LETTER

Improving the Adaptive Steganographic Methods Based on Modulus Function*

Xin LIAO[†a)], **Qiaoyan WEN**[††], ***Nonmembers***, ***and*** **Jie ZHANG**[†††], ***Member***

SUMMARY This letter improves two adaptive steganographic methods in Refs. [5], [6], which utilize the remainders of two consecutive pixels to record the information of secret data. Through analysis, we point out that they perform mistakenly under some conditions, and the recipient cannot extract the secret data exactly. We correct these by enlarging the adjusting range of the remainders of two consecutive pixels within the block in the embedding procedure. Furthermore, the readjusting phase in Ref. [6] is improved by allowing every two-pixel block to be fully modified, and then the sender can select the best choice that introduces the smallest embedding distortion. Experimental results show that the improved method not only extracts secret data exactly but also reduces the embedding distortion.

key words: *steganography, adaptive steganographic methods, pixel-value differencing, modulus function*

1. Introduction

Steganography is a technique of hiding information into innocuous-looking cover media, by slightly modifying the cover, without being suspicious [1]. Images are the most popular cover media, and many image steganographic methods are presented. According to the characteristics of human visual system, it is sensitive to some changes in the pixels of the smooth areas, while it is not sensitive to changes in the edge areas. Thus, several adaptive steganographic methods based on pixel-value differencing (PVD) have been proposed, in which the amount of bits to be embedded in each pixel is variable. In 2003, Wu and Tsai proposed a novel steganographic method, in which the number of bits to be embedded in a pixel is decided by the difference value between two neighboring pixels [2]. Chang and Tseng proposed the difference value between the pixel and its upper and left side pixels to determine how many secret bits should be embedded [3]. Wu et al. presented a new steganographic method combining PVD and LSB substitution [4]. In 2008, Wang et al. proposed a high quality steganograpic method with PVD and modulus function, utilizing the remainders of two consecutive pixels to record the secret information [5]. In 2010, Jung proposed a steganographic method using PVD, modulus function and LSB substitution, i.e., secret data can be embedded on the edge area by PVD and modulus function, and on the smooth area by LSB substitution [6]. Compared with Wang et al.'s method, it provided higher embedding capacity instead of the quality of stego images.

However, the embedding procedures in Refs. [5], [6] perform mistakenly under some conditions, i.e., the recipient cannot extract the secret messages exactly. In Sect. 3.1, the detailed theoretic analyses are given, and we correct these by enlarging the adjusting range of the remainders of two consecutive pixels.

Furthermore, the readjusting phase in Ref. [6] can also be improved. In Sects. 3.2, by allowing every two-pixel block to be fully modified, the sender can select the best choice that introduces the smallest embedding distortion.

Section 4 gives the experimental comparisons between Jung's method and the improved one. It is shown that the improved method not only extracts secret data exactly but also provides better image quality when concealing with the same embedding capacity.

2. Literatures Review

In this section we will briefly review two adaptive steganographic methods in Refs. [5], [6]. Both of them modify the remainders of two consecutive pixels to record the secret data. A gray-value cover image is partitioned into non-overlapping blocks of two neighboring pixels, and all possible difference values between two neighboring pixels range from 0 to 255. A range table R is designed, consisting of 6 contiguous sub-ranges R_k ($k = 1, 2, \cdots, 6$). Each sub-range R_k has its upper and lower bound values u_k and l_k, respectively. The width $w_k = u_k - l_k + 1$ is designed to be a power of 2. $R_1 = [0, 7]$, $R_2 = [8, 15]$, $R_3 = [16, 31]$, $R_4 = [32, 63]$, $R_5 = [64, 127]$, $R_6 = [128, 255]$.

2.1 Embedding Procedure

For each two-pixel block (y_i, y_{i+1}), the embedding procedure is described as follows. We only execute Step 1,3,4,5 in Wang et al.'s method. In order to increase the embedding capacity in the smooth areas, Jung's method embeds the secret data using LSB substitution in the smooth areas, and still uses Wang et al.'s method in the edge areas. The

Manuscript received June 17, 2013.
Manuscript revised July 28, 2013.

