1

A Lossless Data Hiding Method by Histogram Shifting Based on an Adaptive Block Division Scheme

Che-Wei Lee[1] and Wen-Hsiang Tsai[1,2]

[1]*Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan, ROC; E-mail: paradiserlee@gmail.com*
[2]*Department of Information Communication, Asia University, Taichung 41354, Taiwan, ROC; E-mail: whtsai@cs.nctu.edu.tw*

Abstract

A lossless data hiding method based on histogram shifting is proposed, which employs a scheme of adaptive division of cover images into blocks to yield large data hiding capacities as well as high stego-image qualities. The method is shown to break a bottleneck of data-hiding-rate increasing at the image block size of 8×8, which is found in existing histogram-shifting methods. Four ways of block divisions are designed, and the one which provides the largest data hiding capacity is selected adaptively. A series of experiments have been conducted, and superiority of the proposed method is shown by comparing the experimental results with those of other histogram-shifting based methods. A good property of the proposed method observed in the experiment is that data hiding rates are increased without degrading the stego-image quality expressed by the peak signal-to-noise ratio measure.

Keywords: lossless data hiding, data hiding, histogram shifting, block division.

P.S.P. Wang (Ed.), Pattern Recognition and Machine Vision – in Honor and Memory of the Late Professor King-Sun Fu, 1–14.

1.1 Introduction

The development of information hiding techniques provides a way to protect digital media. Such techniques may be employed to embed private or secret information into *cover media* in such a way that the existence of the hidden information is imperceptible but known to the pre-concerted recipient. The cover media can be of various types of digital data, such as image, text, video, etc. Information like private annotations, business logos, and critical intelligence can be embedded into a cover image in an invisible form so that many applications, like ownership claim of digital contents, copyright protection of media, covert communication between parties, etc., can be fulfilled. Information techniques used for covert communication are often called *steganography*, and those for ownership or copyright protection are often called *watermarking*.

In the early phase, conventional steganography [1, 9, 17–19] emphasizes exploring higher hiding payload and pursuing lower quality degradation in the *watermarked image* (also referred to as the *stego-image*, in contrast with the *cover image*). Bender et al. [1] proposed the technique of least-significant-bits (LSBs), in which a secret message is embedded in the least significant bits of the pixel values of cover images. The method yields high data hiding capacities and low computational complexities. Mielikainen [9] proposed a modified method of LSB replacement which embeds as many bits but with a smaller number of changed pixel values. Wu and Tsai [17] proposed a steganographic method for images by pixel-value differencing (PVD). The difference values between pixel value pairs are classified into a number of ranges, and the range into which the difference value between a pair of pixels falls decides how many bits can be embedded into the pixel pair. A method improving that by Wu and Tsai [17] was later proposed by Wu et al. [18], which utilizes the PVD technique to embed data in the smooth areas of the cover image and applies LSB replacement in the edged areas. This hybrid way of data hiding improves the hiding capacity and yields nearly equal stego-image quality, compared with the original PVD method [17]. Afterward, Yang et al. [19] proposed an adaptive k-LSB substitution method in which larger values of k are adopted in edge areas and smaller values of k are used for smooth areas. Also, the range of the PVD values of two consecutive pixels is divided into different levels. And the value of k is adaptively decided according to the level into which the PVD value falls. In this way, larger hiding capacities with higher stego-image quality can be obtained.

Besides the previously mentioned techniques of data hiding in the spatial domain, Wang et al. [15] transformed image block contents into coefficients in the frequency domain by the discrete cosine transform (DCT). The new AC values for the central block of every nine 8×8 image blocks are recomputed. Secret bits are embedded by modifying the magnitude relation between the new AC values and the original ones. In addition to data embedding using the DCT technique, the discrete wavelet transform (DWT) [8, 14, 16] and the discrete Fourier transform (DFT) [11, 12] have also been used.

