block, respectively. Experiment results prove that the proposed method has higher image quality than pervious literature which is based on pixel value differencing.**

***Keywords*— Steganography; Pixel-Value Differencing; Modulus Operation**

1. INTRODUCTION

As the popularity of the computers and the Internet, digital multimedia can be quickly transmitted over the Internet. Copying easily and re-producing unlimitedly are the main advantages of digital data. In recent year, this convenient technique is replacing traditional communication ways. When digital data is published over the Internet, viewer can copy and edit them easily. It produces copyright and data security problems. Therefore, how to protect the digital data becomes more important day by day.

Steganography is one of the popular methods to solve these problems. Steganography is a kind of technique to conceal the secret data into the meaningful multimedia data (as text, image, video, or audio files [1]–[3]), so-called cover-data. The stego-data is the cover-data with the secret data concealed in. Steganographic techniques make the stego-data look like a common multimedia data so that it can avoid the observers' attention. However, steganographic techniques always involve the trade-off between robustness and imperceptibility. In this paper, the robustness is not considered.

There are many steganographic literatures about concealing data in images proposed. The Least Significant Bits (LSBs) substitution is the most well-known steganographic method. It replaces the least bits of pixels in the cover-image with the secret data bits [4]–[6]. Therefore, it is a simple method and the stego-image still has good quality. Some varieties of LSBs were proposed to improve the security and the quality of the stego-image, such as the exhaustive method [7] and the dynamic programming strategy scheme [8]. But the quality of stego-image decreases extremely when the concealing bit of each pixel is equal to or more than four bits. Thien and Lin [9] proposed another steganographic method which uses Modulus Functions (MF) to get better image quality than LSBs.

However, an image is composed of smooth areas and edge areas. According to Human Visual System (HVS), the tolerances of these two areas are different. Human eyes can easily sense the changes in smooth areas. On the contrary, edge areas have higher tolerance than smooth areas. Some studies started considering the property of HVS to hide different number of bits in a pixel [10]–[11]. Chang and Tseng [12] use the side information of upper pixel and left pixel to estimate the number of bits which can be hidden in an image. Wu and Tsai [13] proposed a stegnographic method which is based on Pixel-Value Differencing (PVD). They estimate the location of pixel by calculating the difference of the contiguous pixels. The embedding bits of each two pixel are depended on their difference.

In this paper, we propose a novel steganographic method by separating secret data

and modulus operation. The experimental results show that the proposed method has higher quality than PVD.

2. RELATED WORK

In this section, we review Thien and Lin's Modulus Functions (MF), and Wu and Tsai's Pixel-Value Differencing (PVD).

2.1. MF

Thien and Lin [9] proposed a steganographic method based on Modulus Function (MF). They use modulus operation to hide secret data in the embedding process. Thus they can reduce the distortion while embedding the secret data and raise the image quality. Firstly, they use the following equation to calculate the difference of z and x mod y:

$$w = z - (x \bmod y) \tag{1}$$

Among this equation, x is the pixel value used to conceal secret data, y is the modulus value, z is a decimal secret data, and w is the difference of z and x mod y. After that, they use the following equation to calculate the smallest changing value (c) of x:

$$c = \begin{cases} w & , \text{if } (-\left\lfloor \frac{y-1}{2} \right\rfloor) \le w \le \left\lceil \frac{y-1}{2} \right\rceil \\ w+y & , \text{if } (-y+1) \le w < (-\left\lfloor \frac{y-1}{2} \right\rfloor) \\ w-y & , \text{if } \left\lceil \frac{y-1}{2} \right\rceil < w \le y-1 \end{cases} \tag{2}$$

Finally, use the following equation to get the hidden value x':

$$x' = x + c \tag{3}$$

The embedding process is completed by the above formula. In extraction process, the hidden secret data can be extracted by the following equation:

$$z = x' \bmod y \tag{4}$$

In the subsequent chapter, we use $MODS(x, y, z)$ to represent the above three concealing equations, Eq.(1) to Eq.(3).

