

# Large capacity and high quality reversible data hiding method based on enhanced side match vector quantization

Xiaoxiao Ma · Zhibin Pan · Sen Hu · Lingfei Wang

Received: 20 January 2014 / Revised: 30 July 2014 / Accepted: 8 September 2014 /  
Published online: 19 November 2014  
© Springer Science+Business Media New York 2014

**Abstract** In recent years, many reversible data hiding techniques have been proposed to solve information security problem. In this paper, we propose a novel reversible data hiding method based on enhanced side-match vector quantization (ESMVQ). We embed as many secret bits into the images compressed by ESMVQ which uses the very small state codebook size of 4 to achieve a large hiding capacity and meanwhile ensure the compressed image quality by introducing a complementary state codebook. The experimental results demonstrate that the proposed method has a large embedding capacity and a good image quality. The experimental results also show that our proposed method outperforms the methods of Chang et al. (IEEE Trans on Circ and Syst for Vi Techno 16(10):1301–1308, 2006), Huang et al. (The Image Sci J 61(2):195–203, 2013), Lee et al. (IEEE Trans on Inf Forensic and Secur 5(4):638–648, 2010) and Wang et al. (Inf Sci 246:69–82, 2013).

**Keyword** Data hiding · Hiding capacity · Image quality · Side-match vector quantization (SMVQ) · Enhanced side-match vector quantization (ESMVQ)

## 1 Introduction

With the rapid development of the Internet, there are more and more users to communicate and exchange information via the Internet. As the Internet is primarily a huge open channel, the security problems like interception and modification have become more and more serious. One of the possible solutions is to use encryption techniques which transform secret data into an unrecognizable form and only the legal users can decrypt them with the secret key. However, the unreadable form may arouse much more attentions of illegal users to decrypt the messages and even destroy them. Another of the possible solutions is to use data hiding techniques

---

X. Ma · Z. Pan (✉) · S. Hu · L. Wang  
School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049,  
People's Republic of China  
e-mail: zbpan@mail.xjtu.edu.cn

Z. Pan  
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093,  
People's Republic of China

which embed secret data into a cover media such as image, video or audio to avoid the attention of the illegal users [15]. Data hiding techniques can be used to protect the authentication and ownership of multimedia data, as well as a mean of secret communication for people [15].

Data hiding techniques for image can be roughly divided into three domains: the spatial domain methods [1, 2, 13, 16], the transform domain methods [10] and the compression domain methods [3–5, 8, 11, 14, 17, 19–21]. Data hiding techniques in the spatial domain methods embedded the secret data by modifying the spatial information such as the least-significant-bit (LSB) methods [1, 2] and the histogram modification [13, 16]. Data hiding techniques in the transform domain methods embedded the secret data by shifting the coefficients of the cover image [10]. However, both the spatial domain methods and transform domain methods all need the larger bandwidth and the storage space. In order to save bandwidth or storage space for data transmission or storage, many data hiding methods in compression domain have been proposed [20]. In the compression domain, the secret data is embedded into the compressed-cover images compressed by the compression techniques such as JPEG, vector quantization (VQ) [7] or side-match vector quantization (SMVQ) [9]. As VQ and SMVQ are high-efficient compression techniques, they have been widely used in the different applications such as data hiding techniques [20] and video signals [18]. There are many reversible data hiding algorithms based on VQ and SMVQ proposed in recent years [3–5, 8, 11, 14, 17, 19, 21]. In 2013, Chang et al. proposed a reversible data hiding method based on SMVQ and search order coding (SOC) which embedded 1-bit secret data into each SMVQ index [4]. Recently Pan et al. proposed a reversible data hiding method for VQ indices based on SOC technique to achieve a low bit rate [14]. However, these two methods all had a low hiding capacity. In 2009, Chang et al. proposed a reversible data hiding method for VQ compressed images based on locally adaptive coding method [3]. In 2009, Tsai proposed histogram-based reversible data hiding method for VQ compressed images [17]. Wang and Lu proposed a path optional reversible data hiding method based on the joint neighboring coding (JNC) of the VQ index table in 2009 [21]. In 2006, Chang et al. proposed a data hiding method based on SMVQ coding which embeds 1-bit secret data into SMVQ indices [5]. To improve the image quality and the hiding capacity of Chang et al.'s method [5] in 2006, Huang et al. proposed a new data hiding methods based on SMVQ recently [8]. However, these two methods in Chang et al. [5] and Huang et al. [8] all had a low hiding capacity and the image quality is not very good. In 2010, Lee et al. proposed a reversible data hiding method which embedded the secret data into the SMVQ-index by modifying the SMVQ indices [11]. In 2013, Wang et al. proposed a variant of SMVQ, called adjoining state-codebook mapping (ASCM), to reduce the bit rate and provide a large hiding capacity for secret data [19].