The methods mentioned above yield permanent distortion in the stego-image. In general, a small amount of content distortion is usually imperceptible to human vision. However, such distortion is not preferred in some applications, such as legal documentation, medical imaging, military reconnaissance, high-precision scientific investigation, etc., because it may lead to risks of incorrect decision making. In view of this, a kind of novel data hiding technique, which is referred to as *reversible*, *invertible*, *lossless*, or *distortion-free*, has been developed in recent years. In this study, a reversible data hiding method which produces stego-images with good qualities and high data hiding capacities is proposed.

Reversible data hiding techniques can be employed to restore stego-images to their pristine states after the hidden data are extracted. Such techniques can be classified into three groups: (1) based on data compression [2,3]; (2) based on pixel-value difference expansion [6,13]; and (3) based on histogram shifting [4, 5, 7, 10]. The strategy used in the techniques of the first group is to compress the data to be embedded as well as the related information for data recovery, and then to embed these compressed data directly into the cover image. Celik et al. [3] proposed a high-capacity lossless hiding method in which each image pixel is quantified by a so-called L-level scalar quantization technique, and a lossless compression algorithm called CALIC is applied to compress the residues yielded by the quantified image pixels. Then, the compressed residues integrated with the secret bits are embedded into the quantified image by the LSB replacement technique. The second group of reversible data hiding methods aims to explore the redundancy of pixel values in images. Tian [13] proposed a technique of PVD expansion by performing fundamental arithmetic operations on pairs of pixels to discover hidable space for data embedding. However, not all pairs can be expanded for data hiding. A location map is used to indicate whether and where pairs are expanded or not. An enhanced PVD expansion method proposed by Kim et al. [6] used a refined location map and a new concept of expandability to achieve higher hiding capacities while keeping the resulting image distortion as low as the Tian method.

The last group of reversible data hiding methods, which the proposed method belongs to, is based on histogram shifting. Ni et al. [10] proposed a reversible data hiding method which shifts slightly the part of the histogram between the maximum point (also called the *peak point*) and the minimum one to the right side by one pixel value to create an empty *bin* besides the peak point for hiding the input message. The knowledge of the maximum point and the minimum point of the histogram is necessary for retrieving the hidden data and restoring the stego-image losslessly to the original state. In addition, the coordinates of the pixels whose gray values are equal to the gray value of the minimum point b need be memorized as overhead information when the value of b is not zero. Fallahpour [4] later proposed the idea of decomposing the entire cover image into blocks and using the peak point of the histogram of each block to hide data. Also, Hwang et al. [5] proposed the concept of slightly adjusting the pixel values located at both sides of a histogram peak to embed data. An advantage of this method is that it is unnecessary to record the knowledge of the location of the peak point because the peak location will not be changed after data hiding. But a location map is still required to store the information for restoring the cover image. The hiding capacity, when compared with that of Ni et al. [10], is smaller, and the peak signal-to-noise ratio (PSNR) of the stego-image gets worse in some cases [7]. Later, Kuo et al. [7], similarly to Fallahpour [4], used the block division technique, which is also employed in this study, to increase the hiding capacity of Hwang et al. [5].

An important characteristic of reversible data hiding methods based on histogram shifting is that more peaks imply higher hiding capacities. Therefore, in this study we try to explore the possibility of using a larger number of peaks, instead of just one, in a block to increase the data hiding capability and decrease the distortion in the stego-image. We propose a new reversible histogram-shifting data hiding method which uses an adaptive block division scheme for improving the data hiding capacity and stego-image quality. In the proposed method, each non-overlapping square block in a cover image is divided by four ways of sub-block decompositions. And the way providing the highest data hiding capacity is chosen adaptively. Compared with the existing histogram-shifting based methods, much larger hiding capacities with lower stego-image quality degradations can be achieved.

The remainder of this paper is organized as follows. In Section 1.2, more details of the proposed method are described. Experimental results and some discussions are included in Section 1.3. Conclusions are made finally in Section 1.4.

