Reversible Data Hiding for JPEG Images Based on Histogram Pairs

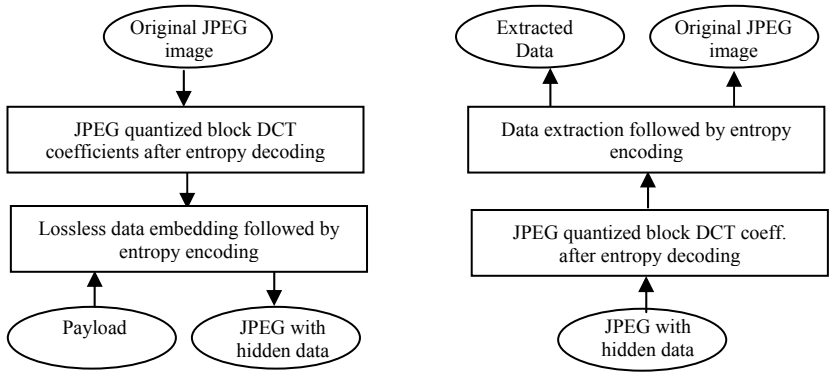
Guorong Xuan[1], Yun Q. Shi[2], Zhicheng Ni[2], Peiqi Chai[1] , Xia Cui[1], Xuefeng Tong[1]
[1]Dept. of Computer Science, Tongji University, Shanghai, China
[2]Dept. of ECE, New Jersey Institute of Technology, Newark, New Jersey, USA
grxuan@public1.sta.net.cn, shi@njit.edu

Abstract
This paper[1] proposes a lossless data hiding technique for JPEG images based on histogram pairs. It embeds data into the JPEG quantized 8x8 block DCT coefficients and can achieve good performance in terms of PSNR versus payload through manipulating histogram pairs with optimum threshold and optimum region of the JPEG DCT coefficients. It can obtain higher payload than the prior arts. In addition, the increase of JPEG file size after data embedding remains unnoticeable. These have been verified by our extensive experiments.

1. Introduction

Reversible, also referred to as invertible or lossless, image data hiding can imperceptibly hide data into digital images and can reconstruct the original image without any distortion after the hidden data have been extracted out. Linking a group of data to a cover image in a reversible way is particularly critical for medical, images (for legal consideration) as well as for military, remote sensing and high-energy physical images (where high accuracy does matter and the original image is of importance).

Reversible data hiding using the histogram shifting technique in spatial domain has first been reported in [1] and in [2], independently. It has been applied to the histogram of integer DCT domain [3], and integer wavelet transform domain [4,5]. In general, the histogram shifting technique has achieved dramatically improved performance in terms of embedding capacity versus visual quality of stego image measured by PSNR (peak signal noise ratio).



(a) Data embedding (b) Data extraction

Figure 1 Block diagram of the proposed lossless data embedding for JPEG image.

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

Among various digital image formats, JPEG (Joint Photographic Experts Group) format is by far used most often currently. How to reversibly hide data into JPEG image file is hence important and useful for many applications including image authentication, secure data system, and linking annotation data to the host image.

However, there are not many reversible data hiding techniques that have been developed for JPEG images so far. Some techniques are reported in [6,7]. In [6], the least significant bit plane of some selected JPEG mode coefficients is losslessly compressed, thus leaving space for reversible data embedding. Consequently the payload is rather limited. In [7], some schemes have been reported which aim at keeping the size of JPEG file after lossless data hiding remaining unchanged. However, the payload is still rather limited (the highest payload in various experiment al results reported in [7] is 0.0176 bpp (bits per pixel). In this paper, we present a novel technique based on histogram pairs applied to some mid- and low-frequency JPEG quantized 8x8 block DCT (discrete cosine transform) coefficients (sometimes referred to as JPEG coefficients in this paper for short) for reversible data hiding. A block diagram of data embedding and extraction is shown in Figure 1. Experimental results are presented to demonstrate its effectiveness. The data embedding capacity range from 0.0004, to 0.001, 0.1, up to 0.5 bpp (bits per pixel) for one-time (or, one-loop) reversible data hiding, while the visual quality of images with hidden data measured by both subjective and objective (PSNR) ways remains high. The increase of size of image file due to data hiding is not noticeable, and the shape of histogram of the mid-and lower-frequency coefficients of DCT remains similar. The proposed technique works for various Q-factors.

