Multimed Tools Appl (2018) 77:20519–20533
https://doi.org/10.1007/s11042-017-5490-3

Reversible data hiding for VQ indices using hierarchical state codebook mapping

Binbin Xia[1] · **Anhong Wang**[1] · **Chin-Chen Chang**[2] · **Li Liu**[1]

Received: 28 March 2017 / Revised: 26 November 2017 / Accepted: 4 December 2017 /
Published online: 13 December 2017

Abstract In this paper, we present an approach to efficiently hide sensitive data in vector quantization (VQ) indices and reversibly extract sensitive data from encrypted code stream. The approach uses two patterns to compress VQ indices. When an index is equal to its upper neighbor's index or left neighbor's index, it is encoded by the corresponding equivalent index; otherwise, it is encoded by a modified VQ codebook mapping named as hierarchical state codebook mapping (HSCM). In the proposed scheme, the hierarchical state codebook mapping (HSCM) is main coding pattern and it is generated according to the side-match distortion method(SMD). By the above two patterns, the size of original code stream is reduced, and more storage space can be used to embed sensitive data. The experimental results indicated that the proposed scheme can achieve a higher embedding capacity than the previous state-of-the-art VQ-index-based data hiding methods.

Keywords Reversible data hiding · Vector-quantization (VQ) · Hierarchical state codebook mapping (HSCM) · Side-match distortion method(SMD)

✉ Anhong Wang
wah_ty@163.com

✉ Chin-Chen Chang
alan3c@gmail.com

Binbin Xia
237531301@qq.com

[1] College of Electronic Information and Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China

[2] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan

1 Introduction

Nowadays, digital images are used extensively as an important media because they contain visual, direct, and abundant information. Based on this feather, data hiding techniques in images are becoming more and more important in order to guarantee the security and confidentiality of our digital information. Traditionally, DES [6] and RSA [15] have been used to encrypt digital information. However, in recent decades, reversible data hiding techniques have been studied extensively because they allow the original cover data to be recovered without introducing distortion after the sensitive data are extracted, which is essential in special applications, such as medical images and document files.

In general, data hiding techniques can be performed in spatial domain, frequency domain, and compression domain. In spatial domain, the modification of pixel values, as is done by the difference expansion method [17, 20], least significant bit modification [1], and histogram modification techniques [13, 16],is used to embed the secret data. In frequency domain, the frequency coefficients of the cover image are modified in order to embed the secret data. In compression domain, first, the original image compressed by some compression algorithms, such as vector quantization (VQ) [2] and side-match vector quantization (SMVQ) [5], and then, the secret data are embedded into the indices that are generated. The compression-domain data hiding schemes compress redundant information involved in digital images, which creates more space for embedding secret information, giving them greater hiding capability than the other two methods.

Vector quantization (VQ)is a well-known image compression technique, and it has been used extensively in compression-domain data hiding schemes, especially those that are reversible, because the VQ indices can be recovered lossless after the secret data have been extracted. In 1999, Lin and Wang [10]proposed the first data hiding method based on VQ-compressed images, in which, two neighboring codewords were considered as a pair for embedding bits(“0” or “1”). Chang et al. (2004) presented a reversible data hiding method for VQ indices that used the search-order-coding (SOC) method [2]. The SOC method exploits correlation of inter-block in compression domain, with the main idea being to replace the current index with the previous index using a pre-defined indicator. However, the indicator bits used in SOC take more storage space. In 2013, Pan et al. improved Chang et al.’s method [2] to further reduce the bit rate [14] by avoiding the indicator. Chang et al. (2009) proposed a reversible data hiding method based on the joint neighboring coding (JNC) scheme based on VQ compressed images [3], and Wang and Lu proposed a new optional path method with the JNC scheme to improve the embedding capacity [18]. In 2009, Chang et al. proposed a reversible data hiding method for VQ indices based on the locally-adaptive coding scheme (LAS) [4], and Yang and Lin (2010) improved Chang et al.’s scheme by replacing the traditional way of processing VQ indices with fractal Hilbert curve [22]. In 2009, Yang and Lin proposed a reversible data hiding method for VQ indices to sort a VQ codebook by using the referred counts [21]. In 2011, Yang et al. proposed a reversible data hiding method for VQ indices based on modified fast correlation vector quantization (MFCVQ) and the Huffman code [23]. In 2013, Lee et al. presented a reversible data hiding method for VQ indices by using the correlation of neighboring blocks to generate specific sub-codebooks for use in encoding and hiding data simultaneously [9]. Wang et al. proposed a reversible data hiding method for VQ indices by using the adjoining state-codebook mapping (ASCM) technique in 2013 [19]. In 2015, Lin et al. [11] improved Wang et al.’s method by combining the SOC algorithm and ASCM to obtain higher embedding capacity.