1.2 Adaptive Block Division for Histogram-shifting Based Data Hiding

As mentioned above, the use of more peaks yields larger hiding capacities. In Ni et al. [10], the number of pixels constituting the peak in the histogram of a cover image is equal to the hiding capacity because only a single peak in a cover image is used. As an improvement, Fallahpour [4] divided the cover image into blocks so as to generate a respective peak for each block. This technique of block division successfully enhances the hiding capacity because the total volume of data that can be hidden in the multiple blocks is generally larger than that which can be hidden in a single cover image, as mentioned previously. Furthermore, the location of the peak in the histogram indicates generally that a great number of pixels are 'centralized' in the neighboring area around the peak point. For this reason, Hwang et al. [5] used the two neighboring points beside the peak point to embed data. On the other hand, the block division technique is also used by Kuo et al. [7] for improving the performance of Hwang et al. [5]. A cover image of size 512×512 was divided into four blocks by Kuo et al. [5].

In this study, to see the trend of data-hiding-rate increasing by block divisions, we divide the cover image into equal-sized sub-blocks from size 256×256 to 2×2 and implemented the method of Kuo et al. [7] to test them. A surprising result was observed in the experimental results, that is, the hiding rate increases from the size of 256×256 through 8×8 and then turns to *decrease* from 4×4 to 2×2. Figure 1.1 shows an example of this trend by bar charts for the image "Lena". On the other hand, the trend of the PSNR values of the stego-image of Lena is also shown as red curve over the bar charts in Figure 1.1. This trend shows that the PSNR value keep *increasing* from the size of 256×256 all way down to 2×2, contrary to the intuition that hiding more data will result in worse stego-image quality. Similar trends of the data-hiding rates and the PSNR values found for some other test images and that of Lena together are summarized in Table 1.1.

From the above observation, the block size of 8×8 is seen to be the best choice for maximizing the data hiding capacity while minimizing the image distortion. However, this size may be contrarily regarded as a *bottleneck* in the data-hiding-rate increasing trend. It is desired to break this bottleneck in this

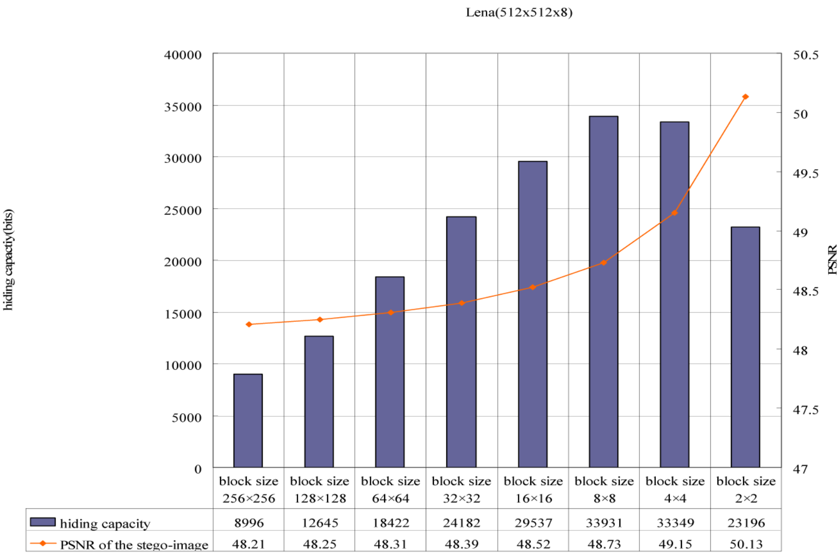


Figure 1.1 Trend of hiding capacities and PSNR values versus block sizes for image ‘Lena’

Table 1.1 Statistics of experimental results of implementation of Kuo et al.’s method [7] which show a bottleneck of data-hiding-rate increasing at block size 8×8