[†]The author is with College of Information Science and Engineering, Hunan University, Changsha, Hunan 410082, China.
[††]The author is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China.
[†††]The author is with School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China.

*This work is supported by National Natural Science Foundation of China (Grant Nos. 61272057, 61170270, 61100203, 61003286), Young Teacher Foundation of Hunan University (Grant No. 531107040701).

a) E-mail: xinliao@hnu.edu.cn (Corresponding author)

DOI: 10.1587/transfun.E96.A.2731

range of the difference value between two consecutive pixels is partitioned into the smooth area and edge area by a predefined division. We execute Step 1-7 in Jung's method.

Step 1: Calculate the difference value $D = |y_i - y_{i+1}|$ of two consecutive pixels (y_i, y_{i+1}).

Step 2: Use the division T to judge the area that the two-pixel block belongs to. If $D > T$, the block belongs to the edge area, go to Step 3. Otherwise, the block belongs to the smooth area, go to Step 6.

Step 3: Find the optimal sub-range R_k of D such that $D \in [l_k, u_k]$. Calculate the number of secret bits $n = \log_2(w_k)$ that can be embedded in the block. Read n bits from the binary secret messages and transform into decimal value b.

Step 4: Calculate the remainder of the sum of two pixels $F_{rem} = (y_i + y_{i+1}) \bmod 2^n$. Let $m = |F_{rem} - b|$ and $m_1 = 2^n - m$. **The adjusting algorithm** is as follows.

case 1: if $F_{rem} > b$, $m \le 2^{n-1}$, and $y_i \ge y_{i+1}$, then $(y'_i, y'_{i+1}) = (y_i - \lceil m/2 \rceil, y_{i+1} - \lfloor m/2 \rfloor)$

case 2: $F_{rem} > b$, $m \le 2^{n-1}$, and $y_i < y_{i+1}$, then $(y'_i, y'_{i+1}) = (y_i - \lfloor m/2 \rfloor, y_{i+1} - \lceil m/2 \rceil)$

case 3: if $F_{rem} > b$, $m > 2^{n-1}$, and $y_i \ge y_{i+1}$, then $(y'_i, y'_{i+1}) = (y_i + \lfloor m_1/2 \rfloor, y_{i+1} + \lceil m_1/2 \rceil)$

case 4: if $F_{rem} > b$, $m > 2^{n-1}$, and $y_i < y_{i+1}$, then $(y'_i, y'_{i+1}) = (y_i + \lceil m_1/2 \rceil, y_{i+1} + \lfloor m_1/2 \rfloor)$

case 5: if $F_{rem} \le b$, $m \le 2^{n-1}$, and $y_i \ge y_{i+1}$, then $(y'_i, y'_{i+1}) = (y_i + \lfloor m/2 \rfloor, y_{i+1} + \lceil m/2 \rceil)$

case 6: if $F_{rem} \le b$, $m \le 2^{n-1}$, and $y_i < y_{i+1}$, then $(y'_i, y'_{i+1}) = (y_i + \lceil m/2 \rceil, y_{i+1} + \lfloor m/2 \rfloor)$

case 7: if $F_{rem} \le b$, $m > 2^{n-1}$, and $y_i \ge y_{i+1}$, then $(y'_i, y'_{i+1}) = (y_i - \lceil m_1/2 \rceil, y_{i+1} - \lfloor m_1/2 \rfloor)$

case 8: if $F_{rem} \le b$, $m > 2^{n-1}$, and $y_i < y_{i+1}$, then $(y'_i, y'_{i+1}) = (y_i - \lfloor m_1/2 \rfloor, y_{i+1} - \lceil m_1/2 \rceil)$

Step 5: Revise y'_i and y'_{i+1} when $y'_i < 0$ or $y'_{i+1} < 0$ or $y'_i > 255$ or $y'_{i+1} > 255$. A detailed implementation was given in Refs. [5], [6].

Step 6: Read 3 bits from the binary secret messages, and convert y_i, y_{i+1} to be y'_i, y'_{i+1} by the 3-bit LSB substitution, respectively.

