Image Reversible Data Hiding Based on Optimum Histogram Pair

Guorong Xuan[1], Yun Q. Shi[2], Peiqi Chai[1] , JianzhongTeng[1], Zhicheng Ni[3], Xuefeng Tong[1]
[1]Dept. of Computer Science, Tongji University, Shanghai, China
[2]Dept. of ECE, New Jersey Institute of Technology, Newark, New Jersey, USA
[3]LSI Corporation, Allentown, Pennsylvania, USA
grxuan@public1.sta.net.cn, shi@njit.edu

Abstract: This paper presents an optimum histogram pair based image reversible data hiding scheme using integer wavelet transform and adaptive histogram modification. This new scheme is characterized by (1) the selection of best threshold T, which leads to the highest PSNR of marked image for a given payload; (2) the adaptive histogram modification, which aims at avoiding underflow and/or overflow, is carried out only when it is necessary, and treats the left side and right side of histogram individually, seeking a minimum amount of histogram modification; and (3) the selection of most suitable embedding region, which attempts to further improve the PSNR of marked image in particular when the payload is low. Consequently, to our best knowledge, it can achieve the highest visual quality of marked image for a given payload as compared with the prior arts of image reversible data hiding. The experimental results have been presented to confirm the claimed superior performance.

Keywords: optimum histogram pair, reversible (lossless) data embedding, integer wavelets, selection of best threshold, adaptive histogram modification, selection of suitable embedding region

1. Introduction

Reversible data embedding, also often called lossless data hiding, requires that not only hidden data can be extracted correctly but also the marked image be inverted back to the original cover image exactly after the hidden data has been extracted out. Recently, Ni et al. [1] proposed the reversible data embedding algorithm based on the spatial domain histogram shifting. Tian[2] proposed the difference expansion method that can achieve a high payload, but it can only be used with integer Haar wavelet. Kamstra and Heijmans [3] improved the PSNR of marked image achieved by Tian in the case of small payload. Xuan et al.[4] proposed the thresholding reversible embedding using the integer wavelet transform (IWT) and histogram modification. In [5], some improvements over [4] have been made about the selection of threshold. However, the three optimality measures taken in this paper have not been developed in [5]. In [6], Yang et al. applied the histogram shifting embedding to the integer discrete cosine transform. The reversible data hiding scheme proposed in this paper♦

♦ This research is supported partly by National Natural Science Foundation of China (NSFC) on the project (90304017).

is based on optimum histogram pairs. It is characterized by selection of optimum threshold T: most suitable embedding region R; and minimum possible amount of histogram modification G, in order to achieve highest PSNR of the stego image for a given data embedding capacity.

The rest of this paper is organized as follows. The principle of reversible data embedding based on histogram pairs is illustrated in Section 2. Integer wavelets and histogram modification are briefly discussed in Section 3. The proposed reversible data hiding algorithm is presented in Section 4 and 5. Section 6 provides experimental results. The performance of three newest algorithms published in 2007 [7,8,9] is cited for comparison. Discussion and conclusion are presented in Section 7.

2. Principle of Histogram Pair

2.1. Reversible data embedding using histogram pair

Histogram h(x) is the number of occurrence as the variable X assumes value x. Since digital image and IWT are considered in this paper, we assume X can only assume integer values. In order to illustrate the concept of histogram pair, we first consider a very simple case. That is, only two consecutive integers *a* and *b* assumed by X are considered, i.e. $x \in \{a,b\}$. Furthermore, let h(a)=m and h(b)=0. We call these two points as a histogram pair, and sometimes denote it by, h=[m,0], or simply [m,0]. Furthermore, we assume m=4. That is, X actually assumes integer value a four times, i.e., X= [a, a, a, a]. Next, let's see how we can reversibly embed bits into X= [a, a, a, a]. Suppose the to-be-embedded binary sequence is D=[1, 0, 0, 1]. In data embedding, we scan the 1-D sequence X= [a, a, a, a] in certain ordering, say, from left to right. When we meet the first a, since we want to embed bit 1, we change a to b. For the next two to-be-embedded bits, since they are bit 0, we do not change a. For the last to-be-embedded bit 1, we change a to b. Therefore, after the four-bit embedding, we have X=[b, a, a, b], and the histogram is now h= [2, 2]. Embedding capacity is C=4. The hidden data extraction, or histogram pair recovery, is the reverse process of data embedding: after extracting the data D=[1, 0, 0, 1], the histogram pair becomes [4, 0] and we can recover X=[a, a, a, a] reversibly.

We define histogram pair here. If for two consecutive non-negative values a and b that X can assume, we have h(a)=m and h(b)=n, where m and n are the numbers of occurrence for x=a and x=b, respectively. When n=0, we call h=[m,n] as a histogram pair. From the above example, we observe that when n=0, we can use this histogram pair to embed data reversibly. We call n=0 in the above defined histogram pair as an "expansion" element, which is ready for reversible data embedding. During the embedding, we scan all of x values, a, in certain order. If bit 1 is to be embedded, we change the a under scanning to b, otherwise, we keep the a unchanged. If a is a negative integer, then h = [m, n] is a histogram pair as $m = 0$ and $n \neq 0$.

2.2. An illustrative example

In this example, D=[1,0,0,1,0,1,1,0,0,1,1,0,1,0,0,1], a 16-bit sequence is to be embedded into a 4×4 image by using the histogram pair scheme. For this image $x \in \{a,b,c,d\}$, and a,b,c,d are four consecutive integers. Assume the image's histogram

is h=[16,0,0,0] as shown in Figure 1(a). We noticed that the two numbers, 16 and 0 having been underlined, form a histogram pair. Let's scan the image from left to right, and from top-to-bottom. The same rule is obeyed, i.e., when bit 0 is to be embedded, a remains, when bit 1 is to be embedded, a changes to b. Figure 1 displays images and histograms before and after the first loop of data embedding, while Figure 2 for the 2nd loop embedding.

Original histogram					After embedding			
h=[16,0,0,0]					h=[8,8,0,0]			
a	a	a	a		b	a	a	b
a	a	a	a		a	b	b	a
a	a	a	a		a	b	b	a
a	a	a	a		b	a	a	b

(a) original image (b) embedding

Figure 1 Image and histogram for 1st loop embedding.

Expansion					Embedding			
h=[8,0,8,0]					h=[4,4,4,4]			
c	a	a	c		d	a	b	c
a	c	c	a		b	c	d	a
a	c	c	a		b	c	d	a
c	a	a	c		d	a	b	c

(a) expansion (b) embedding

Figure 2 Image and histogram for 2nd loop embedding.