In this paper, we propose a novel reversible data hiding method based on enhanced side-match vector quantization (ESMVQ) method which can realize a large hiding capacity and high image quality. ESMVQ compression method, combining the advantage of VQ and SMVQ, uses a very small state codebook size of 4 to achieve a low bit rate and meanwhile ensures the compressed image quality by introducing a complementary state codebook. The secret data is embedded into the ESMVQ-index to achieve a large hiding capacity. The ESMVQ-index can be recovered reversibly after the secret data is extracted. The experimental results demonstrate that our proposed method outperforms the similar works, Chang et al.'s method [5], Huang et al.'s method [8], Lee et al.'s method [11] and Wang et al.'s method [19].

The rest of this paper is organized as follows: In Section 2, we briefly introduce the VQ compression method, the SMVQ compression method and the related SMVQ data hiding method of Chang et al. [5] and Huang et al. [8]. In Section 3, we describe the proposed ESMVQ compression method and the proposed reversible data hiding method based on

ESMVQ with its preprocessing, embedding and extracting processes in detail. In Section 4, we show the experimental results to verify the superiority of the proposed method. Finally, the conclusions are summarized in Section 5.

## 2 Related Work

In this section, a brief introduction to VQ is shown in Section 2.1 and Section 2.2 introduces the SMVQ concept. The SMVQ-based data hiding method of Chang et al. [5] is described in Section 2.3 and finally is the Huang et al.'s method [8].

### 2.1 Vector quantization

Vector quantization [7] is one of the popular lossy data compression techniques in the image processing field [12] because of its low-bit-rate, simple encoding and decoding procedures. VQ can be simply regarded as a mapping function which maps the  $k$ -dimensional space  $R^k$  into a finite subset  $Y = \{Y_0, Y_1, \dots, Y_{N-1}\}$ , where  $Y$  is called codebook and  $Y_j = (Y_{j,0}, Y_{j,1}, \dots, Y_{j,k-1})$  is the  $j^{\text{th}}$  codeword in the codebook  $Y$  and the size of the codebook  $Y$  is  $N$ .

Before encoding, the original image is firstly partitioned into  $k$ -dimensional non-overlapping blocks and then each block is encoded by a  $k$ -dimensional codeword. In the encoding procedure, VQ finds the closest codeword  $Y_{bm}$  for each input vector  $X_i$  from the codebook  $Y$ , where  $Y_{bm}$  is selected according to eq. (1) and (2).

$$d(X_i, Y_j) = \|X_i - Y_j\|^2 = \sum_{l=0}^{k-1} (X_{i,l} - Y_{j,l})^2 \quad (1)$$

$$d(X_i, Y_{bm}) = \underset{0 \leq j \leq N-1}{\text{MIN}} \{d(X_i, Y_j)\} \quad (2)$$

