

# Hiding Multiple Data in Color Images by Histogram Modification

Soo-Chang Pei<sup>1</sup>, and Yi-Chong Zeng<sup>2</sup>

<sup>1</sup>Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, 10617, R. O. C

<sup>2</sup>Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, 10617, R. O. C

E-mail: pei@cc.ee.ntu.edu.tw<sup>1</sup>, d9942010@ee.ntu.edu.tw<sup>2</sup>

## Abstract

*In this paper, we present a new histogram-based data-hiding algorithm that secret data is embedded in the least significant bit of the histogram value. To change the pixel value, it will alter the histogram to accomplish data-hiding work. In the proposed algorithm, it is able to perform data hiding on the one-dimension histogram, two-dimension histogram map and three-dimension histogram cube. Besides, the multiple secret data hiding in various combinations of histogram spaces are successfully demonstrated in our experimental results. In addition, the natural and limited color images are tested in our experiments.*

## 1. Introduction

Data-hiding algorithms often embed data into the least significant bits (LSB) of the image pixel's values. Based on the length of LSB, there are two major methods: fix-length and variable-length LSB data-hiding methods [1, 2]. In order to hide data in the random texture patterns, the texture block coding technique was suggested in [3]. In addition, the approach is able to hide information in the spatial [4] or frequency domain. The major difference between data-hiding and watermarking techniques is its robustness to various attacks, the former one hides the data secretly into the host media, and the data will not be roust to any attack. The later one embeds watermark in the host media, and it needs survive for various attacks.

Histogram, the collective noun, represents the statistical distribution of the pixels in an image. It also widely applies in many research fields, e.g., image enhancement, image indexing and pattern recognition, etc. Several methods to embed data into histogram have presented in [5, 6, 7, 8, 9]. Coltuc et al. developed the watermarking technique by exact histogram specification. The technique changes the image content to result a specified histogram, such as: sine, cosine, sawtooth, notch waves and etc. Furthermore, Mese et al. proposed a method, which is similar to Coltuc's method, to alter the histogram to a desired specific histogram with MSE metric [10]. Here, we present a new histogram-based multiple data-hiding algorithm in color images.

## 2. Histogram-based data-hiding algorithm

In this section, we will describe the proposed histogram-based data-hiding (HBDH) algorithm, which are able to perform it on three different dimension histograms. The hiding procedure and parameters definition for each approach will be discussed in following paragraphs.

### 2.1 1-D HBDH algorithm

Before describing the algorithm, we define some assumptions as follows: the original  $P \times Q$  image is consisted of 8-bit pixel values, calculates its 1-D histogram with  $M$  bins, and  $x^i$  denotes the histogram value at  $i$ -th bin, where  $0 \leq x^i \leq P \times Q$  and  $0 \leq i \leq M-1$ . Assume that the total secret data to be embedded is consisted of  $N$  binary codes,  $y^j$  denotes the code, where  $y^j \in \{0, 1\}$  and  $0 \leq j \leq N-1$ . The modified secret data  $z^j$ , which is recombined from  $B$  bits of  $y^j$ , is prepared to embed in the least  $B$  significant bits of the corresponding histogram  $x^i$ , where  $0 \leq z^j < 2^B$  and  $0 \leq i \leq \lfloor N/B \rfloor - 1$ . The variable  $\lfloor N/B \rfloor$ , the greatest integer, is less than or equal to  $N/B$ ; moreover,  $\lfloor N/B \rfloor \leq M$ .

