





Fidelity-Preserved Reversible Data Hiding in JPEG Images Based on Adaptive Position Selection

Zhen Yue¹ , Hua Ren² , Ming Li³, Cun-liang Liang¹, Ben-zhai Hai¹, and Rui-ping Li¹

¹ Faculty of Education, Henan Normal University, Xinxiang 453007, Henan, China

² Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, School of Computer, Beijing University of Posts and Telecommunications, Beijing 100876, China

renhuahtu@163.com

³ College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China

Abstract. Reversible data hiding (RDH) in JPEG images has gained considerable attention from scholars in recent years. However, it is not easy for JPEG images to exploit the pixel redundancy for RDH design like uncompressed images. In this paper, we propose a fidelity-preserved RDH in JPEG images. The proposed scheme embeds the secret data using the histogram shifting method after considering different distortion costs of non-zero AC coefficients at different positions. According to the block entropy obtained by summing the squares of all pixels of each 8×8 block, the quantified DCT blocks are divided into the smoothed blocks and texture blocks, and the obtained smoothed blocks are sorted in ascending order. The AC coefficients with values 1 and/or -1 in the smoother blocks are used for the embedding, the non-zero AC coefficients are shifted toward the right or the left at most by one, and the others remain invariant. Experiment results have verified the effectiveness of the proposed method.

Keywords: Reversible data hiding · Block division · JPEG · Histogram shifting

1 Introduction

Reversible data hiding (RDH) [1,2], as one particular type of data hiding, is always a core means for content-protected applications where true fidelity is needed, such as medical and military image processing. By concealing secret data into cover images imperceptibly, one can extract the embedded data without any error, and recover the original cover image perfectly. The existing RDH schemes typically can be classified into three types: lossless compression [3,4], difference

expansion [5], and histogram shifting (HS) [6]. The lossless compression-based methods compressed the redundancy characteristics of the original carriers losslessly to provide the space for the embedding process. The difference expansion-based methods divided adjacent two pixels into one pair, and each pair carried one bit of the secrecy streams. The histogram shifting-based approaches selected peak pixels to carry secrecy and shifted the other pixels close to the peak position toward the left or right direction to vacate the space. Although the existing RDH methods have achieved significant breakthroughs in terms of visual quality and embedding capacity, the limitation of the application is apparent because only the uncompressed images can be the original carriers. Compared with existing uncompressed images, the compressed images that possess less redundancy are broadly transmitted on the Internet. Thus, it is quite essential but hard to conceal secrecy into compressed images reversibly. Among different formats of the compressed images, the joint photograph experts group (JPEG) format is of widespread popularity when transmitted [4, 7]. In this light, developing RDH in JPEG images has practical application value in real life.

The existing methods regarding RDH in JPEG images can be divided into four categories: modifying the quantization table [7, 8], modifying the Huffman table [9–11], modifying the quantized DCT coefficients [12–18], and RDH in the encrypted JPEG images [19, 20]. The quantization table modification-based method [8] must perform a preprocessing process before embedding. Some elements of the quantization table are divided by an integer, and the corresponding quantized coefficients are multiplied by the same integer to create the space for embedding. A higher visual quality of the marked image accompanies an evident increase in the file size. The Huffman table modification-based methods [9–11] modified the Huffman table by building the mapping between the variable length codes (VLC). Such methods provided higher visual quality and almost unchanged file size, yet with a low embedding rate, even for a 512×512 image carrying only hundreds of bits. The quantized DCT coefficients-based modification methods [12–21] modified DCT coefficient values to hide secret messages, and the used quantization table was invariant throughout the embedding. The embedding rate of DCT coefficient modification methods is substantial with certain file sizes.

Recently, Huang et al. [13] have presented a novel DCT coefficient modification-based RDH scheme in JPEG images. The scheme utilizes a block selection strategy for selecting the smoothed block and the texture block. The zero AC coefficients in each block remain invariant, and only AC coefficients with the values 1 and -1 are used to carry secret messages. The zero AC coefficients determine the block positions used for the embedding, making correct data extraction and perfect image restoration available. Wedaj et al. [14] improved the method [13] by considering the coefficient distributions and the position modification cost. Unfortunately, the scheme [14] altered the quantization table during embedding. The improvement version [15] utilized all quantized non-zero AC coefficients to embed secret messages. However, the visual quality of the marked JPEG image is poor, and the Huffman table must be modified to

preserve the file size of the final image. Xiao et al. [17] presented a variant of [13] based on multiple two-dimensional histograms, which effectively insured higher visual quality. Based on this trait, they then presented an improvement version with multiple histogram modification in JPEG images [18]. Yao et al. [19] introduced a dynamic allocation method to arrange the embedding bits rationally to implement less distortion in dual JPEG image. Xuan et al. [20] just utilized minimum entropy and histogram-pair to improve the method [13]. Four thresholds, including embedding amplitude threshold, fluctuation threshold, lower frequency threshold, and higher frequency threshold, were considered during embedding. The target frequencies used for embedding might brought multiple shifts when embedding capacity increases, whereas the image distortion and file size are satisfactory. Unfortunately, the method did not consider the coefficient distributions and the modification cost of the AC coefficients position, only manually determining the embedding interval $[T_L, T_H]$, where T_L and T_H represent lower threshold and upper threshold, respectively.

Based on the above discussion, we detail an improvement version based on the method [20]. First, we utilize the block entropies obtained by summing the squares of all pixels of each 8×8 block for generating the smoothed blocks and the texture blocks. After sorting the smoothed blocks in ascending order according to the obtained entropy values, the blocks with lower entropies have priority over the blocks with higher entropies during the embedding. In addition, enlightened by the method [14], the improvement also considers two other aspects: the coefficient distributions and the modification cost of the position. In other words, we calculate the distortion cost in each non-zero AC position by considering all of the obtained smoothed blocks and select non-zero AC coefficients at positions with lower distortions to preferably embed. Compared with the method [20], the main contributions of the proposed scheme are summarized below.