512×512 (No. of blocks)	256×256 (4)	128×128 (16)	64×64 (64)	32×32 (256)	16×16 (1024)	8×8 (4096)	4×4 (16384)	2×2 (65536)
Lena	8996	12645	18422	24182	29537	33931	33349	23196
(PSNR)	(48.21)	(48.25)	(48.31)	(48.39)	(48.52)	(48.73)	(49.15)	(50.13)
Airplane	25555	30296	35109	41780	46945	50920	49690	38718
(PSNR)	(48.38)	(48.43)	(48.54)	(48.68)	(48.86)	(49.15)	(49.71)	(50.94)
Baboon	6044	7731	9474	11454	12426	12909	12042	8309
(PSNR)	(48.18)	(48.20)	(48.23)	(48.27)	(48.33)	(48.46)	(48.78)	(49.65)
Tiffany	16409	21617	28875	35272	40449	44490	43022	30901
(PSNR)	(48.28)	(48.35)	(48.45)	(48.55)	(48.69)	(48.92)	(49.38)	(50.39)
Boat	12931	18897	26875	33562	38382	40379	38234	25955
(PSNR)	(48.25)	(48.32)	(48.41)	(48.50)	(48.62)	(48.81)	(49.19)	(50.13)

study, and the investigation result gives the answer of “*yes*” – by the proposed method the data hiding rate will be increased again beyond the size of 8×8! It is also found in the experimental results of the proposed method that the PSNR value still keeps going up with the decreasing block size, as will be shown later.

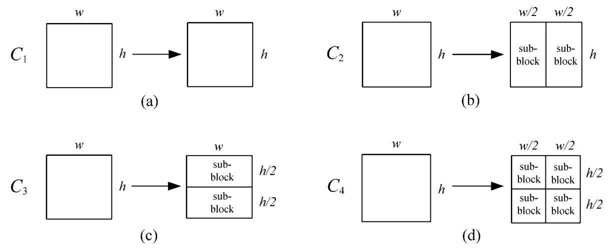


Figure 1.2 Four ways of block divisions

134	126	129	126	133	130	129	131
134	126	129	126	133	130	129	131
134	126	129	126	133	130	129	131
134	126	129	126	133	130	129	131
134	126	129	126	133	130	129	131
132	125	128	130	130	133	131	130
132	126	133	133	130	130	131	127
132	134	133	134	131	134	131	131

Figure 1.3 Pixel values of a block of size 8 × 8 in image 'Lena'

The basic concept behind the proposed method is to divide each 8×8 block by four ways into 4 × 8, 8 × 4, 4 × 4, and the originally-adopted size 8 × 8. The four ways of block divisions are illustrated in Figure 1.2. The best way among the four, which provides the largest volume of space for data hiding, is then chosen adaptively to yield the best data hiding capacity.

Figure 1.3 shows a block in the image 'Lena' which we use to illustrate a case that a block of 8 × 8 can provide a larger data hiding capacity after it is divided into two 4×8 sub-blocks. Originally, the peak point of the entire block is found at the gray value of 126 with a hiding capacity of only 2 bits. This can be seen from the fact that there are only two pixels with gray values 125 and 127 next to 126 at its two sides. But after dividing the block horizontally by the way of block division of C_3 shown in Figure 1.2, totally a hiding capacity of 13 bits can be generated from the upper and the lower sub-blocks with peak points 129 and 131, respectively, as can be seen from the gray values in the two sub-blocks where there are 13 pixels with gray values 128, 130, and 132, which are located next to 129 and 131.

In the following, the proposed method is described in details as two algorithms, the first for data embedding and the second for data extraction.

A *key* is generated by the data embedding algorithm (Algorithm 1). The key is used in the data extraction algorithm (Algorithm 2) for security control; without the key, the embedded message data cannot be extracted successfully.

