

Improved Lossless Data Hiding for JPEG Images Based on Histogram Modification

Yang Du¹, Zhaoxia Yin^{1,2,*} and Xinpeng Zhang³

Abstract: This paper proposes a lossless and high payload data hiding scheme for JPEG images by histogram modification. The most in JPEG bitstream consists of a sequence of VLCs (variable length codes) and the appended bits. Each VLC has a corresponding RLV (run/length value) to record the AC/DC coefficients. To achieve lossless data hiding with high payload, we shift the histogram of VLCs and modify the DHT segment to embed data. Since we sort the histogram of VLCs in descending order, the filesize expansion is limited. The paper's key contribution includes: Lossless data hiding, less filesize expansion in identical pay-load and higher embedding efficiency.

Keywords: Lossless, data hiding, histogram, VLC, JPEG.

1 Introduction

Data hiding is a technique for hiding secret message into digital cover, which can serve as authentication watermarking, annotation, and secret data for covert transmission.

With the rapid development of computer technology, data hiding is widely used in military, commercial, medical, financial, etc. As a branch of data hiding, there are many new steganography algorithms proposed in recent years [Zhang, Qin, Zhang et al. (2018); Ma, Luo, Li et al. (2018); Wang, Li, Shi et al. (2017)]. In some applications, like medical image system, military documents and multimedia archive management, any distortion is intolerable after data embedding. Therefore, the reversible data hiding (RDH) methods are proposed. Recently some reversible data hiding algorithms have been proposed, such as the difference expansion scheme [Tian (2003)] and the histogram shifting (HS) algorithm [Ni, Shi, Ansari et al. (2006)]. As a significant branch in RDH, recently more and more HS-based data hiding methods are proposed [Li, Yang and Zeng (2011); Wang, Li, Shi et al. (2017); Li, Zhang, Gui et al. (2017)]. For example, Li et al. [Li, Yang and Zeng (2011)] can exactly classify the smooth and rough pixels by using the local complexity parameter, and then adaptively embed the data bits into them. Since the smooth pixels can be embedded with more data on average, a better performance can be obtained at the very high capacity. Wang et al. [Wang, Ni, Zhang et al. (2017)] further

¹ Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education, Anhui University, Hefei, 230601, P.R. China.

² Department of Computer Science, Purdue University, West Lafayette, 47906, USA.

³ School of Communication and Information Engineering, Shanghai University, Shanghai 200072, P.R. China.

* Corresponding Author: Zhaoxia Yin. Email: yinzhaoxia@ahu.edu.cn.

improved the optimal bin-selection mechanism by using a developed rate and distortion model and can give a good embedding performance with low complexity by narrowing down the solution space. Li et al. [Li, Zhang, Gui et al. (2017)] proposed the multiple histograms modification (MHM) technique and took the expansion bins of histograms as parameters. Since the modification manner on each histogram can be different, the overall modifications could be optimized by adapting the parameters to the statistical properties of histograms. However, all of these HS-based methods [Ni, Shi, Ansari et al. (2006); Li, Yang and Zeng (2011); Wang, Li, Shi et al. (2017); Li, Zhang, Gui et al. (2017)] introduced above are designed for spatial domain and cannot be applied to compressed images directly.

JPEG is the most popular file format in relation to digital images, which compress the image data effectively. Therefore, we focus on lossless data hiding based on histogram shifting for JPEG images. However, above HS-based methods cannot be applied to JPEG images directly. Consider the less redundancy, modification-sensitive and increased filesize, RDH for JPEG images is more difficult than uncompressed images. Xuan et al. [Xuan, Shi, Ni et al. (2007)] proposed a HS-based method for JPEG images by embedding data into the JPEG quantized 8×8 block DCT coefficients. Xuan et al.'s method can obtain high-payload and unnoticeable filesize extension but the stego image is distorted. Huang et al. proposed a new HS-based method [Huang, Qu, Kim et al. (2016)] that zero coefficients remain unchanged and only coefficients with values 1 and -1 are expanded to carry message bits in.

Recently, Qian and Zhang proposed a lossless and filesize preserved scheme [Qian and Zhang (2012)] in JPEG bitstream by VLC-mapping. They achieved the goal of lossless embedding by replacing the original VLC by a new VLC and mapping the new VLC to the same RLV. Since the original VLC and the new VLC corresponding to the same RLV, the decoded image quality is no distortion entirely. To embed high capacity secret data, inspired by Qian and Zhang's method [Qian and Zhang (2012)], Liu et al. [Liu, Chang, Lin et al. (2016)] proposed a HS-based method for JPEG images, which modify the VLCs in JPEG bitstream directly. They count all the VLCs usage and generate the histogram of the RLVs which corresponding to the VLCs. Then modify the original VLC according to the secret data directly.

