Received: 13 October 2021 | Revised: 28 December 2021 | Accepted: 3 January 2022

IET Image Processing

DOI: 10.1049/ipr2.12426

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Reversible data hiding for JPEG images with a cascaded structure

Wuyue Zhan | **Heng Yao**

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China

Correspondence
Heng Yao, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China.
Email: hyao@usst.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 62172281, 61702332

Abstract

Since Joint Photographic Experts Group is a typical image compression standard, so the reversible data hiding research on Joint Photographic Experts Group has received a lot of attention. There are two main research methods in Joint Photographic Experts Group-reversible data hiding: the method based on modifying DCT coefficients and the method based on variable length coding mapping. The disadvantage of the first one is that the visual loss is always inevitable during the DCT-coefficient modification process, while the second method often has insufficient embedding capacity and excessive file expansion. In order to solve the above problems, this paper proposes a cascaded reversible data hiding scheme, which builds a distortion minimisation scheme based on the DCT coefficients, and then cascades a file expansion suppression scheme based on the variable length coding mapping. For a particular case during the payload distribution, in this paper, all the secret information is embedded into DCT coefficients with a distortion minimisation scheme, and then the auxiliary information generated in the previous process is embedded in the variable length coding sequence with a scheme to optimise the file size expansion. The experimental results demonstrate the superiority of the proposed method in high-quality factor circumstances in terms of visual quality and file expansion.

1 | INTRODUCTION

As an essential branch of image data hiding, reversible data hiding (RDH) is a technology that the secret message data can be imperceptibly embedded into the carrier image. Meanwhile, the carrier image can be restored losslessly after message data extraction. The application of RDH is extensive, especially in the military, medical and other areas that require stego image can be restored losslessly. Numerous RDH methods have been proposed to achieve better performance in recent years. The focus of RDH researches mainly revolves around the following three fundamental strategies: lossless compression [1,2], difference expansion (DE) [3,4], histogram shifting (HS) [5,6], and prediction-error expansion (PEE) [7–9]. In addition, there are some studies focused explicitly on encrypted images [10–14]. However, most of the early developed RDH methods are designed for the uncompressed domain, which is not suitable for the actual situation since the little redundancy of the compressed image makes it difficult to explore the space available for manipulation. It is, therefore, highly desirable to develop RDH methods in the compressed domain.

Joint Photographic Experts Group (JPEG) [15] is one of the most commonly used image file format standards. The wide range of JPEG image applications has attracted many scholars to research RDH in this field. Currently, there are mainly three ways to achieve JPEG RDH: (1) the quantised DCT coefficients modification RDH (DCT-RDH) [16–20], (2) the quantisation table modification [21,22], and (3) the variable length coding (VLC) mapping RDH (VLC–RDH) [23–27]. Generally speaking, directly modifying the quantised DCT coefficients can achieve high embedding capacity but also resulting in noticeable distortion. In 2016, Huang *et al.* [16] proposed a method to modify ±1 alternating current (AC) coefficients in relatively flat image blocks. This method performed well in file expansion suppression and visual quality. In order to quantify the visual distortion caused by the modification of the DCT coefficients, Hou *et al.* [17] deduced that the visual distortion measured by the peak signal-to-noise ratio (PSNR) was only related to the value of the quantisation table corresponding to the selected DCT coefficients. Based on this deduction, an average distortion formula was constructed to determine which frequency bands were suitable for embedding. Motivated

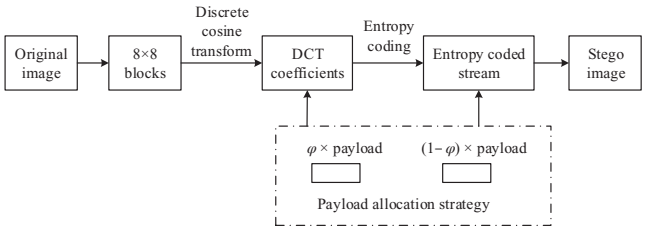


FIGURE 1 Framework of cascaded JPEG RDH method

by [17], He *et al.* [18] established a negative influence model with variable weight coefficient, and this method comprehensively considered the influence of coefficient modification on file expansion and visual quality. Xie et al. [19] proposed to embed secret data into ±1 and ±2 AC coefficients to increase the capacity of JPEG images with low quality factor (QF). It improves the defect of insufficient payload with small QF value, but also generates obvious visual distortion. Recently, Yao et al. [20] proposed a dual-image reversible method for JPEG images based on DCT-coefficient modification. The second type of RDH method is similar to the first one in that the DCT coefficients are modified to carry secret information; however, the second type attempts to make space for embedding by modifying some elements of the quantisation table, which leads to a significant increment of image storage. The third type of JPEG RDH method realised RDH by modifying the entropy-coded stream of JPEG. The advantage is that it can be entirely visually non-destructive; hence, this type of method is also called JPEG lossless data hiding. Specifically, Qian and Zhang [23] proposed a code mapping scheme by replacing the used VLCs with the unused VLCs with the same code length and modifying the file header to record the mapping relationship. Hu et al. [24] and Qiu et al. [25] proposed improved methods to fully use the redundant space of code mapping and offer a larger embedding capacity. Zhang et al. [26] and Yin et al. [27] proposed that the VLC mapping problem can be turned into an optimisation problem with a limited payload. This idea has indeed improved the embedding capacity, and the file expansion is also acceptable.

In recent years, there has been more research on DCT-RDH and VLC-RDH. Both have their own advantages and disadvantages; thus, it would be an interesting study to design a cascaded structure that combines the two types of methods to improve the overall performance. Figure 1 shows a universal framework of cascaded JPEG RDH method, where for a given payload, the adjustment of parameter φ ($\varphi \in [0, 1]$) can control the payload allocation of two types of methods. As can be seen from Figure 1, the selection of DCT–RDH and VLC–RDH embedding schemes and parameter φ will affect the final performance, which mainly refers to the visual quality and file expansion of the image after modification. The user can set reasonable φ according to actual requirements. It should be noted that, first of all, the VLC–RDH method is visually lossless, so the total visual loss only depends on the DCT–RDH scheme, and this loss should be as low as possible. Secondly, the VLC domain usually has a limited embedding capacity; therefore, it is recommended to allocate more loads on the DCT domain. Thirdly, each set of φ requires a complex optimisation process, resulting in different performance results. Therefore, for a real-time practical application, it is not realistic to iterate over and tune φ for optimal performance because modifying the DCT coefficients will cause the subsequent VLC encoding changes; therefore, the image will be unavoidable re-encoded each time when the optimisation operation is performed.

This paper first takes an extreme but reasonable case of cascaded RDH structure to reduce the complexity, that is embed the entire secret information into the DCT coefficients and embed the small amount of auxiliary information generated in the previous process into the entropy-coded stream. Secondly, this paper proposes a distortion minimisation scheme based on the DCT domain, embedding points customisation for each frequency band, and a fast searching method are proposed to find the optimal frequency bands and DCT blocks for embedding. Thirdly, a file expansion suppression scheme based on VLC mapping is cascaded to record auxiliary information generated from the previous process. A method based on Histogram shifting (HS) [5] to reduce coding redundancy and a VLC mapping scheme based on particle swarms optimisation (PSO) [28] is proposed to obtain the optimal VLC mapping relationship. The experimental results show that the proposed method achieved good visual quality on the premise of significantly reducing file expansion. It is worth noting that when the allocation parameter φ of the cascaded framework is set as other values from 0 to 1, a similar RDH operation as described in this paper is still applicable, not tired in words here.

The remainder of this paper is organised as follows. Section 2 introduces the related works about DCT–RDH and VLC–RDH, Section 3 introduces the motivation and implementation process of the proposed RDH method, Section 4 shows the experimental results and presents the detailed analysis, and Section 5 concludes this paper.

2 | RELATED WORKS

This paper propose an RDH framework under a cascaded structure. In order to facilitate the understanding of the proposed method, this section briefly introduce the basic embedding strategies for DCT–RDH and VLC–RDH and their corresponding representative improvement methods, that is [18] and [27], respectively.

2.1 | Introduction of DCT–RDH

2.1.1 | General embedding strategy of DCT–RDH

Divide the original uncompressed image into 8×8 blocks first, and perform discrete cosine transform on each block.