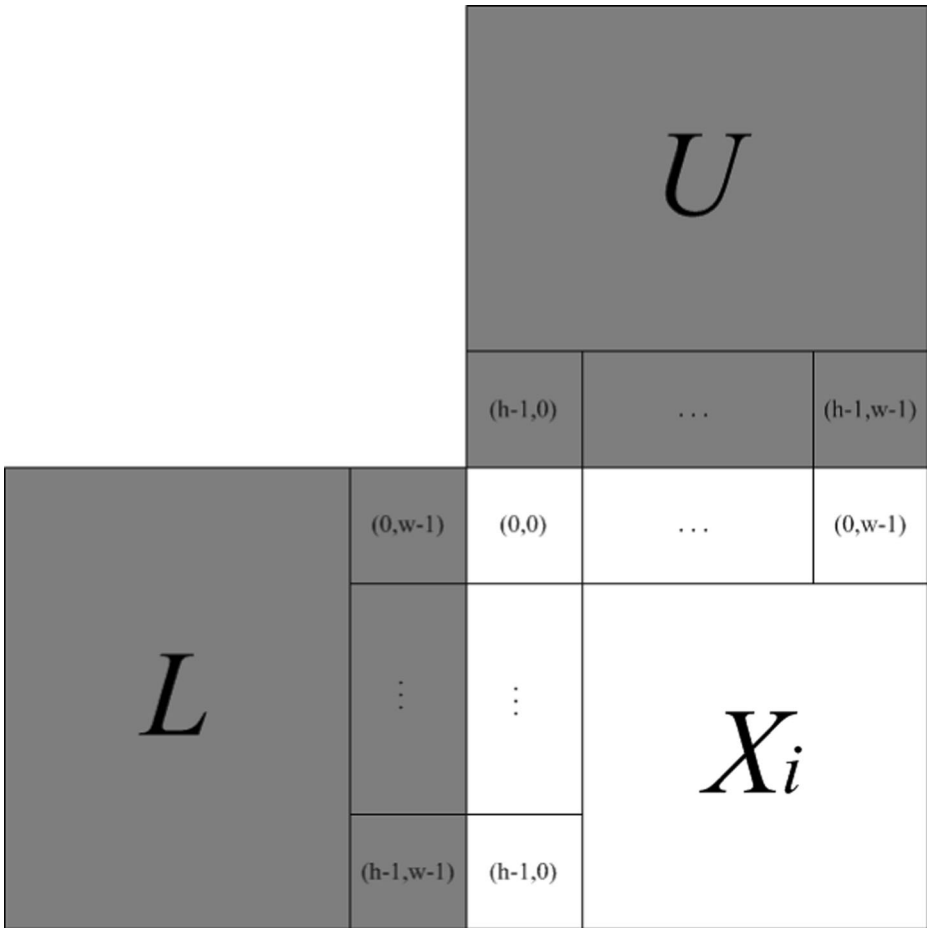
where  $X_i$  is the  $i^{\text{th}}$  input vector,  $Y_j$  is the  $j^{\text{th}}$  codeword in the codebook  $Y$  and  $d(X_i, Y_j)$  is the squared Euclidean distance between the input vector  $X_i$  and the codeword  $Y_j$ . Then VQ encodes the vector  $X_i = (X_{i,0}, X_{i,1}, \dots, X_{i,k-1})$  using the index “bm” of  $Y_{bm}$  in the codebook  $Y$ . In the decoding procedure, VQ can use a simple table look-up technique using received “bm” to reconstruct the compressed image from the VQ index table.

### 2.2 Side-match vector quantization

VQ takes advantage of the high correlation between neighboring pixels within a block but without taking into account the correlation between neighboring blocks. Therefore, the side-match vector quantization technique was proposed by Kim in 1992 [9] to improve the compression ratio of VQ.

In SMVQ method, the first row blocks and the first column blocks are encoded by conventional VQ and the residual blocks are encoded in a raster scanning order using a smaller state codebook (SC) which is generated using a main codebook by online side-match distortion prediction.

For each input vector  $X_i$ , SMVQ first predicts a state codebook for it by using its upper block U and left block L as shown in Fig. 1. SMVQ calculates side-match distortion  $d_{sm}$  for all codewords in the main codebook  $Y$  and then sorts the codewords according to the side-match distortion  $d_{sm}$  in an ascending order and selects the first



**Fig. 1** Relationship of neighboring image blocks in SMVQ

$N_s$  ( $N_s < N$ ) codewords with smallest  $d_{sm}$  to construct a state codebook for vector  $X_i$ . The side-match distortion  $d_{sm}$  is defined in eq. (3).

$$d_{sm}(X_i, Y_j) = \sum_{n=0}^{w-1} (Y_{j(0,n)} - U_{(h-1,n)})^2 + \sum_{n=0}^{h-1} (Y_{j(n,0)} - L_{(n,w-1)})^2 \tag{3}$$

Then SMVQ selects the closest codeword  $Y_{bm}$  for vector  $X_i$  from the state codebook and encodes the vector  $X_i$  with the index of the  $Y_{bm}$  in the state codebook. Since the size  $N_s$  of the state codebook is smaller than the size  $N$  of main codebook  $Y$ , SMVQ can reduce the bit rate obviously.

### 2.3 Chang et al.’s method

In 2006, Chang et al. proposed a reversible data hiding method based on SMVQ [5]. The first row blocks and the first column blocks compressed by VQ did not embed the secret data. For each residual block, the closest codeword  $k_a$  from its state codebook is selected. If the secret

data is “0”, they replaced the block with the codeword  $k_a$ . Otherwise, the closest codeword  $k_b$  for codeword  $k_a$  from the state codebook is selected and an approximate codeword  $k_c$  is computed to replace the block.  $k_c$  is defined as below.

$$k_c = \left\lfloor \frac{2 \times k_a + 1 \times k_b}{3} \right\rfloor \quad (4)$$

In Chang et al.’s method, they can embed just one bit secret data into each block compressed by SMVQ and the image quality is not very good and the hiding capacity is generally rather small.