In this paper, we propose a reversible data hiding scheme by using so-called hierarchical state codebook mapping (HSCM)to improve further the embedding capacity of VQ-index-based data hiding. In our scheme, an input image is encoded by VQ to generate VQ index table. The VQ index table is divided into two kinds of indices, i.e., seed indices and residual indices. The seed indices located in the first row or first column still use indices generated from VQ codebook; but the residual indices are further recoded by two patterns. If the current index is equal to its neighboring upper index or left index, it is represented by the equivalent index; otherwise, it is encoded by using a hierarchical state codebook mapping (HSCM). By the two patterns, the size of residual indices is reduced so that more storage space can be used to embed sensitive data. Experimental results show that our proposed scheme is better than recent related work, Lin et al.'s scheme [11].

The remainder of this paper is organized as follows. Section 2 introduces the back ground of proposed scheme, including VQ, SMD, and recent related work. Section 3 presents the details of our proposed method. The experimental results of our proposed method are presented in Section 4. Finally, our conclusions of this paper are presented in Section 5.

2 Background

Throughout this article, our work mainly focuses on the VQ-index-based reversible data hiding, so we first review the standard concepts and results related with VQ in Sections 2.1. In addition, HSCM of our proposed scheme is generated by side-match distortion method(SMD). SMD is reviewed in Sections 2.2.Then recent related work, Lin et al.'s scheme [11], will briefly be introduced in Section 2.3.

2.1 Vector quantization (VQ)

VQ [7] can be defined as a mapping from k-dimensional Euclidean space $\boldsymbol{R}^k$ to a limited subset of space $\boldsymbol{C}$, where $\boldsymbol{C} = \{\boldsymbol{c_i} | \boldsymbol{c_i} \in \boldsymbol{R}^k;\ i = 1,\ 2,\ \ldots,\ n\}$ is a VQ codebook consisting of $n$codewordsand trained from the training images by using optimization algorithm, such as the Linde-Buzo-Gray (LBG) algorithm [12].

A complete VQ system has two parts, i.e., an encoder and a decoder, as shown in Fig. 1. At the encoding end, an input image is divided into non-overlapping blocks and the block size is $a \times b$. That means each block contains k ($k = a \times b$) pixels and can be regarded as a k-dimensional vector $\boldsymbol{X}$. Then, in VQ encoder, each vector $\boldsymbol{X}$ looks for its closest codeword $\boldsymbol{c_j}(j = 1, 2, \ldots, n)$ in the VQ codebook C by using a Euclidean distance rule. Then, the chosen index j of $\boldsymbol{c_j}$ is to encode the current block. Mathematically, this process can be expressed as:

$$\boldsymbol{c_j} = argmin\|\boldsymbol{X} - \boldsymbol{c_i}\|^2, i = 1, 2, \ldots, n. \tag{1}$$

At the decoding end, the decoder recovers each image block according to the received index by look up the codebook. In general, the size of a block in VQ is usually 4×4 pixels, and the resulting bit rate of the compressed monochrome images is in the range of 0.4–0.6 bit/pixel according to the size of VQ codebook, corresponding to 128, 256 or 512 codewords.

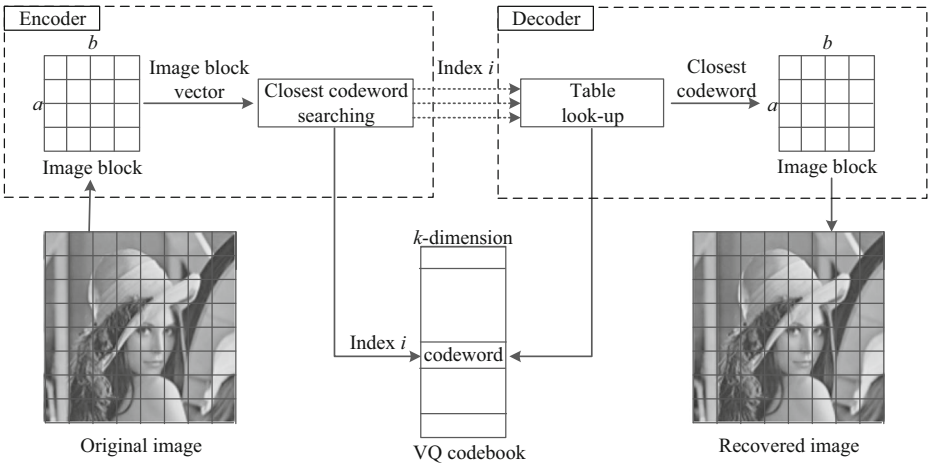

Fig. 1 The VQ encoding and decoding processes

2.2 Side-match distortion method

Side-match distortion (SMD) method was proposed by Kim in 1992 [8] for VQ-compressed images. Because of the correlation between neighboring blocks in images, boundary pixels of current block can be predicted by decoded neighboring blocks. Figure 2 shows the process of SMD with image blocks size $a \times b$, where$Index_U$and$Index_L$are respectivelyupper and left index of $Index_X$. $Block_U$ and $Block_L$ are decoded by VQ decoder. According to the correlation between neighboring blocks in images, the side-match information of $Block_X$ is denoted by $\mathrm{SM} = \left\{x_{11} = \frac{l_{1b}+u_{a1}}{2}, x_{1b(b>1)} = u_{ab}, x_{a1(a>1)} = l_{ab}\right\}$.

