A new steganographic method of the pixel-value differencing

Hui-Chieh Lu, Yen-Ping Chu, Min-Shiang Hwang and Na-I Wu

In this article, a new steganographic method of the pixel-value differencing approach is proposed. In the Wu and Tsai's method, the embedded capacity of secret messages is about 2 times fewer than the least-significant-bit replacement methods. This new method provides more capacity than the Wu and Tsai's method but less imperceptible. Unless this method still provides less capacity than the least-significant-bit replacement methods, it is also refused to dual statistics attacks.

Keywords: Image processing; Data hiding; Steganography; Stego-image

Introduction: Some steganographic methods on spatial domain for embedding data in images have been proposed [1, 2]. Many data embedding techniques are based on the LSB (least-significant-bit) replacement method [1]. However a stegano-analytic method, the dual statistics method, proposed by Fridrich et al. [3] was claimed to detect the existence of data in LSB of the image pixels. Wu and Tsai proposed a pixel-value differencing scheme to resist of the dual statistics attack and it was less imperceptible than the LSB method [2]. But the capacity of the embedding secret data in the Wu and Tsai's method is less than that in the LSB method. In average cases, its capacity is about 2 times fewer than the LSB method.

In this Letter, we propose a new scheme which can increase the embedding data against the method of Wu and Tsai [2] and it is also can resist to the dual statistics attack. Although the quality of stego-image is little bad than the method of Wu and Tsai, it is still imperceptible. Next, we propose our method as follows.

Embedding Algorithm: The cover image used 256 gray scale images in our proposed method. Following are the embedding steps:

1. Divide the cover image into non-overlapping 2 $\times$ 2 blocks in a zigzag manner consecutively, and every block has four pixels, say p_{i1}, p_{i2}, p_{i3} and p_{i4}, as shown in Fig. 1. Assume the gray values of them are g_1, g_2, g_3 and g_4, respectively. Then d_{jk} represents g_k - g_j, where $i, j = 1, 2, 3, 4$.

2. Build a pixel-value differencing range table. In this step, we leverage the method of Wu and Tsai [2]. We classify them into a number of contiguous ranges, say R_i , where $i = 1, 2, ..., n$. The ranges are indexed 1 through n . The lower and upper bound values of R_i are denoted as I_i and u_i respectively, where l_1 is 0 and u_n is 255. This table can refer the length (bits) of data should be embedded on different gray value difference which is computed as $log_2(u_i - l_i + 1)$.

3. Partition every block into three pair pixels, and they are (p_{i1}, p_{i2}) , (p_{i2}, p_{i4}) and (p_{i3}, p_{i4}) , respectively.

4. Embed data into (p_{i2}, p_{i4}) pair. Assume the two pixels are p'_{i2} and p'_{i4} of the stego-image with the new gray values g'_2 and g'_4 after embedding. To compute the new gray values is based on a function $f((g_2, g_4), m)$ which is defined as follows:

$$f((g_2, g_4), m) = (g'_2, g'_4) = \begin{cases} (g_2 - \lceil m/2 \rceil, g_4 + \lfloor m/2 \rfloor), & if\ d_{24}\ is\ odd; \\ (g_2 - \lfloor m/2 \rfloor, g_4 + \lceil m/2 \rceil), & if\ d_{24}\ is\ even, \end{cases} \tag{1}$$

where $m = d'_{24} - d_{24}$, and d'_{24} is denoted as follows:

$$d'_{24} = \begin{cases} l_a + b, & for\ d \geq 0; \\ -(l_a + b), & for\ d < 0, \end{cases} \tag{2}$$

where a is the index of the differencing range table of the pair of (g_2, g_4) , and b is the value of the embedding sub-stream. In the above calculation, some pair of (g_2, g_4) may cause the resulting to fall off the boundaries of the range [0, 255]. Before the embedding process, we must check if the embedding pair is falling-off boundary. So we must run a checking process in advance, and we produce a pair of $(\hat{g}_2, \hat{g}_4)$ from the value of function by Equation (1), $f((g_2, g_4), u_a - d_{24})$. If one of $(\hat{g}_2, \hat{g}_4)$ the values out of the range [0, 255], we say the pair of (g_2, g_4) is falling-off boundary and this pair will not be embedded.