1st loop: There is one histogram pair h= [16,0] that can be used. After embedding, the histogram pair [16,0] changes to [8,8], the histogram changed from h=[16,0,0,0] to h=[8,8,0,0], see Figure 1. The scanned image pixel value sequence changes from X=[a,a,a,a, a,a,a,a, a,a,a,a, a,a,a,a] to X=[b,a,a,b, a,b,b,a, a,b,b,a, b,a,a,b] as shown in Figure1(b). After this loop, 16 bits have been embedded.

2nd loop: Since the resultant pair [8,8] cannot be used to embed data, we first expand the histogram by change b to c in the image, resulting in the new histogram h=[8,0,8,0] as shown in Figure 2. Now, there are two histogram pairs, [8,0] (left) and [8,0] (right). After data embedding, the histogram changes from h=[8,0,8,0] to h=[4,4,4,4], see Figure 2. Both histogram pairs change from [8,0] to [4,4]. The scanned image pixel value sequence changes from X=[c,a,a,c, a,c,c,a, a,c,c,a, c,a,a,c] to X=[d,a,b,c, b,c,d,a, b,c,d,a, d,a,b,c] as shown in Figure2. After data embedding in 2nd loop, another 16 bits are embedded. In total, we have embedded $2\times16=32$ bits into this 4×4 image by these two loops.

2.3. Upper bound of data embedding capacity

The above example has indicated that multiple-loop embedding can embed more data. However, the embedding capacity has an upper bound. When bit 0 and bit 1 are equally distributed among the to-be-embedded sequence, the upper bound of embedding capacity can be calculated. Assume the number of image pixels (or wavelet coefficients) is N, then at most N bits can be embedded for each embedding

loop. Assume the number of different x values is M. The number of loops, r, can be calculated by r = log(M), i.e., M = 2^r. Then, after r loop embedding, the capacity becomes L = r×N. If after r loop embedding, the histogram becomes flat, it means the maximum capacity has been achieved. In the example in Section 2.2, N=16, M=4, hence it can embed r = 2 loops. After twice embedding, in the example, h= [4,4,4,4], i.e., histogram becomes completely horizontal, indicating no data can be embedded any more. In this case, the maximum capacity has been reached, which is L = r×N = 2×16 =32 bits.

2.4. Information theory

Histogram expansion and data embedding will cause the histogram change from up-and-down to relatively more flat. In this way, the entropy increases. On the other hand, data extraction will lead to the opposite, and the entropy decreases to its original value. This process can be shown below.

The entropy can be expressed as $H(x) = -\int h(x)\log(h(x))dx$. It can be proved according to information theory that the probability distribution h(x) will be uniform within a limited range u~v when the maximum entropy H(x) is achieved:

$$h(x) = \arg\max_{h(x)}\left[H(x) - \lambda\left(1 - \int_u^v h(x)dx \right)\right] = \frac{1}{v-u}$$

Proof:

If $d\left[H(x) - \lambda\left(1 - \int_u^v h(x)dx \right)\right] \Big/ dx = 0$, we have $-(1+\log h(x)) + \lambda = 0$. Consider the condition of probability distribution $\int_u^v h(x)dx = 1$, the solution $h(x) = \frac{1}{v-u}$ is obtained.

In summary, after data embedding, the entropy increases and histogram becomes more flat. If the histogram is absolutely flat, the total entropy H(x) is maximum, no more data can be embedded.

3. Integer Wavelets and Histogram Modification

3.1. Integer wavelet transform (IWT)

In this proposed method, data is hidden into IWT coefficients of high-frequency subbands. The motivation of doing so is as follows. (1) The high frequency subband coefficients represent the high frequency components of the image. Human visual system (HVS) is less sensitive to high frequency. Hence, data embedding into high frequency subbands can lead to better imperceptibility of marked image. (2) The histogram distribution of high-frequency subbands is Laplacian-like with a huge peak around zero. This makes high data embedding capacity feasible. (3) Owing to the de-correlation property among the wavelet subbands in the same decomposition level, data embedding using IWT results in higher PSNR than embedding into other transform coefficients such as DCT.

Due to the reversibility constraint, we choose to use integer wavelet transform. Specifically, the integer Haar wavelet transform and integer (5,3) wavelet transform are used in our experimental works. Other types of IWT are usable too. The results

are shown in Section 5, from which we observe that both perform well, however, the integer (5,3) transform performs better, while the integer Haar transform is more simple in implementation.

3.2. Formulas of wavelet transform

Formulas for the above-mentioned two integer wavelet transforms are listed in Table 1.

Table 1 Integer wavelet transform.

Note that $\lfloor x \rfloor$ rounds x to the largest integer not larger than x.

Type of wavelet transform		Formulas	
Integer Haar wavelet transform	Forward transform	Splitting: Dual lifting: Primary lifting:	$s_i \leftarrow x_{2i}$; $d_i \leftarrow x_{2i+1}$ $d_i \leftarrow d_i - s_i$ $s_i \leftarrow s_i + \lfloor d_i/2 \rfloor$
	Inverse transform	Inverse primary lifting: Inverse dual lifting: Merging:	$s_i \leftarrow s_i - \lfloor d_i/2 \rfloor$ $d_i \leftarrow d_i + s_i$ $x_{2i} \leftarrow s_i$; $x_{2i+1} \leftarrow d_i$
Integer (5,3) wavelet transform	Forward transform	Splitting: Dual lifting: Primary lifting:	$s_i \leftarrow x_{2i}$; $d_i \leftarrow x_{2i+1}$ $d_i \leftarrow d_i - \lfloor (s_i + s_{i+1})/2 \rfloor$ $s_i \leftarrow s_i + \lfloor (\hat{d}_{i-1} + \hat{d}_i)/4 \rfloor$
	Inverse transform	Inverse primary lifting: Inverse dual lifting: Merging:	$s_i \leftarrow s_i - \lfloor (\hat{d}_{i-1} + \hat{d}_i)/4 \rfloor$ $d_i \leftarrow d_i + \lfloor (s_i + s_{i+1})/2 \rfloor$ $x_{2i} \leftarrow s_i$; $x_{2i+1} \leftarrow d_i$

3.3. Histogram modification

For a given image, after data embedding into some IWT coefficients, it is possible to cause *underflow* and/or *overflow*, which means that the grayscale values may exceed the range [0, 255] for an 8-bit gray image, thus possibly violating the losslessness constraint. In order to prevent this from happening, we adjust histogram, i.e., we shrink the histogram from one or both sides towards the central portion. In narrowing down a histogram, we need to record the histogram modification for late recovery of the original image, thus resulting in this bookkeeping data, which is to be embedded into the image in addition to the pure payload. That is, both the pure payload and the bookkeeping data are embedded into the image. Instead of adjust histogram as a preprocessing as done in [4] and [5] (meaning that histogram modification is always done at the beginning no matter if necessary or not), we do it only when this is necessary (meaning that the adjustment is done adaptively in data embedding when it is necessary), furthermore the left side and right side of histogram is treated individually. More discussion in this regard will be given below.