The SMD method is used to sort VQ codebook for constructing a state codebook according to the side-match information. The index of this state codebook is actually the code sent by encoder for an image block. Since the length of state codebook is shorter than that of VQ codebook, the size of output index is reduced. Because this side-match information can be obtained by the decoded blocks, it can be obtained at both encoder and decoder without sending extra overhead bits.

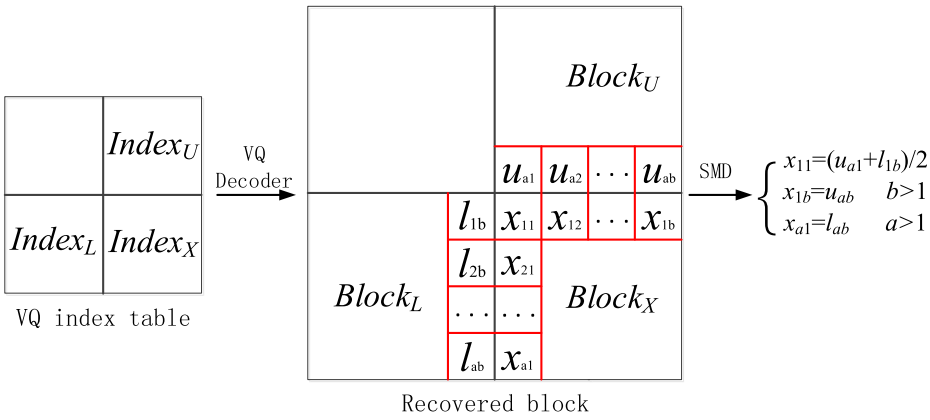

Fig. 2 The process of the side-match distortion method

2.3 Recent related work

In 2015, Lin et al. proposed a reversible data hiding scheme for VQ-compressed images, in which, the size of the VQ index is reduced by selectively using search-order coding (SOC) and state-codebook mapping (SCM) to recode the VQ indices [11]. As a result, more space is freed from the VQ index, and it can be used to improve the hiding capacity for secret data. If one index can be compressed using SOC, the length of the compression code for this index is $(1 + m)$ bits, where one bit represents the indicator, and m bits represent the length of a search-order code. If one index is compressed by SCM, the length of the compression code would be shortened to $(2 + m + \log_2 N_s)$ bits, where two bits represent the indicator, m bits represent the search point(SP) of SOC, and $\log_2 N_s$ bits represent the corresponding index in the state codebook.

The specific steps of Lin et al.'s scheme areas follows.

Input: VQ index table, secret data, VQ codebook, pre-defined value of N_s (length of the state codebook), pre-defined value of w(length of the output index per block), and pre-defined value of m(length of search-order code)
Output: Stego-code stream

Step 1: Input a VQ index for a block as the in-process index scanned by the raster scanning order.
Step 2: Find a search point (SP)that has the same VQ index as the in-process index by the pre-defined search path in the index table; this SP is called a 'matched SP'.
Step 3: If a matched SP is found, SOC is performed to encode the in-process index with the 1-bitindicator "0" followed by the$m$-bit search-order code; then, (w-m-1) secret bits can be embedded.
Step 4: If a matched SP is not found, the in-process index cannot be encoded with any of the 2^m search-order codes, and the following SCM must be performed:

Step 4.1: Use each SP on the search path to generate a corresponding state codebook. In this state codebook, the N_s codewords closest to the current SP are selected from VQ codebook.
Step 4.2: Search all state codebooks generated in Step 4.1 to find a VQ index with the same value as the in-process index.
Step 4.3: If the in-process index is found in the state codebook of the corresponding SP, then it is encoded with the indicator "10," which is followed by m-bit search-order code,$\log_2 N_s$-bitpositionindicatorfor the corresponding state-codebook, and (w-m-$\log_2 N_s$-2) secret bits. Otherwise, the in-process index is preserved, and the indicator "11" is added in front of it.
Step 5: Go to Step 1 and encode the next VQ index until all indices have been processed.