2.2. PVD

Wu and Tsai [13] proposed a steganographic method for image by Pixel-Value Differencing (PVD). In this method, the cover-image is divided into many non-overlapping blocks which are composed of two contiguous pixels. Assume the values of these two contiguous pixels are f_i and f_{i+1}. The difference of these two pixels is e. Then e is computed as $|f_i - f_{i+1}|$. According to the component bit (a) of the cover-image, e must belong to $[0, 2^{\alpha} - 1]$. On the basis of the property of HVS, they conceal more secret data into a block whose difference is close to $2^{\alpha} - 1$. Otherwise, they conceal less secret data into a block whose difference is located within smooth area.

They divide the range of e, 0 to $2^{\alpha} - 1$, into q parts. Each part is denoted as R_j, j=1,2,…,q, as shown in Fig.1, so-called range table. Each block will conceal different bits of secret data based on R_j. Assume the highest and the lowest number of R_j is denoted as u_j and l_j, respectively. Then, they calculate the width (RS_j) of the hidden part to decide the number of concealing bits n, in which $n = \log_2(u_j - l_j + 1)$. They select n bits from the secret bit stream and transfer it to decimal value (b). The destination of PVD is to replace the original difference with a new difference in the same part of R_j. The new difference e' is $l_j + b$, and the concealed f_i' and f_{i+1}' is computed by the following formula:

$$(f_i', f_{i+1}') = \begin{cases} (f_i + \left\lceil \frac{m}{2} \right\rceil, f_{i+1} - \left\lfloor \frac{m}{2} \right\rfloor), \text{if } f_i \ge f_{i+1} \text{ and } e' > e \\ (f_i - \left\lfloor \frac{m}{2} \right\rfloor, f_{i+1} + \left\lceil \frac{m}{2} \right\rceil), \text{if } f_i < f_{i+1} \text{ and } e' > e \\ (f_i - \left\lceil \frac{m}{2} \right\rceil, f_{i+1} + \left\lfloor \frac{m}{2} \right\rfloor), \text{if } f_i \ge f_{i+1} \text{ and } e' \le e \\ (f_i + \left\lceil \frac{m}{2} \right\rceil, f_{i+1} - \left\lfloor \frac{m}{2} \right\rfloor), \text{if } f_i < f_{i+1} \text{ and } e' \le e \end{cases} \tag{5}$$

where $m = |e' - e|$. Repeating the above steps until all blocks of cover-image are concealed, and the stego-image is then produced.

In the extracting process, the PVD method firstly calculate the difference (e') of two pixel values in the stego-image, and $e' = |f_i' - f_{i+1}'|$. Finding the e' is located on which part of R_j. Then, they calculate the width (RS_j) of R_j to evaluate the component bit of the hidden secret data. The concealed decimal data b' is $e' - l_j$. The concealed data (b') is composed by n-bits, in

$R_1 \in [0,7]$
$R_2 \in [8,15]$
$R_3 \in [16,31]$
$R_4 \in [32,63]$
$R_5 \in [64,127]$
$R_6 \in [128,255]$

Fig. 1 Range table: the width of range is 8, 8, 16, 32, 64 and 128.

which $n = \log_2(RS_j)$. Obviously, people cannot extract embedded bit without accurate width of R_j.

3. THE PROPOSED METHOD

In this paper, we similarly propose a steganographic method based on pixel-value differencing. But in our method, we obtain better effect on the image quality than the previous method based on pixel-value differencing. We separate the secret data into two different parts. This process can reduce the distortion of embedding data. And modulus operation is used to conceal the secret data in separated parts. The proposed method divides range table into lower-level and higher-level. When the difference of block belongs to lower-level, the secret data is separated into two different parts. Otherwise, when the difference of block belongs to higher-level, the secret data is separated into two pairs. It is very different from the two parts of lower-level.