If the overflow occurs (at the right side, i.e., the grayscale value is larger than 255), the right end of the histogram will be shrunk towards the center by an amount GR. If the underflow occurs (at the left side, i.e., the grayscale value is smaller than 0), the left end of the histogram will be shrunk towards the center by an amount GL.

Together, the histogram is shrunk by an amount of G, and G=GL+GR. The histogram is narrowed down from "0 to 255" to "GL to (255-GR)". This histogram shrinking uses the histogram pair principle described above as well, specifically it is the reverse process of data embedding (which expands the histogram). This new dynamic way contributes to the superior performance over that of [4] and [5] and will be shown in Sections 5 and 6. There, it is observed that it may not need to do histogram modification for some images with some payloads. When the embedding capacity increases, we may need histogram modification. In addition, the amount of histogram modification for the right side and the left side of histogram may be different. All of these have been implemented with an efficient computer program in this proposed scheme. Figure 3 shown below is an illustration of histogram modification.

4. Proposed Reversible Data Hiding Algorithm

In this section, prior to presenting our proposed new method, we first discuss thresholding method developed in our previous work because it is relevant.

4.1. Thresholding method

To avoid the possible underflow and/or overflow, often only the wavelet coefficients with small absolute value are used for data embedding [4]. This is the so-called thresholding method, which first sets the threshold T depending on the payload and embeds the data into those IWT coefficients with $|x| \leq T$. It does not embed data into the wavelet coefficients with $|x| > T$. Furthermore, for all high frequency wavelet coefficients with $|x| > T$ we simply add T or –T to x depending x is positive or negative so as to make their magnitude larger than 2T. In this way, the data embedding into coefficients with $|x| \leq T$ will not be confused with the large coefficients in which there is no data embedding. Therefore, the so-called thresholding method [4] is in fact the *minimum* thresholding method. The formulas of the thresholding method are shown in Table 2.

Table 2 Formulas of thresholding method for reversible data hiding.

x': gray levels after embedding, $\lfloor x \rfloor$: rounds x to the largest integer not larger than x.

Parts of histogram	Embedding		Recovering	
	after embedding	condition	after recovering	condition
Data embedded region (right side) (positive or zero)	x'=2x+b	x ≤ T	x = $\lfloor (x')/2 \rfloor$, b = x'-2x	x' ≤ 2T-1
Data embedded region (left side) (negative)	x'=2x-b	-T ≤ x	x = $\lfloor (x'+1)/2 \rfloor$ b = x' -2x	-2T-1 ≤ x'
No data embedded region (right edge part) (positive)	x'=x+T+1	x > T	x = x'-T-1	x' > 2T+1
No data embedded region (left edge part) (negative)	x'=x-T-1	x < -T	x = x'+T+1	x' < -2T-1

4.2. Optimum thresholding method based on histogram pairs

As shown below, however, the minimum threshold T does not necessarily lead to the highest PSNR of mark image for a given payload. (This was also reported in [5].) The reason is as follows. If a smaller threshold T is selected, the number of coefficients with |x| > T will be larger. These coefficients need to be moved away from the center of histogram by (T+1) (refer to Table 2) in order to create histogram pairs to embed data. This may lead to a lower PSNR owing to moving a larger end part of the histogram. On the other hand, if a larger T is selected, more coefficients having larger magnitude are to be changed for data embedding, possibly resulting in a lower PSNR of the marked image. Instead of arbitrarily picking up some threshold T (as the starting point for data embedding) and some stopping point S for stopping data embedding as done in [5], it is found that for a given data embedding capacity there does exists an optimum value for T. In this proposed optimum histogram pair reversible data embedding, the best threshold T for a given data embedding capacity is searched with computer program automatically and selected to achieve the highest PSNR for marked image. This will be discussed in Section 5.

The proposed method divides the whole histogram into three parts: (1) the1st part where data is to be embedded; (2) central part – no data embedded and the absolute value of coefficients is smaller than that in the 1st part; (3) the end part – no data embedded and the absolute value of coefficients is larger than that in the 1st part. The whole embedding and extraction procedure can be expressed by the formulae in Table 3. There T is the selected threshold, i.e., start position for data embedding, S is stop position, x is feature (wavelet coefficient) values before embedding, u(S) is unit step function (when S≥ 0; u(S) = 1; when S < 0; u(S) = 0); $\lfloor x \rfloor$ rounds x to the largest integer not larger than x. A simple example is presented in Section 4.5 to illustrate these formulas.

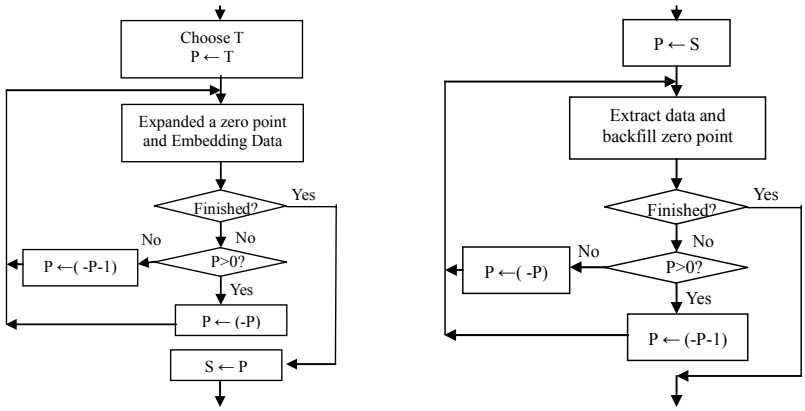
Table 3 Formulas of optimum thresholding method of reversible data hiding.

parts of histogram	Embedding		Recovering	
	after embedding	condition	after recovering	condition
Data embedded region (right side) (positive or zero)	x'=2x+b-|S|	|S| ≤x ≤T	$x=\lfloor(x'+\|S\|)/2\rfloor$, b=x'+|S|-2x	|S| ≤ x'≤ 2T-1-|S|
Data embedded region (left side) (negative)	x'=2x-b +|S|+u(S)	-T ≤ x ≤ -|S|-u(S)	x= $\lfloor(x'-\|S\|-u(S)+1)/2\rfloor$ b=x'-|S|-u(S)-2x	-2T-1+|S|+u(S) ≤ x'≤-|S|-u(S)
No data embedded region (central part) (small absolute value)	x'=x	-|S|-u(S)<x<|S|	x=x'	-|S|-u(S)<x'<|S|
No data embedded region (right edge part) (positive)	x'=x+T+1-|S|	x>T	x=x'-T-1+|S|	x'> 2T+1-|S|
No data embedded region (left edge part) (negative)	x'=x-T-1 +|S|+u(S)	x<-T	x=x'+T+1-|S|-u(S)	x'< -2T-1 +|S|+u(S)

4.3. Data embedding algorithm

The high frequency subbands (HH, HL, LH) coefficients of IWT are used for data embedding in this proposed method. Assume the number of bits to be embedded is L. The data embedding steps are listed below.