The rest of this paper is organized as follows. The principle of histogram pair based lossless data hiding is presented in Section 2. The concept of thresholding is discussed in Section 3. The lossless data embedding and extraction algorithm is described in Section 4. Optimality is analyzed in Section 5. Section 6 contains experimental results. Conclusion and discussion are in Section 6.

2. Principle of Histogram Pair Based Lossless Data Embedding

2.1. Definition of histogram pair

Histogram, h(x), is the number of occurrence (i.e., frequency) of feature x within a set of samples X. Here the samples X in this paper is some selected JPEG quantized 8×8 DCT coefficients，the feature x is the JPEG coefficients' value. The x is either positive, or negative integer, or zero, such as $x\in\{-2,-1,0,1,2,3\}$.

Histogram pair is defined as a part of the histogram, denoted by h=[m,n], where m and n are, respectively, the frequencies of two immediately neighboring feature values $x\in\{a,b\}$ with $a<b$，i.e., $b=a+1$, and one of the two frequencies (m and n) is 0. Histogram pair can be formulated via a process called *histogram expansion*. For example, via expanding, the histogram pair h=[m,0] can be produced. (The detail will be illustrated via three examples in this paper.) Note that the underline is used to mark the histogram pair. The feature value whose frequency (h value) is not 0 is called the feature's original position. The feature value whose h value is 0 is called the feature's expansion position. In this paper, it is defined that when feature value x is greater or equal to 0, the histogram pair is of the format h=[m,0]，which means h(a)=m and h(b)=0；when feature value x is less than 0, the histogram pair is of h=[0,n]，which means h(a)= 0 and h(b)= n.

After the histogram pair is produced, lossless data embedding is possible. Data embedding rule may be as follows. If the to-be-embedded bit is 0，the feature's original position is used. If the to-be-embedded bit is 1, the feature's expansion position is used. Although the other way around also works, this rule is used in this paper. This will be illustrated through the following three examples.

It is observed that after data embedding the histogram becomes more flat. When the histogram is completely flat, it is impossible to further embed data.

2.2. First example of reversible data embedding: Using single histogram pair

Assume samples are X=[a,a,a,a], i.e., the number of samples is M=4, feature values $x \in \{a,b\}$ are greater than 0. There is one histogram pair h=[4,0]. Suppose that the to-be-embedded binary sequence is D=[1,0,0,1] whose length L is equal to 4, i.e., L=4.

During data embedding, we scan the sequence X= [a,a,a,a] in a certain sequencing, say, from left to right. When we meet the first a, since we want to embed bit 1, we change a to its expansion position, b. For the next two to-be-embedded bits, since they are bit 0, we keep a in its original position, i.e., 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=L=4. Data extraction, or histogram pair recovery, is the reverse process of the above mentioned data embedding. After extracting the data D=[1,0,0,1], the histogram pair becomes [4,0] and we recover X=[a,a,a,a] losslessly.

Note that after data embedding, histogram is changed from h=[4,0] to h=[2,2]，histogram is completely flat and hence we cannot embed data any more.

2.3. Second example of reversible data embedding: Using two loops

Given a 3×3 image, the feature values are $x \in \{a,b,c,d\}$, where features are all greater than 0. According to the scan order, say, from left to right and from top to bottom, the samples X become X=[a,a,a,a,a,a,a,a,a]，the total number of samples M=9，histogram is h=[9,0,0,0]，as shown in Figure 2(a). The histogram pair is h=[9,0]. The to-be- embedded bit sequence is D=[0,1,0,0,1,0,1,1,0] and L=9.