- We sort the obtained smoothed blocks in ascending order according to the obtained entropy values. Thus, we can preferably hide the secret messages into the blocks with smaller entropies.
- Among all the smoothed blocks of the embedded secret bits, we consider the distortion cost in each position of one block and preferably choose non-zero AC coefficients at the positions with lower distortions for embedding the secret bits.

The rest parts of the paper are organized as follows. Section 2 briefly describes the previous arts. The proposed method is elaborated in Sect. 3. Experiment results are given in Sect. 4 to show the effectiveness of the proposed method. Finally, the paper is concluded in Sect. 5.

2 Previous Arts

2.1 Overview of JPEG Compression Standard

Joint photograph experts group (JPEG) as a widely used image format is a lossy compression format. The unimportant data of the original image is

missing when compressed, and thus the compressed image is distorted. The existing JPEG encoder consists of three components, namely DCT, quantizer, and entropy encoder. Figure 1 shows the main steps of JPEG compression. The original image is processed into a series of sized 8×8 image blocks, and each block is transformed from spatial domain to frequency domain using a two-dimensional DCT function. These obtained DCT coefficients are quantized and rounded to the nearest integer according to the quantization table. The quantized coefficients are re-arranged in zigzag scanning order and pre-processed using the differential pulse code modulation (DPCM) and run-length en-coding (RLE). The symbol string is Huffman-coded to obtain the final compressed bit streams. After pre-pending the header, someone can generate the final JPEG image.

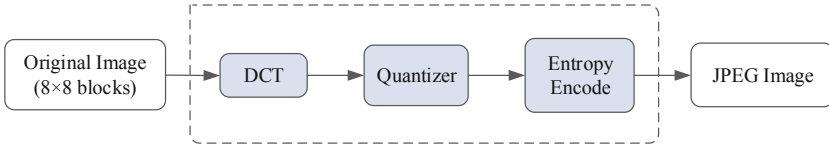


Fig. 1. Block diagram of JPEG encoder.

Mathematical definitions of 8×8 DCT transformation and inverse DCT are formulated as follows:

$$F(u, v) = c(u)c(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \quad (1)$$

$$f(x, y) = \sum_{u=0}^7 \sum_{v=0}^7 c(u)c(v) F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \quad (2)$$

where:

$$c(u) = \begin{cases} \frac{1}{2\sqrt{2}}, & \text{if } u = 0 \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

During compression, the DCT coefficients of each 8×8 block are quantized based on a quantization table. Figure 2 shows an example of DCT block quantization. Figure 2(a) represents the zigzag scanning sketch, Fig. 2(b) is the corresponding position distribution in each 8×8 block, and Fig. 2(c) shows the final quantized DCT coefficients obtained by divided each DCT coefficient with the quantization table as follows.

$$C_q(u, v) = \text{InterRound} \frac{F(u, v)}{q(u, v)} \quad (4)$$

In Eq. 4, $f(x, y)$ denotes the pixel value at block position (x, y) , $c(u) = c(v)$, $u, v = 0, 1, \dots, 7$, $F(u, v)$ and $C_q(u, v)$ denote the original DCT coefficient and

the quantified DCT coefficient at block position of the u -th row and the v -th column, respectively. $q(u, v)$ denotes the corresponding step of quantification table \mathbf{Q} . As shown in Fig. 2, the first coefficient of the quantized block is the DC coefficient, which keeps invariant during the embedding. Except for the first DC coefficient, the remaining ones are the AC coefficients. By remaining the zero AC coefficients invariant, the proposed method shifts the AC coefficients larger than 1 and less than -1 to vacate the room, and uses the AC coefficients with values 1 and -1 for the embedding.

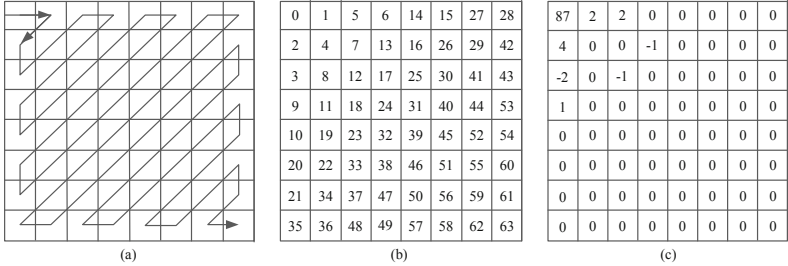


Fig. 2. An example of DCT block. (a) Scanning order; (b) zigzag order; (c) quantized block.

2.2 Overview of Xuan et al. Method

Xuan et al. [20] have presented a novel histogram-based RDH in JPEG images recently. In the method, the minimum entropy is applied to select the smoothed blocks of the embedding process, and the histogram pairs of the smoothed blocks is used to carry the secret message bits. Let us denote the embedding threshold as T , the secrecy messages as b , the quantized AC coefficients as x , the marked quantized coefficients as x' , and the embedding interval as $[T_L, T_H]$. In each block, not all non-zero AC coefficients hide the secret bits, but only the non-zero AC coefficient values that fall into the range of $[T_L, T_H]$ hide the secrecy information. The detail of the embedding process is

$$x' = \begin{cases} x + \text{sign}(x) \times b, & \text{if } |x| = |T| \\ x + \text{sign}(x), & \text{others} \end{cases} \quad (5)$$

where

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases} \quad (6)$$

The extraction is the reverse of the embedding. The details of the data extraction and image recovery are as follows.

$$b = \begin{cases} 0, & \text{if } |x'| = |T| \\ 1, & \text{if } |x'| = |T| + 1 \end{cases} \quad (7)$$

$$x = \begin{cases} x', & \text{if } |x'| = |T| \\ x' - \text{sign}(x'), & \text{if } |x'| \geq |T + 1| \end{cases} \quad (8)$$