5. Embed data into (p_{i1}, p_{i2}) and (p_{i3}, p_{i4}) pairs. Assume the four pixels are p'_{i1} , p'_{i2} , p'_{i3} and p'_{i4} of the stego-image with the new gray values g'_1 , g'_2 , g'_3 and g'_4 after embedding. Because the gray values g'_2 and g'_4 have been embedded some data-bits in Step 4, we will not change their values in this step. To compute the new gray values is based on a function $h((g_j, g'_k), m)$, where $(j, k) = (1, 2), (3, 4)$, which is defined as follows:

$$h((g_j, g'_k), d'_{jk}) = (g'_j, g'_k) = (g'_k - d'_{jk}, g'_k) \tag{3}$$

where d'_{jk} is denoted as follows:

$$d'_{jk} = \begin{cases} l_a + b, & for\ d \geq 0; \\ -(l_a + b), & for\ d < 0, \end{cases} \tag{4}$$

where a is the index of the differencing range table of the pair of (g_j, g_k) , and b is the value of the embedding sub-stream. Since this step may cause the edge area sharp, we set a threshold 'T' in the proposed method. Before embedding data to the (p_{i1}, p_{i2}) and (p_{i3}, p_{i4}) pairs, we must check by the following rule:

If $(u_a > T)$ then

don't embed data-bit(s) to this pair

else

embed data-bit(s) to this pair

6. Repeat Steps 3 - 5, until all secret data-bits are embedded.

In the Step 5, we use a threshold 'T'. It is trade off when we tune this value. The larger 'T' is good for image quality but bad for capacity, and it is contrary when 'T' is smaller. In our experiments, the image quality and capacity are good when 'T' is 16.

Recovering algorithm: When we recover the embedded data, there is no need of the cover image but we need the original the pixel-value differencing range table which is used in the embedding process. Following are the recovering steps:

1. Divide the stego-image into non-overlapping 2 $\times$ 2 blocks in a zigzag manner consecutively, and every block has four pixels, say, p'_{i1} , p'_{i2} , p'_{i3} and p'_{i4}. Assume the gray values of them are g'_1 , g'_2 , g'_3 and g'_4 , respectively.

2. Partition every block into three pair pixels, and they are (p'_{i1}, p'_{i2}) , (p'_{i2}, p'_{i4}) and (p'_{i3}, p'_{i4}) , respectively.

3. Recover data from (p'_{i2}, p'_{i4}) pair. First we must check if it is falling-off boundary by calculating $(\hat{g_2}', \hat{g_4}')$ values. If it is not falling-off boundary, the data-bits can be recovered from this pair. We can get the value of embedding sub-stream by calculating $(|g'_4 - g'_2| - l_a)$, where a is the index of the differencing range table of the pair of (g'_2, g'_4). The total number of sub-stream bits also can be known by indexing the range table, so the sub-stream bits can be recovered.

4. Recover data from (p'_{i1}, p'_{i2}) and (p'_{i3}, p'_{i4}) pairs. First we must check if (g'_{i1}, g'_{i2}) or (g'_{i3}, g'_{i4}) is embedding data and it can be justified by the condition of $(u_a > T)$, where a is the index of the differencing range table of the pair of (g'_j, g'_k) and $(j, k) = (1, 2), (3, 4)$. If it embeds data-bits, we can get the value of embedding sub-stream by calculating $(|g'_j - g'_k| - l_a)$

. The total number of sub-stream bits also can be known by indexing the range table, so the sub-stream bits can be recovered.

5. Repeat Steps 2 - 4, until all secret data-bits are recovered.

Experimental results: In our experiments, we use four images "Lena", "Boboon", "Peppers", and "Jet" with size 512×512 to be the cover images. We select the range widths of 8, 8, 16, 32, 64, and 128 to be the range table in the experiments. The two of the resulting images are shown in Fig. 2.

Finally, the capacities of embedded data and the peaks of the signal-to-noise (PSNR) of the embedding results with comparing to the scheme of Wu and Tsai [2] are shown in Table 1. The capacity is much better than their scheme and the image quality is just little bad than theirs.