## 2.4 Huang et al.’s method

In 2013, Huang et al. proposed a reversible data hiding method based on SMVQ [8] which enlarges the hiding capacity and improves the image quality than that of Chang et al.’s method [5]. As in the Chang et al.’s method, the first row blocks and the first column blocks compressed by VQ did not embed secret data.

### 2.4.1 The 1-bit embedding method

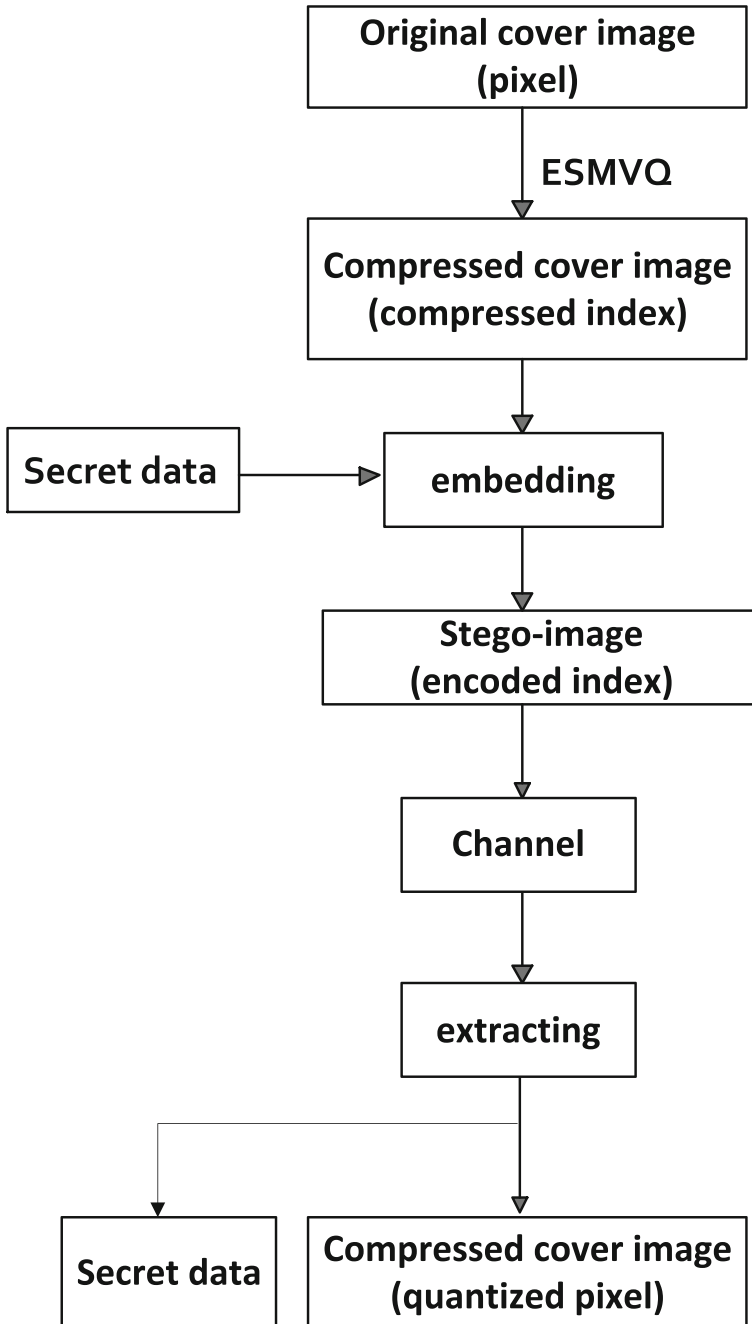
For the blocks compressed by SMVQ, the closest codeword  $k_a$  is found from its state codebook and the closest codeword  $k_b$  for codeword  $k_a$  is found from the main codebook. If the secret data is “0”, the block is replaced with the codeword  $k_a$ . Otherwise, the block is replaced with the approximate codeword  $\lfloor (3 \times k_a + 1 \times k_b) / 4 \rfloor$ .