In Lin et al. proposed scheme, the search-order code length (SOL) (m) and the state codebook length (SCL) (N_s) are two crucial and alterable parameters. Thus, an appropriate combination of m and N_s should be considered such that the indices can be greatly compressed and a lower bit rate can be achieved. In the experimental results, for most of the test images, the optimal SOL-SCL pair can be set as (2,8) when the VQ codebook consists of 256 codewords.

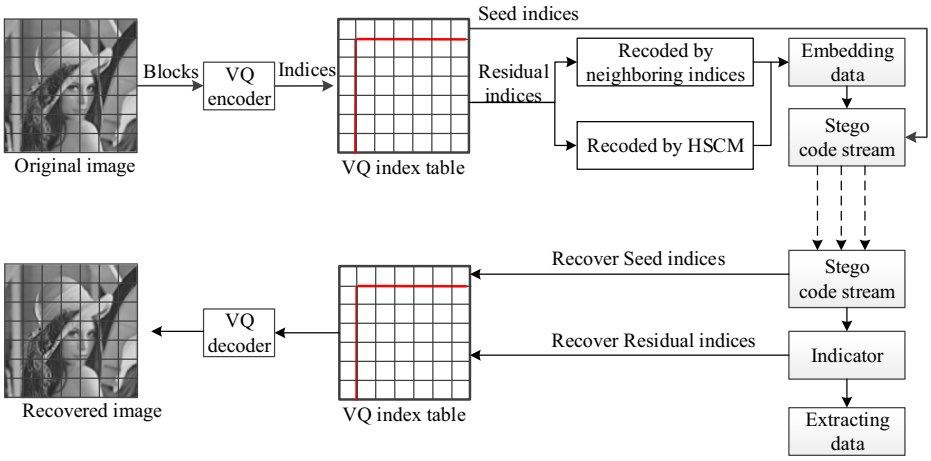

Fig. 3 The complete flowchart of our scheme

3 The proposed scheme

It is evident that there are high correlations between neighboring indices in VQ index table. Based on this method, our goal is to compress the VQ indices further and consequently leave more storage space for data hiding. Figure 3 shows the complete flowchart of our scheme, where seed indices and residual indices are distinguished by red line in VQ index table.

In the proposed scheme, an input image is encoded by VQ to generate VQ index table. The indices of VQ index table are divided into seed indices and residual indices. The seed indices still use its indices from VQ codebook. However, for each residual index, its index is recoded by corresponding neighboring indices or created HSCM. The process of constructing HSCM is described in Section 3.1. The details of encoding and embedding procedures, and decoding and extracting procedures are presented in Sections 3.2 and 3.3, respectively.

3.1 Constructing hierarchical state codebook mapping (HSCM)

In the proposed scheme, the hierarchical state codebook mapping (HSCM) is main pattern to compress residual indices. Figure 4 shows the process of constructing HSCM, in which *ED*

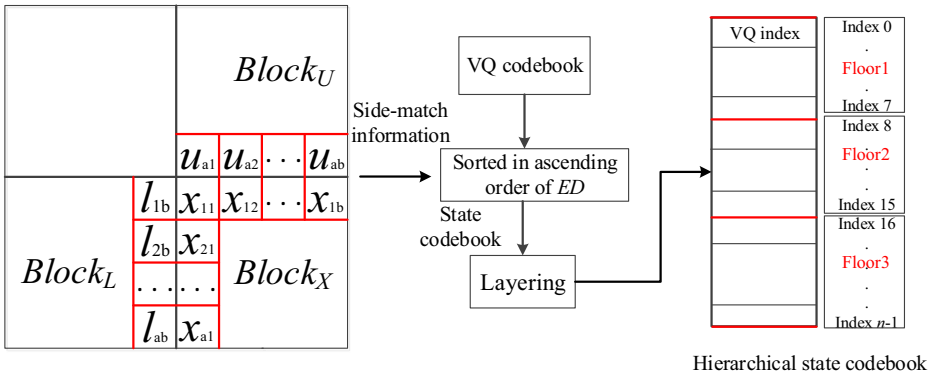

Fig. 4 The process of constructing HSCM.

denotes the Euclidean distance, the size of recovered block is $a \times b$, and n is the size of the VQ codebook. First, the side-match information is extracted according to the model of Fig. 2. Next, the Euclidean distances between the side-match information and each codeword of the VQ codebook are calculated. Then, VQ codebook is stored to form a state codebook in ascending order of Euclidean distance. Finally, the state codebook is layered into three floors to get a hierarchical state codebook.

3.2 Index-encoding and data-embedding phase

In index-encoding and data-embedding phase, seed indices are not changed, while the residual indices are compressed further by using the neighboring upper or left index or HSCM so that more storage space is freed for embedding data. Figure 5 shows the flowchart of index-encoding and data-embedding, and Table 1 shows the encoding rules corresponding to Fig. 5. The detailed index-encoding and data-embedding steps are provided below according to the flowchart and encoding rules.