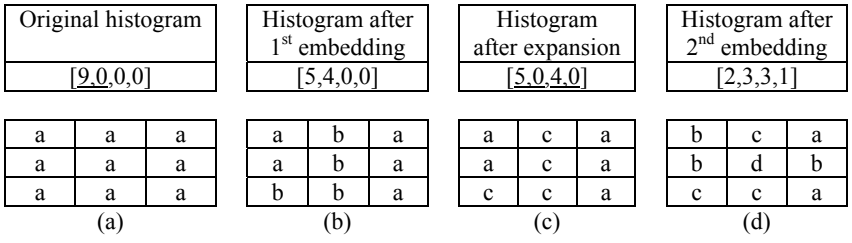


Figure 2　Bit sequence D=[0,1,0,0,1,0,1,1,0] embedded in two loops.

First data embedding "Loop 1"：since the first to-be-embedded bit is 0，use the original feature position a (meaning no change for the first a) , the second bit is 1, use the expansion position (meaning change a to b). In this way, we totally embed 9 bits, after data embedding, the samples become X=[a,b,a,a,b,a,b,b,a], refer to Figure 2(b). After the first embedding loop, the histogram h=[9,0,0,0] becomes h=[5,4,0,0]. The payload is C_1=L=9 bits.

Second data embedding "Loop 2":

Expanding first：the histogram pair h=[4,0] is shifted towards the right-hand side by one position, thus producing the histogram with two histogram pairs h=[5,0,4,0]， and the samples become X=[a,c,a,a,c,a,c,c,a],refer to Figure 2(c).

The second embedding loop will separately use the two histogram pairs in h=[5,0,4,0]， x∈{a,b,c,d} in order to avoid confliction. That is，it first uses the histogram pair with larger absolute feature values, then uses the histogram pair with smaller absolute feature values. In this example, we first embed data into the right histogram pair, then into the left histogram pair. The to-be-embedded bit sequence D=[0,1,0,0,1,0,1,1,0] is separated into two parts accordingly. That is, we first embed the front portion of data D_1=[0,1,0,0] into the histogram pair at the right side h=[4,0], x∈{c, d}, resulting in the corresponding samples X_1=[c,d,c,c]. Then, we embed the remaining data D_2=[1,0,1,1,0] into the left histogram pair h=[5,0], x∈{a,b},resulting in the corresponding samples X_2=[b,a,b,b,a]. After Loop2, the histogram becomes h=[2,3,3,1] and the samples become X=[b,c,a,b,d,b,c,c,a], Figure 2(d). The embedding capacity in Loop 2 is C_2=L=9 bits.

The total capacity after two embedding loops is C=18 bits. After two embedding loops, histogram changes from h=[9,0,0,0] to h=[2,3,3,1]. It is observed that the histogram has changed from rather sharp ([9,0,0,0]) to relatively flat ([2,3,3,1]).

3. Thresholding

3.1. Principle of thresholding

Histogram pair based lossless data hiding seeks for not only higher embedding capacity but also higher visual quality of stego images measured by, say, PSNR (peak signal noise ratio). For instance, we may embed data with sufficient payload for annotation (such as caption) or for security (such as authentication) with reversibility as well as the highest possible PSNR of the stego image with respect to the cover image.

In [4], it was thought that one way to improve the PSNR is to use only a part of JPEG coefficients with small absolute values. In doing so, we need the so-called thresholding technique. The thresholding method is to first set a threshold T, then embed data into those JPEG coefficients, x, with $|x|\leq T$. That is, it does not embed data into the JPEG coefficients with $|x|>T$. In addition, it makes sure that the small JPEG coefficients after data embedding will not conflict (will not be *confused*) with the large JPEG coefficients with ($|x|>T$). That is, for the JPEG coefficients satisfying $|x|\leq T$, histogram pair based data embedding is applied. It requires that after data embedding, the coefficients between $-T\leq x\leq T$ will be separable from the coefficients with $|x|>T$. The *simple* thresholding will divide the whole histogram into two parts:1) the data-to-be embedded region, where the JPEG coefficients absolute value is small；2) no data-to-be embedded region named end regions, where the JPEG coefficients' absolute value is large.