The basis of three proposed approaches is the 1-D HBDH algorithm. In the algorithm, we first determine the amount of add-up/released pixels  $z^j$  to corresponding histogram  $x^i$  within least  $B$  significant bit. The amount  $d$  is determined by,

$$d = \rho^{\alpha^*}(x^i, z^j, B) \quad (1)$$

, where  $\alpha^* = \operatorname{argmin}_{\alpha} |\rho^{\alpha}(x^i, z^j, B)|_{\alpha=\pm 1}$ . The function  $\rho^{\alpha}(u, v, b)$  measures the distance between  $u$  and  $v$  within the least  $b$  significant bits, its mathematical equation defines as

$$\rho^{\alpha}(u, v, b) = \alpha [ \alpha (v - u) \bmod 2^b ] \quad (2)$$

For instance, let  $u=18$ ,  $v=3$  and  $b=2$ , it means that secret data  $v$  (binary code is '11') will be embedded into the least 2 significant bits of the histogram  $u$  (binary code is '10010'). Then, to calculate the distances by (2),  $d=1$  and  $d=-3$  correspond to  $\alpha=+1$  and  $\alpha=-1$ , respectively. In order to decrease the least number of altered image pixel, we choose  $d=1$  which is the minimum absolute distance. Then, the modified value  $u+d=18+1=19$  (binary code is '10011') has been embedded secret data  $v$  in its least 2 significant bits.

If  $d > 0$  at  $m$ -bin histogram, it defines as the add-up procedure starting from  $m$  to  $m+1$  shown in Figure 1(a). We arbitrarily select  $d$  pixels with pixel value  $m+1$  and change them with pixel value  $m$ . If  $d < 0$  at  $m$ -bin histogram, it defines as the released procedure starting from  $m$  to  $m+1$  in Figure 1(b). We arbitrarily select  $|d|$  pixels with pixel value

$m$  and change them with pixel value  $m+1$ . An important issue is to forbid changing previous pixel value in the prior modified histogram. In addition, we prefer to assign one bin,  $m_{\max}$ , with the maximum histogram value, it is only to add-up/release the pixels in the procedure without doing the data-hiding work. Since the number of add-up/released pixels at  $m_{\max}$ -th bin is much smaller than the residuary one, then it is difficult to distinguish the altered pixels in the data hidden image. Therefore, it amounts to  $M-1$  bins of histogram to be changed for data hiding. In the left side of  $m_{\max}$ -th bin, it starts from the first bin to  $m_{\max}$ -th bin in data hiding procedure; on the other hand, in the right side of  $m_{\max}$ -th bin, it starts from the last bin to small  $m_{\max}$ -th bin.

It obviously observes that proposed algorithm is one kind of classical approach to embed information in LSB. Different from the classical approach by slightly modifying the pixel value, we change the global pixel values to alter the histogram in the data hiding work.

## 2.2 2-D HBDH algorithm

For 2-D HBDH algorithm, selecting the two components to construct a 2-D histogram map is a preprocessing step before data hiding. Intuitively, we can arbitrarily select two of trichromatic planes, i.e., red-green, red-blue, green-blue, blue-green, blue-red and green-red histogram maps. We assign the former component to be the horizontal axis, and latter one is the vertical axis of histogram map.

The assumptions of the algorithm are defined as follows. The  $L_x \times L_y$  2-D histogram map is calculated,  $x^{ij}$  denotes the histogram value at  $i$ -th horizontal step and  $j$ -th vertical step, where  $0 \leq x^{ij} \leq P \times Q$ ,  $0 \leq i \leq L_x - 1$  and  $0 \leq j \leq L_y - 1$ . In addition, we assign a 2-D secret data  $\mathbf{y}$ , whose size is the same as histogram map, the  $y^{ij}$  denotes the secret data value, where  $y^{ij} \in \{0, 1\}$ . In addition,  $\mathbf{x}^i$  and  $\mathbf{y}^j$  are defined as the 1-D column histogram and secret data at  $i$ -th horizontal step, respectively.

