An Improved Lossless Data Hiding Scheme in JPEG Bitstream by VLC Mapping

Yang Du[1], Zhaoxia Yin[1,2(✉)], and Xinpeng Zhang[2]

[1] Key Laboratory of Intelligent Computing and Signal Processing,
Ministry of Education, Anhui University, Hefei 230601, People's Republic of China
`yinzhaoxia@ahu.edu.cn`
[2] School of Communication and Information Engineering, Shanghai University,
Shanghai 200072, People's Republic of China
`xzhang@shu.edu.cn`

Abstract. This paper proposed a lossless data hiding scheme by variable length code (VLC) mapping, which focused on embedding additional data into JPEG bitstream. The entropy-coded data in JPEG bitstream consists of a sequence of VLCs and the appended bits. Not all VLCs defined in the JPEG file header are used in the entropy-coded data and the replacement of unused-VLCs do nothing with image decompression. Hence, additional data can be embedded by mapping the unused-VLCs to the used-VLCs. To obtain higher embedding capacity, we improved the mapping rules in this paper. Employing the proposed mapping scheme, larger embedding capacity and no image distortion are both achieved while the filesize of JPEG is preserved after data embedding.

Keywords: Data hiding · JPEG bitstream · VLC · Lossless

1 Introduction

Data hiding is a technique for hiding secret message into digital cover, which developed for the last decades. Data hiding is widely used in military, commercial, medical, financial, etc. People transmit or share multimedia data more and more widely with the rapid development of computer technology. To transmit or share multimedia data quickly in the net, most multimedia data are stored in compressed formats. JPEG is the most popular file format in relation to digital images, which compresses image data effectively. Therefore, the research of data hiding for JPEG images has practical significance.

Traditional data hiding techniques usually change the original data irretrievably, such as modifying the DC components [1], modifying the AC coefficients by histogram shifting [2], etc. However, the distortion due to embedding cannot be accepted in some application areas, like medical, military and law. Hence,

the lossless data hiding techniques are required. As a branch of data hiding techniques, lossless data hiding techniques can embed additional data into cover signal and leave no signal distortion.

Recently some lossless data hiding methods for JPEG images have been proposed. For instance, Fridrich et al. [3] proposed three lossless data embedding methods for embedding data in the quantized DCT coefficients reversibly. Hence, it is possible to reconstruct from the marked JPEG file. Xuan et al. [4] proposed a lossless data hiding method by histogram shifting. The quantized DCT coefficient histogram is shifted to embed high capacity secret data. However, the filesize will increase after data embedding by using the lossless methods introduced above. Then, some lossless and filesize preservation methods for JPEG images appear. A novel lossless method by code mapping is proposed by Mobasseri et al. [5]. The method performs in JPEG bitstream and preserves the image quality and filesize. To improve the embedding capacity, an improved method was proposed by Qian and Zhang [6]. The method improved the code mapping relationships and achieved higher embedding capacity. Recently, Hu et al. [7] improved the mapping relationships further but not optimally.

Similar to the previous methods, we only replace the unused-VLCs for hiding additional data in this paper. Then both the image visual quality and the JPEG filesize are preserved. But the embedding capacity can be improved because of the proposed optimal VLC mapping rules. The rest of this paper is organized as follows. Section 2 briefly introduces the structure of JPEG images and Hu et al.'s method [7]. Section 3 presents the proposed scheme in detail. The experimental results with analysis and comparisons are given in Sect. 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. method, we proposed a lossless and reversible data hiding method. Thus, we introduce the structure of JPEG image at first and Liu et al. method. Since we aim to propose the optimal VLC mapping of JPEG bitstream based on Hu et al.'s method [7] in this paper, we introduce the structure of JPEG bitstream at first in Subsect. 2.1. Because Hu et al. [7] improved the payload based on Qian and Zhang's method [6] Subsect. 2.2 will include the method proposed by Qian and Zhang [6]. Finally, we will introduce Hu et al.'s method in Subsect. 2.3.

2.1   The Structure of JPEG Bitstream

In general, JPEG bitstream 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 the decoding phase, there are two key segments, the Define Huffman Table (DHT) segment and the Start of Scan (SOS) segment. Figure 1 shows the detail structure of the two segments.
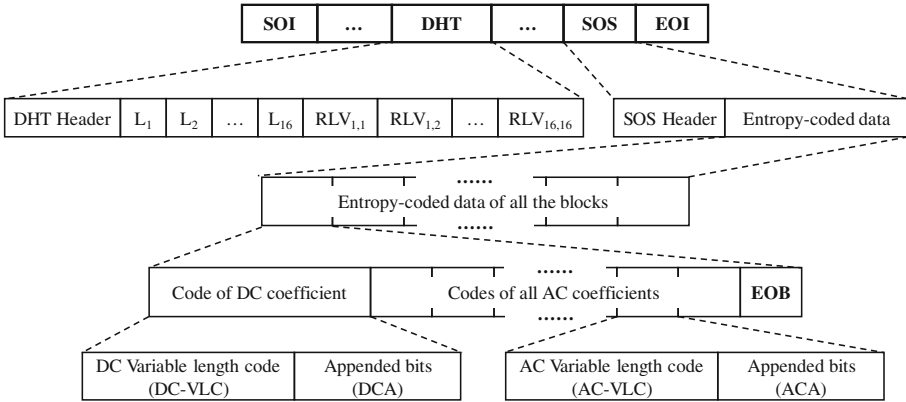
| SOI | ... | DHT | ... | SOS | EOI |