Let $Q_i = \{q_i(u,v)\}(i = 1, 2, \ldots, N, u, v = 0, 1, \ldots, 7)$ represents the quantified DCT coefficients in the ith 8×8 block, where u and v denote the row and column coordinates in each block, respectively, N denotes the total number of blocks. Among all 8×8 image blocks, 63 AC coefficients at different positions form 63 frequency bands. Most DCT–RDH methods apply HS strategy to embed secret information into ± 1 AC coefficients. The basic embedding process of HS strategy is

$$q_i'(u,v) = \begin{cases} q_i(u,v) + \text{sign}(q_i(u,v)) \times b, & \text{if } |q_i(u,v)| = 1, \\ q_i(u,v) + \text{sign}(q_i(u,v)), & \text{if } |q_i(u,v)| > 1, \\ q_i(u,v), & \text{otherwise,} \end{cases} \quad (1)$$

where $(u,v) \neq (0,0)$, $q_i'(u,v)$ is the quantised DCT coefficient after data embedding, $b \in \{0,1\}$ is the to-be-embedded message data bit, and sign$(\cdot)$ is the sign function as

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -1, & \text{if } x < 0. \end{cases} \quad (2)$$

The message data b can be easily extracted, and the original quantised DCT coefficients $q_i(u,v)$ can be restored as follows:

$$b = \begin{cases} 0, & \text{if } |q_i'(u,v)| = 1, \\ 1, & \text{if } |q_i'(u,v)| = 2, \end{cases} \quad (3)$$

$$q_i(u,v) = \begin{cases} q_i'(u,v), & \text{if } |q_i'(u,v)| = 0 \text{ or } 1, \\ q_i'(u,v) - \text{sign}(q_i'(u,v)), & \text{if } |q_i'(u,v)| > 1. \end{cases} \quad (4)$$

2.1.2 | He et al.'s method in DCT domain

Method [18] quantified the file expansion and visual quality of stego image, and constructed a comprehensive influence model with an adjustable preference weight. Users can adjust the parameter to achieve better visual quality or smaller file expansion. Since we only made improvements by referring to its visual distortion model, here only introduce the related part of it.

As deduced in [18], when a quantised AC coefficient at a specific location (u,v) is increased or decreased by one during the data embedding process, the visual distortion measured by PSNR is only related to the corresponding quantisation step $S(u,v)$. Therefore, each frequency band's total distortion $E(u,v)$ can be established as

$$E(u,v) = \frac{\sum_{j=1}^{N} w}{64N} \times S(u,v)^2, \quad (5)$$

where w is defined as

$$w = \begin{cases} \frac{1}{2}, & \text{if } |q_i'(u,v)| = 1, \\ 1, & \text{if } |q_i'(u,v)| > 1, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

It should be noted that $E(u,v)$ includes the distortion caused by the invalid shifting of the AC coefficients whose absolute value is greater than 1, so the concept of average distortion rate $D(u,v)$ is proposed to measure the degree of distortion in each frequency band. Here $D(u,v)$ was obtained by dividing the total distortion $E(u,v)$ by $C(u,v)$, where $C(u,v)$ denotes the amount of ± 1 coefficients at each frequency band corresponding to the location (u,v), that is

$$D(u,v) = \frac{E(u,v)}{C(u,v)} = \frac{S(u,v)^2}{64N}\left(\frac{1}{2} + \frac{\delta_{out}(u,v)}{C(u,v)}\right), \quad (7)$$

where $\delta_{\text{out}}(u,v)$ denotes the total number of quantified coefficients with an absolute value greater than 1 at location (u,v). Under the comprehensive model, the frequency bands with higher evaluation value and blocks with more zero coefficients are finally selected for embedding by traversing all frequency bands and blocks.

However, in [18], the best PSNR value and lowest file expansion are the two extreme cases after weight adjustment, which cannot coexist. This paper attempts to improve the model to better balance these two indicators.

2.2 | Introduction of VLC–RDH

2.2.1 | General embedding strategy of VLC–RDH

In the entropy coding process of JPEG compression, the quantified AC coefficients will be encoded into binary stream using variable length integer (VLI) and VLC technology. Among which, the part of VLC code is used to hide information. In the JPEG guide [15], there are 162 possible VLCs, but not all of them appear in an image. The VLCs that appear in the bitstream are defined as used VLCs, or otherwise are defined as unused VLCs. The example in Figure 2 shows the general fundamental of VLC code mapping scheme. When a mapping group of used VLC v_1 and unused VLC v_2 is created for example, v_1 represents the message data of 0 and v_2 represents 1. Next modify the corresponding symbols of the Huffman table in the file header to mark the mapping relationship of VLCs, that is replace the

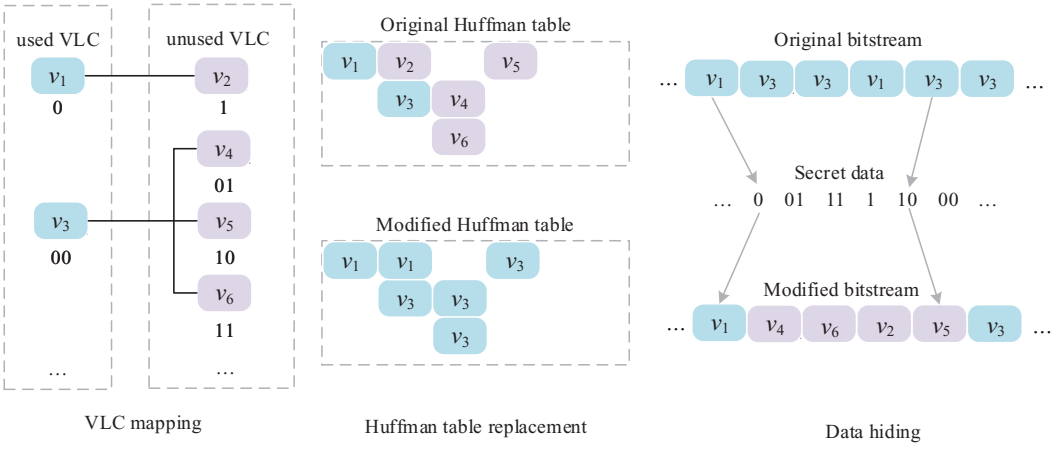


FIGURE 2 An embedding example of VLC code mapping scheme

original symbol of v_2 as v_1. Finally, secret information data can be carried by replacing the mapped VLCs in the bitstream . When there are four VLCs in a mapping group, as shown in the Figure 2, the message data carried by each VLC becomes two bits, and the other embedding processes are the same as above. The number of unused VLCs mapped by one used VLC can only be the format of $2^t - 1, t = 0, 1, \ldots, 7$. Thus the mapping relationship creates a mechanism of secret information representation: each used VLC can represent t-bits binary secret information if the size of the mapping group is 2^t. What is interesting is that the above behaviours can be understood by a universal decoder, and the corresponding modified image can be parsed losslessly.

2.2.2 | Du et al.'s method in VLC domain

Different from the previous methods of seeking the optimal mapping relationship among the same VLC code length, the main innovation of [27] is to establish an optimal mapping relationship among VLCs with different code lengths. The problem is turned to minimising the file expansion under a given payload as follows:

$$\begin{aligned} &\text{Minimise} \quad \textit{File expansion}, \\ &s.t. \quad \textit{Capacity} \geqslant \textit{Payload}, \end{aligned} \tag{8}$$

where *Payload* is a given and fixed value; *File expansion* and *Capacity* are dependent variables only related to the mapping relationship between used and unused VLCs. In order to solve the above optimisation problem, Du *et al.* first applied HS-based embedding method and sorted the frequency set F_{used} in descending order to reduce the coding redundancy, where $F_{\text{used}} = \{f_i | 1 \leq i \leq N_u\}$ denotes the magnitude set of original VLCs, f_i is ith used VLC's number of occurrences, and N_u is the number of used VLCs. Next, four constraint conditions were set to reduce the computational complexity as follows:

(1) The total amount of mapped unused VLCs should be less than or equal to the sum of unused VLCs $N_{\bar{z}}$.
(2) The capacity obtained by selected mapping relationship must be higher than or equal to the given payload.
(3) The number of unused VLCs assigned to ith used VLC should be less than or equal to the one assigned to the previous used VLC.
(4) The number of unused VLCs assigned to each selected used VLC should be selected from the set {0, 1, 3, 7, 15, 31, 63}.

Due to the constraint conditions, only the local optimal solution was obtained. In addition, the specific optimisation algorithm used to solve the problem was not mentioned in [27]. Therefore, a more efficient and specific scheme in the VLC domain will be presented in this paper.

3 | PROPOSED METHOD

The performance indicators of JPEG RDH methods usually include the visual quality and the file expansion size of the stego image. Considering that the visual quality will only be affected in the DCT domain, a minimum distortion optimisation model is first established to embed message data in the DCT domain as Figure 3. Then the auxiliary data generated during the first procedure is then embedded into the entropy-coded stream by a solution that does not generate any visual loss and meanwhile decreases the file size, as shown in Figure 4. Note that the intermediate image and auxiliary data in Figure 3 is the input in Figure 4, so the cascade of the two diagrams constitutes the proposed cascaded framework of DCT–RDH and VLC–RDH.