3.1. Embedding Process

Assume the cover-image used to embed data is a *a*–bits grayscale image. Cover-image will be divided into non-overlapping blocks which are composed of two contiguous pixels values. We use p_i and p_{i+1} to express these two pixels, respectively. The pixels of cover-image are scanned in the zig-zag order. After that, we calculate the difference of each two-pixel block, and $d = |p_i - p_{i+1}|$. Obviously, *d* must belong to $[0, 2^{\alpha} - 1]$ because cover-image is a *a*–bits gray image. We divide $[0, 2^{\alpha} - 1]$ into *q* sub-ranges. Each sub-range is denoted as R_j, and j=1,2,…,q. The result of divided range is so-called range table, as shown in Fig.1. The width of each sub-range (RS_j) is $u_j - l_j + 1$, in which u_j is the highest number of R_j, and l_j is the lowest number of R_j. RS_j is used to decide the amount of embedding bit *n*, and $n = \log_2(RS_j)$. Then, we select *n* bits from the secret data stream and transform it into a decimal value (*DEC*). After embedding secret data is known, we set a threshold (*Div*) to divide $[0, 2^{\alpha} - 1]$ into lower-level and higher-level. In accordance with lower-level and higher-level, we use different methods. If $d < Div$, we use 3.1.1 to hide the secret data. Otherwise, we use 3.1.2 to hide the secret data when $d \geq Div$.

3.1.1. Case 1

If $d < Div$, we divide *DEC* into two parts by following formula:

$$FN = \left\lfloor \frac{DEC}{n} \right\rfloor \qquad SN = DEC \bmod n \tag{6}$$

A modulus operation will be used to hide *FN* and *SN* into p_i and p_{i+1}, respectively. Therefore, we should set the modulus value which will be used in modulus operation. If $d < Div$, FN and SN are both less than *n*. The modulus value used to conceal in p_i and p_{i+1} is *n*. Then, we use the following equation to conceal *FN* and *SN* into p_i and p_{i+1}:

$$(\overline{p}_i, \overline{p}_{i+1}) = (MODS(p_i, n, FN), MODS(p_{i+1}, n, SN)) \tag{7}$$

In which, $\overline{p}_i$ and $\overline{p}_{i+1}$ should exceed the range of pixel value $[0, 2^{\alpha} - 1]$. When pixel exceeds the range, the following equation is used to keep pixel within the accurate range:

$$p_i' = \begin{cases} \overline{p}_i + RG_1 & ,\text{if } \overline{p}_i < 0 \\ \overline{p}_i - RG_1 & ,\text{if } \overline{p}_i > 2^a\text{-}1 \\ \overline{p}_i & ,\text{Otherwise} \end{cases}$$

$$p_{i+1}' = \begin{cases} \overline{p}_{i+1} + RG_2 & ,\text{if } \overline{p}_{i+1} < 0 \\ \overline{p}_{i+1} - RG_2 & ,\text{if } \overline{p}_{i+1} > 2^a\text{-}1 \\ \overline{p}_{i+1} & ,\text{Otherwise} \end{cases} \tag{8}$$

In above equation, RG_1=RG_2=n. After Eq.(8), p_i' and p_{i+1}' are able to exceed the original sub-range of *d*. Therefore, we should calculate the

new difference (d') between p_i' and p_{i+1}' by following equation:

$$d' = \left| p_i' - p_{i+1}' \right| \tag{9}$$

If d' belongs to original R_j, the final embedded pixels p_i''' and p_{i+1}''' are p_i' and p_{i+1}'. If not, we perform the following equation to calculate the smallest distance (*BD*) between d' and original R_j:

$$BD = \begin{cases} d' - u_j , & \text{if } d' > u_j \\ l_j - d' , & \text{if } d' < l_j \end{cases} \tag{10}$$