| DHT Header | L_1 | L_2 | ... | L_{16} | $RLV_{1,1}$ | $RLV_{1,2}$ | ... | $RLV_{16,16}$ | | SOS Header | Entropy-coded data |

Entropy-coded data of all the blocks

| Code of DC coefficient | Codes of all AC coefficients | EOB |

| DC Variable length code (DC-VLC) | Appended bits (DCA) | | AC Variable length code (AC-VLC) | Appended bits (ACA) |

Fig. 1. The structure of the JPEG bitstream.

According to the JPEG guideline, after some preprocessing works, the pixel values of each 8×8 block in the image are transformed into AC/DC coefficients by Discrete Cosine Transformation (DCT). For each block's upper left corner first coefficients, define it as DC coefficients. For the remaining 63 coefficients of each block, define them as AC 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 compress rate, (Run, Length) is encoded in the format of Variable Length Code (VLC), and (Amplitude) is encoded in Variable Length Integer (VLI) format, which calls it as appended bits.

The DHT segment contains the Canonical Huffman table information, which is used for obtaining the VLCs. Figure 1 shows the structure of DHT segment, in which L_i equals the number of the same length VLCs and $RLV_{i,j}$ represents the run/length value (RLV) corresponding to the first j VLC of length i. 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. The statistical results indicate that not all of the VLCs appear in the entropy-coded data. As discussed in Sect. 1, many researchers make use of this condition to embed data [6,7].

2.2   Qian and Zhang's Method

Qian and Zhang proposed a lossless data hiding method [6] by Huffman code mapping, which can preserve the modified image with no distortion and provide more embedding capacity than [5]. In addition, their method can preserve the filesize with not changed.
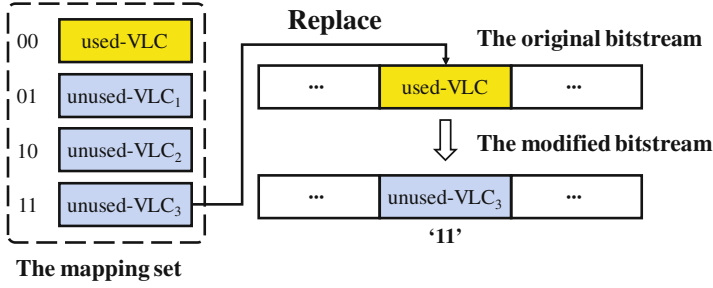
Fig. 2. An embedding instance by VLC mapping.

As mentioned in Sect. 1 and Subsect. 2.1 that not each kind of VLC appears in the entropy-coded data. Qian and Zhang used this condition to establish mapping relationships between used-VLCs and unused-VLCs. Then replace the used-VLC by the any VLC in the mapping sets to embed data. Figure 2 shows an instance about embedding data by VLC mapping. In Fig. 2, the mapping set includes four VLCs, used-VLC, $unused-VLC_1$, $unused-VLC_2$ and $unused-VLC_3$. Therefore, the four VLCs can stands for all the 2^2 situations of two binary bits. Then find the used-VLC in the entropy-coded data and replace it by any of the mapping set. For instance, if we replace the used-VLC by the $unsed-VLC_3$, the data '11' will be embedded.

Their embedding phase can be described as following steps.

Step 1: Parse the JPEG bitstream and extract all the VLCs in the entropy-coded data.

Step 2: Establish the mapping relationships according to the statistical results of the used-VLCs and unused-VLCs.

Step 3: Modify the corresponding run/size value in the DHT segment according to the mapping relationships.

Step 4: Replace the VLCs in the entropy-coded data with the corresponding unused-VLCs to embed data.

To preserve the filesize with not changed, Qian and Zhang only replace the used-VLC by the unused-VLC with same length. Therefore, all the VLCs are classified into 16 categories:

$$\{C_1, C_2, \ldots, C_{16}\}$$

by their code length and category C_i has L_i VLCs of length $i(i = 1, 2, \ldots, 16)$, which can be represented in the following equation:

$$C_i = \{VLC_{i,1}^u, \ldots, VLC_{i,p_i}^u; VLC_{i,1}^n, \ldots, VLC_{i,q_i}^n\} \tag{1}$$

where $VLC_{i,1}^u, \ldots, VLC_{i,p_i}^u$ are used-VLCs, $VLC_{i,1}^n, \ldots, VLC_{i,q_i}^n$ are unused-VLCs, and $p_i + q_i = L_i$.