3.1 | Secret data hiding scheme with minimum distortion

The histogram of quantised AC coefficients of most JPEG images is very steep, coefficients valued 0 occupy the major-

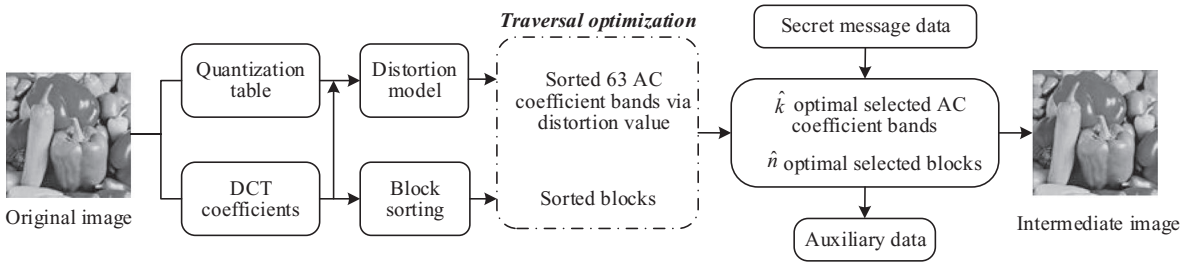


FIGURE 3 Secret data hiding scheme with minimum distortion in DCT coefficients

ity, and other coefficients are approximately symmetrically distributed centred on 0 coefficients. Most traditional DCT–RDH methods embed secret information into ±1 AC coefficients, but the invalid shifting of coefficients whose absolute value is greater than 1 will bring unavoidable losses. This paper proposes a fast customisation algorithm for embedding points in each frequency band and then selects the optimal embedding frequency bands and blocks to minimise visual distortion.

3.1.1 | Embedding points customisation for each frequency band

Unlike the general embedding strategy do as Equations (1)–(4), different embedding points will be customised for 63 frequency bands as a set of $\{a_i, b_i | i = 1, 2, \dots, 63\}$. Considering the symmetry of the coefficient distribution and the significant loss when shifting 0 coefficient, here, we set the range of a_i and b_i to be $\{-R, \dots, -2, -1\}$ and $\{1, 2, \dots, R\}$, respectively, where R is a positive integer to constrain the solution space. Compared with Equation (1), the embedding strategy is modified as

$$q_i'(u,v) = \begin{cases} q_i(u,v) + 1, & \text{if } q_i(u,v) > b_i, \\ q_i(u,v) + b, & \text{if } q_i(u,v) = b_i, \\ q_i(u,v), & \text{if } a_i < q_i(u,v) < b_i, \\ q_i(u,v) - b, & \text{if } q_i(u,v) = a_i, \\ q_i(u,v) - 1, & \text{if } q_i(u,v) < a_i. \end{cases} \tag{9}$$

Secret information can be extracted as $b = 0$, if $q_i'(u,v) = a_i$ or b_i; $b = 1$, if $q_i'(u,v) = a_i - 1$ or $b_i + 1$. The original image can also be restored as

$$q_i(u,v) = \begin{cases} q_i'(u,v), & \text{if } q_i'(u,v) = a_i \text{ or } b_i, \\ q_i'(u,v) - 1, & \text{if } q_i'(u,v) > b_i, \\ q_i'(u,v) + 1, & \text{if } q_i'(u,v) < a_i. \end{cases} \tag{10}$$

The motivation of customising different embedding points for each frequency band is to reduce invalid shifting of AC coefficients, but less invalid shifting does not mean higher visual quality of stego image. A coefficient with larger absolute value may locate at high-frequency region, which will bring greater quantisation distortion. Besides, it may require more frequency bands for embedding to afford enough capacity, so the total distortion is not easy to compare. In order to quantify the visual loss caused by the modification of each frequency band, we refer to [18] distortion model as shown in Equation (7). Here, we redefine the average distortion rate D_{pro} as the proposed evaluation parameter. In addition to the change in the embedding points, the structures of the two average distortion rates are the same.

$$D_{\text{pro}}(u,v) = \frac{S(u,v)^2}{64N}\left(\frac{1}{2} + \frac{\delta'_{\text{out}}(u,v)}{C_{\text{pro}}(u,v)}\right), \tag{11}$$

where $\delta'_{\text{out}}(u,v)$ denotes the total number of coefficients with values greater than b_i or smaller than a_i at location (u,v). $C_{\text{pro}}(u,v)$ denotes the total number of coefficients with value a_i or b_i at location (u,v). For each frequency band, the (a_i, b_i) with smallest D_{pro} value among R^2 combinations will be selected as the customised embedding points.

3.1.2 | Selection of frequency bands and DCT blocks

When the payload is less than the maximum capacity of an image, some rules should be set to select the embeddable region to avoid modifying unnecessary coefficients, so as to improve the performance. Based on the customised embedding points for each frequency band, the frequency bands with lower $D_{\text{pro}}(u,v)$ and DCT blocks with more zero-value AC coefficients should be considered as the candidate region for embedding. On the one hand, frequency bands with a lower $D_{\text{pro}}(u,v)$ usually locate at medium or low frequency region and bring less invalid shifting of coefficients. On the other hand, the DCT blocks with more zero-value AC coefficients usually belong to a smoother region of an image, and the modification on these blocks often leads to lower visual loss.

Specifically, first, arrange the frequency bands in descending order according to the $D_{\text{pro}}(u,v)$, and sort the DCT blocks in descending order of zero-value AC coefficient numbers. Traverse all the first k frequency bands start from the starting band, denoted as B_f, that can afford enough capacity. The simulation

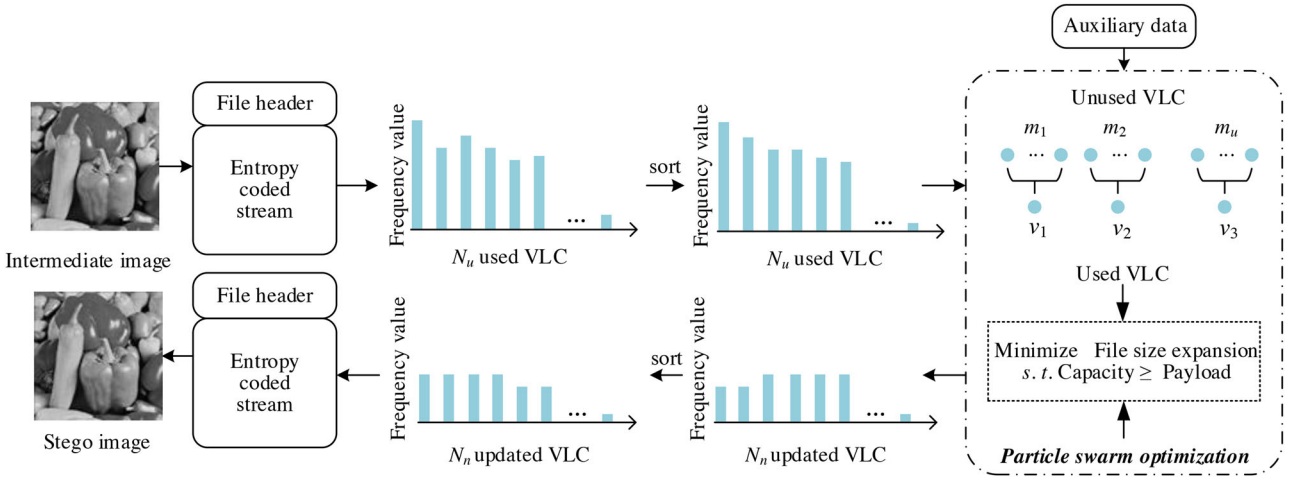


FIGURE 4 Auxiliary data hiding scheme for file expansion optimisation

embedding is performed on the selected frequency bands and corresponding blocks with more zero-value AC coefficients, and the final PSNR values of the obtained modified images are used as the criterion for judging the optimal frequency bands and DCT blocks. Different from [18], we try to change the starting band B_f in the range of $1, 2, \ldots, S_{\text{th}}$ to expand the solution space in a reasonable range, where S_{th} is not recommended to be set too large, and we set it as 5 in our experiments. Furthermore, compared with 63 frequency bands, the number of DCT blocks is very large, calculating the cumulative capacity of each block consumes many times. For an image sized 512×512, it takes 4096 times at most. Therefore, the dichotomy method is used when accumulating the capacity of blocks. The time complexity for searching in [18] is $O(n)$, while the proposed one is $O(\log_2 5n)$.

In the end, the corresponding generated image is referred to as the intermediate image. Note that some auxiliary information is generated during this stage, and it will be embedded in the entropy-coded stream of the intermediate image in the next stage. The specific description of the auxiliary information will be described in Section 3.3.

3.2 | Auxiliary data hiding scheme for file expansion optimisation

As analysed in Section 2.2.1, the RDH schemes in the VLC domain apply the VLC mapping strategy, so the used VLCs will be different after data embedding which causes the file size change. Excessive file expansion leads to increased storage, and the image will be easily perceived as an abnormal signal during transmission, which should be avoided.