The embedding threshold T is an embedding sequence, which can be expressed by

$$[T, -T, T - 1, -T + 1, \dots, 2, -2, 1, -1, 0] \text{ for } T > 0 \quad (9)$$

$$[T, -T - 1, T + 1, -T - 2, T + 2, \dots, 2, -2, 1, -1, 0] \text{ for } T < 0 \quad (10)$$

The embedding threshold T relies on a given payload size, and a larger payload leads to more histogram pairs to participate in the data embedding. The non-zero AC coefficients carry the secret bits. For example, assume two histogram pairs as $[0, -1]$ and $[1, 0]$, the embedding threshold T as 1. Xuan et al. [20] firstly embed a part of the payload into the AC coefficients with the value 1, and then embed the remaining into the AC coefficients with the value -1 . The embedding process is implemented according to the arranged sequence of Eq. 5 and Eq. 6. However, two issues are non-negligible. One is that the method [20] only uses the minimum entropy to obtain the smoothed blocks, and it does not include an entropy-based block order which can preferably choose these blocks with smaller entropies to embed. The other is that the method does not consider the distortion cost on each coefficient position in each block. Although selecting the embedding interval $[T_L, T_H]$ to embed can reduce certain embedding distortion, it is not the best choice.

3 The Proposed Method

In this section, a novel RDH scheme in JPEG images is proposed to achieve fidelity preservation of protected images. In the following, we detail the minimum entropy-based block selection and block order, AC coefficient position selection and position order, the data embedding, and the data extraction and restoration successively.

3.1 Minimum Entropy-Based Block Selection and Block Order

Suppose that an original image is divided into a series of equal-size $n \times n$ blocks. If the block size n is 8, there are a total of 4096 blocks for a sized 512×512 image. The entropy E under Gaussian distribution is defined as follows.

$$E = \ln \sqrt{2\pi e \sigma^2} \quad (11)$$

$$\sigma^2 = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (I(i, j) - \mu)^2 \quad (12)$$

where e denotes a natural constant, $I(i, j)$ is the pixel value in the i -th row and the j -th column, n is set as 8 when taking usual JPEG compression, and

represents the mean value of one block. Next, a fluctuation value F measures the smoothness of each coefficient block:

$$F = \frac{1}{n^2 - 1} \left(\sum_{i=1}^n \sum_{j=1}^n x(i, j)^2 - x(1, 1)^2 \right) \quad (13)$$

where $x(i, j)$ is the coefficient value in the i -th row and the j -th column, $x(1, 1)$ is the DC coefficient value. Since only AC coefficients are utilized to measure the block smoothness, the first DC coefficient should not be included. The computation of the smoothness of each block in the transform domain by using Eq. 13 is exactly equal to that in the spatial domain by using Eq. 12. In other words, we can have

$$F = \frac{1}{n^2 - 1} \left(\sum_{i=1}^n \sum_{j=1}^n x(i, j)^2 - x(1, 1)^2 \right) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (I(i, j) - \mu)^2 = \sigma^2 \quad (14)$$

Thus, instead of exploiting AC coefficients to classify the smoothed block and the texture block, the pixels variance of one block is utilized in the spatial domain. For a given threshold T_F , only the blocks smaller than T_F are selected for embedding, whereas the blocks larger than T_F are skipped. After all entropies of smoothed blocks are obtained, we sort the entropies in ascending order to preferably choose the blocks with smaller entropies for data embedding.

3.2 AC Coefficient Position Selection and Position Order

In this section, we consider the AC coefficient position selection and position order before embedding, and we will not modify the used quantization table. The purpose of selecting the AC coefficient positions is to find the positions with smaller distortion costs when considering the coefficient distributions. The core of position order is to preferably embed secret messages into non-zero AC coefficients at positions with smaller distortion costs.

To select smaller distortion-causing position, the shiftable AC coefficients, the em-beddable AC coefficient, and the quantization table are used to calculate a metric called embedding efficiency η_i for each position $i \in \{1, 2, 3, \dots, 63\}$.

$$\eta_i = \sum_{n=1}^{N_1} \frac{E(i, n)}{(S(i, n) + E(i, n)/2) \times Q_i^2} \quad (15)$$

where $E(i, n) \in \{0, 1\}$ means that the AC coefficient at position i of the n -th block is embeddable 1 or not 0, $S(i, n) \in \{0, 1\}$ denotes the AC coefficient at position i of the n -th block is shiftable 1 or not 0, Q_i denotes the quantization table entry at position i , and $N_1 \in [0, 4096]$ represents the total number of the smoothed block depending on the threshold T_F . According to Eq. 15, one can easily know that $E(i, n)$ actually computes the total embedding capacity at position i , and $S(i, n) + E(i, n)/2$ computes the total modified amount at position i (assuming the payload is pseudo-random, approximately half of the

embeddable will cause modification), i.e., the total distortion. Thus, the best position to embed will have the largest embedding efficiency, and vice versa. Once all embedding efficiencies are calculated for 63 positions, the values will be sorted from the highest to the lowest. For example, if the positions (4,9,14,3,2,7) are determined for embedding, the secret message bits will be successively hide into the positions (2,3,4,7,9,14) to ensure perfect data extraction. It is quite different from the method [20], which selects continuous embedding position to modify, such as (4,5,6,7,8,9).

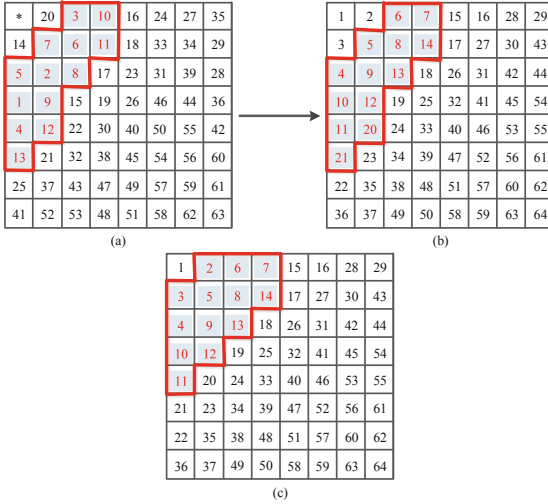


Fig. 3. Comparison on location selection for data embedding of our scheme and [20]. (a) Priority selection of embedding location for our scheme; (b) the corresponding selected embedding region; (c) the selected embedding region for [20].