To obtain more payload, they have proposed two mapping manners:

(1) For the case $p \geq q$ and $q > 0$, VLCs in each category are mapped by one-to-one manner.

$$C_i = \{\{VLC_{i,1}^u \leftrightarrow VLC_{i,1}^n\}, \ldots, \{VLC_{i,q_i}^u \leftrightarrow VLC_{i,q_i}^n\}\} \tag{2}$$

where "$\leftrightarrow$" represents the mapping relationship.

(2) For the cases $q \geq p$ and $p > 0$, VLCs in each category are mapped in the manner of one-to-k. This One-to-k manner can stand for $log_2(k+1)$ binary bits data, where $log_2(k+1)$ is a positive integer.

Qian and Zhang's method is mapping the same number of the unused-VLCs to each used-VLC in each category. The mapping manner could be described as follows:

$$M_i = \left\{ \begin{array}{l} \{VLC_{i,1}^u \leftrightarrow \{VLC_{i,1}^n, \ldots, VLC_{i,k_i}^n\}\}, \ldots, \\ \{VLC_{i,p_i}^u \leftrightarrow \{VLC_{i,(p_i-1) \times k_i-1}^n, \ldots, VLC_{i,p_i \times k_i}^n\}\} \end{array} \right\} \tag{3}$$

where $k_i = 2^{\lfloor log_2(q_i/p_i+1) \rfloor - 1}$ and "$\lfloor x \rfloor$" stands for the floor function.

After establishing the mapping relationships, the additional data can be successfully embedded into JPEG bitstream. However, there are still free space to obtain more payload.

2.3 Hu et al. Method

Hu et al. have discovered the potential free space to improve the payload based on Qian and Zhang's method [6]. They have considered the statistical results of used-VLCs and unused-VLCs and explored to increase the payload. The key contribution of Hu et al.'s method is improving the one-to-k mapping manner by considering the frequency of each used-VLC in the entropy-coded data.

For the VLCs, the largest category is C_16 which has 125 VLCs, and the total number of VLCs is 162. Hence use one-to-k manner can represent 6 ($\lfloor log_2(125 + 1) \rfloor$) data in C_16. In each category C_i, Hu et al.'s method assumed the number of one-to-$(2j-1)$ manner mapping sets is $m_{i,j}$ ($1 \leq j \leq 6$). The selection of $m_{i,j}$ should satisfy the following condition:

$$\mathbf{max} \quad \mathbf{Z} = \sum_{j=1}^{6} j \cdot m_{i,j}$$

$$\mathbf{s.t.} \quad \left\{ \begin{array}{l} \sum_{j=1}^{6} m_{i,j} \leq p_i \\ \sum_{j=1}^{6} (2^j - 1) \cdot m_{i,j} \leq q_i \\ m_{i,j} \geq 0, j = 1, 2, \ldots, 6 \\ m_{i,j} \in N^*, j = 1, 2, \ldots, 6 \end{array} \right. \tag{4}$$

After [4] is solved, all code mapping relationships are established which could be described as:

$$M_i = \begin{cases} \{VLC_{i,1}^u \leftrightarrow \{VLC_{i,1}^n, \ldots, VLC_{i,63}^n\}\}, \ldots, \\ \{VLC_{i,m_{i,6}}^u \leftrightarrow \{VLC_{i,63\cdot(m_{i,6}-1)+1}^n, \ldots, VLC_{i,63\cdot m_{i,6}}^n\}\}, \\ \{VLC_{i,m_{i,6}+1}^{(u} \leftrightarrow \{VLC_{i,63\cdot m_{i,6}+1}^n, \ldots, VLC_{i,63\cdot m_{i,6}+31}^n\}\}, \ldots, \\ \{VLC_{i,\sum_{j=1}^6 m_{i,j}}^u \leftrightarrow VLC_{i,\sum_{j=1}^6[(2^j-1)\cdot m_{i,j}]}^n\} \end{cases} \quad (5)$$

where $VLC_{i,j}^u$ are the used-VLCs after sorted in descend order and $VLC_{i,1}^n, \ldots, VLC^{(n)}{}_{i,j}$ are unused-VLCs.

This means that the first $m_{i,6}$ sorted used VLCs follow the one-to-63 manner, the next $m_{i,5}$ sorted used VLCs follow the one-to-31 manner, ... , and the last $m_{i,1}$ sorted used VLCs follow the one-to-one manner.

Since each used-VLC each time occurs in the entropy-coded segment can be embedded $log_2(k+1)$ bits data, Hu et al.'s method maps more unused-VLCs to the used-VLC which occurs more in the entropy-coded data. By considering each used-VLC's frequency (occurrences number) in the mapping relationships, Hu et al.'s method gained more payload than Qian and Zhang's method. Whereas only consider the first six high frequency VLCs may ignore the potential free space. The statistical results indicate that the frequency of each used-VLC in the same category also vary distinctly. Hence, the payload can be further increased by considering the frequency of each used-VLC. That is the key point of the proposed method that will be described in detail in Sect. 3.