3.2. Optimum thresholding

Our experimental works have indicated that the smallest threshold T does not necessarily lead to the highest PSNR for a given data embedding capacity. Instead, it

is found that for a given data embedding capacity there is an optimum value of T. This can be justified as follows. If a smaller threshold T is selected, the number of coefficients with |x|>T will be larger. This implies that more coefficients with |x|>T need to be moved away from 0 in order to create histogram pair(s) to losslessly embed data. This may lead to a lower PSNR and more side information (hence smaller embedding capacity). Therefore in this proposed optimum histogram pair lossless embedding, the best threshold T for a given data embedding capacity is selected to achieve the highest PSNR. Discussion about the optimum parameters are presented in Section 5, and experimental results are shown in Section 6.

4. Histogram Pair Based Reversible Data Hiding Algorithm

4.1. Data embedding algorithm

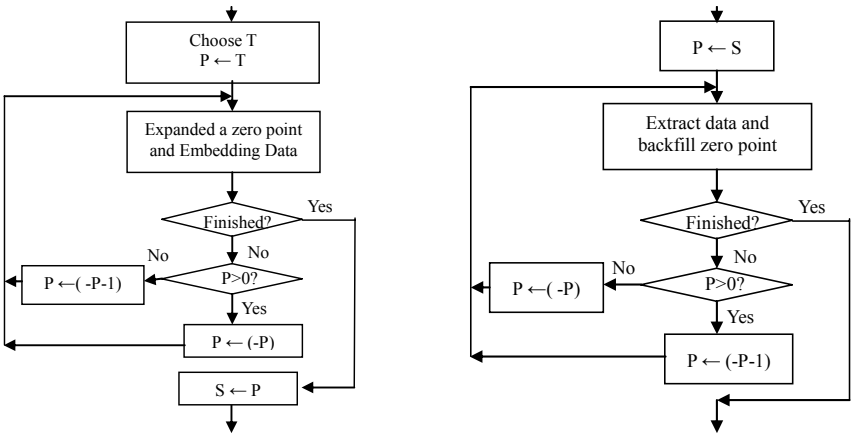
Assume the length of the to-be-embedded data is L. The data embedding steps are listed below. (See Figure 3 (a).)

(1) Set a threshold T>0, to let the number of the mid- and low-frequency JPEG coefficients within [-T,T] be greater than L. And set the $P \leftarrow T$.

(2) In the JPEG coefficient histogram, move the portion of histogram with the coefficient value 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 into P and $P+1$ according to the to-be-embedded bit is 0 or 1, respectively. .

(3) If some of the to-be-embedded bits have not been embedded at this point, let $P \leftarrow (-P)$, and move the histogram (less than P) to the left-hand side by one unit to leave a zero-point at the value $(P-1)$. And embed data into P and $(P-1)$ according to the to-be-embedded bit is 0 or 1, respectively. .

(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 step (2) to continue to embed the remaining to-be-embedded data.



(a) embedding (b) extracting

Figure 3 Flowchart of proposed lossless data embedding and extracting.

4.2. 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* (positive). The data extraction steps are as follows. (Refer to Figure 3 (b).)

(1) Set $P \leftarrow S$.

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

(3) If the amount of extracted data is less than *C*, set $P \leftarrow (-P-1)$. Continue to extract data until *(P-1)* becomes a zero-point. Then move the histogram (less than *P-1*) to the right-hand side by one unit to eliminate the zero-pint.