(1) For a given data embedding capacity, apply our algorithm to the given image, to search for an optimum threshold T as shown in Figure 7 in Section 5. And set the *P* ← T. Where T is like a starting value.



(a) embedding (b) extracting

Figure 4 Flowchart of proposed reversible data embedding and extracting.

(2) In the histogram of high frequency wavelet coefficients, move the portion of histogram with the coefficient values greater than P to the right-hand side by one unit to make the histogram at $P+1$ equal to zero (call $P+1$ as a zero-point). Then embed data in this point.

(3) If some of the to-be-embedded bits have not been embedded yet, let $P \leftarrow (-P)$, and move the histogram (less than P) to the left-hand side by 1 unit to leave a zero-point at the value ($-P$-1). And embed data in this point.

(4) If all the data have been embedded, then stop embedding and record the P value as the stop value, S. Otherwise, $P \leftarrow (-P-1)$, go back to (2) to continue to embed the remaining to-be-embedded data, where S is a stop value. If the sum of histogram for x $\in$ [-T,T] is equal L, the S will be zero.

4.4. Data extraction algorithm

The data extraction is the reverse of data embedding. Without loss of generality, assume the stop position of data embedding is S, S > 0.. Steps are as follows.

(1) Set $P \leftarrow$ S.

(2) Decode with the stopping value P. Extract all the data until P+1 becomes a zero-point. Move all the histogram greater than P+1 towards the left-hand by one unit to cover the zero-point.

(3) If the extracted data is less than L, set $P \leftarrow (-P-1)$. Continue to extract data until it becomes a zero-point in the position (P-1). Then move histogram (less than P-1) towards the right-hand side by one unit to cover the zero-point.

(4) If all the hidden bits have been extracted, stop. Otherwise, set $P \leftarrow -P$, go back to (2) to continue to extract the data.

4.5. A simple yet complete example

In this simple yet complete example, the to-be-embedded bit sequence D=[1 10 001] has six bits and will be embedded into an image by using the proposed histogram pair scheme with threshold T=3, and stop value S= 2. The image 5×5 shown in Figure 5 (a) has 12 distinct feature (grayscale) values, i.e., x ∈ {-5,-4,-3,-2,-1,0,1,2,3,4,5,6}.

0	4	0	-4	1
0	2	-2	3	-1
4	-3	0	2	-3
-1	-2	0	-1	0
-2	1	2	-1	1

(a)

0	6	0	-5	1
0	2	-2	4	-1
6	-3	0	2	-3
-1	-2	0	-1	0
-2	1	2	-1	1

(b)

0	6	0	-5	1
0	2	-2	5	-1
6	-4	0	2	-3
-1	-2	0	-1	0
-2	1	3	-1	1

(c)

Figure 5 5×5 wavelet subband (a) original one, (b) after 3 expanding, (c) after 6-bit embedding (what marked is how the last 3 bits are embedded).

The grayscale values of this image have the histogram h_0= [0,1,2,3,4,6,3,3,1,2,0,0] (as shown in 1st row of Figure 6). As said before, for x ≥ 0, the histogram pair is of form h=[m,0], for x<0, the histogram pair is h=[0,n]. The 2nd row of Figure 6 is expanded image histogram: h_1 (expanded), it has three histogram pairs. The 1st

histogram pair is in the far-right-hand side h= [1,0]; the 2nd histogram pair is in the left-hand side h= [0,2]; the 3rd histogram pair is in the right-hand side near the center h= [3,0]. The 3rd row of Figure 6 is the image histogram after data embedding: h_2 (bits embedded).

Figure 6 and Table 5 use solid (orange) line squares to mark the third histogram pair. The first histogram pair [1,0] is used to embed the 1st bit 1, the second histogram pair[0,2] is used to embed the next two bits 1,0 , and the third histogram pair [3; 0] is used to embed three bits: 0,0,1.

During expanding, we are first making h(4)=0, then making h(-4)=0, finally making h(3)=0. Note that h(3) = 0 at this time makes h(4) = 0 "shifting" to h(5) = 0. During each zero-point creation, the histogram shifting towards one of two (left and right) ends is carried out, the resultant histogram becomes h_1 = [1,0,2,3,4,6,3,3,0,1,0.2] (refer to Figure 5(b) and 2nd row of Figure 6). There histogram pairs are thus produced: in the right-most h=[1,0], in the left h=[0,2] and in the right (near center) h=[3,0].

After data embedding with bit sequence D=[1 10 001] and the selected scanning order, the histogram becomes h_2 =[1,1,1,2,4,6,3,2,1,0,1,2] (refer to Figure 5(c) and 3rd row of Figure 6). The three histogram pairs changed: in the right most from h=[1,0] to h=[0,1], in the left from h=[0,2] to h=[1,1], and in the right (near center) from h=[3,0] to h=[2,1].

After embedding, the grayscale values changed too. For example, embedding the last three bits (001) causes the right histogram pair (near center) to change from h=[3,0] to h=[2,1], and three grayscale values marked with small rectangles to change from X=[2, 2, 2] to X=[2, 2, 3] (refer to Figure 5 (c) and 3rd row of Figure 6).

Through this example, it becomes clear that the threshold can also be viewed as the starting point to implement histogram pair reversible data hiding. The formulas associated with this example are shown in Table 4, which provides a specific illustration of general formulas in Table 3.

Table 4 Formulas of reversible data hiding in the example in Section 4.5.