In the proposed VLC code mapping scheme, the payload P_a is the length of the auxiliary data generated in the previous section, and the visual quality will not change after modification. Once the embedding capacity C_E that the image can afford is not less than P_a, the only thing we need to consider is the suppression of file expansion E_f after modification:

$$\begin{aligned} &\text{Minimise} \quad E_f, \\ &s.t. \quad C_E \geqslant P_a. \end{aligned} \tag{12}$$

3.2.1 | Structure of file expansion suppression

In order to suppress file expansion, we try to reduce the coding redundancy of VLC codes first. Since most JPEG images use a default Huffman table to encode, there is some coding redundancy, that is the shortest VLC is not always used to represent the highest frequency information. Let $F_{\text{used}} = \{f_i | 1 \leq i \leq N_u\}$ denotes the magnitude set of the original image's used VLCs. We apply the descending operation twice to the frequency histogram as in Figure 5. The first sorting operation is to prioritise the VLCs with higher occurrences for mapping, which will bring less file expansion. The new histogram after the first sorting is denoted as $F'_{\text{used}} = \{f'_i | 1 \leq i \leq N_u\}$. The second sorting ensures that, after embedding, shorter VLCs represent the data with higher occurrences.

The core of the problem described in Equation (13) is to find the appropriate mapping number m_i for each used VLC corresponding to f'_i. Note that m_i here contains the used VLC itself, so it must be the form of $2^t, t = 0, 1, \ldots, 7$. Once m_i is determined, the simulated embedding operation will be performed to F'_{used}. The previously unused VLCs will appear in the intermediate image after simulated embedding, so the number of used VLCs in the intermediate image is updated as N_n, and a new used VLCs' histogram $F''_{\text{used}} = \{f''_i | 1 \leq i \leq N_u\}$ is generated. It defaults that the secret information satisfies identical and independent distribution. Thus each used VLC will be replaced with the mapped VLCs with equal probability during the VLC code replacement process. After simulated embedding, f''_i can be represented by $f''_i = \frac{f'_i}{m_i}$. As mentioned before, after

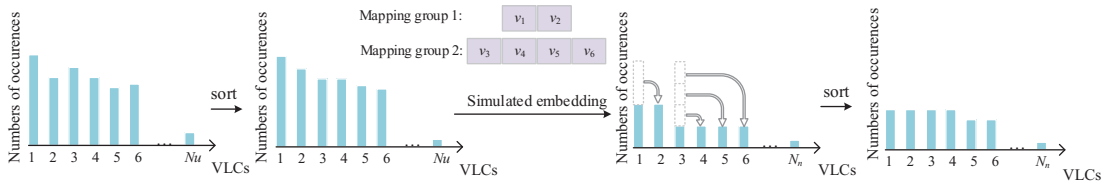


FIGURE 5 An example of HS-based VLC code mapping operation

all the secret information embedding is finished, the descending operation will be applied again. Denote the final histogram as $F^*_{\text{used}} = \{f^*_i | 1 \le i \le N_u\}$, and the final file size change is as follows:

$$E_f = \sum_{i=1}^{N_u} f^*_i \times L_i - \sum_{i=1}^{N_u} f'_i \times L_i, \tag{13}$$

where $L = \{L_i | i = 1, 2, \dots, 162\}$ represents 162 VLCs' code lengths defined by the JPEG standard. When m_i is set as 2^k, each mapped VLC can carry k-bit secret. Thus the corresponding capacity C_E can be described as

$$C_E = \sum_{i=1}^{N_u} f'_i \times \log_2 m_i. \tag{14}$$

Finally, the VLC codes corresponding to F^*_{used} will be parsed into stego-image, and all the auxiliary information data can be embedded into the stego-image under low file expansion.

3.2.2 | The solution details of optimis4ation

The optimal VLC mapping relationship is the solution of Equation (13), which is an NP-hard problem. It takes super polynomial time to seek the optimal mapping value m_i from the N_u used VLCs. In the proposed VLC mapping scheme, we use a PSO algorithm [28] to seek the optimal mapping relationship within a limited number of searches. The implementation details of PSO are listed in Algorithm 1, where individual I_i in the iterative process is a vector connected in series with m_i encoded by three bits of binary code. The output P_{best} is the individual corresponding to the highest PSNR value.

In the implementation of the particle swarm algorithm, the inertia value $w_{\max}$ and $w_{\min}$ are set as 0.9 and 0.4, respectively. The speed value $S_{\max}$ and $S_{\min}$ are 5 and −5, respectively. The learning factor C_1 and C_2 are set to 1.8. Furthermore, the number of iterations G is 200. There are 100 individuals in each population, and each individual is a $3 \times N_u$ vector connected in series with the mapping numbers encoded by three bits of binary code. In each iteration, a velocity vector $S_i = [\, s_1, s_2, \dots, s_{3\times N_u}]$ equals in length to each individual $I_i = [\, i_1, i_2, \dots, i_{3*N_u}]$, S_i is generated as $S_i = w \times S_i + C_1 \times \text{rand} \times (I_{\text{best}} - I_i) + C_2 \times \text{rand} \times (G_{\text{best}} - I_i)$. If $s_i > 0.5$, i_i will be updated as 1, or otherwise be updated as 0.

ALGORITHM 1 Framework of the PSO

Input: Individual size P_{size}; Max generation number G; Inertia weights $w_{\max}$ and $w_{\min}$; Speed $S_{\max}$ and $S_{\min}$; Learning factors C_1 and C_2

Output: Best individual P_{best}

Initialize P_{size} individuals $I = \{I_1, I_2, \dots, I_{P_{\text{size}}}\}$. Initialise the global optimal individual P_{best} and corresponding PSNR value G_{best};
for $i = 1, i \le G, i++$ **do**
  for $j = 1, j \le P_{\text{size}}, j++$ **do**
    // Update jth individual I_j based on $w_{\max}$, $w_{\min}$, $S_{\max}$, $S_{\min}$, C_1, C_2
    for $k = 1, k \le \text{length}(I_j), k++$ **do**
      $w = w_{\max} - (w_{\max} - w_{\min}) \times \frac{i}{G}$;
      $S[k] = w \times S[k] + C_1 \times \text{rand} \times (I_{\text{best}} - I_i) + C_2 \times \text{rand} \times (G_{\text{best}} - I_i)$;
      if $S[k] > S_{\max}$ **then**
        $S[k] = S_{\max}$;
      end if
      if $S[k] < S_{\min}$ **then**
        $S[k] = S_{\min}$;
      end if
      if $S[k] > 0.5$ **then**
        $I_j[k] = 1$;
      else
        $I_j[k] = 0$;
      end if
    end for
  end for
  Find the best individual in I and compare its PSNR with G_{best}, keep the better one as new P_{best};
end for

3.3 | Embedding, Extraction, and Restoration

In this subsection, the specific implementation details of data embedding, data extraction and image recovery are presented as follows.

3.3.1 | Implementation details of data embedding

The process of data embedding includes two parts in the proposed cascaded structure. First, the secret data is embedded into the DCT domain, then the entropy coding is carried out on the

(a) Baboon (b) Sailboat (c) Peppers (d) House (e) Lena

FIGURE 6 Five standard test images

modified DCT coefficients, and then the auxiliary data generated in the previous step is embedded into the entropy-coded stream. The detailed data embedding procedure is listed as follows:

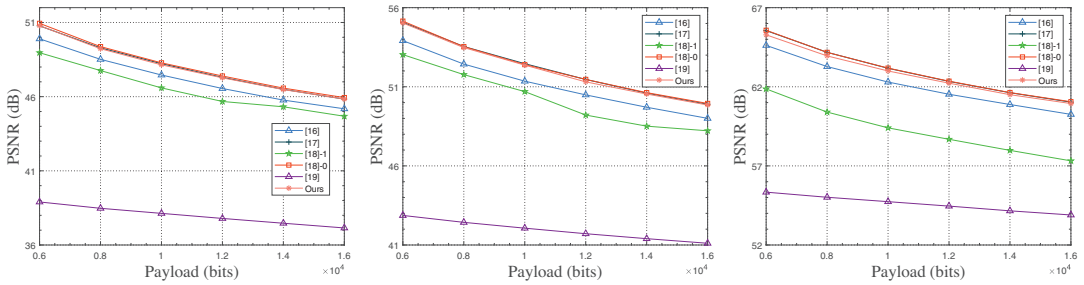
(1) Parse the JPEG image to decode the quantised DCT coefficients, quantisation table, and Huffman table.
(2) Customise embedding points for each AC frequency bands, the pair of embedding points corresponding to the lowest average distortion rate $D_{\text{pro}}(u, v)$ is selected for each frequency band.
(3) Arrange the 63 AC frequency bands in ascending order according to $D_{\text{pro}}(u, v)$.
(4) Calculate the number of zero AC coefficients in each 8×8 block of the cover image, and then sort the blocks in descending order.
(5) Initiate B_f as 1, and implement selection strategy of the optimal bands and blocks:
   - Find the minimum value k, which satisfies that capacity provided by the first k AC frequency bands starting from the B_f frequency band is not less than payload.
   - Find the minimum value n by dichotomy method, such that the capacity provided by the corresponding k frequency bands in the first n sorted blocks is not less than payload. Embed secret data in the corresponding k frequency bands and n blocks according to Equation (10), and record the PSNR of the obtained intermediate image.
   - Let $k = k + 1$ perform the previous two steps until k equals 63. Record the optimal parameters k and n corresponding to the highest PSNR value.
   - Let $B_f = B_f + 1$, perform the previous three steps until B_f equals 5. Denote $\widehat{k}$ and $\widehat{n}$ as the optimal parameters corresponding to the highest PSNR value among all the solutions.
(6) In the selected $\widehat{k}$ blocks and $\widehat{n}$ frequency bands, the message data is sequentially embedded into the ± 1 coefficients in the order of zero-run length from 0 to 62, until it is completely embedded. The zero-run length of the last embedded coefficient is recorded as $R_{\widehat{z}}$.
(7) Encode the DCT coefficients by Huffman table to obtained the intermediate image. The auxiliary information includes 12 bits of the length of auxiliary data, the record of embedding types for each frequency band, map of $\widehat{k}$ selected frequencies, six-bit $R_{\widehat{z}}$, and the length of message data.
(8) Count the VLC histogram of the intermediate image, and find the approximate optimal mapping vector P_{best} according to the PSO algorithm. Encode the binary string of P_{best} into the mapping value m_i for each frequency bands.
(9) Perform VLC code replacement and file header modification, and embed the auxiliary information into the entropy-coded stream.
(10) Merge the new JPEG bitstream using the modified Huffman table.