3   Proposed Method

The proposed method can further increase the payload based on Hu et al.'s method [7] by improving the VLCs' mapping relationships. The improved algorithm, the data embedding and data extraction procedures are introduced in the following sections. Figure 3 illustrates the framework of our proposed method. First, parse the JPEG bitstream and extract all the VLCs. Then establish the VLC mapping relationships by our proposed mapping rules. According to the optimal mapping relationships, we can embed more additional data.

3.1   Algorithm of Optimized VLC Mapping

According to the JPEG guideline, there are 162 kinds of VLCs to represent AC coefficients and 12 kinds of VLCs to represent DC coefficients. The statistical results show that the VLCs for DC coefficients are all used in the entropy-coded data, but for AC coefficients, not all of the VLCs are used. Hence, the algorithm is suitable for AC coefficients. The unused-VLCs' corresponding RLVs can be modified to the same value as the used-VLCs' corresponding RLVs, thus the unused-VLCs can be mapped to the used-VLCs. In order to preserve the image filesize, the code length of the used-VLCs and the mapped unused-VLCs must be consistent.
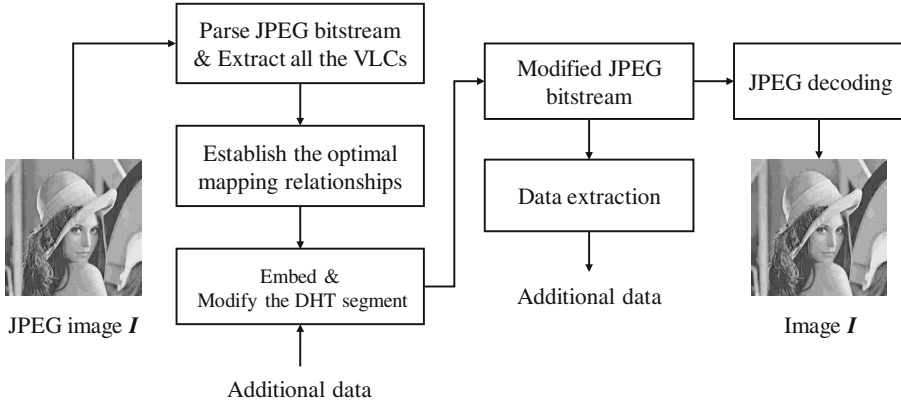
Fig. 3. The framework of proposed method.

The steps of the VLCs' classifying, occurrences recording and VLCs sorting are the same as Hu et al.'s method. After the above steps completed, each category can be represented in the following form:

$$C_i = \{VLC_{i,1}^{(u)'}, \ldots, VLC_{i,p_i}^{(u)'}; VLC_{i,1}^n, \ldots, VLC_{i,q_i}^n\} \tag{6}$$

where $VLC_{i,1}^{(u)'}, \ldots, VLC_{i,p_i}^{(u)'}$ are the used-VLCs after sorted in descend order and $VLC_{i,1}^n, \ldots, VLC^{(n)}{}_{i,q_i}$ are unused-VLCs.

After the above steps are completed, the mapping relationships will be established. Each category's mapping rule depends on p_i and q_i. For the case $p \geq q$ and $q > 0$, C_i employs the one-to-one mapping rule. For the case $q \geq p$ and $p > 0$, C_i employs the one-to-k mapping rule. If both the two cases are not satisfied, C_i does not employ any mapping rule. A detailed description of the different mapping rules is given in the following subsection.

One-to-One Mapping Rule. For the case $p \geq q$ and $q > 0$, VLCs in each category are mapped by one-to-one rule which the same as Hu et al.'s method:

$$C_i = \{\{VLC_{i,1}^{(u)'} \leftrightarrow VLC_{i,1}^n\}, \ldots, \{VLC_{i,q_i}^{(u)'} \leftrightarrow VLC_{i,q_i}^n\}\} \tag{7}$$

where "$\leftrightarrow$" represents the mapping relationship. Figure 4 illustrates the one-to-one mapping rule. As shown in Fig. 4, each unused-VLC is mapped to a different used-VLC.