	Embedding		Recovering	
	after embedding	condition	after recovering	condition
central	x'=x x'=[-2,-1,0,1]	if -2<x<2 x=[-2,-1,0,1]	x=x' , x=[-2,-1,0,1]	if-2-u(S)<x'<|S| x'=[-2,-1,0,1]
right end	x'=x+2, x'=[6]	If x>3 x=[4]	x=x'-2, x=[4]	If x'> 5, x'=[6]
left end	x'=x-1, x'=[-5]	If x<-3 x=[-4]	x=x'+1, x=[-4]	If x'< -4 , x'=[-5]
right to-be-embedded	x'=2x+b-2 b=0: x'=[2,4] b=1: x'=[3,5]	if 2≤x≤3 b=0: x=[2,3] b=1: x=[2,3]	x=floor((x'+2)/2),b=x'+2-2x b=0: x=[2,3] b=1: x=[2,3]	if 2≤x'≤5 b=0: x'=[2,4] b=1: x'=[3,5]
left to-be-embedded	x'=2x-b+3 b=0: x'=[-3] b=1: x'=[-4]	if -3≤x≤-3 b=0: x=[-3] b=1: x=[-3]	x=floor((x'-2)/2),b=x'-3 -2x b=0: x=[-3] b=1: x=[-3]	if -4≤x'≤-3 b=0: x'=[-3] b=1: x'=[-4]

Table 5 Formulas of histogram pair data embedding example (T=3，S= +2).
(6 bit sequence D=[1 10 001]) (What marked is how the last 3 bits are embedded.)

X	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
h_0 (original)		1	2	3	4	6	3	3	1	2		
h_1 (extended)	1	0	2	3	4	6	3	3	0	1	0	2
h_2 (embedded)	1	1	1	2	4	6	3	2	1	0	1	2
embedded (ordering)	no em-bed-ding	[1 0] embed -ded (second)		no embedding				[001] em-bedded (third)		[1] em-bedded (first)		no embed-ding

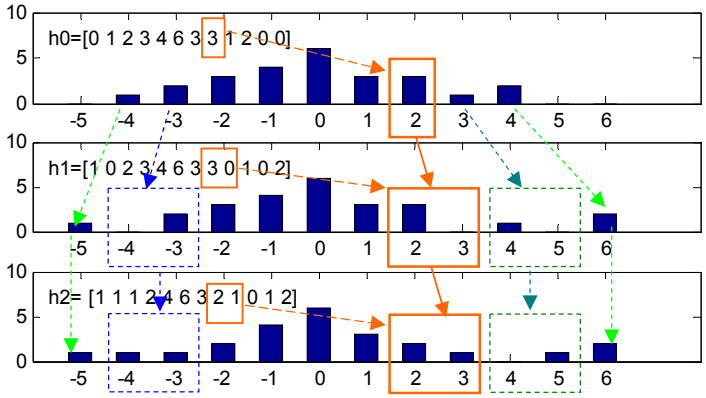


Figure 6 Histogram pair data embedding example (T=3，S= +2).
6 bit sequence D=[1 10 001]. What marked is how the last 3 bits are embedded.

4.6. Data embedding capacity

Data embedding capacity L can be calculated as follows. When the stopping value S is negative： $L=\sum_{-T}^{S} h(X)+\sum_{-S}^{T} h(X)$; when S is positive or zero： $L=\sum_{-T}^{-S-1} h(X)+\sum_{S}^{T} h(X)$; and when S is zero： $L=\sum_{-T}^{-1} h(X)+\sum_{0}^{T} h(X)=\sum_{-T}^{T} h(X)$.

5. Selection of Optimum Parameters

For a given required data embedding capacity, the proposed method selects the optimum parameter to achieve the highest possible PSNR. The optimum parameters include: the best threshold T, the adaptive histogram modification value, G (in spatial domain), and the suitable data embedding region R. That is, optimal parameters can be selected as follows.

$$[T,G,R]=\arg\max_{T,G,R}\left(PSNR\right)$$

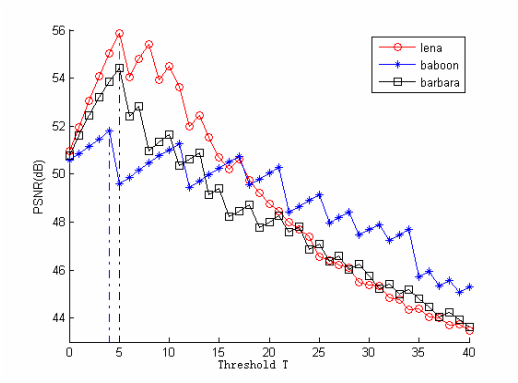


Figure 7 Selection of the best threshold T.

(1) Best threshold T: Figure 7 shows the PSNR of marked image vs the threshold T with embedding capacity 0.02 bpp. It is observed there does exist an optimal threshold T for each of the three commonly used images: Lena, Baboon and Barbra.

(2) Adaptive modification value G: the histogram modification is carried out in this method adaptively "on real time", instead of as a necessary preprocessing in [4][5]. That is, after data embedding into each wavelet coefficient, underflow and/or overflow is checked. If underflow and/or overflow occurs, and it occurs from the left side (<0), the left end of the histogram will be shrunk towards the center by an amount GL. GR is similarly handled. Our experiments have shown that on three frequently used test images: Lena, Barbra and Baboon, only when the embedding data rate is high than certain amount it needs histogram modification (G>0). Otherwise, there is no need for histogram modification. This adaptive histogram modification leads to the higher PSNR of marked image for a given payload.

(3) Suitable data embedding region R: in order to improve the PSNR when the payload is small (e.g., <0.1 bpp), we choose to only embed data into the HH subband , i.e., R={HH}. When the payload is large, all three high frequency subbands are used, i.e., R={HH,HL,LH}. This measure further enhances the PSNR for a given payload. Our experimental works have shown that the PSNR achieved by this new method is obviously better than that by the method [5] when the payload is rather small.

6. Experiments

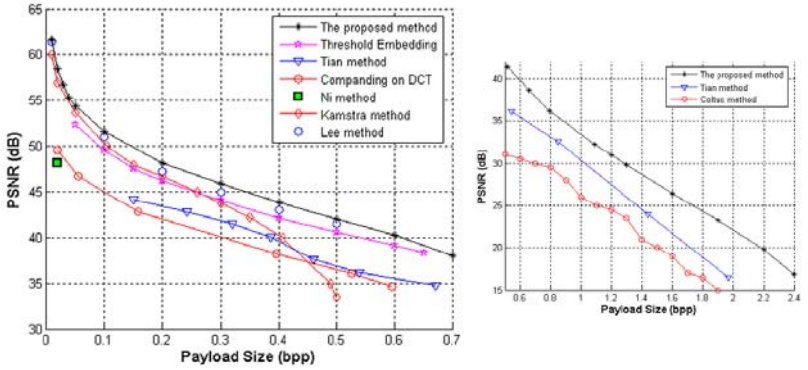
In this section, we first present our experimental results by using the proposed method, which consists of two parts. The first part is experiments on three commonly used images, i.e., Lena, Barbra and Baboon images. The second part is one of the JPEG2000 test images, i.e., Woman image. Then, the results are compared with prior arts. Finally experimental results using Haar wavelet transform and using (5,3) integer wavelet transform are presented and compared.