Although the embedding capacity is expanded, there are two problems exist in Liu et al.'s method [Liu, Chang, Lin et al. (2016)]. The first problem is that only modifying the VLC directly in the entropy-coded data will cause the decoded results occur error. Another problem is the X-axis of histogram generated according to Liu et al.'s method [Liu, Chang, Lin et al. (2016)] is not logical. In other words, the adjacent X-values of histogram are no correlation.

To solve the two problems, we propose a novel HS-based method and modify the JPEG file header to preserve the image quality with no distortion as Qian et al.'s method [Qian and Zhang (2012)]. In addition, since we sort the RLVs in terms of the corresponding VLCs' frequency in the entropy-coded data, the stego image filesize is closer to cover image than Liu et al.'s method [Liu, Chang, Lin et al. (2016)]. The rest of this paper is organized as follows. Section 2 briefly introduces the structure of JPEG images and Liu et al.'s method [Liu, Chang, Lin et al. (2016)]. Section 3 presents the proposed scheme in

detail. Detail experimental results with analysis and comparisons are given in Section 4. Section 5 concludes the paper.

2 Related works

Since the proposed method works on JPEG bitstream, the knowledge of JPEG image structure is required. Furthermore, inspired by Liu et al.’s method [Liu, Chang, Lin et al. (2016)], we proposed a lossless and reversible data hiding method. Thus, we introduce the structure of JPEG image at first and Liu et al.’s method [Liu, Chang, Lin et al. (2016)].

2.1 The structure of JPEG bitstream

A JPEG image consists of a sequence of segments, each beginning with a marker. Each marker begins with a 0xFF byte followed by a byte indicating what kind of marker it is. For example, the DHT (Define Huffman Table) segment begins with 0xFFD8 and the SOS (Start of Scan) segment begins with 0xFFDA. The structure of JPEG image is shown in Fig. 1.

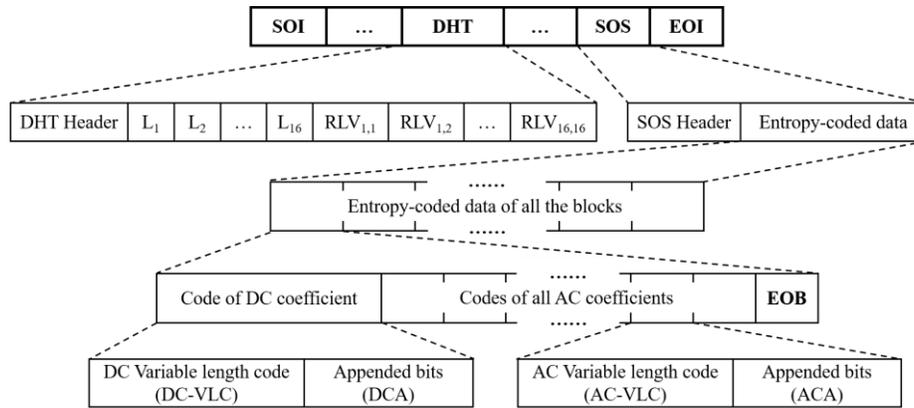


Figure 1: The structure of the JPEG bitstream

According to the JPEG guideline, after some preprocessing works, the pixels in the image are transformed into AC/DC coefficients. To compress the image filesize, the AC coefficients are encoded in Run-Length encoding format as intermediate symbols, (Run, Length) and (Amplitude). Further, to improve the compression rate, (Run, Length) is encoded in the format of VLC, and (Amplitude) is encoded in VLI (Variable Length Integer) format.

The DHT segment contains the Canonical Huffman table information, which is used for obtaining the VLCs. Each VLC is corresponded to a specific RLV that represents an AC coefficient in entropy-coded data. The VLCs are encoded by Canonical Huffman code, for instance, if the run/length value is “0/4”, the corresponding VLC is “1011”. For AC coefficients of luminance component, 162 different VLCs are corresponded to the run/length value from “0/1” to “F/A” accompanied with “0/0” (End of Block) and “F/0” (Zero Run Length). The length of VLC is between 2 to 16 bits.

According to JPEG guideline, if only change the VLC, the decoded results would be different compared with the cover image's decoded results. This is due to the VLC's corresponding RLV has changed, which affects the decoded results. To preserve the run/size not changed after the replacement of the corresponding VLC, we modify the DHT segment.