3.3.2 | Implementation details of data extraction and image recovery

(1) Parse the Huffman table in the file header of the modified JPEG image to obtain the VLC mapping relationship.
(2) According to the mapping relationship of the used and unused VLCs, the auxiliary information in the bitstream can be extracted, and the intermediate image can also be recovered.
(3) Parse the intermediate image to obtain the quantisation table, the quantised DCT coefficients, and then arrange the 8×8 DCT blocks in descending order of the number of zero quantised coefficients.
(4) According to the auxiliary information extracted, calculate the smallest $\widehat{n}$ blocks that provide the required payload in the selected $\widehat{k}$ frequency bands.
(5) According to the extracted $R_{\widehat{z}}$, embedding points for $\widehat{k}$ frequency bands, and the length of payload, the original JPEG image can be recovered, and secret data is extracted in the order of zero run length from 0 to $R_{\widehat{z}}$ in the selected $\widehat{k}$ frequency bands and $\widehat{n}$ blocks according to Equation (11).

4 | EXPERIMENTAL RESULTS

Five 512×512 standard test images, including *Baboon*, *Sailboat*, *Peppers*, *House*, and *Lena* were involved in our experiments, as shown in Figure 6. Compress the five uncompressed test images into JPEG images with QFs of 30, 50, and 90 respectively, and set the corresponding payloads as {6000, 7000, 8000, 9000}, {6000, 8000, 10,000, 12,000}, {6000, 10,000, 14,000, 18,000}, respectively. Furthermore, 50 images randomly selected from the BOSSbase standard image library (http://agents.fel.cvut.cz/boss) were also involved in demonstrating the general applicability of the proposed method.



(a) QF = 30 (b) QF = 50 (c) QF = 90

FIGURE 7 Average PSNR values of 50 images from BOSSbase standard image library

Payloads from 6000 to 16,000 with the step of 2000 are tested on batch-test images with QFs of 30, 50, and 90, respectively. All the JPEG images in the experiment are encoded by default Huffman table.

In order to evaluate the performance of the proposed scheme, five representative algorithms, including [16–19,27], were used for comparative experiments. It should be noted that this paper embeds all the secret information in the DCT domain and embed only a small amount of auxiliary information into the VLC coded stream, so we mainly compare with the RDH methods in the DCT domain here. However, to ensure the comprehensiveness of the comparative experiment, we also added one recent work in VLC domain [27]. The VLC mapping schemes are lossless, the corresponding PSNR value are Inf and structural similarity (SSIM) are all valued 1, so we only compare file expansion with [27].

4.1 | Comparison of visual quality

The PSNR and SSIM were used as indicators to evaluate the visual quality of stego images. The PSNR and SSIM results shown in Tables 1 and 2 are obtained by processing the images under the above setting parameters, where '[18]-0' means the weight α in the scheme of [18] was set to 0, that is, the increment of file size was allowed, so as to obtain the best visual quality. '[18]-1' means that the weight α was set to 1; that is, the visual quality is sacrificed to achieve minimum file expansion. In the proposed embedding point customisation, R is set as 2 in the following experiment. The proposed method performs almost best on PSNR and SSIM when the QF is high, especially when QF is 90. When the QF is 30, [18]-0 has higher PSNR values and slightly better than [17] and ours. That is because our distortion minimisation scheme in the DCT domain customises embedding points for the frequency band; the advantages of the proposed customisation scheme will be more apparent when the image has a high texture.

The average results of PSNR and SSIM compared with other methods are shown in Figures 7 and 8. It can be observed that the average PSNR and SSIM values of stego images in [16] are the lowest under different QF values and payloads. Both [17,18]-0 and the proposed method have similar PSNR and SSIM values, which represent the highest level. The reason for the significant performance gap between [19] and other methods is that [19] embed secret information into ± 1 and ± 2 to enlarge the capacity, which sacrifices the visual quality. Although AC coefficients with an absolute value higher than 1 were also embedded in our method, the distortion caused by different embedding points was accurately quantified, and performance indicators when customising the embedding strategies were better balanced than [19].

4.2 | Comparison of file size

The file expansion of the RDH scheme should not be too significant after data embedding. Table 3 shows the file expansion result of the proposed scheme and other methods. The file size preservation of proposed method is almost the lowest among [16–19], which are all the RDH methods in DCT domain. It shows that the proposed method is very effective in minimising file expansion. When compared with the VLC–RDH method [27], [27] has the lowest file expansion among all the six methods when the QF is low and payload is not large, but the performance of [27] gradually become worse with the payload and QF increase. When the QF is 90 or 50, the proposed method has lower file expansion than [27] on most data. In general, our method has an excellent performance in restraining file expansion, especially when the payload is relatively large and QF is higher than 50. It should be noted that, when [18] set the weight as 0 and get the almost best PSNR performance, the corresponding file expansion is large. The proposed method is very close to the optimal PSNR when the quality factor is low, and gradually grow to the optimum when the quality factor is high. At the same time, the proposed method can also achieve excellent file expansion suppression effects, which indicate that the overall performance of our solution is relatively high.

The comparison of average file expansion obtained by batch processing of 50 images is shown in Figure 9. As the QF increases, the file expansion of the marked images embedded with the same payload of secret data will be larger. Under batch testing, the proposed method is still superior to other four DCT–RDH solutions in terms of file expansion. The proposed method is slightly worse than [27] when QF is 30, but better than [27] when QF is 50 or 90.

TABLE 1 The PSNR values (dB) of the marked images with different QFs while embedded with different lengths of secret data. (Best results are in bold)