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

4.3. Third example of histogram pair based lossless data hiding

In this simple but 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 dimensionality of the image is 5×5 as shown in Figure 4 (a). The image has 12 distinct feature (grayscale) values, i.e., $x \in \{-5,-4,-3,-2,-1,0,1,2,3,4,5,6\}$. 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 5). As said before, for $x \geq 0$, the histogram pair is of form h=[m,0], for x<0, the histogram pair is h=[0,n]. The second row of Figure 5 is expanded image histogram: h_1 (expanded), it has three histogram pairs. The first histogram pair is in the far-right-hand side h= [1,0]; the second histogram pair is in the left-hand side h= [0,2]; the third histogram pair is in the right-hand side near the center h= [3,0]. The third row of Figure 5 is the image histogram after data embedding; h_2 (bits embedded).

Figure 5 and Table 1 use red line square 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 1st making h(4)=0, then making h(-4)=0, finally making h(3)=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 Figure4(b) and 2nd row of Figure 5). 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].

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 4 5×5 image (a) original image, (b) image after 3-bit expanding, (c) image after 6-bit embedding. (What marked is how the last 3 bits are embedded.)

After data embedding with bit sequence D=[1 10 001] with the selected scanning order (from right to left and from top to bottom), the histogram becomes h_2 =[1,1,1,2,4,6,3,2,1,0,1,2] (refer to Figure 4(c) and 3rd row of Figure 5). 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, some histogram pairs have been changed. For example, embedding the last three bits 0,0,1 causes the histogram pair at the right-hand side (near center) to change from h=[3,0] to h=[2,1], and three coefficients marked with small rectangles (in red) to change from [2,2,2] to [2,2,3] (refer to Figure 4 (c) and 3rd row of Figure 5).

Through this example, it becomes clear that the threshold can also be viewed as the starting point to implement histogram pair lossless data hiding.

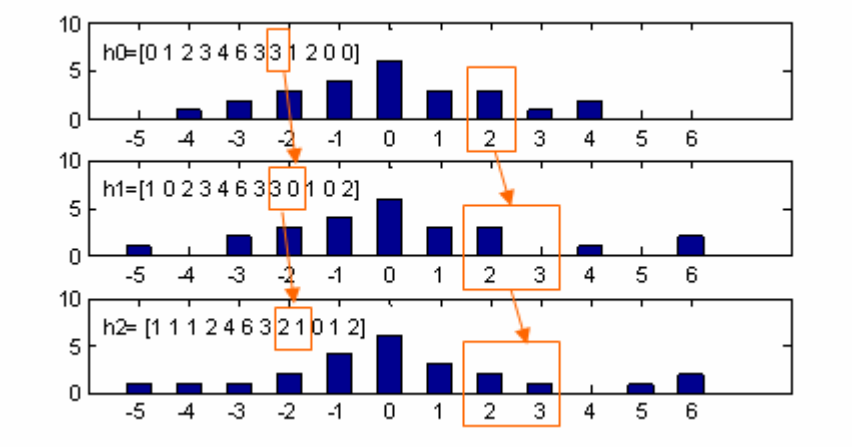


Figure 5 Histogram pair data embedding example（T=3，S= +2）
(6 bit sequence D=[1 10 001]). (What marked is how the last 3 bits are embedded.)

4.4. Formulae of lossless data hiding based on histogram pairs

The proposed method divides the whole histogram into three parts: (1) the part where data to be embedded; (2) central part – no data embedded and the absolute value of coefficients is small; (3) end part – no data embedded and the absolute value of coefficients is large. The whole embedding and extraction procedure can be expressed by the formulae in Table 1.

Table 1 Formulae of lossless data hiding based on histogram pairs

parts of histogram	Embedding		Recovering	
	after embedding	condition	after recovering	condition
Data to be 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 to be 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)
Central part (small absolute value)	x'=x	-|S|-u(S)<x<|S|	x=x'	-|S|-u(S)<x'<|S|
Right edge part (positive)	x'=x+T+1-|S|	x>T	x=x'-T-1+|S|	x'> 2T+1-|S|
Left edge part (negative)	x'=x-T-1 +|S|+u(S)	x<-T	x=x'+T+1-|S|-u(S)	if x'< -2T-1 +|S|+u(S)

where T is selected threshold, i.e., start position, S is stop position, x is feature (JPEG coefficient) values before embedding, x' is feature values after 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.