Table 1. Various PSNRs for embedding data into Lena under quality factor QF = 80.

Lena		Xuan et al. method [20]	Proposed method
PL (bpp)	T_F	PSNR (dB)	PSNR (dB)
0.01	22	54.32	54.56
0.02	70	50.44	50.52
0.03	150	48.13	48.23
0.04	220	46.26	46.73
0.05	1200	44.49	44.54

To describe the effectiveness of this strategy easily, Fig. 3 shows a comparison on location selection for data embedding of our scheme and [20]. The proposed method adopts the entropy sorting strategy, whereas the scheme [20] manually selects the fixed embedding area. Figure 3(a) is the priority selection of

embedding location according to AC coefficient position selection. The priorities marked with red color region are the final selected priorities for data embedding. Figure 3(b) is the selected coefficient region of our scheme according to the priority in Fig. 3(a), and Fig. 3(c) is the manually selected region of the scheme [20]. Depending on these two ways used for ours and the scheme [20], The PSNR results under different embedding rates are listed in Table 1, where PL means the embedded payload. The results imply our sorting strategy outperforms the manually selected fixed region [20]. In our scheme, only the lower entropies are used for data embedding, which leads to lower visual distortion for the directly decrypted image. The manually selected fixed region [20] may contain the unexpected modified coefficients, thus incurs a worse result, even under various embedding rates.

For histogram shifting-based RDH in JPEG images, some side information should be processed to ensure true reversibility. The side information of the proposed method includes the size of the secret messages, the length of the threshold T_F , and the position map information. The maximum payload size needed to transmit is $\log(W \times H)$, where W and H are the weight and the height of an image. For an image sized 512×512 , the maximum length needed to be recorded should be 18 bits. The threshold T_F is exploited to select the smoothed blocks. The maximum length of the value is 13 bits. The binary position map (the value 1 is embeddable and the value 0 is not embeddable) records the embedding positions. Since there are a total of 63 AC coefficient positions in 8×8 blocks, the size of the binary position map should be 63 bits. To sum up, the total size of side information is 94 bits, which can be embedded in the first 94 LSBs of the DC coefficients. The original 94 LSBs of the DC coefficients are collected and regarded as a part of the payload to facilitate the perfect recovery of these selected DC coefficients.

3.3 Embedding, Extraction and Restoration

The proposed method embeds secret message bits into these AC coefficients with the values 1 and -1 . Non-zero AC coefficients close to the AC coefficients with the values 1 and -1 are shifted to vacate the room. All zero AC coefficients are invariant during the embedding. The following depicts the embedding algorithm of the proposed method.

$$x' = \begin{cases} x + \text{sign}(x) \times b, & \text{if } |x| = 1 \\ x + \text{sign}(x), & \text{others} \end{cases} \quad (16)$$

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases} \quad (17)$$

where x and x' denote the quantized non-zero AC coefficients and the marked quantized non-zero AC coefficients, $b \in \{0, 1\}$ denotes the secret messages to be embedded, and $\text{sign}(\bullet)$ denotes a sign function. According to Eq. 16 and Eq. 17, we can embed secret messages into the AC coefficients with the values 1 and -1 .

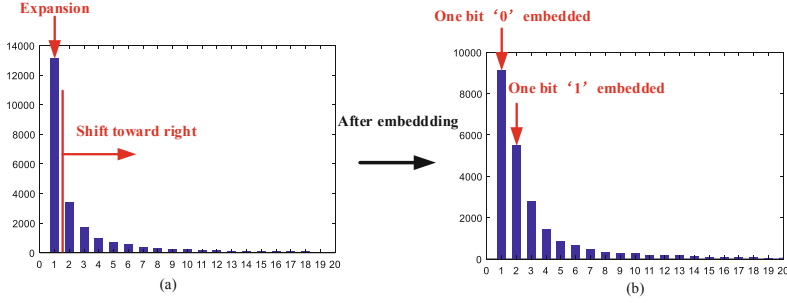


Fig. 4. Local details of histogram of the non-zero AC coefficients of the Lena image. (a) the original histogram; (b) the shifted histogram.

In the following, we take an example to explain the histogram shifting-based embedding process. Suppose the quality factor $QF = 80$, $T_F = 300$, the selected embedding position as $(10, 9, 6, 11, 4, 8, 5, 13, 12, 7, 14, 20, 21, 3, 19, 15, 18, 17, 25)$. Figure 4 shows the local details of the histogram shifting process, where the ordinate axes represent the statistic amount on the above-specified positions and the horizontal axes represent different non-zero AC coefficients on the selected positions. Figure 4(a) shows the original histogram of the non-zero AC coefficients in image Lena, and Fig. 4(b) shows the corresponding shifted histogram. We can see that a majority of the AC coefficients gather in the value 1, which indicates the amount of the secret messages embedded.

When needed to extract the embedded secret messages and restore the original coefficients, the following formulas can be carried out.

$$b = \begin{cases} 0, & \text{if } |x'| = 1 \\ 1, & \text{if } |x'| = 2 \end{cases} \quad (18)$$

$$x = \begin{cases} x', & \text{if } |x'| = 1 \\ x' - \text{sign}(x'), & \text{if } |x'| \geq 2 \end{cases} \quad (19)$$

Data Embedding Steps. In the embedding process, AC coefficients with the values -1 and 1 are modified to hide secret messages.

Step 1: Divide the original image into a series of equal-size non-overlapping 8×8 blocks, and there are total 4096 blocks for an image sized 512×512 . Perform DCT quantization on each block, and obtain the quantized AC coefficients.