Algorithm 1 – Data Embedding.
Input: a cover image I divided into blocks with size $n \times n$, and a message bit string M to be embedded.
Output: a stego-image I' with M embedded, a key K in the form of an integer number sequence, and a location map S_M.
Steps:

Step 1 (*Block decomposition*) divide each block B in I by the four ways C_1, C_2, C_3, and C_4, respectively, as shown in Figure 1.4.

Step 2 (*Estimation of data hiding capacities*) compute the data hiding capacity for each case C_i of block divisions, $i = 1, 2, 3$, and 4, by the following way.

  2.1 For each sub-block I_j in C_i, perform the following operations.

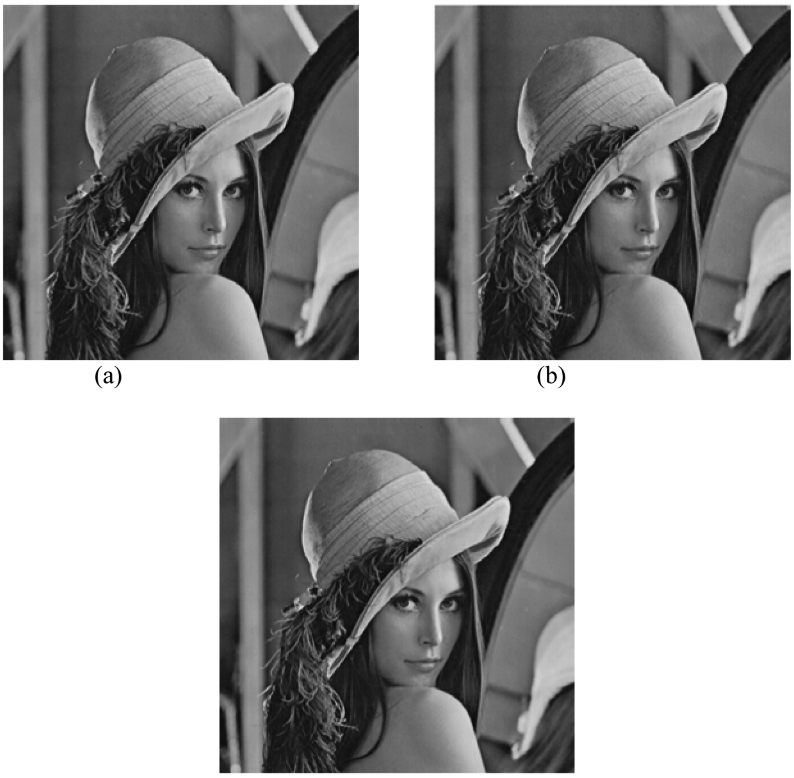
    (a) Generate the histogram h of I_j.
    (b) Find the peak in h and its location x_0.
    (c) Sum up $h(x_0 - 1)$ and $h(x_0 + 1)$ as d_j.

  2.2 Sum up all d_j to get the data hiding capacity D_i for C_i.

Step 3 (*Optimal decision*) select the C_i with the maximum capacity D_i among D_1 through D_4, denote it as C_m, and record integer m as a *component k* in the key K for the currently-processed block B.

Step 4 (*Data embedding*) for each sub-blocks I_j in C_m, perform the following operations.

  4.1 Generate the histogram h of I_j.
  4.2 Find the peak in h and its location x_0.
  4.3 Collect all pixels in I_j with gray values smaller than x_0 as a set S_L, and those with gray values larger than x_0 as a set S_R.
  4.4 Collect all pixels in S_L whose gray values are zero as a set S_L^0, and all those whose gray values are 255 in S_R as a set S_R^{255}.
  4.5 (*Histogram shifting*) decrement the gray value of each pixel in S_L by one except those in S_L^0, and increment the gray value of each pixel in S_R by one except those in S_R^{255}.
  4.6 (*Location map generation*) put S_L^0 and S_R^{255} into a set S_M, and call it a *location map*.