		Payload with QF = 30				Payload with QF = 50				Payload with QF = 90			
Image	**Scheme**	**6000**	**7000**	**8000**	**9000**	**6000**	**8000**	**10,000**	**12,000**	**6000**	**10,000**	**14,000**	**18,000**
Baboon	[16]	37.68	36.62	35.69	35.03	41.30	39.23	37.83	36.04	48.08	45.01	42.94	41.17
	[17]	38.05	37.09	36.34	35.45	41.64	39.86	38.27	36.95	48.32	45.25	43.11	41.37
	[18]-0	**38.07**	**37.12**	**36.35**	**35.51**	**41.71**	**39.92**	**38.39**	**37.17**	**49.37**	**46.10**	**43.85**	**42.06**
	[18]-1	32.09	31.64	31.35	30.59	41.07	39.48	37.91	36.12	47.27	39.50	38.07	36.44
	[19]	32.57	32.18	31.77	34.73	36.42	37.89	36.62	35.51	42.11	40.32	39.40	38.39
	ours	37.74	36.78	35.94	35.10	**41.71**	39.77	38.36	37.11	49.34	45.98	43.70	41.99
Sailboat	[16]	37.63	36.64	36.08	35.18	42.13	40.21	38.55	37.44	50.43	46.92	45.20	43.38
	[17]	**38.33**	**37.26**	**36.41**	35.43	43.06	**40.99**	39.12	**37.73**	51.61	47.80	45.52	43.75
	[18]-0	38.23	37.21	36.32	**35.48**	**42.97**	40.92	**39.21**	**37.73**	53.62	50.38	47.98	45.87
	[18]-1	37.92	36.98	34.63	33.98	42.55	38.21	35.94	36.25	52.15	48.29	46.12	43.77
	[19]	32.82	32.45	36.08	33.99	37.58	40.07	38.54	38.08	47.78	44.73	43.70	42.51
	ours	38.10	37.00	36.03	35.10	42.85	40.64	39.00	37.53	**53.64**	**50.44**	**48.07**	**45.87**
Peppers	[16]	37.51	36.71	35.67	34.83	42.70	41.04	39.02	37.53	51.81	48.92	47.20	45.42
	[17]	**38.40**	**37.30**	**36.32**	**35.17**	**43.50**	**41.55**	**39.84**	**38.23**	53.24	50.08	47.79	45.90
	[18]-0	38.23	37.13	36.23	35.02	43.41	41.43	39.74	38.19	54.01	51.31	49.38	47.76
	[18]-1	38.05	36.77	35.86	34.93	42.96	41.09	39.28	37.37	52.96	50.29	48.01	46.21
	[19]	32.68	35.44	35.14	34.41	37.91	42.13	39.44	38.26	47.67	46.12	45.35	44.26
	ours	38.09	36.98	35.97	35.03	43.44	41.43	39.71	38.15	**54.05**	**51.34**	**49.41**	**47.78**
House	[16]	36.35	35.54	34.90	34.11	39.82	38.23	36.88	35.90	52.02	48.98	46.35	43.91
	[17]	37.35	**36.50**	35.74	34.93	41.33	39.47	37.94	36.64	53.14	49.47	46.47	44.31
	[18]-0	**37.42**	36.48	**35.76**	**34.95**	**41.35**	**39.53**	**38.04**	**36.85**	53.44	**50.11**	47.12	44.78
	[18]-1	36.42	35.48	35.27	34.09	38.35	36.84	36.21	35.77	51.04	41.54	40.80	40.18
	[19]	32.25	35.15	33.39	32.85	35.49	36.36	35.40	34.73	48.98	46.70	44.86	43.30
	ours	37.27	36.33	35.56	34.72	41.24	39.41	37.89	36.65	**53.45**	**50.11**	**47.28**	**44.79**
Lena	[16]	37.75	36.91	35.78	34.97	42.28	40.61	38.62	37.32	53.19	50.63	49.03	47.43
	[17]	**38.23**	37.12	**36.22**	**35.16**	42.83	40.94	**39.16**	37.50	54.01	51.23	49.27	47.63
	[18]-0	38.15	**37.15**	36.18	35.12	42.89	**41.04**	**39.16**	**37.63**	54.46	51.85	50.02	48.33
	[18]-1	37.98	36.91	35.67	35.01	42.76	39.85	38.96	37.36	53.68	51.09	49.43	47.83
	[19]	32.73	36.22	35.27	34.63	37.54	41.76	39.50	38.06	49.07	50.20	47.43	46.43
	ours	37.99	36.91	35.88	35.11	**42.96**	41.00	39.01	37.24	**54.51**	**51.90**	**50.08**	**48.34**

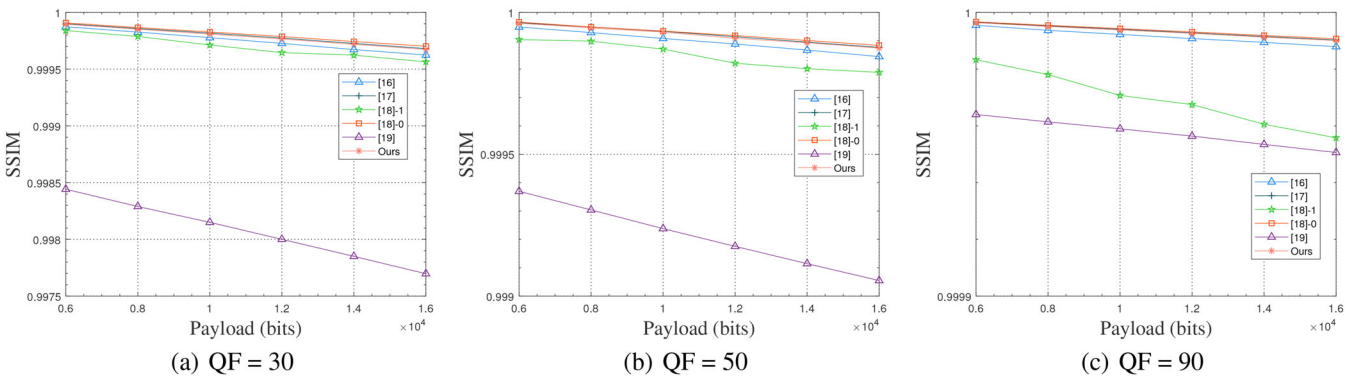
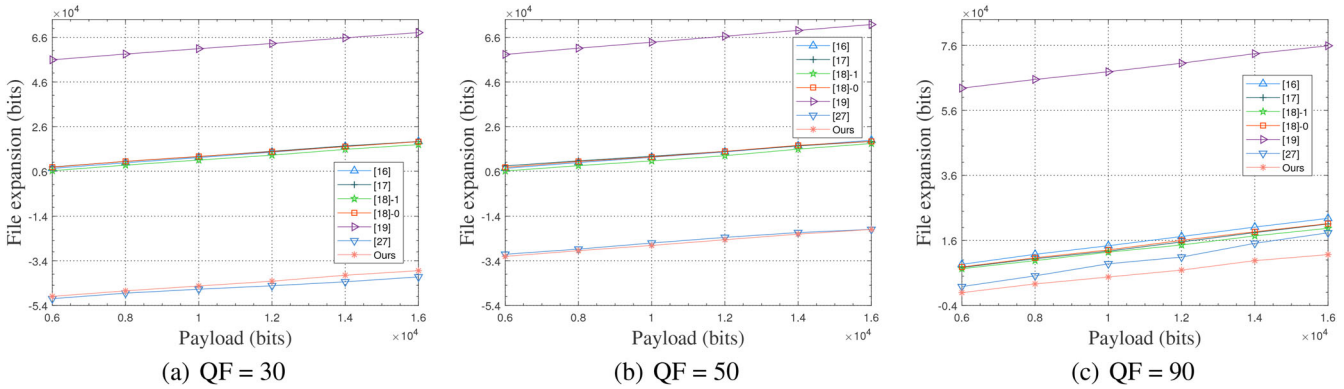


FIGURE 8 Average SSIM values of 50 images from BOSSbase standard image library

TABLE 2 The SSIM values of the marked images with different QFs while embedded with different lengths of secret data. (Best results are in bold)