Step 2: Scan the total 4096 blocks from left to right from top to bottom. For each selected DCT coefficient block, use Eq. 14 to compute the entropy F , if $F > T_F$, obtain all smoothed blocks. And, sort the obtained entropies of all blocks from the low-est to the highest.

Step 3: Select non-zero AC coefficients at positions with lower distortion cost according to Eq. 15, and sort the corresponding positions from the highest to the lowest.

Step 4: Embed the message length L (represented by l_1 , 18 bits), the threshold T_F (represented by l_2 , 13 bits), and the position map (represented by l_3 , 63 bits) into the first 94 LSBs of the DC coefficients.

Step 5: Encrypt and shuffle the secret message using a pseudo-random sequence, and embed the encrypted secret message into the AC coefficients with the value 1 and -1 according to Eq. 16.

Step 6: After all secret messages are embedded, entropy-encode the obtained coefficients to get the marked JPEG file.

Extraction and Restoration Steps. Step 1: Entropy-decode the marked JPEG file to obtain the quantized DCT coefficients.

Step 2: Extract the message length L , the threshold T_F , and the position map from the first 94 LSBs of the DC coefficients.

Step 3: Scan the total 4096 blocks from left to right from top to bottom. For each selected DCT coefficient block, use Eq. 14 to compute the entropy F , if $F > T_F$, obtain all smoothed blocks. And, sort the obtained entropies of all blocks from the low-est to the highest.

Step 4: Extract the secret messages according to the position map from all sorted smoothed blocks according to Eq. 18, and restore the original coefficients according to Eq. 19.

Step 5: After extracting all secret messages, inversely shuffle and decrypt the extracted secret messages to recover the original secret messages. Entropy-encode the restored coefficients again to obtain the original JPEG file.

4 Experiment Results and Analyses

In our experiments, the secret messages are generated randomly, and the JPEG images are compressed with the optimal Huffman table using the IJG toolbox [22]. All of our experiments are based on the compressed JPEG images, and the obtained four compressed images (QF = 80) are shown in Fig. 5.

4.1 Effectiveness of the Proposed Method

In order to illustrate the effectiveness of the proposed method, the PSNR results and the file size of the marked images separately with QF = 70, 80, 90 under different embedding capacities are listed in Table 2. As expected, with the increase of payload bits, the PSNR value is correspondingly decreased. What is more, the PSNR values are increased with the growth of quality factor QF. The reason is straightforward because a larger QF will have a higher visual quality. In addition, PSNRs of the marked images are still acceptable when the payload size becomes 16000 bits, thus the proposed method is satisfactory.

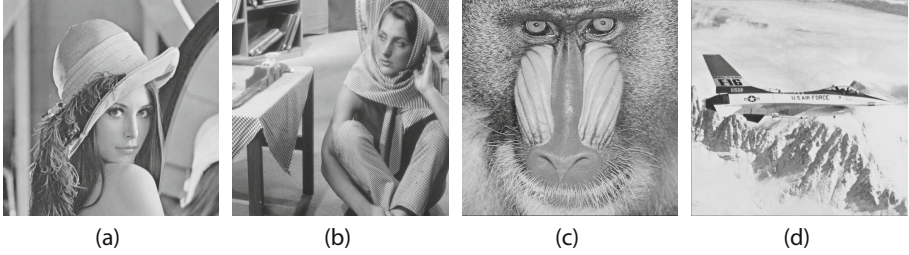


Fig. 5. Four compressed JPEG images. (a) Lena; (b) Barbara; (c) Baboon; (d) Airplane.

Besides, the threshold T_F of the proposed method determines the total number of the selected smoothed blocks. The smoothed blocks ensure the preferentially selected positions for embedding. Therefore, the settings of the threshold affect the PSNR values and file size of marked images. Figure 6 presents the PSNR values and file size of the marked Lena under three payloads by setting different T_F values. As seen in Fig. 6, when keeping the same payload bits, the PSNR values are decreased with the increase of T_F , whereas there will remain almost stable if the threshold T_F is large enough. Moreover, as for the embedding of the same payload, the file size of the marked Lena will become fluctuant within a small range when the T_F becomes larger. Thus, we suggest setting the threshold T_F into the scope $[20,450]$ to a better PSNR value and preservation of file size.

Table 2. PSNR results and file size separately with QF = 70, 80, 90 under different embedding capacity.

Image	QF	PL (bits)					
		6000	8000	10000	12000	14000	16000
Lena	70	48.04,6976	45.84,10272	44.40,12256	42.79,16128	41.51,19488	40.14,22888
	80	50.90,6928	49.15,9000	47.21,12104	46.11,14560	44.93,18232	43.86,22112
	90	54.87,6600	53.35,9072	52.06,11336	50.83,13856	49.93,16560	48.98,20088
Barbara	70	46.38,7712	44.33,10840	42.63,13936	41.11,17560	39.95,20880	38.76,24112
	80	49.43,6920	47.20,10272	45.65,13208	44.07,17096	42.80,20232	41.42,24336
	90	53.56,6976	52.00,9992	50.40,13000	49.09,16208	47.80,20216	46.68,23280
Baboon	70	45.23,7792	43.37,10392	41.80,13768	40.41,16816	39.17,20712	37.99,23304
	80	47.03,7480	45.24,10400	43.57,13672	42.12,17168	40.74,21704	39.83,24168
	90	49.63,8552	47.74,12016	45.88,16744	44.24,22128	43.39,25032	42.47,28816
Airplane	70	48.40,6520	45.96,10024	43.94,12960	42.20,16128	40.56,19632	39.14,22648
	80	51.09,6576	49.29,9032	47.16,12080	45.74,15696	44.41,18600	42.91,21832
	90	55.22,7240	53.60,8936	52.30,11672	51.23,14256	50.28,16888	49.01,20480