Input: VQ index table, secret data, pre-defined value of n(the size of VQ codebook), pre-defined value of $\log_2 n$ (the size of original index), and pre-defined value of m(the size of index in the hierarchical state codebook)
Output: Stego code stream CS

Step 1: Input an index X processed in the raster scanning order from the VQ index table. V_X denotes the real value of index X, and V_U and V_L denote the real values of neighboring

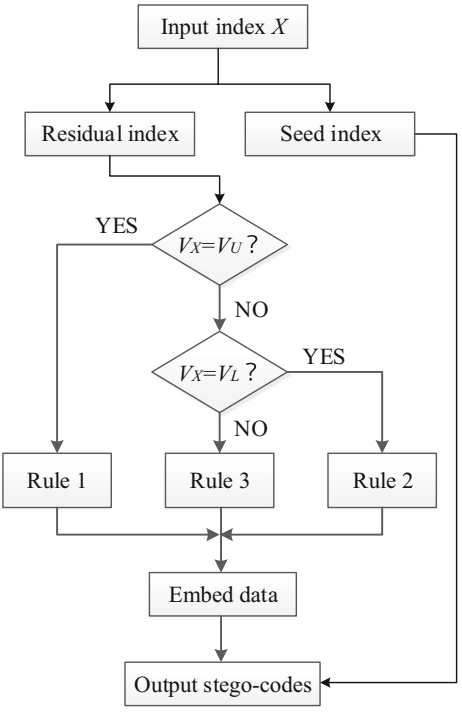

Fig. 5 The flowchart of the index encoding and data embedding

Table 1 The encoding rules corresponding to Fig. 5

Rules	Conditions		Encoding bits
Rule 1	$V_X = V_U$		00
Rule 2	$V_X \neq V_U$and $V_X = V_L$		01
Rule 3	$V_X \neq V_U$ and $V_X \neq V_L$	Floor 1	10 + m-bitindex in hierarchical state codebook
		Floor 2	110 + m-bit index in hierarchical state codebook
		Floor 3	111 + $\log_2 n$-bit original index

upper index U and left index L, respectively. If the input index X is seed index, it is represented by 8-bit index value; otherwise, the input index X is residual index and go to Step 2.
Step 2: If V_X is equal to V_U, the index X is encoded by "00" as indicator, and ($\log_2 n$-2) secret bits can be embedded. If V_X is not equal to V_U, but it is equal to V_L, the index X is encoded by "01" as indicator, and ($\log_2 n$-2) secret bits can be embedded. Otherwise, the following HSCM is used to encode the index X.
Step 3: HSCM.

Step 3.1: For each residual index, the hierarchical state codebook is constructed according to Fig. 4 by using side-match information extracted from the neighboring upper index and left index.
Step 3.2: Find the location of index value V_X in hierarchical state codebook.
Step 3.3: If V_X is located on Floor 1, the indicator bits are "10", the current index X is encoded by the corresponding m-bits index on Floor1 of the hierarchical state codebook, and ($\log_2 n$-2-m) secret bits can be embedded. If the index value X is located on Floor 2, the indicator bits are "110", the index X is encoded by the corresponding m-bits index, and ($\log_2 n$-3-m) secret bits can be embedded. If the index value X is located on Floor 3, the index X is not changed, the indicator bits are "111", and no secret bit is embedded.
Step 4: Repeat Steps1 through 3 until all the residual indices are processed.
Step 5: All seed indices and all processed residual indices are merged to the output watermarked code stream CS.

3.3 Index-decoding and data-extracting phase

In index-decoding and data-extracting phase, the watermarked code stream is sent to the receiver. The receiver can extract the secret data and recover the original VQ index table by using the following algorithm.

Input: Watermarked code stream CS
Output: Original VQ index table, secret data

Step 1: Construct an empty VQ index table. Recover seed indices from CS according to a certain position located in the leftmost and topmost of empty VQ index table.

Fig. 6 Six tested standard gray images

Step 2: Recover residual indices from *CS* according to their position in VQ index table. Read two bits from the remaining *CS*:

Step 2.1: If the two bits are "00", the current index value V_X is equal to the neighboring upper index value V_U. So, the upper index value U is used to recover the current index X, and the following ($\log_2 n$-2)secret bits can be extracted.
Step 2.2: If the two bits are "01", the current index value V_X is equal to the neighboring left index value V_L. So, the left index value L is used to recover the current index X, and the following ($\log_2 n$-2) secret bits can be extracted.
Step 2.3: If the two bits are "10", the current original index X is located on Floor1 of the hierarchical state codebook. Construct the hierarchical state codebook and read the next m bits from the remaining *CS*. The corresponding position' value of the m-bits index on Floor1 is the original index value V_X, and the following ($\log_2 n$-2-m) secret bits can be extracted.