(a) (b) (c)

Figure 1.4 An example of experimental results. (a) Cover image of Lena. (b) Stego-image of Lena generated from Algorithm 1. (c) Losslessly-recovered image of Lena generated from Algorithm 2.

4.7 (*Bit embedding*) take a data bit m_ℓ sequentially from M, scan the pixels in I_j in a raster-scan order, take an unprocessed pixel with gray value v, and perform the following task, until all bits in M are exhausted:

(a) if $v = x_0 - 2$, then
increment v by one if $m_\ell = 1$ or keep v unchanged if $m_\ell = 0$;

(b) if $v = x_0 + 2$, then
decrement v by one if $m_\ell = 1$ or keep v unchanged if $m_\ell = 0$.

Step 5 (*Key generation*) concatenate the values k's of all the blocks in I in the block-processing order into an integer number sequence $k_1k_2k_3\cdots$ as the desired key K.

The proposed data extraction process is described in the following as an algorithm, which is basically a reverse process of Algorithm 1. The input includes the output data yielded by Algorithm 1 and the number of message data bits embedded in the stego-image.

Algorithm 2 – Data Extraction.
Input: a stego-image I' with blocks of size $n \times n$, the number N of message data bits, the location map S_M, and the key K.
Output: the message data M embedded in I' and the original cover image I losslessly recovered.
Steps:

Step 1 (*Initialization*) set $M = \varepsilon$ (an empty string).
Step 2 Take out the blocks in I' in order, and divide each of them, denoted by B', in the way of the block division specified by the corresponding component k in the key K.
Step 3 (*Data extraction*) for each sub-blocks I'_j in B', perform the following operations.

3.1 Generate the histogram h' of I'_j.
3.2 Find the peak in h' and its location x'_0.
3.3 (*Bit extraction*) scan the pixels in I'_j in a raster-scan order, take an unprocessed pixel with gray value v, and perform the following task, until all the N message bits are extracted:

(a) if $v = x'_0 - 2$ or $x'_0 + 2$, then extract a bit "0" and append it to the end of M;
(b) if $v = x'_0 - 1$, then extract a bit "1," append it to the end of M, and decrement v by one;
(c) if $v = x'_0 + 1$, then extract a bit "1," append it to the end of M, and increment v by one.

3.4 Collect all pixels in I'_j with gray values smaller than x'_0 as a set S'_L, and those with gray values larger than x'_0 as a set S'_R.
3.5 (*Reverse histogram shifting*) perform the following steps to recover the cover image I losslessly.

(a) Increment the gray value of each pixel in S'_L by one, and decrement the gray value of each pixel in S'_R by one.

(b) Use the content (S_L^0, S_R^{255}) of the location map S_M to restore the original gray values of the pixels by setting the gray value of each pixel in S_L^0 to be 0 and that of each pixel in S_R^{255} to be 255.

By Algorithm 2, the hidden data can be extracted successfully, and the original cover image recovered *losslessly*.

1.3 Experimental Results

Each test image used in our experiment is a grayscale one of the size 512×512 with gray values ranging from 0 through 255. For the purpose of comparing our results with those of other methods, we implemented additionally the algorithms of Ni et al. [10], Hwang et al. [5], and Kuo et al. [7].

As mentioned previously, as the size of blocks in each test image becomes smaller, the PSNR of the resulting stego-image keeps increasing but the data hiding capacity increases until a bottleneck at the block size of 8 × 8, as shown in Table 1.1 and as illustrated by Figure 1.1. Therefore, we processed the tested images, starting mainly from the block size of 8 × 8; and chose adaptively for each block one of the four sizes 8 × 8, 4 × 8, 8 × 4, and 4 × 4 to optimize the data hiding capacity, using Algorithms 1 and 2. Some experimental results are shown in Tables 1.2 and 1.3.