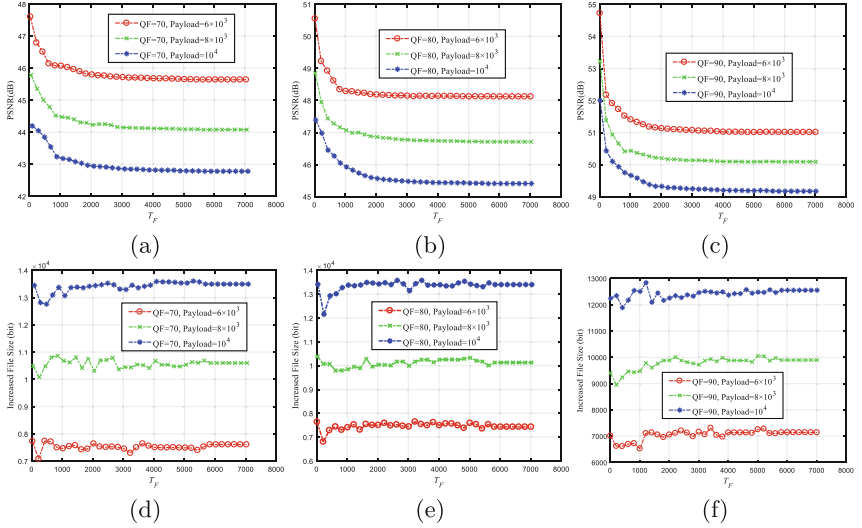


Fig. 6. PSNR values and file size of the marked Lena under three payloads by setting different T_F values. (a, d) PSNRs and file size of marked Lena with QF = 70; (b, e) PSNRs and file size of marked Lena with QF = 80; (c, f) PSNRs and file size of marked Lena with QF = 90.

4.2 Comparisons

Since the proposed method is an improvement version based on the method [20], the proposed method is compared with the method [20]. Tables 3, 4, 5, 6 show the comparison results under quality factor QF = 70 and QF = 80. The total selected position number of our method is consistent with the length of the embedding interval $[T_L, T_H]$ of the Xuan et al. method. The compressed images containing Lena, Barbara, Baboon, Airplane, are considered under the assumptions of the same T_F and the same payload size.

Note that in Tables 3, 4, 5, 6, parameters PL, FS, and Rate performance represent the embedded payload, the increased file size to the original JPEG image (in bits), and the rate of increased file size to the payload, respectively. The first column also lists the original JPEG size with byte representation. From which, we can observe that for different QFs, the original JPEG sizes with the same image are different. Moreover, with the increase of embedded payload bits,

the PSNR values of different images with QF = 70 are always lower than QF = 80, and the two comparison methods show consistency in this trend. In addition, from Tables Tables 3, 4, 5, 6, we can observe that the PSNR results obtained by the proposed method are better than that of Xuan et al. [20]. This is because the non-zero coefficients at the positions with smaller distortion costs are always preferentially selected to modify for embedding in our method; on the contrary, Xuan et al. method [20] only makes the modifications on the non-zero AC in the given embedding interval $[T_L, T_H]$. The FS and rate performance of the marked JPEG images in our method are not entirely better than that of the Xuan et al. method, because the proposed method always preferentially selects the non-zero coefficients at positions with smaller distortion costs to modify, which ensures the visual quality of marked JPEG images but does not always ensure the embedding capacity. In other words, the non-zero coefficients at positions with smaller distortion costs do not con-currently signify that the total embedding numbers in these positions are higher.

Table 3. Comparison results with Lena.

	PL (bpp)	T_F	PL (bits)	Xuan et.al method [20]			Proposed method		
				PSNR (dB)	Inc. FS (bits)	Rate (%)	PSNR (dB)	Inc. FS (bits)	Rate (%)
QF=70 (original JPEG size 29294 bytes)	0.01	20	2621	51.37	3384	29	52.71	3464	32
	0.02	30	5243	47.59	6712	28	48.73	6360	21
	0.03	80	7864	45.08	10,096	28	46.07	9720	24
	0.04	300	10,486	43.06	14,088	34	43.70	13,592	30
	0.05	300	13,107	39.69	17,880	36	42.05	17,840	36
QF=80 (original JPEG size 37937 bytes)	0.01	22	2621	54.32	2776	6	54.56	3120	19
	0.02	70	5243	50.44	5824	11	50.52	5960	14
	0.03	150	7864	48.13	9560	22	48.23	9648	23
	0.04	220	10,486	46.26	13,512	29	46.73	12,856	23
	0.05	1200	13,107	44.49	17,536	34	44.54	17,680	35

Table 4. Comparison results with Barbara.

	PL (bpp)	T_F	PL (bits)	Xuan et.al method [20]			Proposed method		
				PSNR (dB)	Inc. FS (bits)	Rate (%)	PSNR (dB)	Inc. FS (bits)	Rate (%)
QF=70 (original JPEG size 38880 bytes)	0.01	40	2621	50.74	3368	29	51.71	3240	24
	0.02	65	5243	46.7	6656	27	47.26	7424	42
	0.03	265	7864	42.42	10,520	34	44.31	10,376	32
	0.04	420	10,486	41.09	14,504	38	42.32	14,576	39
	0.05	450	13,107	39.55	18,560	42	39.64	19,616	50
QF=80 (original JPEG size 48335 bytes)	0.01	22	2621	53.03	3360	28	54.28	2944	12
	0.02	50	5243	48.94	7624	45	50.38	6288	20
	0.03	170	7864	46.37	10,920	39	47.31	10,048	28
	0.04	210	10,486	42.75	16,040	53	45.27	14,248	36
	0.05	250	13,107	40.4	19,456	48	42.23	19,608	50

Table 5. Comparison results with Baboon.