One-to-k Mapping Rule. For the case $q \geq p$ and $p > 0$, the mapping rule will change to one-to-k manner. Hu et al.'s method assign the corresponding unused-VLCs to used-VLCs according to their sorted order. Both of Qian and Zhang' method [6] and Hu et al.'s method [7] are particular cases. Different from Qian and Zhang's method and Hu et al.'s method, the proposed method takes each
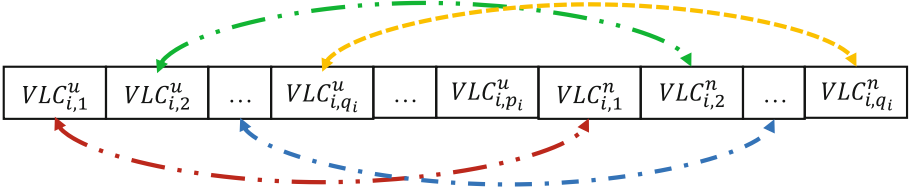
Fig. 4. The diagram of the one-to-one mapping rule.

used-VLC's frequency into consideration. However, how many unused-VLCs to be mapped to each used-VLC is a pure integer nonlinear programming ($PINLP$) problem. Assuming the number of unused-VLCs which mapped to each used-VLC in the category C_i is $x_{i,j}$ and the frequency of each used-VLC is $f_{i,j}$, the selection of $x_{i,j}$ should satisfy the following condition:

$$\mathbf{max} \quad Z_i = \sum_{j=1}^{p_i} f_{i,j} \cdot log_2(x_{i,j} + 1)$$

$$\mathbf{s.t.} \quad \begin{cases} \sum_{j=1}^{p_i} x_{i,j} \leq q_i \\ x_{i,j} \geq 0 \\ x_{i,j} \leq q_i \\ x_{i,j} \in N^* \\ log_2(x_{i,j} + 1) \in N^* \end{cases} \tag{8}$$

where Z_i represents the embedding capacity of the category C_i and $log_2(x_{i,j}+1)$ means $log_2(x_{i,j} + 1)$ additional data can be represented by each VLC belonging to the corresponding mapping set. For instance, if there are 63 unused-VLCs mapped to a used-VLC, 6 additional data can be represented by each of the same used-VLCs. According to Eq. (8), there is a positive correlation between $x_{i,j}$ and $f_{i,j}$. After Eq. (8) is solved, all code mapping relationships are established which can be described as:

$$M_i = \begin{cases} \{VLC_{i,1}^u \leftrightarrow \{VLC_{i,1}^n, \dots, VLC_{i,x_{i,1}}^n\}\}, \\ \{VLC_{i,2}^{(u} \leftrightarrow \{VLC_{i,x_{i,1}+1}^n, \dots, VLC_{i,x_{i,1}+x_{i,2}}^n\}\}, \dots, \\ \{VLC_{i,p_i}^{(u} \leftrightarrow \{VLC_{i,q_i-x_{i,p_i}}^n, \dots, VLC_{i,q_i}^n\}\}, \end{cases} \tag{9}$$

Figure 5 illustrates the one-to-k mapping rule. In Fig. 5, $VLC_{i,1}^u$ is mapped to k_1 different unused-VLCs. Thus the mapping rule for is one-to-k_1. In the same way, the mapping rule for $VLCVLC_{i,k}^u$ is one-to-k_2.
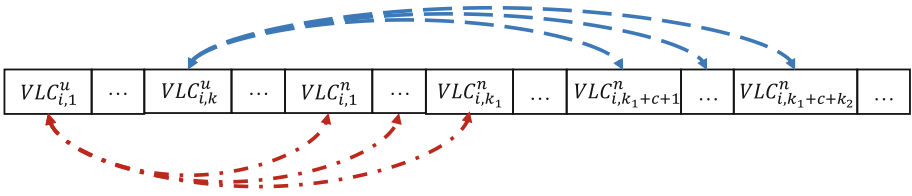
Fig. 5. The diagram of the one-to-k mapping rule.

3.2   Data Embedding and Extraction

The embedding procedure can be summarized as follows:

Input: An original JPEG bitstream and a secret bitstream.

Output: A stego JPEG bitstream.

Step 1: Parse the entropy-coded data, extract all of the used-VLCs and unused-VLCs.

Step 2: Establish the mapping relationships based on the mapping method mentioned in Sect. 3.1.

Step 3: Modify the corresponding RLVs in DHT segment, embed these by replacing the VLCs. Figure 6 shows the process of embedding additional data by replacing VLCs. In Fig. 6, $VLC_{i,m}^{u}$ is mapped to three different unused-VLCs $VLC_{i,k}^{n}, VLC_{i,l}^{n}$, and $VLC_{i,n}^{n}$. After replacing the three unused-VLCs' corresponding RLVs by $VLC_{i,m}^{u}$'s corresponding RLV $RLV_{i,m}^{u}$, these four VLCs can represent the same RLV. We can then embed additional data by replacing the original VLC by the four VLCs. Figure 7 shows one of the groups of additional
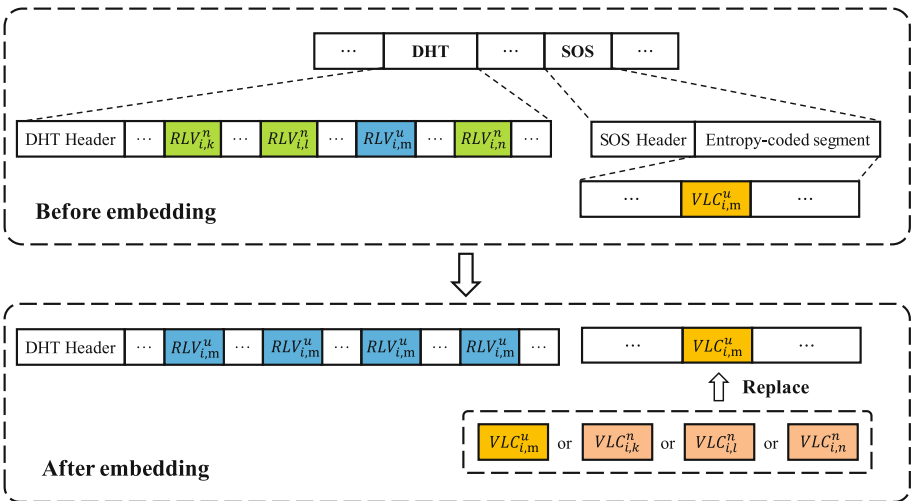