2.2 Liu et al.'s method

To obtain high embedding capacity for JPEG images, Liu et al. [Liu, Chang, Lin et al. (2016)] proposed a data embedding method based on histogram modification in entropy-coded data, which different from VLC mapping-based proposed by Qian et al. [Qian and Zhang (2012)]. The detail embedding steps can be summarized as follows:

Input: An origin JPEG bitstream and a secret bitstream.

Output: A stego JPEG bitstream, p and z .

Step 1: Parse the JPEG image and extract all the used-VLCs and unused-VLCs. Generate the histogram of run/size which begins with "0/0" and ends with "F/A". The Y -axis in the histogram represents the occurrence number of each run/size's corresponding VLC in the entropy-coded data.

Step 2: Find a peak point p and a zero point z and save the values of p and z as overhead information. The zero point z is the nearest zero point to p .

Step 3: Scan all of the VLCs in the entropy-coded data sequentially. Shift all the run/length values to right by one unit between p and z (not including p and z). For instance, the run/length "0/2", "0/3", "0/4" and "0/5" are shifted to "0/3", "0/4", "0/5" and "0/6". The corresponding VLC will be changed according to the shifted run/length. For instance, the VLC="1011" (corresponding run/length value is "0/4"), will be changed to "11010" (corresponding run/length value is "0/5"). Once the VLC that corresponds to run/length value equal to p is attained, replace the VLC or not depend to the secret bit that will be embedded. If the secret bit is "0", the VLC will be maintained and if the secret bit is "1", the VLC will be replaced by another VLC that corresponds to the run/length value equals to $p+1$.

Step 4: Generate a stego JPEG bitstream.

After the above steps, a stego JPEG bitstream is generated. However, the stego JPEG bitstream will cause the results occur error during JPEG decoding. This is because only VLCs in the entropy-coded data are modified. The size value of the modified VLC's corresponding RLV is not equal to the length of original appended bits. In addition, the RLV sorting manner is also not optimal for the histogram of RLV. Because the VLC uses Huffman coding so sorting by VLC code length order can make the file expansion generated after embedding data effectively reduce. Based on the above analysis, we propose an improved lossless data hiding scheme for JPEG images.

3 Proposed method

Inspired by Qian et al.'s scheme [Qian and Zhang (2012)], our method modified the DHT segment of JPEG header based on Liu et al.'s scheme [Liu, Chang, Lin et al. (2016)], resulting in a normal JPEG image decoding and no distortion of image quality. Thus, it is

a lossless data hiding scheme. At the same time, the file expansion of the secret JPEG bitstream is significantly reduced from that of Huang et al. [Huang, Qu, Kim et al. (2016)] and Liu et al. [Liu, Chang, Lin et al. (2016)] and the performance is better than the above. The overall framework of the proposed method is depicted in Fig. 2, including the data embedding process and extraction process. The details of the proposed method and the procedures of data embedding and extraction are presented in the following subsections severally.

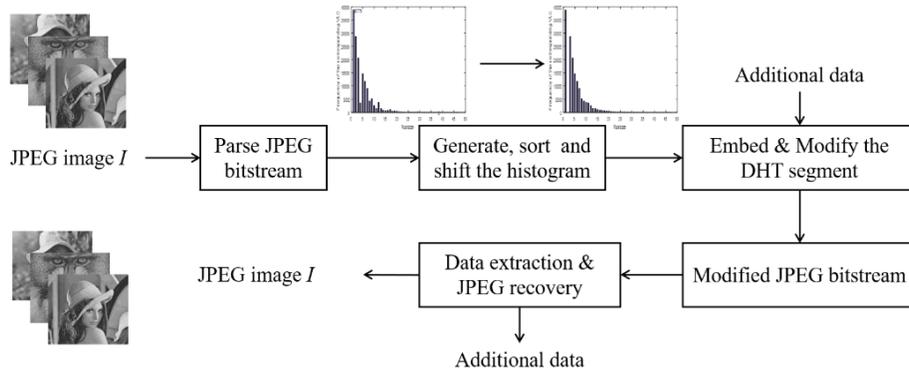


Figure 2: The overall framework of the proposed method

3.1 Lossless data embedding based on histogram modification

The histogram establishment process of our method is different from that of Liu et al. [Liu, Chang, Lin et al. (2016)]. Before generating the histogram, all VLCs in the entropy-coded data are first extracted and their frequency of occurrence is counted. According to the order of the RLVs in the DHT segment, a histogram of frequency distributions of VLCs is generated. We use the “Baboon” image with a quality factor of 80 as an example and compare it with Liu et al.’s histogram establishment scheme [Liu, Chang, Lin et al. (2016)]. Fig. 3 is a histogram of frequency distribution of VLC.