	PL (bpp)	T_F	PL (bits)	Xuan et.al method [20]			Proposed method		
				PSNR (dB)	Inc. FS (bits)	Rate (%)	PSNR (dB)	Inc. FS (bits)	Rate (%)
QF=70 (original JPEG size 62536 bytes)	0.01	70	2621	48.78	3224	23	49.9	3064	17
	0.02	220	5243	44.46	6880	31	45.41	6528	25
	0.03	200	7864	42.21	10,472	33	43.43	10,128	29
	0.04	300	10,486	40.29	15,264	46	41.45	14,168	35
	0.05	360	13,107	38.52	19,208	47	39.4	18,544	41
QF=80 (original JPEG size 78677 bytes)	0.01	190	2621	50.25	3400	30	50.41	2832	8
	0.02	200	5243	46.66	6872	31	47.2	6360	21
	0.03	430	7864	44.02	11,304	44	44.31	11,424	45
	0.04	440	10,486	41.35	15,192	45	42.89	14,440	38
	0.05	360	13,107	39.16	17,552	34	41.47	18,792	43

Table 6. Comparison results with Airplane.

	PL (bpp)	T_F	PL (bits)	Xuan et.al method [20]			Proposed method		
				PSNR (dB)	Inc. FS (bits)	Rate (%)	PSNR (dB)	Inc. FS (bits)	Rate (%)
QF=70 (original JPEG size 30636 bytes)	0.01	22	2621	51.41	3400	30	52.6	2720	4
	0.02	40	5243	47.2	6944	32	48.96	6016	15
	0.03	50	7864	44.34	10,584	35	46.09	9624	22
	0.04	150	10,486	42.11	14,240	36	43.43	13,664	30
	0.05	320	13,107	39.48	18,096	38	41.07	18,136	38
QF=80 (original JPEG size 38344 bytes)	0.01	14	2621	54.01	3080	18	54.93	3288	25
	0.02	40	5243	50.7	6304	20	51.38	6144	17
	0.03	120	7864	47.92	9512	21	48.52	9224	17
	0.04	270	10,486	45.92	13,384	28	46.32	13,000	24
	0.05	470	13,107	44.18	17,904	37	44.51	17,728	35

In addition, we compare the PSNR values and the increased file size under different embedding capacity with the methods [13, 14, 20]. Figure 7 shows the comparison results of different embedding capacity with QF = 70. Figure 8 shows the comparison results of different embedding capacity with QF = 80. Huang et al. method [13] originally uses non-zero AC coefficients with the values -1 and 1 to carry secret bits as well as other non-zero coefficients to vacate room for embedding, and the zero AC coefficients are used to select smoothed blocks. The improvement method [20] adopts histogram-pair based to embed secret messages into a given embedding interval $[T_L, T_H]$. If two histogram pairs $[0, -1]$ and $[1, 0]$ are used to carry secret messages, the method [20] firstly embeds a part of the payload into the AC coefficients with the value 1 , subsequently embeds the remaining part of payload into the AC coefficients with the value -1 . Therefore, we compare the three state-of-the-art methods [13, 14, 20] in the following. As seen, the PSNR values of the proposed method are larger than the other three methods [13, 14, 20], and the file sizes of the marked images are almost less than the methods [13, 14, 20]. In particular, these advantages will become more obvious when the compression rate is lower (compared with QF = 70 and QF = 80). It is because the proposed method always embeds the secret messages into the non-zero AC coefficients with the values 1 and -1 with lower distortion cost.

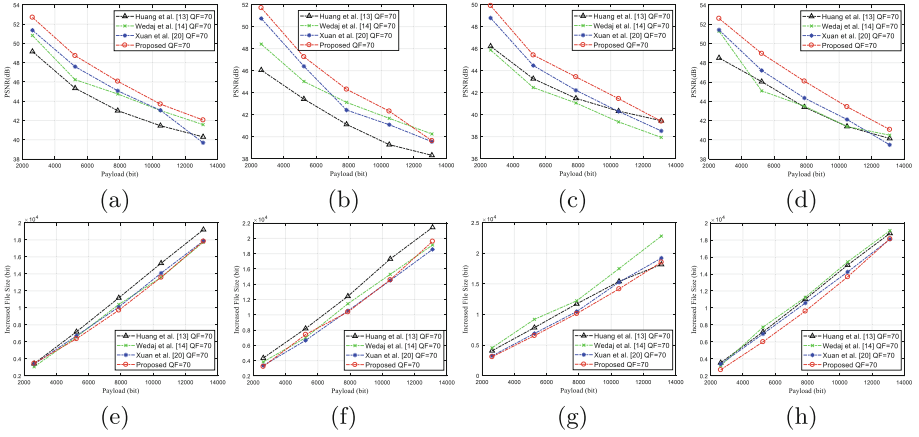


Fig. 7. Comparisons of different embedding capacity with $QF = 70$. (a, e) Lena; (b, f) Barbara; (c, g) Baboon; (d, h) Airplane.

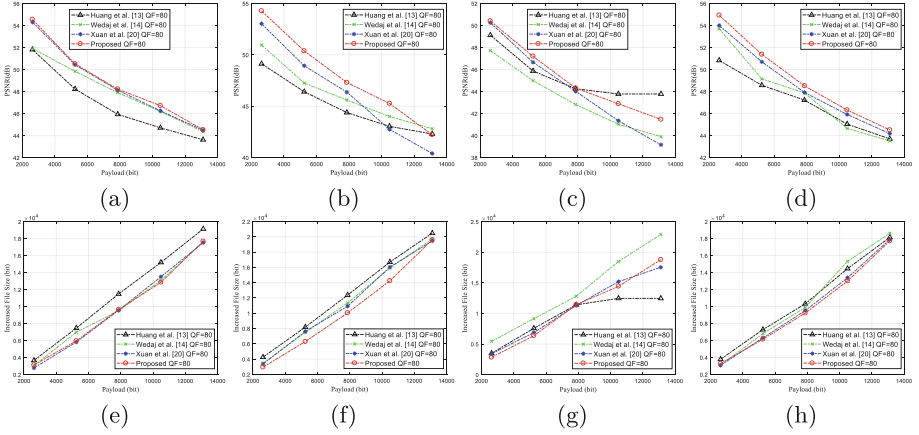


Fig. 8. Comparisons of different embedding capacity with $QF = 80$. (a, e) Lena; (b, f) Barbara; (c, g) Baboon; (d, h) Airplane.