Fig. 6. The diagram of embedding additional data by replacing VLCs.

VLC	RLV		VLC	RLV	Data
$VLC_{i,m}^{u}$	$RLV_{i,m}^{u}$		$VLC_{i,m}^{u}$	$RLV_{i,m}^{u}$	00
$VLC_{i,k}^{n}$	$RLV_{i,k}^{n}$	$\Rightarrow$	$VLC_{i,k}^{n}$	$RLV_{i,m}^{u}$	01
$VLC_{i,l}^{n}$	$RLV_{i,l}^{n}$		$VLC_{i,l}^{n}$	$RLV_{i,m}^{u}$	10
$VLC_{i,n}^{n}$	$RLV_{i,n}^{n}$		$VLC_{i,n}^{n}$	$RLV_{i,m}^{u}$	11

Fig. 7. One of the group of additional data.

data. If we use the group as described in Fig. 7, when we replace the original VLC with $VLC_{i,k}^{n}$, additional data '01' will be embedded in the bitstream.

On the receiver side, the data extraction procedure can be summarized as follows:

Input: A stego JPEG bitstream.
Output: An original JPEG bitstream and a secret bitstream.
Step 1: Read the DHT segment and find the RLVs with same values.
Step 2: Classify the same RLVs into a set and record the RLVs, the corresponding VLCs and each corresponding additional data.
Step 3: Parse the JPEG bitstream and extract the embedded data according to the records from the entropy-coded data.

4 Experimental Results and Analysis

4.1 Experimental Results

To test the performance of our proposed method, firstly we used the images from the USC-SIPI database as mentioned in [8]. Figure 8 shows the ten test

| Baboon | Boat | Bridge | Elaine | F-16 |
| Gray21 | Lena | Peppers | Splash | Tiffany |

Fig. 8. The ten test grey-scale images

Table 1. Embedding capacity comparison with JPEG quality factors from 10 to 90 (bits)

Image	Method	10	20	30	40	50	60	70	80	90
Baboon	Hu et al.	6596	3475	1753	1755	1245	1272	1317	713	585
	Proposed	6596	3475	1754	1789	1271	1305	1358	757	675
Boat	Hu et al.	950	549	681	547	647	738	617	960	1776
	Proposed	950	549	691	571	687	811	679	1000	1929
Bridge	Hu et al.	1802	982	666	829	1007	691	795	569	555
	Proposed	1802	989	713	860	1035	714	817	596	615
Elaine	Hu et al.	341	264	200	414	553	326	407	787	1739
	Proposed	341	264	200	414	553	326	408	788	1745
F16	Hu et al.	791	764	904	863	721	670	527	366	332
	Proposed	791	764	904	961	886	860	741	920	928
Gray21	Hu et al.	235	502	904	982	884	892	1383	1398	2658
	Proposed	235	472	904	982	884	929	1510	1675	3207
Lena	Hu et al.	564	326	262	293	363	195	285	343	562
	Proposed	564	326	262	293	370	197	286	351	583
Peppers	Hu et al.	567	518	679	427	429	550	363	261	1537
	Proposed	567	427	717	492	556	644	450	633	1553
Splash	Hu et al.	496	597	1062	1330	1340	1429	1042	881	456
	Proposed	496	597	1086	1344	1380	1545	1184	1153	719
Tiffany	Hu et al.	4721	1152	656	864	736	786	856	761	625
	Proposed	4721	1152	682	908	782	880	997	956	744

grey-scale images. First, we convert these images to grey-scale images and then compress these to generate the JPEG images with different quality factors.

Since we only replace the used-VLCs by the same length unused-VLCs, the filesize of the modified JPEG bitstream is not changed. Moreover, because the corresponding RSV of the replaced VLC is the same as the original VLC, the decoded results are not changed. Thus, the decoded image has no distortion left. For the reason that our proposed method is lossless with filesize not changed, we only compare the embedding capacity with the other earlier method.

The embedding capacity results of both our proposed method and Hu et al.'s method are shown in Table 1, which the JPEG quality factors are from 10 to 90. From Table 1, we can see the improvement of capacity by our proposed method.