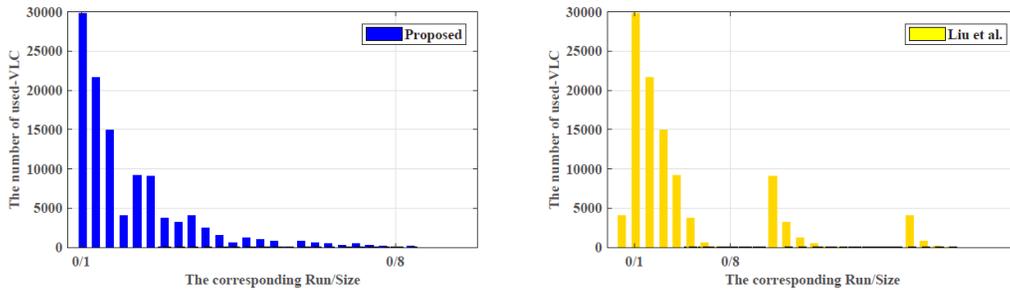


Figure 3: The original histogram of VLCs (“Baboon” image, QF=80)

3.1.1 Optimization of RLV ordering

Considering that, the encoder default Huffman code table is used in the JPEG file header, which is not the optimal Huffman code table for the current image. Therefore, there are still encoding redundancies that cannot be ignored in JPEG images. To reduce unnecessary encoding redundancies as much as possible, we adjusted the order of the RLVs according to the feature of the Huffman coding before the histogram was modified. The strategy we adopted is as follows:

1. Find the peak point p , and then look for the nearest zero point z from the peak point p according to the frequency distribution information of the VLC in the histogram. A large number of statistical results show that p is generally “0/1”, which is the ranked first RLV in the default Huffman code table. In Fig. 3, p is “0/1” and z is “0/8”.
2. Adjust the order of the RLVs corresponding to the VLC between the peak point p and the zero point z . After both p and z are determined, theoretically we can already perform histogram modification operations. However, considering that the ranking of the VLC frequency distribution between the peak point and the zero point is not strictly in accordance with the order of the RLVs in the default Huffman code table, the RLV sequence needs to be optimized. In other words, we need to readjust the RLV order in the DHT segment according to the descending order of the current VLC frequency distribution. The theoretical basis for this is that Huffman coding defines a shorter code length for frequently occurring codes, so adjusting the coding sequence based on the current VLC frequency of occurrence is equivalent to optimizing the Huffman coding, which means that we further reduce encoding redundancies. Fig. 4 shows a sorted histogram with the “Baboon” image with a quality factor of 80 as an example.

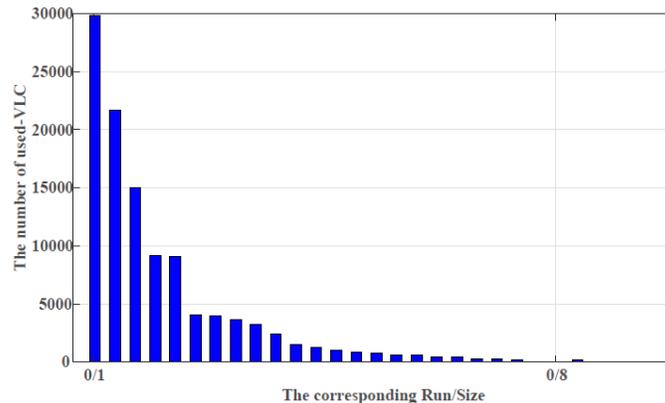


Figure 4: The sorted histogram of VLCs (“Baboon” image, QF=80)

Before optimization, 162 VLCs are divided into n categories according to code length. The length of the i -th category is L_i , that is:

$$C_i = \{VLC_1^i, VLC_2^i, \dots, VLC_{L_i}^i\} \quad (1)$$

Reorder the VLC and the corresponding RLV between the peak point p and the zero point z according to the descending order, then the VLC between p and z can be recorded as:

$$C'_i = \{VLC_1^{(i)'}, VLC_2^{(i)'}, \dots, VLC_{L_i}^{(i)'}\} \quad (2)$$

Where $VLC_1^{(i)'}, VLC_2^{(i)'}, \dots, VLC_{L_i}^{(i)'}$ is the VLC sequence of the i -th category adjusted by descending ordering. Assume that reduced encoding redundancy is R , and then R can be represented as:

$$R = \sum_{i=np}^{nz} \sum_{j=1}^{L_i} num(VLC_j^{(i)'}) \cdot (l-i) \quad (3)$$

Where np is equal to the category number of the category where the peak point is located; nz is equal to the previous category number of the category where the zero point is located; l is the code length of VLC_j^i corresponding to RLV_j^i before modification; $num(VLC_j^{(i)'})$ denotes the frequency of occurrence of $VLC_j^{(i)'}$ in the entropy-encoded data segment.