Table 1.2 Comparison of results of proposed method and Kuo et al.'s method [7]

Method		Kuo et al.'s method		Proposed method (initial block size of 8×8)	
Block size		Blocks of size 8×8		Combination of blocks with sizes 8×8, 8×4, 4×8 and 4×4	
Hiding capacity	Quality of stego-image	Bits	PSNR	Bits	PSNR
Lena (512×512)		33931	48.73	41257	48.90
Airplane (512×512)		50920	49.15	59397	49.37
Baboon (512×512)		12909	48.46	17455	48.57
Tiffany (512×512)		44490	48.92	52297	49.10
Boat (512×512)		40379	48.81	46833	48.95

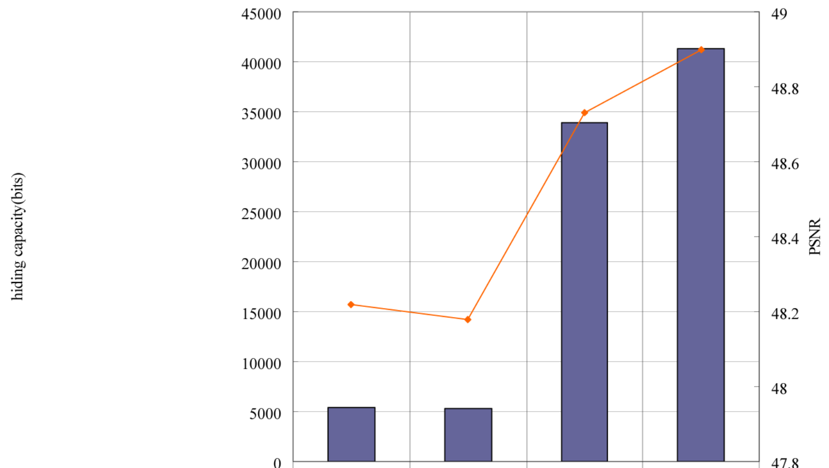
Table 1.3 Comparison of results of proposed method and related methods

Method		Ni's		Hwang's		Kuo's		Proposed method	
Image or block size		Image of size 512×512		Image of size 512×512		Blocks of size 8×8		Combination of blocks with sizes 8×8, 8×4, 4×8 and 4×4	
Hiding Capacity	Quality of stego-image	bits	PSNR	bits	PSNR	bits	PSNR	bits	PSNR
Lena (512×512)		5409	48.22	5304	48.18	33931	48.73	41257	48.90
Airplane (512×512)		18700	48.45	17502	48.28	50920	49.15	59397	49.37
Baboon (512×512)		5932	48.28	5793	48.18	12909	48.46	17455	48.57
Tiffany (512×512)		10153	48.30	9942	48.21	44490	48.92	52297	49.10
Boat (512×512)		10546	49.25	9709	48.21	40379	48.81	46833	48.95

As can be seen from Table 1.2, the resulting data hiding capacity and the PSNR value for each of the five test images are both raised, compared with those yielded by Kuo et al. [7]. For example, for the image of Lena, the bottleneck of the data hiding capacity is 33931 bits with the PSNR 48.73 dB, and the resulting capacity of the proposed method now is 41257 bits with the PSNR value being 48.90 dB. In Figure 1.4, the image processing results of the test image Lena are shown. Both the imperceptibility of the data hidden in the stego-image and the reversibility of the proposed method (i.e., the capability of *lossless* data recovery) can be verified from the images.

A more complete comparison of our results with those the related methods, including Ni et al. [10] and Hwang et al. [5], in addition to Kuo et al. [7], is shown in Table 1.3. As can be seen, the proposed method produces better results both in hiding capacities and in image qualities. For a clearer visualization of the data included in Table 1.3, we use a bar-chart diagram to show the results of image Lena in Figure 1.5 as an example.