5. Optimum Parameters of Histogram Pair Based Lossless Data Hiding

5.1. Selection of a part of 8x8 block DCT coefficients for data embedding

In order to make data embedding unperceivable and make the PSNR of stego image higher when embedding a certain amount of data, we choose lower- and mid-frequency JPEG coefficients to embed data in the implementation of our invented technology. Among all of the JPEG quantized DCT coefficients, which part of coefficients used for data embedding will lead to the best performance? We did some experimental investigation.

When scanning all of the 8x8 block DCT coefficients in the zigzag order [8], we can mark all of the JPEG coefficients within the 8x8 block in a sequential order, from 1, 2, …, up to 64. In order to examine which part of mid- an d low-frequency JPEG coefficients can bring out optimum performance, we select two sequential numbers along the zigzag scan order, i.e., two numbers, o_1 and o_2, with $1<o_1<o_2<64$. For each pair of o_1 and o_2 we can formulate a histogram and apply our histogram pair based lossless data hiding algorithm to this histogram. The performance comparison with various selections of o_1 and o_2, i.e., different JPEG coefficient regions has been investigated. In addition, different T values and S values are other two parameters that do affect the performance.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

(a) R= {16,36}

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

(b) R={4,36}

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

(c) R={16,49}

Figure 6 Selected DCT coefficients for data embedding.

Table 2 Achieved PSNR as 0.01 bpp data embedded into Lena

R: region	PSNR	comments
R={1,36}	34.7741	
R={16,36}	34.4574	
R={4,36}	36.5374	Selected in this paper
R={4,49}	36.2801	
R={16,49}	32.4435	

Table 2 listed five different pairs of o_1 and o_2, i.e., five different JPEG coefficients regions, among which three regions, {16,36}, {4,36} and {16,49} have been marked in red lines and shown in Figure 6. Our experimental works have shown that when the payload is fixed as 0.01 bpp (bit per pixel) for the 512x512 Lena image, the region {4,36} provides the highest PSNR for the stego image. Therefore, this region has been used in our experimental works, reported in Section 6.

5.2. Optimum threshold

Obviously the PSNR will change as T changes, When the PSNR reaches its peak, the corresponding T is the optimum threshold. The traditional definition of PSNR [8] is used in our experimental works. Note that the PSNR is calculated from the stego image versus the JPEG image after JPEG decompression.

Of course, optimum can also be addressed as search for optimum T and R such that the payload is largest for a given PSNR value. Figure 7 displays the relation between PSNR and the threshold T when embedding 500 bits into 512x512 Lena, Barbara and Baboon images (0.00191 bpp) with the selected JPEG coefficient region R={4,36}. It is observed that the optimum T values for these three images are 5, 8, and 14, respectivelv.

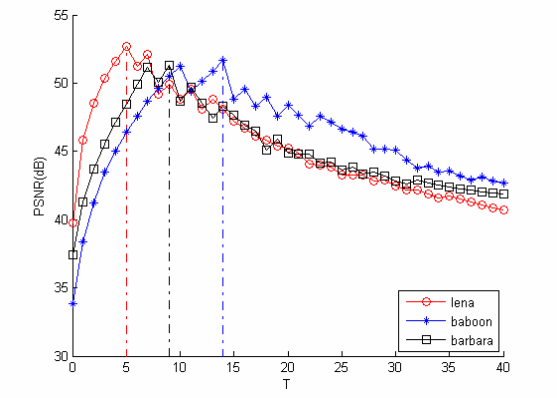


Figure 7 Selecting of T values for three commonly use images as 500 bits are embedded into 512x512 JPEG images (0.00191 bpp) and the region of JPEG coefficients being R={4,36}.