3.1.2 Modification of histogram of VLCs

After reordering the RLVs, we can embed data by histogram modification. First, the RLV between the peak point and the zero point is shifted rightward, e.g. “0/2” is modified to “0/3”. After the right-shift operation is completed, the frequency of the nearest VLC to the right of the peak point p is 0, that is, the zero point at this time becomes $p+1$. The following figure shows the right-shifted histogram of a “Baboon” image with a quality factor of 80.

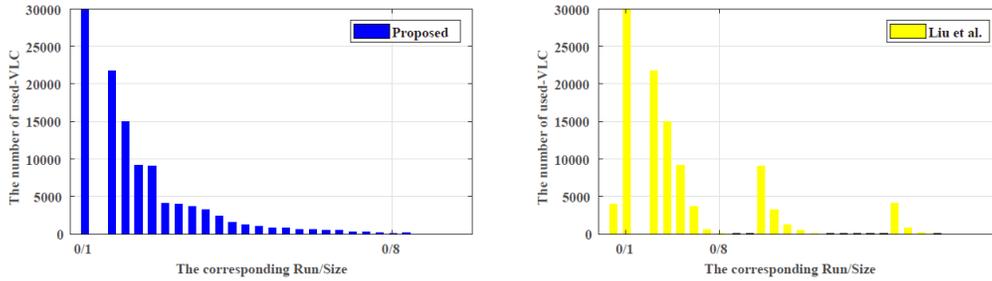


Figure 5: The shifted histogram of VLCs (“Baboon” image, QF=80)

3.1.3 Modification of DHT segment

In order to ensure that the modified JPEG image generated after data embedding has no distortion compared to the original image, we also modified the RLV sequence in the DHT segment while right-shifting the VLC histogram. Because the RLVs at different positions in the DHT segment correspond to different VLCs, we also shift the RLV in the

corresponding position by one unit at the same time according to the order in which the VLC is shifted right. In this way, the RLV corresponding to each VLC before and after the histogram modification is unchanged, which ensures that the result obtained by the decoding process is consistent with the original image and ensures that the image is free of any distortion. In addition, we modify the RLV with a value of $p+1$ to p so that when embedding data, regardless of embedding “0” or “1”, the RLV corresponding to VLC is p , ensuring that the decoding result is unchanged.

3.1.4 Calculation of the total filesize expansion

During the modification of the histogram, the filesize expansion is unavoidable. Because the length of the VLC may not be equal before and after the right shift of the RLV, if the modified VLC is longer than the original VLC, JPEG file size will increase. However, because we optimized the RLV ordering, the VLCs are sorted in descending order. Only the VLC with the lowest frequency in each category will increase the code length after modification, so it can effectively reduce the file expansion. Formula (4) denotes the file expansion caused by the modification of the histogram:

$$E = \sum_{i=np}^{nz} num(VLC_{L_i}^{(i)}) \cdot (j-i) \quad (4)$$

Where j represents the category number of the next category of C_i .

As mentioned in 3.1.1, the RLV sorting optimization is performed before the histogram is modified so the encoding redundancy is reduced. This part of the reduction is R , then the total file expansion (FE) can be represented as:

$$FE = \sum_{i=np}^{nz} num(VLC_{L_i}^{(i)}) \cdot (j-i) - \sum_{i=np}^{nz} \sum_{j=1}^{L_i} num(VLC_j^{(i)}) \cdot (l-i) \quad (5)$$

3.2 Data embedding & extraction

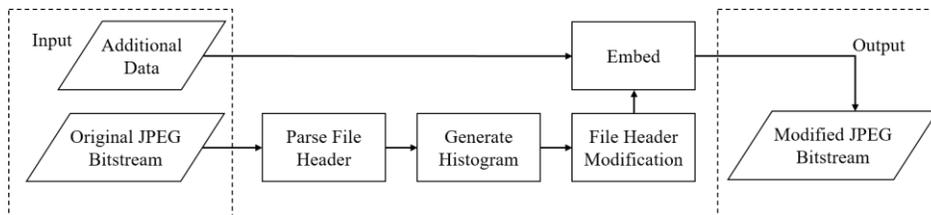


Figure 6: Data embedding process

The process of data embedding is shown in Fig. 6. The main process of the embedding phase is as follows:

Input: Secret bitstream and original JPEG bitstream.

Output: Modified JPEG bitstream.