In order to observe the property of modulus operation, the changing number should be a multiple of modulus value *n* or be more than or equal to *BD*. Therefore, Eq.(11) is used to calculate the changing number of p_i' and p_{i+1}'.

$$\begin{aligned} CH_1 &= \left\lceil \frac{BD}{MG_1} \right\rceil \times MG_1 \\ CH_2 &= \left\lceil \frac{BD}{MG_2} \right\rceil \times MG_2 \end{aligned} \tag{11}$$

In above equation, $MG_1 = MG_2 = n$. Eq.(12) is used to distinguish that p_i' and p_{i+1}' should be added or be subtracted.

$$(p_i'', p_{i+1}'') = \begin{cases} (p_i' + CH_1, p_{i+1}' - CH_2), \\ \quad \text{if } d' < l_j \ \ \& \ \ p_i' > p_{i+1}' \\ (p_i' - CH_1, p_{i+1}' + CH_2), \\ \quad \text{if } d' < l_j \ \ \& \ \ p_i' \le p_{i+1}' \\ (p_i' - CH_1, p_{i+1}' + CH_2), \\ \quad \text{if } d' > u_j \ \ \& \ \ p_i' > p_{i+1}' \\ (p_i' + CH_1, p_{i+1}' - CH_2), \\ \quad \text{if } d' > u_j \ \ \& \ \ p_i' \le p_{i+1}' \end{cases} \tag{12}$$

Then, we make p_i''' and p_{i+1}''' to (p_i'', p_{i+1}') and (p_i', p_{i+1}''), and compare these two combinations with (p_i, p_{i+1}) by Eq.(13) and Eq.(14). The above steps can make us reduce the distortion as much as possible. Finally, we get the concealed pixels p_i''' and p_{i+1}'''.

For example, we assume (p_i, p_{i+1}) is (12,19) which belongs to R_1. We know the embedding bit is 3-bits. The secret data *DEC* is assumed to be 7. According to Eq.(6), we are aware that $FN = 2$ and $SN = 1$. Calculated by Eq.(7), Eq.(8), and

$$(p_i''', p_{i+1}''') = \begin{cases} (p_i'', p_{i+1}'), \\ \quad \text{if } p_i'' \in [0, 2^a\text{-}1] \,\&\, p_{i+1}'' \in [0, 2^a\text{-}1] \\ \quad \ \& \ MSE_i <= MSE_{i+1} \\ (p_i', p_{i+1}''), \\ \quad \text{if } p_i'' \in [0, 2^a\text{-}1] \,\&\, p_{i+1}'' \in [0, 2^a\text{-}1] \\ \quad \ \& \ MSE_i > MSE_{i+1} \\ (p_i', p_{i+1}''), \\ \quad \text{if } p_i'' \notin [0, 2^a\text{-}1] \,\&\, p_{i+1}'' \in [0, 2^a\text{-}1] \\ (p_i'', p_{i+1}'), \\ \quad \text{if } p_i'' \in [0, 2^a\text{-}1] \,\&\, p_{i+1}'' \notin [0, 2^a\text{-}1] \end{cases} \tag{13}$$

$$\begin{aligned} MSE_i &= (p_i'' - p_i)^2 + (p_{i+1}' - p_{i+1})^2 \\ MSE_{i+1} &= (p_i' - p_i)^2 + (p_{i+1}'' - p_{i+1})^2 \end{aligned} \tag{14}$$

Eq.(9), we know p_i' and p_{i+1}' is 11 and 19. New difference (d') is $\left| p_i' - p_{i+1}' \right| = \left| 11 - 19 \right| = 8$. The new difference exceeds the original R_1 to R_2. For that reason, we use Eq.(10) to calculate the smallest distance between d' and original R_1. And we can know that $BD = d' - u_j = 8 - 7 = 1$. According to Eq.(11) and Eq.(12), we know (p_i'', p_{i+1}'') is (14,16) that both belong to $[0, 255]$. Therefore, we should find the best change out. Based on Eq.(13) and Eq.(14), we get MSE_i is 4, and MSE_{i+1} is 10. Finally, the embedded pixel (p_i''', p_{i+1}''') is (14, 19).