		Payload with QF = 30				Payload with QF = 50				Payload with QF = 90			
image	**scheme**	**6000**	**7000**	**8000**	**9000**	**6000**	**8000**	**10000**	**12000**	**6000**	**10000**	**14000**	**18000**
Baboon	[16]	0.9970	0.9962	0.9954	0.9946	0.9987	0.9980	0.9972	0.9958	0.9997	0.9995	0.9991	0.9987
	[17]	**0.9973**	0.9966	0.9960	0.9951	**0.9988**	0.9982	**0.9975**	0.9966	0.9997	0.9995	0.9992	0.9988
	[18]-0	**0.9973**	**0.9967**	**0.9960**	**0.9952**	**0.9988**	**0.9983**	**0.9975**	**0.9968**	**0.9998**	**0.9996**	**0.9993**	**0.9989**
	[18]-1	0.9895	0.9884	0.9877	0.9854	0.9987	0.9981	0.9973	0.9959	0.9997	0.9981	0.9974	0.9962
	[19]	0.9906	0.9897	0.9887	0.9942	0.9961	0.9972	0.9963	0.9953	0.9990	0.9984	0.9981	0.9976
	ours	0.9971	0.9964	0.9956	0.9947	**0.9988**	0.9982	**0.9975**	0.9967	**0.9998**	**0.9996**	**0.9993**	**0.9989**
Sailboat	[16]	0.9988	0.9984	0.9982	0.9978	**0.9996**	0.9993	0.9990	0.9987	0.9999	**0.9999**	0.9998	0.9997
	[17]	**0.9989**	**0.9986**	**0.9984**	0.9979	**0.9996**	**0.9994**	**0.9991**	0.9988	**1.0000**	**0.9999**	0.9998	0.9997
	[18]-0	**0.9989**	**0.9986**	0.9983	**0.9980**	**0.9996**	**0.9994**	**0.9991**	0.9988	**1.0000**	**0.9999**	**0.9999**	**0.9998**
	[18]-1	0.9988	**0.9986**	0.9975	0.9971	**0.9996**	0.9989	0.9982	0.9983	**1.0000**	**0.9999**	0.9998	0.9997
	[19]	0.9962	0.9959	0.9982	0.9971	0.9987	0.9993	0.9990	**0.9989**	0.9999	0.9998	0.9997	0.9996
	ours	**0.9989**	**0.9986**	0.9982	0.9978	**0.9996**	**0.9994**	**0.9991**	0.9987	**1.0000**	**0.9999**	**0.9999**	**0.9998**
Peppers	[16]	0.9980	0.9977	0.9970	0.9964	0.9994	0.9991	0.9986	0.9981	0.9999	**0.9999**	0.9998	0.9997
	[17]	**0.9984**	**0.9980**	**0.9974**	**0.9967**	**0.9995**	0.9992	**0.9989**	0.9983	0.9999	**0.9999**	0.9998	0.9997
	[18]-0	0.9983	0.9979	**0.9974**	0.9966	**0.9995**	0.9992	0.9988	0.9983	**1.0000**	**0.9999**	**0.9999**	**0.9998**
	[18]-1	0.9983	0.9977	0.9972	0.9965	0.9994	0.9991	0.9987	0.9980	0.9999	**0.9999**	0.9998	0.9997
	[19]	0.9941	0.9969	0.9966	0.9960	0.9982	**0.9993**	0.9987	**0.9984**	0.9998	0.9997	0.9997	0.9996
	ours	0.9983	0.9978	0.9972	0.9966	**0.9995**	0.9992	0.9988	0.9983	**1.0000**	**0.9999**	**0.9999**	**0.9998**
House	[16]	0.9968	0.9962	0.9956	0.9947	0.9986	0.9979	0.9972	0.9965	**0.9999**	0.9998	**0.9997**	0.9994
	[17]	**0.9975**	**0.9969**	0.9963	**0.9956**	**0.9990**	0.9984	**0.9978**	0.9970	**0.9999**	0.9998	**0.9997**	**0.9995**
	[18]-0	**0.9975**	**0.9969**	**0.9964**	**0.9956**	**0.9990**	**0.9985**	**0.9978**	**0.9972**	**0.9999**	**0.9999**	**0.9997**	**0.9995**
	[18]-1	0.9969	0.9961	0.9959	0.9947	0.9980	0.9971	0.9967	0.9964	**0.9999**	0.9990	0.9988	0.9987
	[19]	0.9919	0.9958	0.9938	0.9929	0.9961	0.9968	0.9960	0.9954	0.9998	0.9997	0.9995	0.9994
	ours	0.9974	0.9968	0.9962	0.9954	**0.9990**	0.9984	**0.9978**	0.9970	**0.9999**	**0.9999**	**0.9997**	**0.9995**
Lena	[16]	**0.9989**	**0.9981**	**0.9973**	**0.9962**	**0.9997**	**0.9992**	**0.9985**	0.9975	0.9999	**0.9999**	0.9998	0.9997
	[17]	0.9980	0.9974	0.9968	0.9959	0.9993	0.9989	0.9984	0.9976	0.9999	**0.9999**	0.9998	**0.9998**
	[18]-0	0.9979	0.9974	0.9968	0.9959	0.9993	0.9989	0.9984	0.9977	**1.0000**	**0.9999**	**0.9999**	**0.9998**
	[18]-1	0.9978	0.9973	0.9964	0.9958	0.9993	0.9986	0.9983	0.9975	0.9999	**0.9999**	0.9998	**0.9998**
	[19]	0.9929	0.9968	0.9960	0.9954	0.9976	0.9991	**0.9985**	**0.9979**	0.9998	**0.9999**	0.9998	0.9997
	ours	0.9979	0.9973	0.9965	0.9959	0.9993	0.9989	0.9983	0.9975	**1.0000**	**0.9999**	**0.9999**	**0.9998**



(a) QF = 30 (b) QF = 50 (c) QF = 90

FIGURE 9 Average file expansion values of 50 images from BOSSbase standard image library

TABLE 3 The file expansion values (bits) of the marked images with different QFs while embedded with different lengths of secret data (Best results are in bold.)

		Payload with QF = 30				Payload with QF = 50				Payload with QF = 90			
image	**scheme**	**6000**	**7000**	**8000**	**9000**	**6000**	**8000**	**10,000**	**12,000**	**6000**	**10,000**	**14,000**	**18,000**
Baboon	[16]	8728	10,120	11,520	13,368	8144	10,824	14096	17,416	9960	16,976	23,856	30,800
	[17]	8608	9992	11,024	12,984	8104	10,712	13,672	16,480	9288	16,528	23,608	29,984
	[18]-0	7616	8944	10,000	11,392	6976	10,000	13,224	16,320	9192	15,968	22,928	29,184
	[18]-1	7136	7792	9632	11,888	6704	9232	12,240	14,992	6672	11,800	17,256	21,032
	[19]	14,112	15,440	16,288	10,432	13648	10,552	13,024	16,608	19,240	27,200	34,592	41,904
	[27]	**−512**	**1232**	**1512**	4056	**1312**	7720	7928	13,544	**−3088**	**5248**	**7792**	19,320
	ours	192	1344	2280	**4008**	1344	**3960**	**6912**	**9448**	−1376	5400	12,560	**16, 688**
Sailboat	[16]	8288	9552	10,816	12,488	8552	11,224	14,328	17,064	8672	15,456	21,688	28,832
	[17]	7800	9272	10,480	12,328	7760	11,032	13,976	16,584	8544	15,032	21,456	28,968
	[18]-0	7496	8592	9888	12,480	7464	9640	13,288	16,040	7680	13,048	18,720	25,680
	[18]-1	7152	8312	9424	11,072	7264	9616	12,256	15,592	7024	12,704	19,296	25,200
	[19]	13,376	14,880	8112	11,288	13,504	9592	11,672	13,864	11,128	17,408	23,720	30,568
	[27]	4984	**5264**	**7024**	**8400**	5976	10,016	11,688	14,648	3960	11,496	18,112	27,496
	ours	**3968**	5336	10,520	10,824	**5656**	**8096**	**10, 888**	**13, 760**	**−448**	**5376**	**11, 816**	**17, 632**
Peppers	[16]	8192	9696	10,984	12,888	8192	11,032	14,440	17,184	8888	15,072	20,616	27,432
	[17]	7768	9504	10,784	12,464	8024	10,672	13,800	16,760	7920	13,728	19,864	26,312
	[18]-0	7856	9432	10,472	12,752	7504	10,296	13,552	16,568	6456	11,736	16,152	22,240
	[18]-1	7440	8968	9896	12,480	7656	9880	12,768	15,592	7056	12,360	18,600	24,624
	[19]	13,920	8752	9568	11,168	13,712	8456	11,224	13,712	13,968	16,264	21,248	27,480
	[27]	**2392**	**3856**	**5384**	**6512**	5520	8384	10,752	13,808	672	11,800	19,520	29,592
	ours	4416	7576	5912	6927	**4288**	**6568**	**10, 296**	**12, 504**	**−2176**	**3080**	**7968**	**13, 288**
House	[16]	9624	10,728	12,016	13,720	9232	11,976	15,120	18,208	8424	14,496	21,600	28,960
	[17]	8792	10,280	11,408	13,064	8792	11,776	14,576	17,592	8512	14,552	21,296	28,720
	[18]-0	8624	9704	10,864	12,232	8112	10,752	13,640	17,256	8024	14,312	20,416	28,168
	[18]-1	8072	9248	10,344	11,720	7592	9512	12,816	16,336	6768	12,640	17,528	23,384
	[19]	15,056	9384	11,272	12,448	15,288	13,056	16,280	19,424	11,352	17,784	24,432	31,712
	[27]	3392	4864	5936	**6664**	**5728**	9216	**11, 680**	14,504	7752	11,400	**16, 448**	26,848
	ours	**2720**	**3976**	**5840**	7376	8816	**8232**	12,680	**13, 864**	**4752**	**10, 184**	16,536	**22, 472**
Lena	[16]	8384	9736	11,048	13,264	8240	11,080	14,280	17,024	8832	13,888	19,680	25,432
	[17]	8016	9704	10,808	12,768	8136	10,648	14,232	16,888	7296	13,632	18,984	24,656
	[18]-0	7864	9200	10,976	12,120	7368	10,192	13,272	16,744	6440	12,168	17,008	22,920
	[18]-1	7504	8680	10,672	11,848	7224	10,048	13,024	16,024	6440	11,368	16,896	21,824
	[19]	14,416	8464	9912	11,552	13,808	7672	11,240	14,152	13,624	11,760	19,368	24,776
	[27]	**2896**	**4032**	**5512**	**6760**	5544	8520	10,912	13,496	8896	16,096	24,216	32,752
	ours	3112	4408	5520	7905	**4376**	**6824**	**10, 016**	**13, 080**	**5256**	**9568**	**14, 968**	**19, 872**

4.3 | Analysis of the file size change in file expansion optimisation scheme

In the file size improvement scheme based on VLC, the auxiliary information includes 12 bits of the binary bits representation of the auxiliary information length which will always be placed at the front of secret data, 63 bits map of selected frequency bands, six bits of the binary representation of $R_{\hat{k}}$, $\log_2 64N$ bits of the binary representation of the secret information length, five bits of the start searching band B_f, and the record of embedding type of the selected frequency bands, which is $\widehat{k} \times c$ bits, c is an integer no less than $\log_2(R^2)$. For example, when R is set to 3, there are nine types of embedding points combination, and c will be the value of 4. Usually, the length of the auxiliary information is small. In the case of low embedding amount, the file expansion optimisation of the five test images in the second