Conclusion: We propose a new steganographic method for information hiding. It is based on the pixel-value differencing approach. The scheme is good for capacity and it is resist to the dual statistics attack.

Reference

1. Walton, S.:'Image authentication for a slippery new age'. *Dr. Dobb's J.: Software Tools Profess. Program*, 1995, 20, pp. 18-26.

2. Wu, D.C., and Tsai, W.H.:'A steganographic method for images by pixel-value differencing'. *PatternRecognitionLetters*, 2003, 24, pp. 1613-1626.

3. Fridrich, J, Goljan, M., and Du, R.:'Reliable detection of LSB steganography in grayscale and color images'. *Proc. ACM Workshop on Multimedia and Security*,2001,pp. 27-30

Authors' affiliations:

Hui-Chieh Lu and Yen-Ping Chu (Department of Computer Science, National Chung Hsing University, Taiwan 402, R.O.C.)

Min-Shiang Hwang (Department of Management Information System, National Chung Hsing University, Taiwan 402, R.O.C.)

Na-I Wu (Graduate Institute of Networking and Communication Engineering, National Chaoyang University of Technology, Taiwan, R.O.C.)

Responsible for correspondence: Prof. Min-Shiang Hwang (Email: mshwang@nchu.edu.tw)

Figure captions:

Fig.1 The non-overlapping 2×2 block

Fig.2 Resulting images

a "Lenna" cover image

b "Lenna" stego-image

c "Baboon" cover image

d "Babbon" stego-image

Table captions:

Table 1 Experimental results

Figure 1

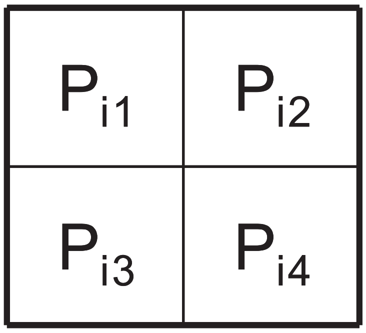

Figure 2

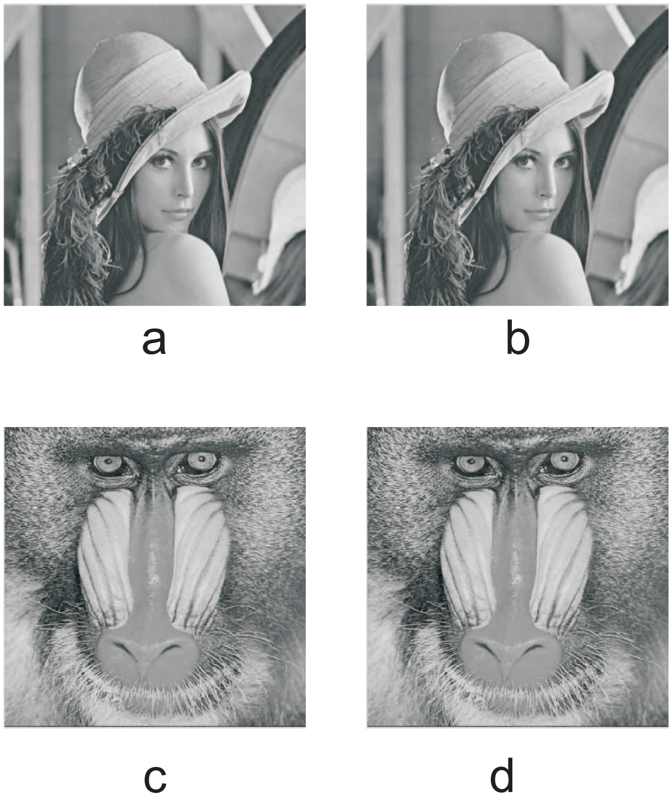

Table 1

Images(512×512)	Wu and Tsai's scheme[2]		Our scheme	
	Capacity (bytes)	PSNR(dB)	Capacity(bytes)	PSNR(dB)
Lena	50801	33.58	70154	32.18
Baboon	56889	27.53	62208	25.92
Peppers	50769	32.63	70497	30.91
Jet	51242	31.42	70520	31.10