Step 7: This step is called "**the readjusting phase**". Calculate the new difference value $D' = |y'_i - y'_{i+1}|$. If $D' > T$, readjust y'_i and y'_{i+1} as follows.

case 1: if $y'_i \ge y'_{i+1}$, then $(y'_i, y'_{i+1}) = (y'_i - 8, y'_{i+1} + 8)$.

case 2: if $y'_i < y'_{i+1}$, then $(y'_i, y'_{i+1}) = (y'_i + 8, y'_{i+1} - 8)$.

2.2 Extracting Procedure

We can quickly extract the secret data without the original image. Partition the stego image into two-pixel blocks, which is identical with the embedding procedure.

For Wang et al.'s method, calculate the number of secret bits n that can be extracted from the two-pixel block. Calculate the remainder of the sum of two pixels $b = (y'_i + y'_{i+1}) \bmod 2^n$. Transform b into the binary secret data.

For Jung's method, calculate the difference value D. Use the division T to judge the area that the two-pixel block belongs to. If $D > T$, the block belongs to the edge area, and extract secret data using Wang et al.'s method. Otherwise, the block belongs to the smooth area, and extract 3 secret bits from the 3-bit LSB of y'_i and y'_{i+1}, respectively.

3. Theoretic Analyses and Improvement

3.1 Improvement for the Adjusting Algorithm

The adjusting algorithm was proposed to alter remainder of the sum of two consecutive pixels so as to record the secret data exactly and reduce the embedding distortion. However, it has a loophole. Here we will show that, under some conditions, the recipient cannot extract the secret data exactly, i.e., the original adjusting algorithm is invalid.

For all the cases 1-8 in Step 4, we find that

$$D' = |y'_i - y'_{i+1}| = \begin{cases} D & \text{if } m \equiv 0 \pmod 2 \\ D - 1 & \text{if } m \equiv 1 \pmod 2 \end{cases} \quad (1)$$

In addition, the widths of sub-ranges R_i are 8, 8, 16, 32, 64, 128, and the numbers of secret bits n that can be embedded are $3, 3, 4, 5, 6, 7$. All the l_k are the even numbers.

Theorem 1: Suppose that two pixels y_i and y_{i+1} satisfy their difference value $D = l_k$ $(3 \le k \le 6)$, then the original adjusting algorithm is invalid when the secret data b such that $2^{n-1} \le b < 2^n$ and $b \equiv 1 \pmod 2$.

Proof: F_{rem} and $(y_i + y_{i+1})$ have the same parity. Because the parity of the sum of two numbers is the same as that of their difference, F_{rem} and $D = |y_i - y_{i+1}|$ have the same parity. Since $D = l_k$ and l_k is an even number, F_{rem} is an even number. $b \equiv 1 \pmod 2$, so $m = |F_{rem} - b|$ is an odd number. From Eq. (1), we have $D' = D - 1 = l_k - 1 = u_{k-1}$, so $D' \in [l_{k-1}, u_{k-1}]$ and $n' = \log_2 w_{k-1} = n - 1$ $(3 \le k \le 6)$.

In the embedding procedure, the sender modifies the remainders of the sum of two consecutive pixels, such that $(y'_i + y'_{i+1}) \bmod 2^n = b$. Thus, $(y'_i + y'_{i+1}) = 2^n \times q + b, q \in \mathbf{Z}$.

In the extracting procedure, the recipient calculates

$$\begin{aligned}(y'_i + y'_{i+1}) \bmod 2^{n'} &= (2^n \times q + b) \bmod 2^{n-1} \\ &= \begin{cases} b & \text{if } \ 0 \le b < 2^{n-1} \\ b - 2^{n-1} & \text{if } \ 2^{n-1} \le b < 2^n \end{cases}\end{aligned}$$

He will mistake $b - 2^{n-1}$ for the secret data when $2^{n-1} \le b < 2^n$ and $b \equiv 1 \pmod 2$. *Q.E.D*

The original adjusting algorithm makes the difference values smaller under some conditions, resulting in failure to extraction. To avoid this problem, we enlarge the adjusting range of the remainders of two consecutive pixels. Step 4 should be modified as follows.