TABLE 4 The file expansion values E_f (bits) after embedded with 87-bits of auxiliary information of the five test images in file expansion optimisation scheme

QF \ Image	Baboon	Sailboat	Peppers	House	Lena
20	−8656	−4312	−5800	−6872	−6432
50	−5072	−880	−2712	−1952	−2680
90	−10, 208	−8408	−9880	−3440	−1472

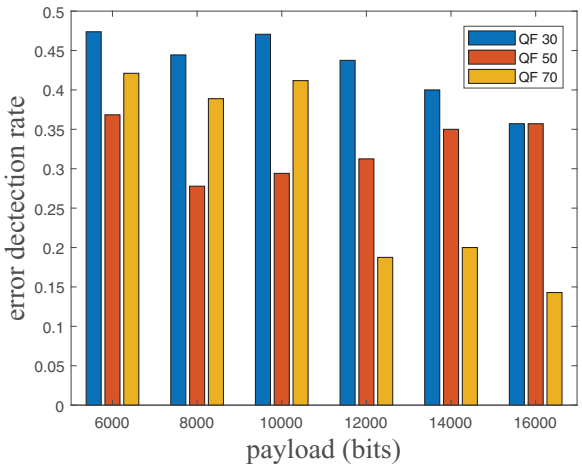


FIGURE 10 Error detection rate under DCTR feature extraction

stage of the cascade scheme is shown in Table 4. As shown in Table 4, where R is set as 2, the length of the auxiliary information corresponding to the five 512×512 test images is $12 + 63 + 6 + 5 + \hat{k} \times 2 + \log_2(64 \times 4096) = 104 + \hat{k} \times 2$. The file expansions after the VLC code mapping operation are all negative values, which greatly optimises the image storage.

4.4 | Security analysis

Security analysis was also briefly evaluated in this paper, where the steganalysis method used in our experiment was DCTR [29]. The features are designed as first-order statistics of quantised noise residuals obtained from the decompressed JPEG image using 64 kernels of the DCT coefficients. We randomly selected 330 images from the BOSSbase library and compressed them into JPEG images with QFs of 30, 50, and 70. Three hundred images were selected and separated into six training sets equally. Different training sets were embedded with from 6000 to 16,000 bits of information via the proposed method. The remaining 30 images were used as the test set. The error detection rate results (i.e. the classification accuracy rate achieved by the SVM classification method) are shown in Figure 10. It can be seen that the error detection rate remained above 0.35 when QF was 30. However, with the increase of QF, the security of the RDH scheme shows a declining trend. It is worth pointing out that as a particular branch of data hiding, the application scenario of RDH is relatively unique in that the information transmission channel is always supposed to be safe enough. Most people pay attention to whether the proposed method can extract secret information errorlessly and restore the original host image losslessly. Therefore, security is not an essential consideration in most approaches.

5 | CONCLUSION

This paper proposes a novel RDH scheme with a cascaded structure, where all secret message data is embedded into the DCT domain through customised embedding point strategy and embedding region selection strategy. The cascaded auxiliary information embedding scheme optimised the code stream structure to a great extent and therefore reduced the file storage. The benefit of the cascaded structure is that it combines the advantages of DCT–RDH and VLC–RDH simultaneously. On the one hand, the DCT–RDH method can achieve a larger embedding capacity; on the other hand, the VLC–RDH method can effectively suppress the file expansion. The experimental results demonstrate the efficacy of the proposed method with respect to stego-image visual quality and file size increment suppression.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (62172281, 61702332). The authors would like to thank anonymous reviewers for their valuable suggestions, which helped to improve this paper.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding for all image formats. Secur. Watermarking Multimed. Content. IV 4675, 572–583 (2002)
2. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless generalized-LSB data embedding. IEEE Trans. Image Process. 12(2), 253–266 (2005)
3. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Technol. 13(8), 890–896 (2003)
4. Alattar, A.: Reversible watermark using the difference expansion of a generalized integer transform. IEEE Trans. Image Process. 13(8), 1147–1156 (2004)
5. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. 16(3), 354–362 (2006)
6. Li, X., Li, B., Yang, B., Zeng, T.: General framework to histogram-shifting-based reversible data hiding. IEEE Trans. Image Process. 22(6), 2181–2191 (2013)
7. Ou, B., Li, X., Zhao, Y., Ni, R., Shi, Y.-Q.: Pairwise prediction-error expansion for efficient reversible data hiding. IEEE Trans. Image Process. 22(12), 5010–5021 (2013)
8. Wang, J., Chen, X., Ni, J., Mao, N., Shi, Y.: Multiple histograms-based reversible data hiding: framework and realization. IEEE Trans. Circuits Syst. Video Technol. 30(8), 2313–2328 (2019)
9. Yao, H., Mao, F., Tang, Z., Qin, C.: High-fidelity dual-image reversible data hiding via prediction-error shift. Signal Process. 170, 107447 (2020)
10. Qin, C., Zhang, W., Cao, F., Zhang, X., Chang, C.-C.: Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection. Signal Process. 153, 109–122 (2018)

11. Qin, C., Qian, X., Hong, W., Zhang, X.: An efficient coding scheme for reversible data hiding in encrypted image with redundancy transfer. Inform. Sci. 487, 176–192 (2019)
12. Yu, M., Liu, Y., Sun, H., Yao, H., Qiao, T.: Adaptive and separable multi-ary reversible data hiding in encryption domain. EURASIP J. Image Video Process. 2020, 16 (2020)
13. Qin, C., Jiang, C., Mo, Q., Yao, H., Chang, C.-C.: Reversible data hiding in encrypted image via secret sharing based on GF (p) and GF (28). IEEE Trans. Circuits Syst. Video Technol. (2021) https://doi.org/10.1109/TCSVT.2021.3091319
14. Qin, C., Zhang, X.: Effective reversible data hiding in encrypted image with privacy protection for image content. J. Visual Commun. Image Represent. 31, 154–164 (2015)
15. ITU: Information technology - digital compression and coding of continuous - tone still images - requirements and guidelines. CCITT Recommendation (1992)
16. Huang, F., Qu, X., Kim, H.J., Huang, J.: Reversible data hiding in JPEG images. IEEE Trans. Circuits Syst. Video Technol. 26(9), 1610–1621 (2016)
17. Hou, D., Wang, H., Zhang, W., Yu, N.: Reversible data hiding in JPEG image based on DCT frequency and block selection. Signal Process. 148, 41–47 (2018)
18. He, J., Chen, J., Tang, S.: Reversible data hiding in JPEG images based on negative influence models. IEEE Trans. Inf. Forensics Secur. 15, 2121–2133 (2020)
19. Xie, X.Z., Lin, C.C., Chang, C.C.: A reversible data hiding scheme for JPEG images by doubling small quantized AC coefficients. Multimed. Tools Appl. 78(9), 11443–11462 (2019)
20. Yao, H., Mao, F., Qin, C., Tang, Z.: Dual-JPEG-image reversible data hiding. Inform. Sci. 563, 130–149 (2021)
21. Chang, C.-C., Lin, C.-C., Tseng, C.-S., Tai, W.-L.: Reversible hiding in dct-based compressed images. Inform. Sci. 177(13), 2768–2786 (2007)
22. 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)
23. Qian, Z., Zhang, X.: Lossless data hiding in JPEG bitstream. J. Syst. Softw. 85(2), 309–313 (2012)
24. Hu, Y., Kan, W., Lu, Z.M.: An improved VLC-based lossless data hiding scheme for JPEG images. J. Syst. Softw. 86(8), 2166–2173 (2013)
25. Qiu, Y., He, H., Qian, Z., Li, S., Zhang, X.: Lossless data hiding in JPEG bitstream using alternative embedding. J. Visual Commun. Image Represent. 52, 86–91 (2018)
26. Zhang, C., Ou, B., Tian, H., Qin, Z.: Reversible data hiding in JPEG bitstream using optimal VLC mapping. J. Visual Commun. Image Represent. 71, 102821 (2020)
27. Du, Y., Yin, Z., Zhang, X.: High capacity lossless data hiding in JPEG bitstream based on general VLC mapping. IEEE Trans. Dependable Secure Comput. 1-1 (2020)
28. Schutte, J.F., Reinbolt, J.A., Regly, B.J.F., Haftka, R.T., George, A.D.: Parallel global optimization with the particle swarm algorithm. Int. J. Numer. Methods Eng. 61(13), 2296–2315 (2004)
29. Holub, V., Fridrich, J.: Low-complexity features for JPEG steganalysis using undecimated DCT. IEEE Trans. Inf. Forensics Secur. 10(2), 219–228 (2015)

How to cite this article: Zhan W, Yao H. Reversible data hiding for JPEG images with a cascaded structure. IET Image Process. 2022;16:1486–1499. https://doi.org/10.1049/ipr2.12426