Table 2 The bit rate without any secret bit when the size of VQ codebook is 256 and SOL-SCL pair is (2, 8) in Lin et al.'s scheme

Method	Image					
	Lena	F16	Peppers	Girl	Boat	Baboon
Proposed	0.2852	0.3119	0.2707	0.3089	0.2974	0.4479
Lin et al.'s	0.3165	0.3253	0.3284	0.3662	0.3883	0.4675
Traditional VQ	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000

Step 2.4: If the two bits are "11", construct the hierarchical state codebook and read the next one bit from the remaining *CS*; if the next one bit is "0", the current original index X is located on Floor2 of the hierarchical state codebook. Read the next m bits to recover the original index X(same as Step 2.3) and the following ($\log_2 n$-3-m) secret bits can be extracted; if the next one bit is "1", the original index X is not changed. Read the next $\log_2 n$ bits to recover index X.

Step 3: Repeat Step 2 until all residual indices have been recovered and all secret bits have been extracted.

Step 4: The original VQ index table is recovered and secret data is extracted.

4 Experimental results

In this section, we present the results of the experiments that were conducted to demonstrate the effectiveness and feasibility of our scheme. Figure 6 shows the six 512 × 512 grayscale images that were used as the test images, i.e., 'Lena', 'F16', 'Peppers', 'Girl', 'Boat', and 'Baboon'. The simulation environment was equipped with a 2.66 GHz Intel (R) Core (TM) 2 Quad CPU computer with 4 GB RAM and the MATLAB platform. In order to facilitate compression with recent related work, the common processing parameters are set up. Each image was segmented into non-overlapping blocks sized 4 × 4 and encoded by the VQ algorithm. The codebook used in our experiments is with the size of 256 and trained by the well-known Linde-Buzo-Gray (LBG) algorithm [8]. The secret stream was generated randomly in advance by a random number generator. Taking into account the correlation between image blocks, mapping to the index table, there is also correlation between adjacent indexes, so most of the index is located in the first two layers of hierarchical state codebook. Therefore, the hierarchical state codebook is set up to three floors.

Herein, two criteria were used to evaluate the compression efficiency and embedding efficiency of the proposed scheme, i.e., the bit rate and the embedding rate, which are defined as follows.

$$\text{Bit } rate = \frac{Size\ of\ (code\ stream)}{Size\ of\ (original\ image)}, \tag{2}$$

$$Embedding\ rate = \frac{Size\ of\ (embedded\ secret\ data)}{Size\ of\ (code\ stream)} \times 100\%. \tag{3}$$

Table 2 compares the performance of bit rate without any secret bit. In traditional VQ compression that consists of 256 codewords (i.e., the size of the codebook is 256), each index contains 8 bits in the index table for a 4 × 4 image block, so the bit rate of VQ compression is 0.5 bpp. In our scheme, we counted the number of different indices used, as shown in Table 3.Compared with Lin et al.'s scheme, the state codebook generated in the proposed scheme was particularly layered in order to reduce the length of output code. For most of the residual indices, because of the strong correlations of neighboring indices in VQ index table, they were encoded by a2-bit code or the corresponding indices on Floor1 or Floor2 of the hierarchical state codebook. While only a few residual indices could be found on Floor3 of the hierarchical state codebook. Therefore, the proposed scheme provided a lower bit rate than Lin et al.'s scheme.

Table 3 The number of compression indices using two neighboring indices and HSCM

Image	Different compression index				
	$V_X = V_U$ (00)	$V_X = V_L$ (01)	HSCM		
			Floor 1	Floor 2	Floor 3
Lena	5840	1492	5855	1364	1938
F16	2480	2769	7127	1539	2214
Peppers	5025	2172	6269	1222	1441
Girl	4794	2006	4422	2150	2757
Boat	5086	2601	4194	1428	2820
Baboon	1400	924	4852	2403	6550

Table 4 lists the bit rates of test images generated by using the proposed scheme and the scheme in [10] for different embedding capacities. Note that the bit rate increased as the embedding capacity increased, because larger numbers of embedded secret bits results in adaptively increasing the length of the output encoding block. The comparative results show that our proposed method is better than the scheme in [10] in bit rate when embedded in the same capacity. Table 5 compares the embedding rates of the test images. It is obvious that the proposed scheme provides a much higher embedding rate than the scheme in [10]. Especially for the image 'Boat', with a payload of 20,000, the embedding rate was as high as 20.4%, which was more than 16.4% better than the embedding rate provided by [10]. This benefit is directly related to the fact that our scheme achieves a higher compression of VQ indices.