Step 4: Let $y'_i = y_i + j_1, y'_{i+1} = y_{i+1} + j_2$, $|j_1|, |j_2| \le 2^n$, $j_1, j_2 \in \mathbf{Z}$. Search (y'_i, y'_{i+1}) such that

(1) $D' = |y'_i - y'_{i+1}| \in [l_k, u_k]$.

(2) $b = (y'_i + y'_{i+1}) \bmod 2^n$.

(3) The value of $(y'_i - y_i)^2 + (y'_{i+1} - y_{i+1})^2$ is minimized.

(4) $y'_i, y'_{i+1} \in [0, 255]$.

It is shown that the searching range of each pixel is 2^{n+1}, and the maximum choice is 2^8. Thus, for a $M \times N$ grayscale image, the computational costs are not more than $2^8 \times M \times N$, and the computation complexity is $O(MN)$.

Table 1 Experimental comparisons based on the same embedding capacity for ten cover images.

	Jung's ($T = 15$)			Improved ($T = 15$)			Jung's ($T = 31$)			Improved ($T = 31$)		
Covers	Capacity	PSNR	IF	Capacity	PSNR	IF	Capacity	PSNR	IF	Capacity	PSNR	IF
Lena	768612	37.75	0.999353	768612	40.32	0.999642	783898	37.75	0.999352	783898	40.37	0.999646
Baboon	729526	37.48	0.999345	729526	39.55	0.999593	776644	37.39	0.999330	776644	39.64	0.999602
Peppers	774985	37.49	0.999306	774985	39.89	0.999600	784623	37.65	0.999331	784623	40.21	0.999629
Toys	772678	37.28	0.999208	772678	39.41	0.999514	782788	37.25	0.999202	782788	39.42	0.999515
Boat	760145	37.35	0.999438	760145	39.53	0.999659	781037	37.34	0.999436	781037	39.59	0.999664
Girl	771137	37.75	0.999451	771137	40.47	0.999707	785255	37.81	0.999459	785255	40.55	0.999712
Gold	768850	37.77	0.999262	768850	40.38	0.999595	784304	37.79	0.999265	784304	40.45	0.999602
Zelda	778303	37.85	0.999215	778303	40.60	0.999583	785623	37.87	0.999218	785623	40.63	0.999586
Barb	738908	36.88	0.999267	738908	38.52	0.999498	774500	36.75	0.999246	774500	38.53	0.999499
Tiffany	772946	37.69	0.999720	772946	40.28	0.999846	784216	37.69	0.999720	784216	40.30	0.999847
Average	**763609**	**37.53**	**0.999357**	**763609**	**39.90**	**0.999624**	**782288**	**37.53**	**0.999356**	**782288**	**39.97**	**0.999630**

Table 2 Experimental comparisons based on the same embedding capacity for two image databases.

	$T = 7$			$T = 15$			$T = 31$			$T = 63$		
	Capacity	PSNR	IF	Capacity	PSNR	IF	Capacity	PSNR	IF	Capacity	PSNR	IF
Jung's	695992	31.29	0.991495	744394	31.78	0.992774	766974	32.55	0.994335	773217	33.88	0.996189
Improved	695992	32.11	0.991818	744394	32.64	0.993099	766974	33.53	0.994675	773217	35.13	0.996543

Usually, the computing complexity is acceptable in image steganographic methods.

3.2 Improvement for the Readjusting Phase

The readjusting phase was executed if and only if the new difference value $D' > T$, and there are only two ways of readjusting. In fact, for each two-pixel block, the readjusting phase can be executed, regardless of whether $D' > T$ or not. The modification directions can also be exploited fully, other than two original ways.

In the improved one, we execute the readjustment for every pixel pair and allow nine different ways of modification, so the sender can select the best choice that introduces the smallest distortion. Step 7 is modified as follows.

Step 7: Let $\hat{y}_i = y'_i + j_1 \times 8, \hat{y}_{i+1} = y'_{i+1} + j_2 \times 8$, here $j_1, j_2 \in \{0, 1, -1\}$. Search $(\hat{y}_i, \hat{y}_{i+1})$ such that

(1) $\hat{D} = |\hat{y}_i - \hat{y}_{i+1}| \le T$.