The optimum histogram pair based lossless embedding is that which leads to the highest PSNR for a given payload C with some choice of the threshold T, the JPEG coefficient region R. That is, for a given payload, the optimum histogram pair lossless embedding can be expressed as:

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

6. Experimental results

6.1 Experimental results with JPEG Lena images having different Q-factors

When 1000 bits are embedded into the 512x512 JPEG Lena images (0.0038 bpp) with

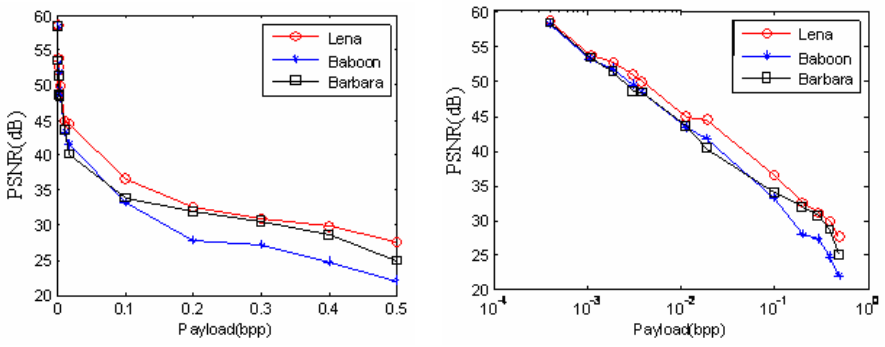
different Q-factors (30, 40, 50, 60, 70, 80, 90, 100), the experimental results with the range of JPEG coefficients R={4,36} are listed in Table 3. From these results, it is seen that the proposed technology works for various Q-factors well.

Table 3 Experimental results with 1000 bits embedded into the 512x512 Lena JPEG image (0.0038 bpp) with various Q-factors and the selected JPEG coefficient region R={4,36}

Q	PSNR	T	S	Time (sec)	Original image file size (bits)	image size after embedding (bits)	image size increase after embedding (bits)	image size increase after embedding (%)
30	41.7598	2	-2	0.337	15,159	15,423	264	1.7%
40	44.5994	2	2	0.179	18,027	18,216	189	1.0%
50	45.3630	2	2	0.185	20,919	21,142	223	1.1%
60	46.0381	2	2	0.184	24,076	24,359	283	1.2%
70	47.9433	4	-4	0.364	29,294	29,403	109	0.4%
80	49.8880	5	-5	0.375	37,937	38,120	183	0.5%
90	52.9991	7	-7	0.392	59,197	59,419	222	0.4%
100	58.2008	9	9	0.358	162,247	162,437	190	0.1%

It is observed that when 1000 bits are embedded into 512x512 Lena image, for Q-factor ranging from 30 to 100, the PSNR ranges from 41 dB to 58 dB. Furthermore, before and after the data embedding, the amount of image file size increase ranges from 283 bits to 109 bits. The execution time is reasonably fast. These results indicate satisfactory performance.

6.2. Experimental results with three commonly used JPEG images having Q-factor 80



(a) Payload (bpp) in regular form (b) Payload (bpp) in log form

Figure 8 PSNR versus payload for three commonly used JPEG images.

We have conducted extensive experiments with various payloads. What reported here are experiments on Lena.jpg (512x512), Barbara.jpg (512x512), Baboon.jpg (512x512) with Q-factor 80. The random bits produced by MATLAB are used as payload. They are embedded into the JPEG coefficients in the region R={4,36}. Figure 8 depicts the PSNR versus various payloads. From these results, it is observed that the proposed technology works well with these three commonly used images.

The more detailed experimental results on the 512x512 Lena JPEG images with Q-80 are presented in Table 4 and Figure 9. Similar good results have achieved for Barbara and Baboon image as well. Due to the page limit, the corresponding tables and figures are not shown here.