We divide the 2-D histogram map into several 1-D ones  $\mathbf{x}^i$  along the horizontal direction, and the length of 1-D histograms is  $L_y$ . By the same token, 2-D secret data is divided into several 1-D ones  $\mathbf{y}^j$ , and its length is  $L_x$  as well. The goal of hiding procedure is to embed  $\mathbf{y}^j$  into the LSB of the corresponding  $\mathbf{x}^i$  for each  $i$ . Therefore, we utilize 1-D HBDH algorithm to alter the value of add-up/released pixels in  $\mathbf{x}^i$ , consequently, it yields a modified histogram map with hiding data. Obviously, the column histogram  $\mathbf{x}^i$  is modified by 1-D HBDH algorithm, and it further alerts the pixel value of the vertical component in 2-D histogram map.

## 2.3 3-D HBDH algorithm

A 24-bit color image is consisted of the pixels with three 8-bit trichromatic planes. We adopt these components to construct a 3-D histogram cube, it is illustrated a histogram cube of color Lena image in Figure 4(a). The  $L_u \times L_v \times L_w$  3-D histogram cube is calculated,  $x^{ijk}$  denotes the histogram value at  $i$ -th step of  $u$ -axis,  $j$ -th step of  $v$ -axis and  $k$ -th step

of  $w$ -axis, where  $0 \leq x^{ijk} \leq P \times Q$ ,  $0 \leq i \leq L_u - 1$ ,  $0 \leq j \leq L_v - 1$  and  $0 \leq k \leq L_w - 1$ . We assign a 3-D secret data, whose size is the same as histogram cube; the variable  $y^{ijk}$  denotes the secret data value, where  $y^{ijk} \in \{0, 1\}$ . In addition, we define  $\mathbf{x}^k$  and  $\mathbf{y}^k$  to be 2-D histogram map and 2-D secret data at  $k$ -th step of  $w$ -axis, respectively. We utilize the 2-D HBDH algorithm to embed  $\mathbf{y}^k$  into  $\mathbf{x}^k$ . The modified histogram cube is eventually produced with 3-D hiding data.

From the above discussions, we summarize 10 cases, the possible combinations of number of 1-D, 2-D and 3-D secret data embed in the color image, are listed in Table 1. For example in case 3, there are one 1-D and two 2-D secret data, embeds into one 1-D histogram and two 2-D histogram maps, respectively. For multiple data hiding, low dimensional data hiding is performed in higher priority than the high ones. For example, we modify the 1-D histogram of red plane by 1-D HBDH algorithm. To avoid altering the 1-D data hidden histogram of red plane once again, we must select red-green and red-blue histogram maps to embed two 2-D secret data by using 2-D HBDH algorithm. It will only modify the pixel values of green and blue planes.

## 3. Experimental results

In this section, we demonstrate that our proposed algorithms are able to embed data in different dimension histograms. The tested image is size of  $256 \times 256$  Lena images with 24-bit resolution. In Table 2, it is listed PSNRs of hidden image by embedding three secret data with different sizes in case 7.

The secret data amount to 1530 bits which embed in the least 2 significant bits of histogram value ( $B=2$ ) in each color plane. The host and hidden images are shown in Figure 2(a) and Figure 2(b), respectively. The hidden image PSNR is 46.00dB. Implement the case 3 to embed one 1-D and two 2-D data into the three different histograms, then the hidden image is displayed in Figure 2(c) with PSNR=48.16dB. We calculate the 1-D histogram from green plane, and employ green-red and green-blue histogram maps to embed two 2-D data respectively. Their original histogram maps, modified histogram maps and extracted data of two histogram maps are shown in Figure 3(a) and Figure 3(b). The parameters are assigned to  $L_x=256$ ,  $L_y=256$  and  $B=1$ . For case 10, we utilize the 3-D HBDH algorithm to embed one 3-D secret data. In the algorithm, it extracts each 2-D plane data, and embeds it into the corresponding red-green histogram map by 2-D HBDH algorithm along the blue-axis. Figure 2(d) shows the hidden image with PSNR=45.43dB, its original histogram cube, modified one and extracted 3-D data are illustrated in Figure 4(a)-(c). The parameters are assigned to  $L_u=128$ ,  $L_v=128$  and  $L_w=128$ .