5 Conclusion

In this paper, an improvement of the work [20] is proposed. In the process of block selection, we first obtain the entropy value of each block then use it to sort the smooth blocks in ascending order. In the position selection of smoothed blocks, instead of the fixed embedding selection utilized by [20], the distributions of embeddable, unchangeable and shiftable coefficients and the modification cost of the position in each block are considered. Thus, the proposed method can preferentially embed the secret messages into these 1 and -1 coefficients with lower distortion cost. Experiments have verified that the proposed method is

better than the method [20], especially when the compression rate QF is lower. However, this proposed method evenly divides the whole image into 8×8 blocks and does not consider the intra-block structural characteristics, which may limit the embedding capacity. In the future, we will consider using adaptive block selection strategy to divide the image into blocks of different sizes, such that the characteristics of the compressed image can be fully exploited.

Acknowledgements. We thank all the appending authors for their contributions to this article.

Funding Statement. This work is supported by the National Natural Science Foundation of China (No. 61602158), the Science and Technology Research Project of Henan Province (No. 212102210413). Start-up grant for doctoral research at Henan Normal University (QD2021096).

Conflict of Interest. The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Chung, K.L., Huang, Y.H., Chang, P.C., Mark, H.Y.: Reversible data hiding-based approach for intra-frame error concealment in H.264/AVC. *IEEE Trans. Circ. Syst. Video Technol.* **20**(11), 1643–1647 (2010)
2. Ren, H., Niu, S.: Separable reversible data hiding in homomorphic encrypted domain using POB number system. *Multimedia Tools Appl.* **81**(2), 2161–2187 (2021). <https://doi.org/10.1007/s11042-021-11341-w>
3. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding-new paradigm in digital watermarking. *EURASIP J. Adv. Signal Process.* **2002**(2), 185–196 (2002)
4. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding for all image formats. *SPIE Secur. Watermarking Multimedia Contents IV* **4675**, 572–583 (2002)
5. Tian, J.: Reversible data embedding using a difference expansion. *IEEE Trans. Circ. Syst. Video Technol.* **13**(8), 890–896 (2003)
6. Ni, Z.C., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. *IEEE Trans. Circ. Syst. Video Technol.* **16**(3), 354–362 (2006)
7. Chang, C.C., Lin, C.C., Tseng, C.S., Tai, W.L.: Reversible hiding in DCT-based compressed images. *Inf. Sci.* **177**(13), 2768–2786 (2007)
8. Wang, K., Lu, Z.M., Hu, Y.J.: A high capacity lossless data hiding scheme for JPEG images. *J. Syst. Softw.* **86**(7), 1965–1975 (2013)
9. Qiu, Y.Q., He, H., Qian, Z.X.: Lossless data hiding in JPEG bitstream using alternative embedding. *J. Vis. Commun. Image Represent.* **52**, 86–91 (2018)
10. Zhang, C., Ou, B., Tian, H.W., Qin, Z.: Reversible data hiding in JPEG bitstream using optimal VLC mapping. *J. Vis. Commun. Image Represent.* **71**, 1–12 (2020)
11. Zhang, C., Ou, B., Tang, D.: An improved VLC mapping method with parameter optimization for reversible data hiding in JPEG bitstream. *Multimedia Tools Appl.* **79**(27), 19045–19062 (2020)
12. Nikolaidis, A.: Reversible data hiding in JPEG images utilizing zero quantized coefficients. *IET Image Process.* **9**(7), 560–568 (2015)

13. Huang, F.J., Qu, X., Kim, H.J., Huang, J.W.: Reversible data hiding in JPEG images. *IEEE Trans. Circ. Syst. Video Technol.* **26**(9), 1610–1621 (2016)
14. Wedaj, F.T., Kim, S., Kim, H.J., Huang, F.: Improved reversible data hiding in JPEG images based on new coefficient selection strategy. *EURASIP J. Image Video Process.* **2017**(1), 1–11 (2017). <https://doi.org/10.1186/s13640-017-0206-1>
15. Liu, Y., Chang, C.C.: Reversible data hiding for JPEG images employing all quantized non-zero ac coefficients. *Displays* **51**, 51–56 (2018)
16. Xuan, G., Li, X., Shi, Y.Q.: Histogram-pair based reversible data hiding via searching for optimal four thresholds. *J. Inf. Secur. Appl.* **39**, 58–67 (2018)
17. Xiao, M.Y., Li, X.L., Zhao, Y.: Reversible data hiding for JPEG images based on multiple two-dimensional histograms. *IEEE Signal Process. Lett.* **28**, 1620–1624 (2021)
18. Xiao, M.Y., Li, X.L., Ma, B., Zhang, X.P., Zhao, Y.: Efficient reversible data hiding for JPEG images with multiple histograms modification. *IEEE Trans. Circ. Syst. Video Technol.* **31**(7), 2535–2546 (2021)
19. Yao, H., Mao, F.Y., Qin, C.A., Tang, Z.J.: Dual-JPEG-image reversible data hiding. *Inf. Sci.* **563**, 130–149 (2021)
20. Xuan, G., Li, X., Shi, Y.Q.: Minimum entropy and histogram-pair based JPEG image reversible data hiding. *J. Inf. Secur. Appl.* **45**, 1–9 (2019)
21. Hou, D., Wang, H., Zhang, W., et al.: Reversible data hiding in JPEG image based on DCT frequency and block selection. *Signal Process.* **148**, 41–47 (2018)
22. Independent JPEG Group. <http://www.ijg.org/>. Accessed 12 Jan 2020