It also can be observed from the above experimental data that the data hiding rates are increased by the proposed method without degrading the



	Ni's method	Hwang's method	Kuo's method	Proposed method
hiding capacity	5409	5304	33931	41257
the PSNR value of the stego-image	48.22	48.18	48.73	48.9

Figure 1.5 Illustration of comparison of hiding capacity versus image distortion in mentioned methods using image ‘Lena’

stego-image quality expressed by the peak signal-to-noise ratio measure, which is an unusual phenomenon in data hiding research results.

1.4 Conclusions

A lossless data hiding method based on histogram shifting using an adaptive block division scheme has been proposed, which not only embeds large-volume data into cover images but also produces stego-images with high qualities. The bottleneck for data-hiding-rate increasing tendency at the block size of 8×8 found in the existing methods is broken by the proposed method. Experimental results show the effectiveness of the proposed method. Future researches may be directed to investigating more block division types and recursive division schemes for further improvement on the data hiding capacity, applying the histogram shifting technique to other information hiding methods and applications, etc.

References

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding, *IBM Systems Journal*, 35(3–4):313–336, 1996.

[2] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber. Reversible data hiding. In *Proceedings International Conference on Image Processing*, Vol. 1, pages 71–76, 2002.

[3] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber. Lossless generalized-LSB data embedding, *IEEE Trans. Image Process.*, 14(2):253–266, 2005.

[4] M. Fallahpour and M.H. Sedaaghi. High capacity lossless data hiding based on histogram modification. *IEICE Electron. Express*, 4(7):205–210, 2007.

[5] J.H. Hwang, J.W. Kim, and J.K. Choi. A reversible watermarking based on histogram shifting. *Proceedings IWDW 2006*, Lecture Notes in Computer Science, Vol. 4283, pages 348–361. Springer, 2006.

[6] H.J. Kim, V. Sachnev, Y.Q. Shi, J. Nam, and H.G. Choo. A novel difference expansion transform for reversible data embedding. *IEEE Trans. Information Forensics and Security*, 3:456–465, 2008.

[7] W.C. Kuo, D.J. Jiang, and Y.C. Huang. A reversible data hiding scheme based on block division. In *Proceedings CISP 2008*, Vol. 1, pages 365–369, 2008.

[8] A. Lumini and D. Maio. A wavelet-based image watermarking scheme. In *Proceedings International Conference on Information Technology*, 2000.

[9] J. Mielikainen. LSB matching revisited. *IEEE Signal Processing Letters*, 13(5):285–287, 2006.

[10] Z. Ni, Y.Q. Shi, N. Ansari, and W. Su. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.*, 16(3):354–362, 2006.

[11] S. Pereira and T. Pun. Robust template matching for affine resistant image watermarks. *IEEE Trans. Image Process.*, 9:1123–1129, 2000.

[12] C.-M. Pun. A novel DFT-based digital watermarking system for images. *Proc. Signal Processing*, 2, 2006.

[13] J. Tian. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.*, 13(8):890–896, 2003.

[14] S.-H. Wang and Y.-P. Lin. Wavelet tree quantization for copyright protection watermarking. *IEEE Trans. Image Process.*, 13:154–165, 2004.

[15] Y.N. Wang and A. Pearmain. Blind image data hiding based on self reference. *Pattern Recognition Lett.*, 25:1681–1689, 2004.

[16] Y.P. Wang, M.-J. Chen, and P.-Y. Cheng, Robust image watermark with wavelet transform and spread spectrum techniques. In *Proceedings Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, Vol. 2, pages 1846–1850, 2000.

[17] D.C. Wu and W.H. Tsai. A steganographic method for images by pixel-value differencing. *Pattern Recognition Lett.*, 24(9–10):1613–1626, 2003.

[18] 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. *Proc. Inst. Elect. Eng., Vis. Images Signal Process.*, 152(5):611–615, 2005.

[19] C.-H. Yang, C.-Y. Weng, S.-J. Wang, and H.-M. Sun. Adaptive data hiding in edge areas of images with spatial LSB domain systems. *IEEE Trans. Information Forensics and Security*, 3(3), 2008.