In order to demonstrate the performance of proposed scheme further, we compared our proposed scheme with other reversible data hiding schemes [3, 19, 22, and]. Table 6 lists the embedding performances off our reversible data hiding schemes. Compared with Chang et al.'s scheme [3], our proposed scheme achieves a higher capacity, a lower bit rate, and a higher embedding rate. Compared with Yang et al.'s scheme [22] and Wang et al.'s scheme [19], our proposed scheme also achieves a lower bit rate and a higher embedding rate when embedded in the same capacity. As a result, our proposed scheme has a higher capacity, a lower bit rate, and a higher embedding rate among the four reversible data hiding schemes.

Table 4 The bit rates of the proposed scheme and the scheme in [10] under different embedding capacity

Capacity	Method	Image					
		Lena	F16	Peppers	Girl	Boat	Baboon
5000	Proposed	0.3043	0.3310	0.2898	0.3279	0.3165	0.4670
	Lin et al.'s	0.3356	0.3444	0.3475	0.3853	0.4073	0.4866
10,000	Proposed	0.3233	0.3500	0.3088	0.3470	0.3355	0.4860
	Lin et al.'s	0.3547	0.3634	0.3665	0.4044	0.4265	0.5056
20,000	Proposed	0.3615	0.3882	0.3661	0.3852	0.3737	0.5242
	Lin et al.'s	0.3928	0.4016	0.4047	0.4425	0.4646	0.5438
40,000	Proposed	0.4378	0.4645	0.4233	0.4615	0.4500	0.6005
	Lin et al.'s	0.4691	0.4779	0.4810	0.5188	0.5409	0.6201

Table 5 The embedding rates of the proposed scheme and the scheme in [10] under different embedding capacity

Capacity	Method	Image					
		Lena	F16	Peppers	Girl	Boat	Baboon
5000	Proposed	6.3%	5.8%	6.6%	5.8%	6.0%	4.1%
	Lin et al.'s	5.7%	5.5%	5.5%	5.0%	4.7%	3.9%
10,000	Proposed	11.8%	10.9%	12.4%	11.0%	11.4%	7.9%
	Lin et al.'s	10.8%	10.5%	10.4%	9.4%	9.0%	7.5%
20,000	Proposed	21.1%	19.7%	22.0%	19.8%	20.4%	14.6%
	Lin et al.'s	19.4%	19.0%	18.9%	17.2%	16.4%	14.0%
40,000	Proposed	34.9%	32.9%	36.1%	33.1%	33.9%	25.4%
	Lin et al.'s	32.5%	31.9%	31.7%	29.4%	28.2%	24.6%

The experimental results discussed above indicate that proposed scheme based on HSCM is more feasible and more effective with respect to compression ability and high payload.

5 Conclusions

In this paper, we proposed a reversible data hiding scheme for VQ indices using hierarchical state codebook mapping. In our scheme, if one of the neighboring upper index or left index values is not equal to the current index value, the hierarchical state codebook mapping is used to recode the current index. Based on our scheme, the data are embedded into the space freed from VQ index. The experimental results indicated that the size of output code stream was greatly reduced after VQ indices were recoded, so data embedding capacity remained high. Comparing the performance of our scheme with that of some previously proposed schemes, we conclude that our scheme achieves the best compression rate and embedding rate.

Table 6 the embedding performance comparisons of four data hiding schemes

Image	Performance	Method			
		Chang et al.'s scheme [3]	Yang et al.'s scheme [22]	Wang et al.'s scheme [19]	Proposed scheme
Lena	Capacity	16,129	16,156	16,156	16,156
	Bit rate	0.5152	0.4310	0.3934	0.3468
	Embedding rate	11.95%	14.32%	15.66%	17.77%
F16	Capacity	16,129	16,161	16,161	16,161
	Bit rate	0.5063	0.4009	0.4175	0.3735
	Embedding rate	12.19%	15.04%	14.76%	16.50%
Peppers	Capacity	16,129	16,147	16,147	16,147
	Bit rate	0.5055	0.4396	0.4292	0.3323
	Embedding rate	12.15%	13.89%	14.36%	18.54%
Baboon	Capacity	16,129	16,141	16,141	16,141
	Bit rate	0.5940	0.6312	0.5262	0.5095
	Embedding rate	10.36%	9.75%	11.71%	12.10%

Acknowledgements This work has been supported in part by National Natural Science Foundation of China (No.61272262 and No.61210006), International Cooperative Program of Shanxi Province (No.2015031003-2),Research Project Supported by Shanxi Scholarship Council of China (2014-056) and Program for New Century Excellent Talent in Universities (NCET-12-1037), Scientific and Technological Innovation Team of Shanxi Province (No. 201705D131025), Collaborative Innovation Center of Internet+3D Printing in Shanxi Province.