We modify each histogram value gradually and embed secret data in corresponding bin simultaneously. It is different to previous techniques, Coltuc's and Mese's methods, which estimate the histogram transformation from original to specified histogram. In order to evaluate our method, we first implement the exact histogram

specification [10] to transform the original histogram of Figure 2(a) to specified histogram of Figure 2(b). Figure 5(a) shows the hidden image with PSNR=46.00dB, and its image quality exactly equals to our method in Figure 2(b). Consequently, our method is an optimal histogram modification. Additionally, we rudely modify the least 2 significant bits of original histogram value without using proposed method, and the hidden image is shown in Figure 5(b) with PSNR=45.55dB. The image quality is lower than 46.00dB by using our method.

The proposed method also successfully embeds data into 1-D histogram for the limited color image with only a few dominant colors. In Figure 6(a), it displays logo of National Taiwan University, and merely contains 16 colors. Then the hidden image, which is shown in Figure 6(b) with PSNR=44.85dB, is embedded three 1-D secret data in case 7. Our method is suitable to hide multi-dimensional data in the dense bins of 2-D histogram map and 3-D histogram cube. However 16 colors sparsely distribute over the histogram map, the limited color image successfully hides 1-D secret data, but not suitable for 2-D and 3-D secret data hiding.

#### 4. Conclusion

A new data-hiding algorithm based on histogram is presented in this paper. The algorithm is performed on the 1-D histogram, 2-D histogram map and 3-D histogram cube. From the above experimental results, it demonstrates that the proposed algorithm is simple and efficient to accomplish the data hiding work.

#### 5. Reference

- [1] D.C. Wu, and W.H. Tsai, "Data hiding in images via multiple-based number conversion and lossy compression", *IEEE Trans. on Consumer Electronics*, vol. 44, no. 4, 1998, pp. 1406-1412.
- [2] 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 ICIP'99*, vol.1, 1999, pp. 286-290.
- [3] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding", *IBM Systems Journal*, vol. 35, no.3&4, 1996, pp. 313-336.
- [4] D.C. Wu and W.H. Tsai, "Image hiding in spatial domain using an image differencing approach", *IEE Proceedings, Vision, Image and Signal Processing*, vol. 147, no. 1, 2000, pp. 29-37.
- [5] D. Coltuc and Ph. Bolon, "Robust watermarking by histogram specification", *IEEE ICIP'99*, Kobe, Japan, 24-28 Oct. 1999, pp. 236-239.
- [6] D. Coltuc and Ph. Bolon, "Watermarking by histogram specification", *SPIE Proceedings, Conf. on Security and Watermarking of Multimedia Contents*, vol. 3657, San Jose, CA, Jan. 1999, pp.252-263.
- [7] D. Coltuc and Ph. Bolon, "Color image watermarking in HSI space", *IEEE ICIP'2000*, Vancouver, Canada, 10-13

Sept. 2000, pp. 698-701.

[8] D. Coltuc, J.M. Chassery, and Ph. Bolon, "Image authentication by exact histogram specification", *IEEE MMSP'02*, Cannes, 3-5 Oct. 2001, pp.701-710.

[9] D. Coltuc, Ph. Bolon, and J.M. Chassery, "Fragile and robust watermarking by histogram specification", *SPIE Proceedings, Conf. on Security and Watermarking of Multimedia Contents*, vol. 4675, April 2002, pp.701-710.

[10]M. Mese, P.P. Vaidyanathan, "Optimal histogram modification with MSE metric", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 7-11 May 2001, pp.1665-1668.

Table 1. There are 10 cases, the possible combinations of number of 1-D, 2-D and 3-D secret data are embed into 24-bit color image.