6.1. Experimental results and performance comparison

6.1.1. On three commonly used test images: Lena, Barbra and Baboon

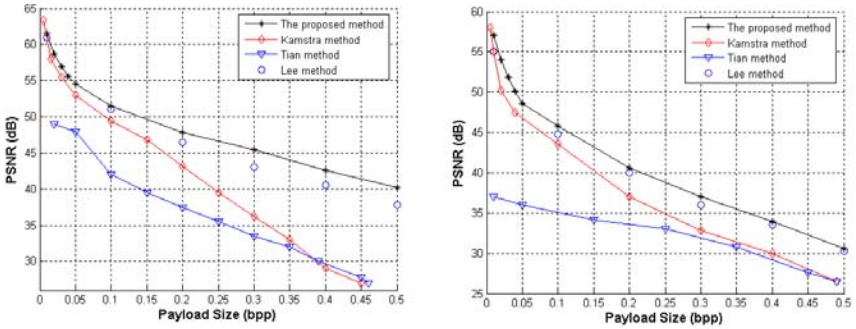
For Lena, Barbra and Baboon three commonly used images (all in 512x512), we compare the results of our proposed histogram pair technique with that results of other techniques, i.e., techniques of [1,2,3,4,5,6,7,8,9]. The results of performance comparison are shown in Figures 8, 9 and 10, respectively. What shown in Figure 10(a) are the original Lena image and the marked Lena images generated by using the proposed method with the same group of three different payloads as used in [2], in Figure 10(b), the corresponding test results of [9] are shown. From these figures, it is observed that the PSNR achieved by our proposed method for the same embedded payload is the highest among the nine methods for these three commonly used test images.

Note that the platform used in all of experiments reported in Section 6 is as follows. CPU: Pentium 2.8GHz, RAM: 512M, OS: Windows XP SP2, and Matlab 7.0.



(a) (b)

Fig. 8. (a) Performance comparison on Lena (b) Comparison of multiple-time data embedding into Lena image among [2], [8] and the proposed method.



(a) (b)

Fig. 9. (a) Comparison on Barbara (b) Comparison on Baboon

In Table 6, the three parameters, GR, GL, and G, are listed when applying our proposed adaptive histogram modification method to the above-mentioned three

commonly used images with various embedding rates. It is interesting to observe the following. That is, our extensive experiments have shown that only when the embedding data rate for Lena is greater than 1.0 bpp, for Barbara greater than 0.57 bpp, and for Baboon greater than 0.008 bpp, it needs histogram modification (G>0). Otherwise, there is no need for histogram modification when reversibly embedding data into these three commonly used images.

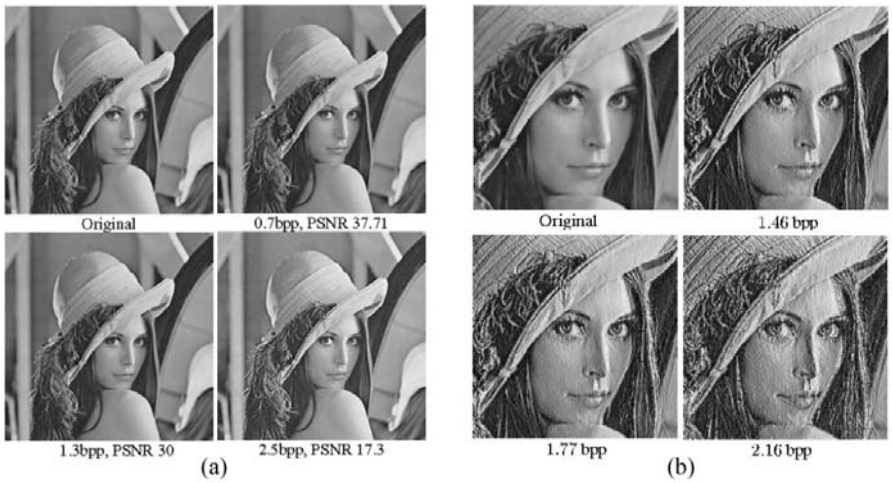


Fig.10 (a) Original and marked Lena image with three different payloads by the proposed method (b) Performance on Lena image reported in [9]

Table 6 GL and GR values used in our experiments.

Figure 6,7,10 Lena					Figure 8 Barbara					Figure 9 Baboon				
bits	bpp	GL	GR	G	bits	bpp	GL	GR	G	bits	bpp	GL	GR	G
3132	0.0119	0	0	0	2819	0.0108	0	0	0	1037	0.0040	0	0	0
6744	0.0257	0	0	0	6041	0.0230	0	0	0	2089	0.0080	0	0	0
11986	0.0457	0	0	0	13064	0.0498	0	0	0	5082	0.0194	2	0	2
21206	0.0809	0	0	0	20095	0.0767	0	0	0	14936	0.0570	6	0	6
36421	0.1389	0	0	0	31729	0.1210	0	0	0	31148	0.1188	8	0	8
58672	0.2238	0	0	0	42639	0.1627	0	0	0	55990	0.2136	10	0	10
82720	0.3156	0	0	0	66702	0.2544	0	0	0	80122	0.3056	13	0	13
109009	0.4158	0	0	0	98430	0.3755	0	0	0	99015	0.3777	15	0	15
135062	0.5152	0	0	0	119672	0.4565	0	0	0	125005	0.4769	18	0	18
172983	0.6599	0	0	0	133784	0.5103	0	0	0	141066	0.5381	22	0	22
207273	0.7907	0	0	0	150320	0.5734	0	0	0					
285027	1.0873	0	0	0	174944	0.6674	0	3	3					
317999	1.2131	2	4	6	180910	0.6901	5	15	18					
336236	1.2826	6	22	28										
430482	1.6422	22	24	46										
510079	1.9458	42	48	90										

The PSNR versus payload, and parameters for multiple loop data embedding on Lena image are shown in Table 7.

Table 7 Multiple loop of data embedding into Lena image.