To compare the embedding capacity of these methods further, we has also tested the entire 1338 images from UCID image database [9]. Figure 9 shows the average payload of these 1338 JPEG images with different quality factors by these methods. Table 2 lists the average payload of the three methods. As shown in Fig. 9, the higher the quality factor, the more obvious the improvement of our proposed method. Table 3 shows the specific improvements of our proposed method compared to the other methods. Therefore, to indicate the improvement clearly, we has generated three scatter plots Figs. 10, 11 and 12, which the quality

factors are 70, 80, and 90 respectively. From the three scatter plots, we can see that the improvement is steady and clear.

4.2   Analysis

From Table 3 and Fig. 9, we can see the improvement of our proposed method is much obvious for the high-QF JPEG images. However, from Table 2, the average payload of the high-QF JPEG images is less than the low-QF JPEG images, regardless of obtained by using any of the three methods. The reason can be concluded as the following two points.
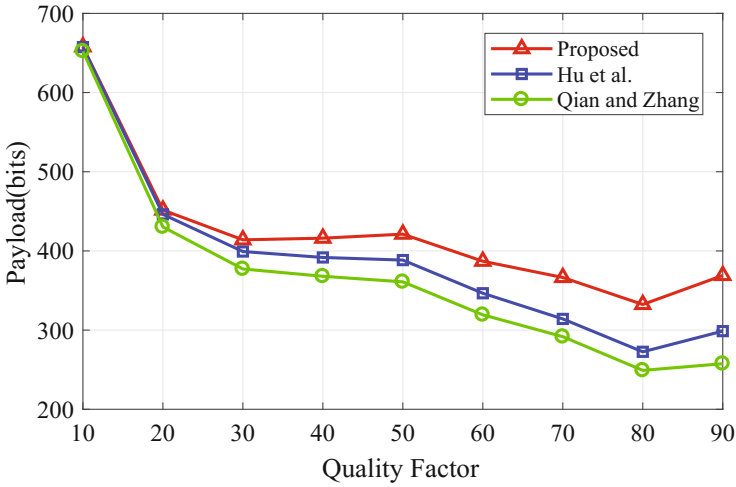

Fig. 9. The comparison of average payload of the JPEG images from UCID.

Table 2. The average payload of the three methods.

Quality factor	Payload(bits)		
	Proposed	Hu et al.	Qian and Zhang
10	1115.667	1114.771	1104.443
20	703.194	692.469	660.613
30	628.033	598.496	554.406
40	632.213	583.499	535.963
50	642.400	576.840	521.901
60	573.980	493.093	438.634
70	532.930	428.234	383.388
80	464.750	345.194	298.259
90	538.026	397.435	315.119

Table 3. The improvements of proposed method compared to the other methods.

Quality factor	Increased than Qian(%)	Increased than Hu(%)
10	1.016	0.080
20	6.446	1.549
30	13.280	4.935
40	17.958	8.349
50	23.088	11.365
60	30.856	16.404
70	39.005	24.448
80	55.821	34.634
90	70.738	35.375
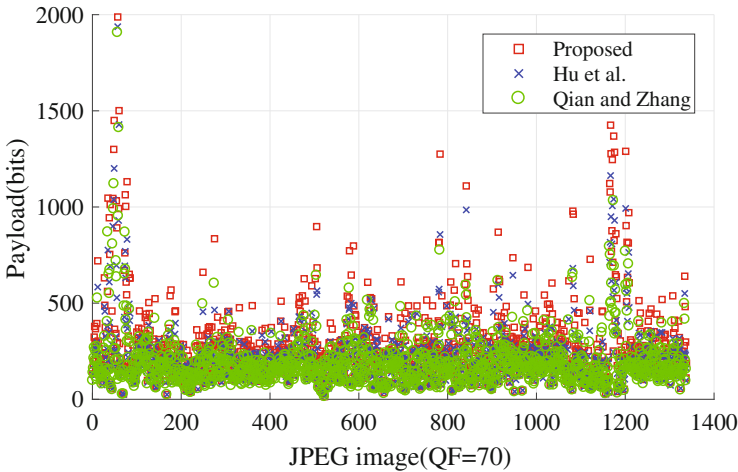
Fig. 10. The payload of each JPEG image with QF = 70 from UCID.

First point: The shorter the length of VLC, the higher the frequency (occurrence number) in the entropy-coded data generally. This is because the VLC is encoded by the format of Canonical Huffman Code.

Second point: The lower the QF, the higher possibility of unused-VLCs with shorter length exists. The QF stands for the compressed degree of a JPEG image. The high-QF means the compressed degree is low. The lower the QF, the more zero-AC coefficients, which means the less kinds of used-VLCs in the entropy-coded data. Correspondingly the possibility of unused-VLCs with shorter length exists is more.