Cases	1-D Data	2-D Data	3-D Data
Case 1	1	×	×
Case 2	1	1	×
Case 3	1	2	×
Case 4	1	×	1
Case 5	2	×	×
Case 6	2	1	×
Case 7	3	×	×
Case 8	×	1	×
Case 9	×	2	×
Case 10	×	×	1

Table 2. The PSNRs of hidden image by embedding three secret data with different size in case 7.

Parameter	Data Size	PSNR
B	(bits)	(dB)
1	765	45.34
2	1530	40.13
3	2295	34.84

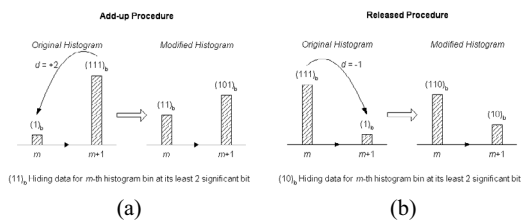


Figure 1. Original color Lena image and hidden images. (a) Add-up procedure; (b) released procedure.



Figure 2. Original color Lena image and hidden images. (a) Original color Lena image; (b) hidden image in case 7 with PSNR=46.00dB; (c) hidden image in case 3 with PSNR=48.16dB; (d) hidden image in case 10 with PSNR=45.43dB.

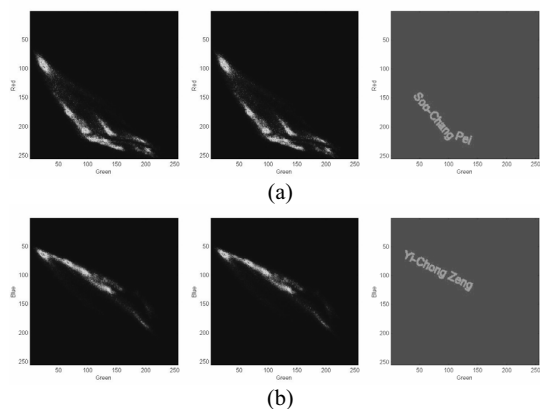


Figure 3. Original 2-D histogram maps, modified ones and secret data are arranged from leftmost column to rightmost one. (a) 1st results of Fig.2(c) in green-red histogram map (in case 3: one 1-D and two 2-D secret data); (b) 2nd results of Fig.2(c) in green-blue histogram map (in case 3: one 1-D and two 2-D secret data).

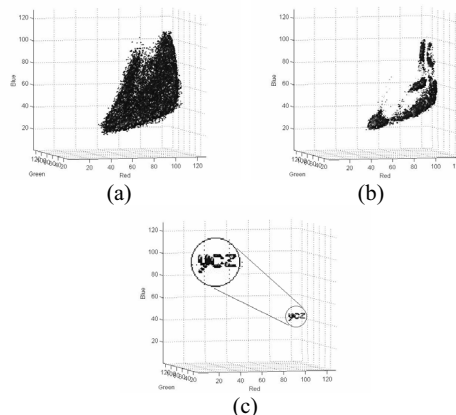


Figure 4. Original 3-D histogram cube, modified one and 3-D secret data. (a) Original 3-D histogram cube of Figure 2(a); (b) modified 3-D histogram cube of Figure 2(d); (c) extract 3-D secret data, which illustrates three alphabets "ycz".



Figure 5. Two hidden images. (a) It transforms the original histogram of Figure 2(a) to specified histogram of Figure 2(b) by exact histogram specification [10] (PSNR=46.00dB); (b) to modify the least 2 significant bits of original histogram value without using proposed method embeds 1530-bit secret data in color image. (PSNR=45.55dB)

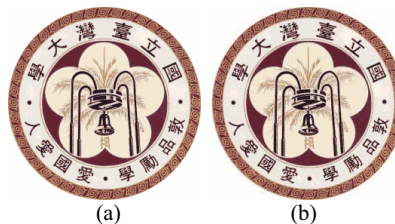


Figure 6. Original and hidden limited color images (a) Original logo of National Taiwan University; (b) hidden image in case 7 with PSNR = 44.85 dB.