Pay-load (bpp)	PSNR	Number of Loop	T for 1st Loop	T for 2nd Loop	T for 3rd Loop	T for 4th Loop	GL for 1st Loop	GR for 1st Loop	Book-keeping for 1st Loop	Time (second)
1.3	29.7709	3	-6	-12	-3	No	6	6	33	2.8600
1.6	26.3727	3	-7	-22	-4	-4	35	42	1356	6.6100
1.9	23.2216	4	-8	-28	-14	-4	70	80	7512	11.0930
2.2	19.8060	4	-8	-28	-28	-9	127	147	27558	17.7180
2.4	16.8659	5	-9	-26	-40	-26	84	87	1264	42.4530

6.1.2. Woman image

Woman image, one of the popularly used JPEG2000 test images, and its histogram have been shown in Figure 11. The histograms of three popularly used images in reversible data hiding research, i.e., Lena, Barbra and Baboon images are shown in Figure 12 (a), (b) and (c), respectively. It is observed that the both right and left ends of the histogram of Woman image have been occupied and with some high peaks, while the histograms of the three commonly used test images in the reversible data hiding community have both ends empty. Furthermore, Lena image has two ends empty most widely, while Barbra and Baboon images have two ends relatively narrower. These characteristics of three histograms explain why the histogram shrinking from both right and left ends are not necessary when the embedding rate is low for these three images. That is, GR and GL values are both equal to 0 as embedding rates are low. Hence, it is felt that it is necessary to test reversible data hiding algorithms on Woman image as well in order to have complete and objective performance evaluation of various reversible data hiding schemes.

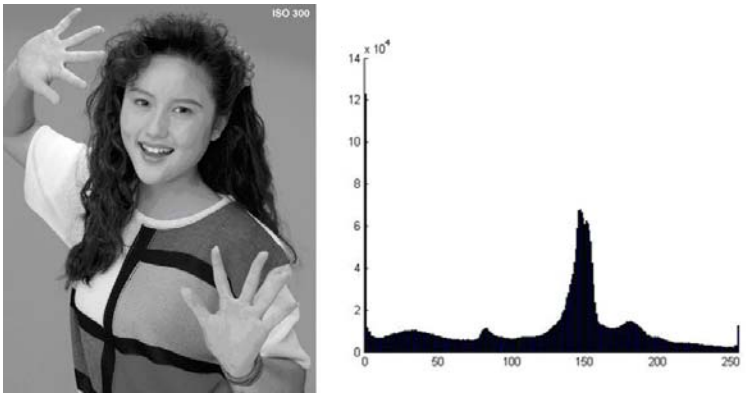
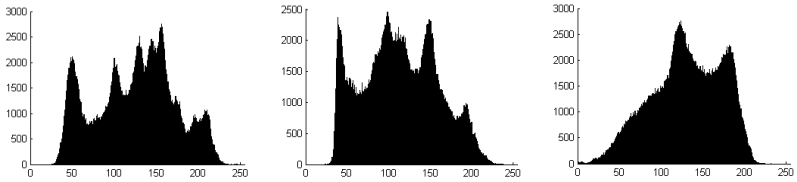


Figure 11 (a) Woman image, (b) its histogram.



(a) (b) (c)

Figure 12 Histograms of (a) Lena, (b) Barbra, and (c) Baboon images.

Table 8 Performance of location map method [2] on Woman image.

Payload (bpp)	PSNR (db)	T	Location map (bit)	Time (sec.)
0.09	43.6	10	809040	6.1
0.15	41.1	15	737024	6.3
0.19	39.1	20	679128	6.1
0.22	37.6	25	628520	7.0
0.24	36.4	30	582576	6.3
0.31	33.3	50	448088	6.1
0.36	30.8	inf	341328	6.0

Table 9 Performance of proposed method on Woman image.

Payload (bpp)	PSNR (db)	T	S	Bookkeeping (bit)	GL	GR	Time (sec)
0.01	57.5	3	-3	19176	1	1	12.8
0.10	49.1	2	0	62882	3	3	20.5
0.15	48.1	1	0	71616	3	5	20.4
0.20	45.9	1	0	98486	5	5	23.1
0.25	44.4	2	0	130909	6	8	26.6
0.30	42.9	2	0	170844	7	9	51.2
0.35	41.0	3	0	242053	10	11	69.7
0.40	39.4	4	0	301941	12	13	78.1

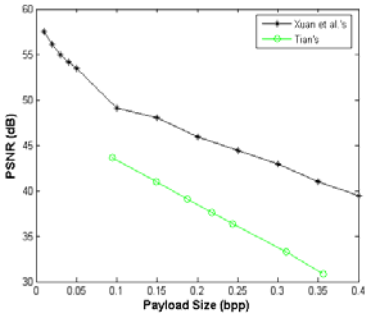


Figure 13 Payload vs PSNR on Woman image.

6.2. Comparison by using integer (5,3) and Haar wavelets

Table 10 and Table 11 list the PSNR of marked image versus payload (bpp), and parameters of G, T, and S of the proposed reversible data embedding when using integer (5,3) and Haar wavelet, respectively. As expected, integer (5,3) wavelet provides higher PSNR for a given payload than integer Haar wavelet, while integer Haar is simpler in implementation. Specifically, as payload is as low as 0.05 bpp, for these three commonly used test images, the PSNR of marked versus original images achieved by using integer (5,3) is, respectively, 1.7 dB, 2.5 dB and 0.5 dB higher than that by using Haar IWT; as payload is 0.5 bpp, the PSNR by integer (5,3) is, respectively, 2.6 dB, 3.1 dB, and 1,5 dB higher higher than that by using Haar IWT.

Table 10 PSNR of reversible data embedding using (5,3) IWT.

Payload (bpp)	Lena				Barbara				Baboon			
	T	S	PSNR	G	T	S	PSNR	G	T	S	PSNR	G
0.05	3	-2	54.4	0	2	1	54.6	0	3	0	48.6	3
0.1	2	1	51.6	0	2	1	51.5	0	3	0	45.9	3
0.2	1	0	48.2	0	2	-1	47.8	0	4	0	40.6	6
0.3	2	0	46.0	0	2	0	45.3	0	6	0	37.0	10
0.4	3	0	43.9	0	4	0	42.6	0	9	0	34.0	18
0.5	4	0	42.1	0	5	0	40.2	0	16	0	30.6	30

Table 11 PSNR of reversible data embedding using Haar IWT.