Step 1: Parse the JPEG entropy-coded segment and the DHT segment and count the frequency of occurrence of each VLC.

Step 2: The frequency distribution histogram of the VLC is generated according to the order of the RLV values in the DHT segment. The location of the peak point and zero point are determined based on the length of the secret bitstream.

Step 3: Reorder the order of the RLVs in the DHT segment according to the optimized Huffman encoding operation mentioned in 3.1. Then perform histogram modification operation and DHT segment modification operation.

Step 4: Scan all the VLCs in the entropy-coded data and modify the VLC in the entropy-coded data according to the information that needs to be embedded. When the VLC corresponding to the RLV with the value of p is scanned, the data embedding operation can be performed. If “0” is embedded, the current VLC remains unchanged; if “1” is embedded, the current VLC is modified to the original VLC that correspond RLV is $p+1$. The entropy-coded data segment is scanned sequentially until all the data in the secret bitstream are embedded, then generate the modified JPEG bitstream.

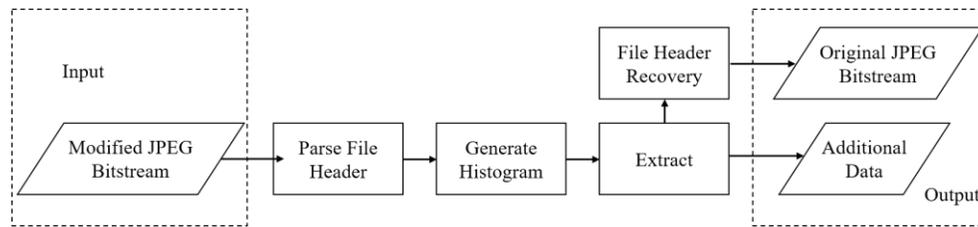


Figure 7: Data extraction process

The process of data extraction is shown in Fig. 7. The main flow of the secret data extraction and recovery phase is as follows:

Input: Modified JPEG bitstream.

Output: Secret bitstream and original JPEG bitstream.

Step 1: Parse the DHT segment of the JPEG file header and calculate the corresponding VLC value according to the RLV ranking order. Find the first two VLCs that correspond to the same RLV, namely the peak point and the nearest zero point.

Step 2: Scan all the VLCs in the entropy-coded data segment and extract all secret data according to the peak point and the length of the secret bitstream.

Step 3: After the secret data extraction is completed, the modified RLV sequence is restored to the original state according to the default Huffman code table and the corresponding VLC is modified to generate a recovered JPEG bitstream.

4 Experimental results and analysis

In order to test the performance of the proposed method and compare with other two methods, we have tested all the images from UCID image database. These images are converted to grayscale and further compressed to JPEG files with different quality factors. Fig. 8 shows the comparison of the payload and filesize expansion rate (FER) of the three schemes under different quality factors. Denote the original JPEG stream length as L , the FER can be represented as:

$$FER = \frac{FE}{L} \times 100\% \tag{6}$$

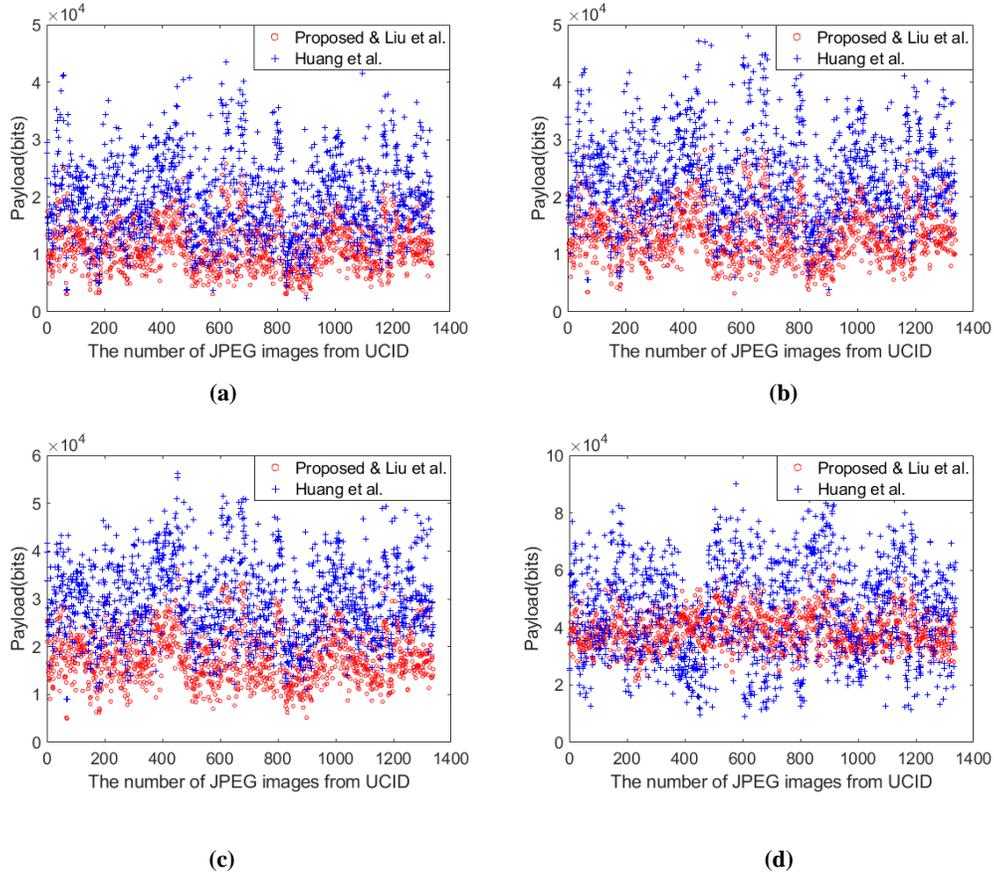
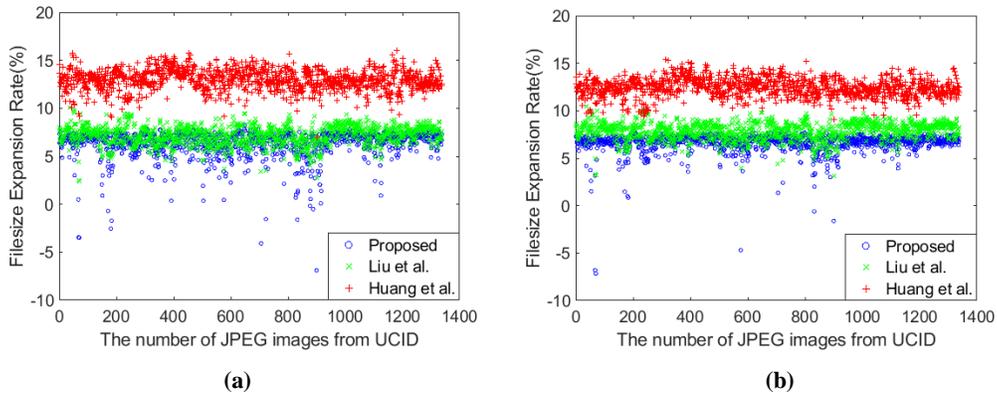


Figure 8: Comparisons of payload in UCID Image Library with different quality factors. (a) QF=70. (b) QF=80. (c) QF=90. (d) QF=100



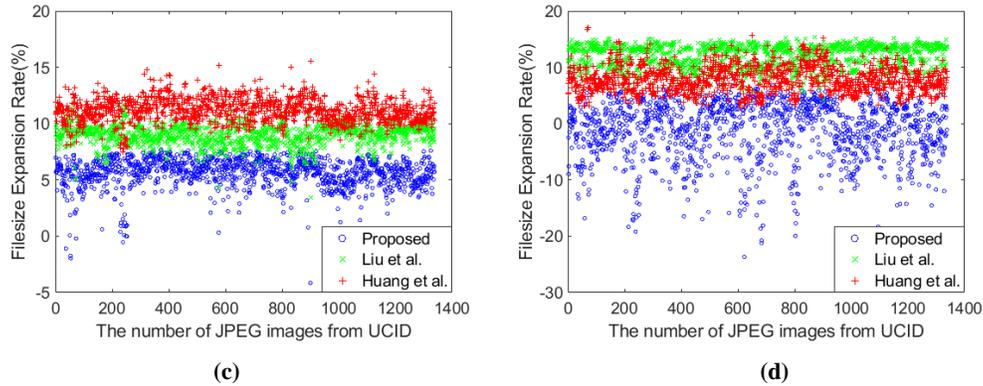


Figure 9: Comparisons of FER in UCID Image Library with different quality factors. (a) QF=70. (b) QF=80. (c) QF=90. (d) QF=100

According to the results in Fig. 8, Huang et al.’s method [Huang, Qu, Kim et al. (2016)] has the highest embedding capacity in full embedding among the three methods. For the FER, it can be seen from Fig. 9 that the FER of Huang et al.’s method [Huang, Qu, Kim et al. (2016)] is higher than other methods.