Table 4 Experimental results on 512x512 Lena JPEG image with Q-factor 80

No.	Payload (bpp)	Payload (bits)	PSNR	T (begin)	S (stop)	Time (sec)	JPEG (bit)
1(Original)	0	0	∞	-	-	-	37937
2	0.0004	100	58.4707	12	12	0.1820	37956
3	0.0011	300	53.7209	9	-9	0.3730	37980
4	0.0019	500	52.6613	5	5	0.1790	38024
5	0.0030	800	51.0275	6	-6	0.3820	38119
6	0.0038	1000	49.8880	5	-5	0.3750	38120
7	0.0114	3000	44.9344	2	-2	0.3420	38913
8	0.0191	5000	44.4357	2	-2	0.3670	38913
9	0.1000	26214	36.5374	2	-1	0.7450	42884
10	0.2000	52429	32.5016	0	0	0.1870	52486
11	0.3000	78643	30.9442	0	0	0.2040	57733
12	0.4000	104858	29.7305	1	0	0.5970	60792
13	0.5000	131072	27.6371	6	0	2.4740	65090

(a) Original

(b) 0.0191 bpp/44.4 dB

(c) 0.1 bpp/36.5 dB

(d) 0.5 bpp/27.6 dB

Figure 9 Experimental results of Lena JPEG image with Q-80.

7. Conclusion

1. Lossless data hiding can be obtained by using the proposed histogram pair technique. The histogram pair based lossless data hiding technique can be applied to JPEG quantized 8x8 block DCT coefficients.
2. The selection of optimum threshold "T" and optimum JPEG coefficient region "R" for data embedding can further improve the PSNR of stego image with a given payload.
3. In addition, the proposed histogram pair JPEG image lossless data hiding technique does not increase the size of JPEG image file noticeably.
4. Specifically, when 1000 bits are embedded into 512x512 Lena JPEG image (0.0038 bpp) with Q-factor ranging from 30 to 100, the PSNR of the resultant stego images ranges from 41 dB to 58 dB. Furthermore, after the data hiding, the increase of the JPEG file size ranges from 1.7% to 0.1%. The amount of image-file-size increase ranges from 264 bits to 190 bits. These results indicate satisfactory performance.
5. Compared with prior-arts [6,7], it appears that the proposed scheme can achieve higher payload.
6. It is expected that this new scheme can be applied to the I-frame of MPEG video as well.

References

[1] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *Proceedings of IEEE International Symposium on Circuits and Systems,* vol. 2, pp. 912-915, Bangkok, Thailand, May 2003; *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, March 2006.

[2] A. Leest, M. Veen, and F. Bruekers, "Reversible image watermarking", *Proceedings of IEEE International Conference on Image Processing*, vol.2, pp.731-734, Rochester, NY, USA, September 2003.

[3] B. Yang, M. Schmucker, W. Funk, C. Busch, S. Sun, "Integer DCT-based reversible watermarking for images using companding technique," *Proceedings of SPIE EI, Security and watermarking of Multimedia Content, Electronic Imaging*, San Jose, CA, USA, 2004.

[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," *Proc. IEEE International Conference on Multimedia & Expo (ICME05)*, Amsterdam, Netherlands, July 6-8, 2005.

[5] G. Xuan, Q. Yao, C. Yang, J. Gao, P. Chai, Y. Q. Shi, Z. Ni, "Lossless data hiding using histogram shifting method based on integer wavelets," *Proc. International Workshop on Digital Watermarking (IWDW06)*, November 2006, Jeju, Korea.

[6] J. Fridrich, M. Goljan, and R. Du, "Lossless data hiding for all image formats," *Proc. of SPIE, Electronic Imaging 2002, Security and Watermarking of Multimedia Contents IV*, vol. 4675, San Jose, California, pp. 572-583, 2002.

[7] J. Fridrich, M. Goljan, Q. Chen, and V. Pathak, "Lossless data embedding with file size preservation," *Proc. SPIE Electronic Imaging 2004, Security and Watermarking of Multimedia Contents*, San Jose, California, January 2004.

[8] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards,* CRC Press, Boca Raton, FL, 1999.