3.1.2. Case 2

If $d \ge Div$, we divide the embedding secret data *DEC* into two parts by the following equation:

$$\begin{aligned} TD &= \left\lfloor \frac{DEC}{10} \right\rfloor \\ OD &= DEC \bmod 10 \end{aligned} \tag{15}$$

The same as 3.1.1, in order to conceal *TD* and *OD*, we should calculate the modulus value firstly. When $d \ge Div$, the following equation is used to calculate the modulus value *FM* and *SM* of p_i and p_{i+1}, respectively.

$$\begin{aligned} FM &= \left\lfloor \frac{RS}{10} \right\rfloor + 1 \\ SM &= 10 \end{aligned} \tag{16}$$

Then, the following equation uses modulus operation to conceal *TD* and *OD* into p_i and p_{i+1}, respectively.

$$(\bar{p}_1, \bar{p}_2) = (MODS(p_1, FM, TD), MODS(p_2, SM, OD)) \quad (17)$$

Using Eq.(8) ensures that p_i' and p_{i+1}' are both in the accurate range $\left[0, 2^{\alpha}-1\right]$, and $RG_1 = FM$, $RG_2 = SM$. According to Eq.(9), we can calculate d'. If d' and d are in the same range, the concealing process is finished. If not, Eq.(10) will be used again to get the smallest distance (BD) between d' and the original range of d. Using Eq.(11) and Eq.(12) gets the changing pixels p_i'' and p_{i+1}'', in which $MG_1 = FM$ and $MG_2 = SM$. Then, two combinations (p_i'', p_{i+1}') and (p_i', p_{i+1}'') are produced. Comparing these two combinations with (p_i, p_{i+1}) by Eq.(13) and Eq.(14), we can find which one of these two combinations cause less distortion. Finally, we get the embedded pixels p_i''' and p_{i+1}'''.

3.2. Extracting Process

In extracting process, we need the stego-image and accurate divided range table (as shown in Fig.1). In another words, the embedded data can not be extracted without accurate divided range talble. Initially, we partition stego-image into non-overlapping two pixel blocks as in the embedding process. Then, we calculate the difference (d'') of each two pixel block in the stego-image by the following equation:

$$d'' = \left| p_i''' - p_{i+1}''' \right| \quad (18)$$

According to the range table, we can find which part is d'' belongs to, and calculate the width of located R_j. The above processes are suitable for 3.2.1 and 3.2.2. In next two paragraphs, we detail the extracting processes of two cases, respectively.

3.2.1. Case 1

The modulus value of 3.1.1 is embedding bit (n), which is calculated by $n = \log_2(RS_j)$. And the embedded bits of p_i''' and p_{i+1}''' can be extracted by the following modulus operation:

$$FN' = p_i''' \bmod n \qquad SN' = p_{i+1}''' \bmod n \quad (19)$$

Finally, the embedded bits can be restructured by the following equation:

$$FN' \times n + SN' = DEC' \quad (20)$$

3.2.2. Case 2

According to the embedding process, we know the modulus value (SM') of p_{i+1}''' is a fixed number which is equal to 10. Depending on the width of RS_j, the modulus value of p_i''' is calculated by the following equation:

$$FM' = \left\lfloor \frac{RS_j}{10} \right\rfloor + 1 \quad (21)$$

Then, the concealed bits of p_i''' and p_{i+1}''' are gotten by the following modulus operation:

$$TD' = p_i''' \bmod FM' \qquad OD' = p_{i+1}''' \bmod SM' \quad (22)$$

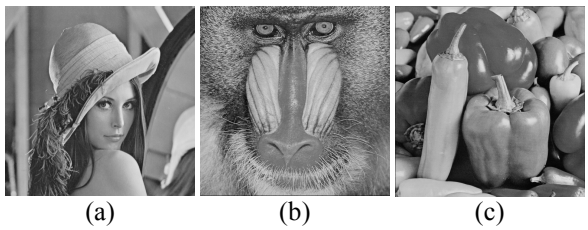
Finally, we can extract the embedded bits by the following equation:

$$TD' \times 10 + OD' = DEC' \quad (23)$$

4. EXPERIMENT RESULTS

In our experiments, we embed the same secret data bits, which are generated at random, into the cover-image by making use of PVD and our proposed method, respectively. We use three images, which are Lena, Baboon, and Peppers (as shown in Fig.2), to be cover-images. The size of cover-images is 512×512. We use the range table (as shown in Fig.1) to compare Wu and Tsai's method with our proposed method. The threshold Div of our method is 32.

The embedding capacity and the PSNR value of PVD and our proposed method are shown in Table.1. The results are the average of embedding 50 sets random bit stream into the cover-image. Obviously, our proposed method has extremely higher PSNR value than PVD when the embedding capacity is similar.



(a) (b) (c)

Fig.2 Cover-images: (a) Lena, (b) Baboon, (c) Peppers.

TABLE.1
THE COMPARISONS WITH PVD

Cover Image	PVD		Proposed Method(*Div*=32)	
	Capacity	PSNR	Capacity	PSNR
Lena	51222	41.1	51223	48.42
Baboon	57117	36.95	57138	46.09
Peppers	50793	41.29	50909	48.48

5. CONCLUSIONS

In this paper, we proposed a steganographic method based on modulus operation. We divide the embedding secret data into two parts depending on *Div*. Then, a modulus operation is used to hide these two parts into the first and the second pixel of two pixel block, respectively. We reduce the distortion by partitioning the secret data into two parts and using modulus operation. The experiment results demonstrate our proposed method has higher image quality than pervious literature which is based on pixel value differencing.

REFERENCES

[1] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proceedings of the IEEE*, vol. 86, No. 6, pp. 1064–1087, Jun. 1998.

[2] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking digital image and video data: A state-of-the-art overview," *IEEE Signal Processing Magazine*, vol. 17, No. 5, pp. 20–46, Sept. 2000.

[3] C. C. Chang, J. C. Chuang, and Y. P. Lai, "Hiding data in multitone images for data communications," *IEE Vision, Image and Signal Processing*, vol. 151, no. 2, pp. 137–145, Apr. 2004.

[4] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color and gray-scale images," *IEEE Multimedia* vol. 8, no. 4, pp. 22–28, Oct. 2001.

[5] C. K. Chan, and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, Mar. 2004.

[6] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Image Processing*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[7] R. Z. Wang, C. F. Lin, and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," *Pattern Recognition*, vol. 34, no. 3, pp. 671–683, March 2001

[8] C. C. Chang, J. Y. Hsiao, and C. S. Chan, "Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy," *Pattern Recognition*, vol. 36, no. 7, pp. 1583–1595, Jul. 2003.

[9] C. C. Thien and J. C. Lin, "A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function," *Pattern Recognition*, vol. 36, no. 12, pp. 2875–2811, Dec. 2003.

[10] P. L. Lin, "Robust transparent image watermarking system with spatial mechanisms," *Systems and Software*, vol. 50, no. 2, pp. 107–116, Feb. 2000.

[11] C. C. Chang, Y. S. Hu, and T. Z. Lu, "A watermarking-based image ownership and tampering authentication scheme," *Pattern Recognition Letters*, vol. 27, no. 5, pp.439–446, Apr. 2006.

[12] C. C. Chang, and H. W. Tseng, "A steganographic method for digital images using side match," *Pattern Recognition Letters*, vol. 25, no.12, pp. 1431–1437, Sep. 2004.

[13] D. C. Wu, and W. H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 27, no. 9-10, pp. 1613–1626, Jun. 2003.