References

1. Celik MU, Sharma G, Tekalp AM, Sable E (2005) Lossless generalized-LSB data embedding. IEEE Trans Image Process 14(2):253–266
2. Chang CC, Chen GM, Lin MH (2004) Information hiding based on search-order coding for VQindices. Pattern Recogn Lett 25(11):1253–1261
3. Chang CC, Kieu TD, Wu WC (2009) A lossless data embedding technique by joint neighboringcoding. Pattern Recogn 42(7):1597–1603
4. Chang CC, Kieu TD, Chou YC (2009) Reversible information hiding for VQ indices based onlocally adaptive coding. J Vis Commun Image Represent 20(1):57–64
5. Chang CC, Nguyen TS, Lin CC (2015) A reversible compression code hiding using SOC and SMVQ indices. Inf Sci 300:85–99
6. Diffie W, Hellman ME (1977) Exhaustive cryptanalysis of the NBS data encryptionstandard. Computer 10(6):74–84
7. Gray R (1984) Vector quantization. IEEE ASSP Mag 1(2):4–29
8. Kim T (1992) Side match and overlap match vector quantizers for images. IEEE Trans Image Process 1(4): 170–185
9. Lee JD, Chiou YH, Guo JM (2013) Lossless data hiding for VQ indices based on neighboringcorrelation. Inf Sci 221:419–438
10. Lin YC, Wang CC (1999) Digital images watermarking by vector quantization. Proc Natl Comput Symp 3: 76–87
11. Lin CC, Liu XL, Yuan SM (2015) Reversible data hiding for VQ-compressed images based on search-order coding and state-codebook mapping. Inf Sci 293:314–326
12. Linde Y, Buzo A, Gray RM (1980) An algorithm for vectorquantizer design. IEEE Trans Commun 28(1): 84–95
13. Ni Z, Shi YQ, Ansari N, Su W (2006) Reversible data hiding. IEEE Trans Circuits Syst Video Technol 16(3):354–362
14. Pan ZB, Ma XX, Deng XM, Hu S (2013) Low bit-rate information hiding method based onsearch-order-coding technique. J Syst Softw 86(11):2863–2869
15. Rivest R, Shamir A, Adleman L (1978) A method for obtaining digital signatures andpublic-key cryptosystems. Commun ACM 21(2):120–126
16. Tai WL, Yeh CM, Chang CC (2009) Reversible data hiding based on histogram modification ofpixel differences. IEEE Trans Circuits Syst Video Technol 19(6):906–910
17. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol 13(8):890–896
18. Wang JX, Lu ZM (2009) A path optional lossless data hiding scheme based on VQ jointneighboring coding. Inf Sci 179(19):3332–3348
19. Wang WJ, Huang CT, Liu CM, Su PC, Wang SJ (2013) Data embedding for vector quantization imageprocessing on the basis of adjoining state-codebook mapping. Inf Sci 246:69–82
20. Wu HC, Lee CC, Tsai CS (2009) A high capacity reversible data hiding scheme with edgeprediction and difference expansion. J Syst Softw 82(12):1966–1973
21. Yang CH, Lin YC (2009) Reversible data hiding of a VQ index table based on referred counts. J Vis Commun Image Represent 20(6):399–407
22. Yang CH, Lin YC (2010) Fractal curves to improve the reversible data embedding for VQ-indicesbased on locally adaptive coding. JVis Commun Image Represent 21(4):334–342
23. Yang CH, Wu SC, Huang SC, Lin YK (2011) Huffman-code strategies to improve MFCVQ-basedreversible data hiding for VQ indices. J Syst Softw 84(3):388–396

Binbin Xia He was born in Hebei Province, China, in 1991.He is currently pursuing the M.E. degree in Taiyuan University of Science and Technology. His current research interests include data hiding, and image processing.

Anhong Wang She received B.E and M.E. degrees from Taiyuan University of Science and Technology (TYUST) respectively in 1994 and 2002, and Ph. D degree in Institute of Information Science, Beijing Jiaotong University in 2009. She became an associate professor with TYUST in 2005 and became a professor in 2009. She is now the director of Institute of Digital Media and Communication, Taiyuan University of Science and Technology. Her research interest includes image/video coding, compressed sensing, and secret image sharing.

Chin-Chen Chang received his B.E. degree in applied mathematics in 1977 and the M.E. degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph. D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. Since February 2005, he has been a Chair Professor of Feng Chia University. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression and data structures.

Li Liu received her B.E. degree in communication engineering in 2002, from Lanzhou Railway University and M.E. degree in communication and information system in 2006, from Lanzhou Jiaotong University. Now, she is a Ph. D student in Northwestern Polytechnical University. Her current research interests include information hiding and secret sharing.