Table 1: Comparison of the three methods (QF=80)

Methods	Payload(bits)	FER (%)	PSNR
Huang et al. [Huang, Qu, Kim et al. (2016)]	23177	12.4255	34.7230
Liu et al. [Liu, Chang, Lin et al. (2016)]	13833	7.8310	0
Proposed	13833	6.6051	∞

Table 2: Comparison of the three methods (QF=90)

Methods	Payload(bits)	FER (%)	PSNR
Huang et al. [Huang, Qu, Kim et al. (2016)]	29579	11.1738	37.1107
Liu et al. [Liu, Chang, Lin et al. (2016)]	18089	9.0070	0
Proposed	18089	5.6061	∞

Tab. 1 and Tab. 2 respectively list the average values of all the indexes of all three methods when QF=80 and QF=90. The peak signal-to-noise ratio (PSNR) is an objective criterion for evaluating the visual quality of the image between the modified JPEG image and the original JPEG image. The higher the PSNR value, the higher the visual quality of the modified JPEG image. Since our proposed method has no effect on the image quality, i.e. lossless, the PSNR value can be regarded as ∞ . However, since Liu et al.’s method [Liu, Chang, Lin et al. (2016)] only modifies the value of VLC in the entropy-coded data segment during histogram modification, while the corresponding RLV has not been

modified, which means that the additional bit information length after VLC does not correspond to the size value of RLV. The decoder may generate exceptions when decoding. Therefore, PSNR is set to 0.

The key point of our proposed method is lossless, but our method also can obtain high payload, which is not inferior to Huang et al.'s RDH method [Huang, Qu, Kim et al. (2016)]. In addition, the filesize expansion of our method is acceptable and less than Liu et al.'s method.

5 Conclusions

In this paper, a lossless and high-payload data hiding scheme for JPEG images is proposed. After introducing the HS-based method in JPEG bitstream, we propose a novel method that optimize histogram modification and modify the DHT segment to preserve the image quality with no distortion. After sorting the histogram and replace the corresponding VLCs, we obtain less filesize expansion. Key contribution is lossless, less filesize expansion in identical payload.

Acknowledgement: This research work is partly supported by National Natural Science Foundation of China (61502009, 61525203, 61472235, U1636206, 61572308), CSC Postdoctoral Project (201706505004), Anhui Provincial Natural Science Foundation (1508085SQF216), Key Program for Excellent Young Talents in Colleges and Universities of Anhui Province (gxyqZD2016011), and Anhui university research and innovation training project for undergraduate students.

References

- Huang, F.; Qu, X.; Kim, H. J.** (2016): Reversible data hiding in JPEG images. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1610-1621.
- Li, X.; Yang, B.; Zeng, T.** (2011): Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 20, no. 12, pp. 3524-3533.
- Li, X.; Zhang, W.; Gui, X.; Yang, B.** (2017): Efficient reversible data hiding based on multiple histograms modification. *IEEE Transactions on Information Forensics & Security*, vol. 10, no. 9, pp. 2016-2027.
- Liu, X. L.; Chang, C. C.; Lin, C. C.; Yuan, S. M.** (2016): A high-payload, reversible data hiding scheme based on histogram modification in JPEG bitstream. *Imaging Science Journal*, vol. 64, no. 7, pp. 364-373.
- Ma, Y.; Luo, X.; Li, X.; Bao, Z.; Zhang, Y.** (2018): Selection of rich model steganalysis features based on decision rough set α -positive region reduction. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Ni, Z.; Shi, Y. Q.; Ansari, N.; Su, W.** (2006): Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362.
- Qian, Z.; Zhang, X.** (2012): Lossless data hiding in JPEG bitstream. *Journal of Systems & Software*, vol. 85, no. 2, pp. 309-313.
- Tian, J.** (2003): Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 13, no. 8, pp. 890-896.

Wang, J.; Li, T.; Shi, Y. Q.; Lian, S.; Ye, J. (2017): Forensics feature analysis in quaternion wavelet domain for distinguishing photographic images and computer graphics. *Multimedia Tools and Applications*, vol. 76, no. 22, pp. 23721-23737.

Wang, J.; Ni, J.; Zhang, X.; Shi, Y. Q. (2017): Rate and distortion optimization for reversible data hiding using multiple histogram shifting. *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 315-326.

Xuan, G.; Shi, Y. Q.; Ni, Z.; Chai, P.; Cui, X. et al. (2007): Reversible data hiding for JPEG images based on histogram pairs. *Image Analysis and Recognition*, pp. 715-727.

Zhang, Y.; Qin, C.; Zhang, W.; Liu, F.; Luo, X. (2018): On the fault-tolerant performance for a class of robust image steganography. *Signal Processing*, vol. 146, pp. 99-111.