(2) The value of $(\hat{y}_i - y_i)^2 + (\hat{y}_{i+1} - y_{i+1})^2$ is minimized.

(3) $\hat{y}_i, \hat{y}_{i+1} \in [0, 255]$.

It is shown that the searching range of each pixel is 3. For a $M \times N$ grayscale image, the computational costs are not more than $3 \times M \times N$. The computation complexity is also acceptable.

4. Experimental Results

In this section, experimental results are presented to demonstrate the correctness and effectiveness of the improved method. Random bit streams and ten grayscale images are used as the secret data and cover images. The peak signal to noise ratio (PSNR) and image fidelity (IF) are utilized to evaluate the quality of stego images [7]. For a $M \times N$ grayscale image, the PSNR and IF values are defined as follows:

$$PSNR = 10 \times \log_{10} \frac{255 \times 255 \times M \times N}{\sum_{i=1}^{M} \sum_{j=1}^{N} (p_{i,j} - q_{i,j})^2} (dB) \tag{2}$$

$$IF = 1 - \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (p_{i,j} - q_{i,j})^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} p_{i,j}^2} \tag{3}$$

where $p_{i,j}$ and $q_{i,j}$ denote the pixel values in row i and column j of the cover image and the stego image, respectively.

Table 1 shows the comparisons results of ten cover images based on the same embedding capacity, where the divisions $T = 15$ and $T = 31$. It is shown that our improved method can obtain higher PSNR and IF values.

To further evaluate the performance, two image databases are used in the experiments. 2000 test images include (but not limited to) landscapes, plants, animals, people and buildings.

(1) UCID Database [8]: randomly download 1338 color images with size of 384×512 or 512×384.

(2) Ground Truth Database [9]: randomly download 662 color images, and resize them to 756×504 or 504×756.

Table 2 shows the comparisons results of two images databases based on the same embedding capacity, where the divisions $T = 7, 15, 31, 63$. It is shown that the improved method is effective.

5. Conclusion

In this letter, two adaptive steganographic methods based on modulus function are improved. Firstly, we point out the recipient in Refs. [5], [6] cannot extract the secret data exactly under some conditions, and then correct these by enlarging the adjusting range of the remainders of two consecutive pixels. Furthermore, the readjusting phase in Ref. [6] is improved by fully exploited the modification of pixels.

The improved method majors in more significant promotions in the terms of correctness and effectiveness. In future, besides the merits achieved in this letter, we will attempt to modify it to achieve stronger security.

References

[1] R.J. Anderson and F.A.P. Petitcolas, "On the limits of steganography," IEEE J. Sel. Areas. Commun., vol.16, pp.474–481, 1998.
[2] D.C. Wu and W.H. Tsai, "A steganographic method for images by pixel-value differencing," Pattern Recognit. Lett., vol.24, no.9-10, pp.1613–1626, 2003.
[3] C.C. Chang and H.W. Tseng, "A steganographic method for digital images using side match," Pattern Recognit. Lett., vol.25, no.12, pp.1431–1437, 2004.
[4] H.C. Wu, N.I. Wu, C.S. Tsai, et al., "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," IEE Proc.-Vis., Imag. and Sign. Process., vol.152, no.5, pp.611–615, 2005.
[5] C.M. Wang, N.I. Wu, C.S. Tsai, et al., "A high quality steganography method with pixel-value differencing and modulus function," J. Syst. Softw., vol.81, no.1, pp.150–158, 2008.
[6] K.H. Jung, "High-capacity steganographic method based on pixel-value differencing and LSB replacement methods," Imag. Sci. J., vol.58, no.4, pp.213–221, 2010.
[7] M. Kutter and F.A.P. Petitcolas, "A fair benchmark for image watermarking systems," Proc. SPIE Conf. Security Watermarking of Multimedia Contents, vol.3657, pp.226–239, San Jose, CA, 1999.
[8] G. Schaefer and M. Stich, "UCID-An uncompressed colour image database," Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia, pp.472–480, 2004.
[9] University of Washington, Object and Concept Recognition for Content-Based Image Retrieval.