Payload (bpp)	Lena				Barbara				Baboon			
	T	S	PSNR	G	T	S	PSNR	G	T	S	PSNR	G
0.05	2	2	52.7	0	2	-2	52.1	0	7	-6	48.1	1
0.1	2	-2	50.1	0	2	1	46.9	0	5	3	43.6	3
0.2	3	-2	46.9	0	2	0	46.1	0	4	0	38.4	8
0.3	2	0	43.7	0	3	0	42.1	0	8	-1	35.0	15
0.4	4	-1	41.8	0	4	-1	40.2	0	11	0	31.9	21
0.5	6	-1	39.5	0	7	-1	37.1	0	17	0	29.1	28

7. Discussion and Conclusion

(1) In this paper, it has been shown that the minimum threshold T in reversible data hiding does not necessarily provides the best performance in terms of the visual quality of marked image measured by PSNR versus data embedding capacity.

(2) An optimum histogram pair based image reversible data embedding scheme using integer wavelet transform is presented in this paper. It uses the new concept of histogram pair to reversibly embed data into image. Furthermore, it is characterized by the selection of best threshold, adaptive histogram modification parameters and suitable embedding region. The experimental results have demonstrated its superior performance in terms of the visual quality of marked image measured by PSNR versus data embedding capacity, to our best knowledge, over all of the prior arts including [1,2,3,4,5,6,7,8,9].

(3) Different from the method in [5], our proposed method uses histogram pair to reversibly embed data, which provides more flexibility in implementation. The procedure of histogram pair is also used in adaptive histogram modification in an

inverse way. That is, the same histogram pair technique is used in data embedding as well as in histogram modification to avoid overflow and underflow.

(4) Furthermore, the new method systematically and computationally selects the best threshold, the adaptive histogram modification parameters GR and GL, and the most suitable data embedding region.

(5) One example in selecting suitable data embedding region is for reversible data embedding into the Lena image as the payload is low. Our experimental works have shown that the PSNR of marked image achieved by this new method is distinctly better than that achieved by the method [5] because the new method first selects integer wavelet coefficients in the HH subband for data embedding as the payload is low instead of embedding data arbitrarily into the wavelet coefficients of the HL, LH and HH subbands.

Furthermore if the payload is further low such that not all of the high-frequency coefficients in the HH subband need to be used for data embedding, an additional measure is adopted to further raise the PSNR of marked image versus original image. That is, the magnitudes of all of the wavelet coefficients in the immediate next-level high-high wavelet subband, denoted by HH_1 are examined so as to arrange these coefficients in an non-increasing order according to their magnitudes. In addition to the proposed optimum histogram pair scheme described above, the data is sequentially embedded into those wavelet coefficients in the original-level HH subband, which correspond to the magnitude non-increasing order arranged in the immediate next-level HH_1 subband. This measure has been shown effective in further raising the PSNR of marked Lena and Barbara images as the payload is rather low.

(6) The computational complexity, including optimal histogram pair based data embedding and possible histogram modification, is shown affordable for possible real applications. Specifically, for data embedding ranging from 0.01 bpp to 1.0 bpp into three commonly used images: Lena, Barbara and Baboon, the execution time varies from 0.25 to 2.68 seconds with the platform of CPU: Pentium 2.8GHz, RAM: 512M, OS: Windows XP SP2, and Matlab: 7.0.

(7) If the data embedding rate is not high (e.g., as shown in Table 6, not higher than 1.0873 bpp for Lena image, 0.5734 bpp for Barbara image, and 0.0080 bpp for Baboon image), the amount of histogram modification G = 0, meaning that the histogram shrinking is not needed. This means that when the data embedding rate is not larger than a specific amount for a given image, our proposed reversible data hiding method does not need to calculate point-by-point to avoid the possible underflow and/or overflow as required in the methods proposed in [2] or [3]. Under this circumstance, the proposed reversible data hiding becomes very simple in implementation.

(8) To the authors' best knowledge, for the first time in the literature, one of the JPEG2000 test images, Woman, has been used in experimental study for reversible data hiding. This is necessary since the histogram of Woman image is quite different from these three frequently used test images: Lena, Barbra and Baboon images. The histogram of Woman has peaks at both grayscale values 0 and 255, while the histograms of the three frequently used images do not have these two end values occupied. It is shown that our proposed method can still effectively and efficiently work for Woman image.

(9) In our experimental works, the proposed method has been applied to both integer

(5,3) and Haar wavelet transforms. It is shown that using integer (5,3) wavelet brings out better performance in terms of data embedding rate versus PSNR. The proposed method can also be applied to any other integer wavelet transforms, demonstrating its flexibility is using integer wavelet transforms. Note that the reversible method in [2] can only use Haar transform.

Acknowledgement

Authors sincerely thank Ms. Haifeng Xiao and Mr. Patchara Sutthiwan for their time and efforts put into the preparation of this paper at the very early and very final stages, respectively.

References

1. Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, March 2006, pp. 354-362.
2. J. Tian, "Reversible data embedding using a difference expansion," IEEE Transactions on Circuits and Systems for Video Technology, Aug. 2003, pp.890-896.
3. L. Kamstra, H. J.A.M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," IEEE Transactions on Image Processing, vol. 14, no. 12, Dec. 2005, pp. 2082-2090.
4. G. Xuan, Y. Q. Shi, C. Yang,, Y Zheng, D. Zou, P. Chai, "Lossless data hiding using integer wavelet transform and threshold embedding technique," IEEE International Conference on Multimedia and Expo (ICME05), Amsterdam, Netherlands, July 6-8, 2005.
5. G. Xuan, Y. Q. Shi, Q. Yao, Z. Ni, C. Yang, J. Gao, P. Chai, "Lossless data hiding using histogram shifting method based on integer wavelets," International Workshop on Digital Watermarking (IWDW06), Nov. 8 - 10, 2006, Jeju Island, Korea.
6. B. Yang, M. Schmucker, W. Funk, C. Busch and S. Sun, "Integer DCT-based reversible watermarking for images using companding technique," Proceedings of SPIE Security and Watermarking of Multimedia Content, Electronic Imaging. San Jose, CA, USA, January 2004.
7. S. Lee, C. D. Yoo, and T. Kalker, "Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform," IEEE Transactions on Information Forensics and Security, Vol.2, Issue3, Part 1, Sept. 2007
8. D. Coltuc and J. M. Chassery, "Very fast watermarking by reversible contrast mapping," IEEE Signal Processing Letters, Vol. 14, No. 4, pp.255-258, April 2007.
9. D. Coltuc, "Improved capacity reversible watermarking," International Conference on Image Processing (ICIP07), September 16-19, 2007, San Antonio, Texas ,USA, III, pp.249-252.