Base the two points we can embed more data into low-QF JPEG images than high-QF JPEG images. Because the payload is determined by both the frequency of used-VLC and the amount of information, which each used-VLC can stands
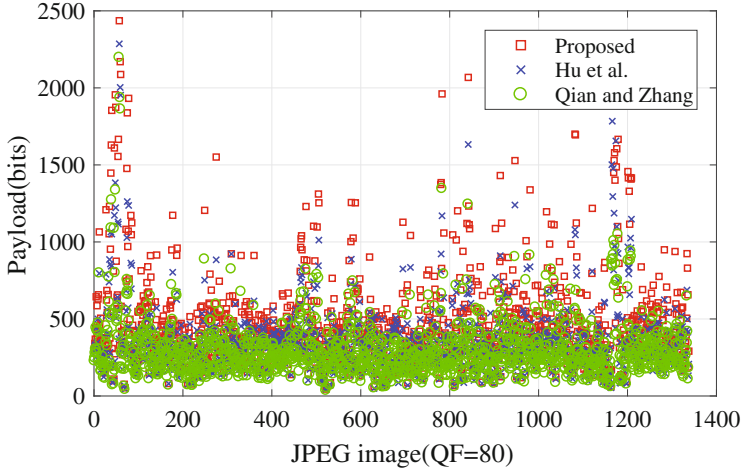
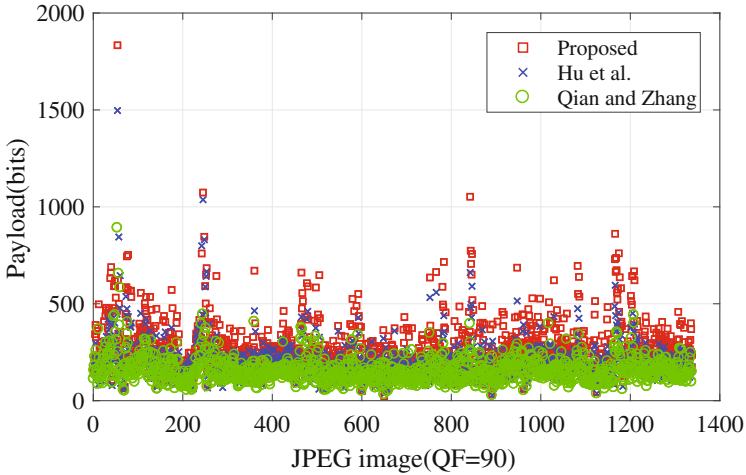**Fig. 11.** The payload of each JPEG image with QF = 80 from UCID.

Fig. 12. The payload of each JPEG image with QF = 90 from UCID.

for. However, the frequency of used-VLC affects more. Because the possibility of the unused-VLC with shorter length exists in low-QF JPEG images is higher, the used-VLC with shorter length is more possible to establish mapping relationship. In addition, the frequency of the used-VLC with shorter length is much high, so we can embed more data than high-QF JPEG images.

However, the high-QF JPEG images are used more widely than low-QF JPEG images, which is because the distortion is less than low-QF JPEG images. Therefore, for most JPEG images in daily life, our method can obtain obvious improvement than the other methods.

5   Conclusions

In this paper, a lossless data hiding scheme for JPEG images is proposed. After introducing the established VLC mapping algorithm for JPEG bitstream, we develop the optimal VLC mapping according to the statistical results of VLCs. Thus, the proposed method obtains more free space to embed additional data.

References

1. Wong, P.H.W., Au, O.C.L., Wong, J.W.C.: Data hiding and watermarking in JPEG-compressed domain by DC coefficient modification. In: Proceedings of SPIE-The International Society for Optical Engineering, pp. 237–244 (2000)
2. Huang, F., Qu, X., Kim, H.J., et al.: Reversible data hiding in JPEG images. IEEE Trans. Circuits Syst. Video Technol. **26**(9), 1610–1621 (2016)
3. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding for all image formats. In: International Society for Optics and Photonics, Electronic Imaging, pp. 572–583 (2002)
4. Xuan, G., Shi, Y.Q., Ni, Z., Chai, P., Cui, X., Tong, X.: Reversible data hiding for JPEG images based on histogram pairs. In: Kamel, M., Campilho, A. (eds.) ICIAR 2007. LNCS, vol. 4633, pp. 715–727. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74260-9_64
5. Mobasseri, B.G., Berger, R.J., Marcinak, M.P., et al.: Data embedding in JPEG bitstream by code mapping. IEEE Trans. Image Process. **19**(4), 958–966 (2010)
6. Qian, Z., Zhang, X.: Lossless data hiding in JPEG bitstream. J. Syst. Softw. **85**(2), 309–313 (2012)
7. Hu, Y., Wang, K., Lu, Z.M.: An improved VLC-based lossless data hiding scheme for JPEG images. J. Syst. Softw. **86**(8), 2166–2173 (2013)
8. The USC-SIPI Image Database. http://sipi.usc.edu/database/
9. The UCID Image Database. http://vision.doc.ntu.ac.uk/