### 2.4.2 The 2-bit embedding method

For the blocks compressed by SMVQ, the closest codeword  $k_a$  is found from its state codebook and the closest codeword  $k_b$  for codeword  $k_a$  is found from the main codebook. If the secret data is “00”, the block is replaced with the codeword  $k_a$ . If the secret data is “01”, the block is replaced with the approximate codeword  $\lfloor (7 \times k_a + 1 \times k_b) / 8 \rfloor$ . If the secret data is “10”, the block is replaced with the approximate codeword  $\lfloor (6 \times k_a + 2 \times k_b) / 8 \rfloor$ . Otherwise, the block is replaced with the approximate codeword  $\lfloor (5 \times k_a + 3 \times k_b) / 8 \rfloor$ .

## 3 The proposed method

In this section, a novel technique, called enhanced side-match vector quantization which combines the advantage of VQ and SMVQ by using a very small state codebook and a complementary state codebook, is proposed to achieve a high compression rate and large embedding capacity. To a certain extent, our proposed ESMVQ can overcome the large encoded distortion weakness of conventional SMVQ and the high bit rate weakness of conventional VQ. Then our proposed method compresses the image using ESMVQ to get ESMVQ compressed images and then embeds the secret data into the ESMVQ compressed indices to form a code stream CS. The code stream CS is transmitted to receiver. The secret data is extracted from code stream CS and the indices are recovered reversibly to reconstruct the ESMVQ compressed images. The general process of our proposed reversible data hiding method is shown in Fig. 2. The proposed enhanced side-match vector quantization is described in Section 3.1.



**Fig. 2** The general process of proposed reversible data hiding method

Section 3.2 gives the proposed reversible data hiding method based on ESMVQ with its preprocessing, embedding and extracting processes in detail.

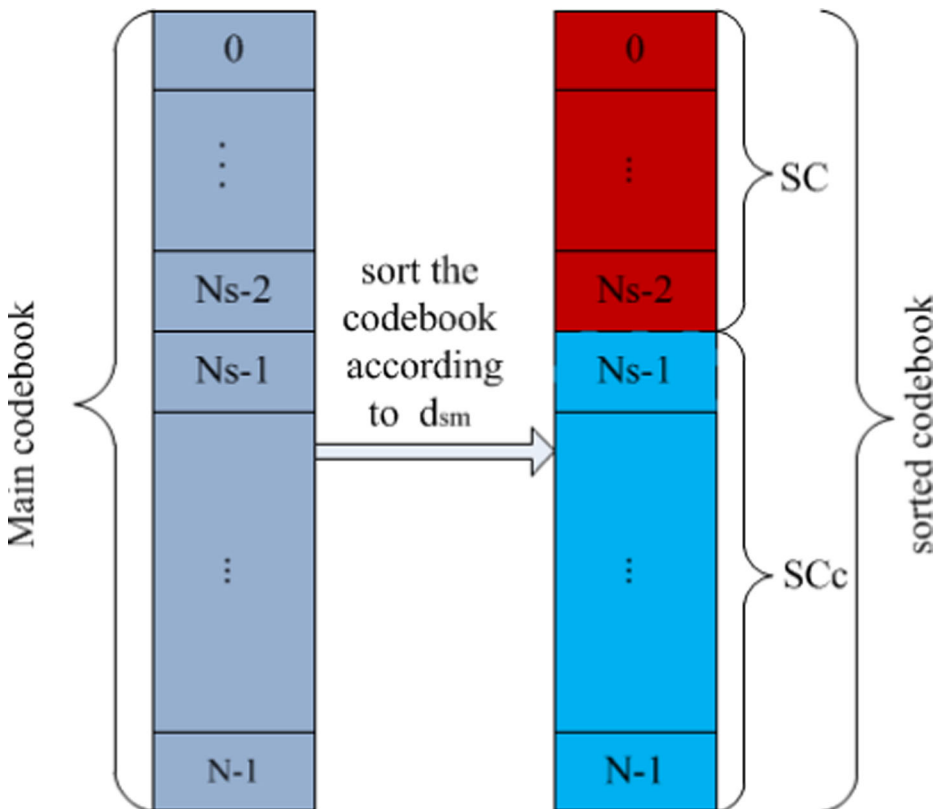
### 3.1 Enhanced side-match vector quantization

Although SMVQ can achieve a lower bit-rate than the conventional VQ with the same size of the main codebook, the image quality using SMVQ is generally much poorer than that using the conventional VQ. In this section, in order to achieve a higher image quality than conventional SMVQ, an enhanced side-match vector quantization is proposed to encode the vector by using the state codebook (SC) and its complementary state codebook (SCc). The relationship of the state codebook and its complementary state codebook in a sorted main codebook is shown in Fig. 3.

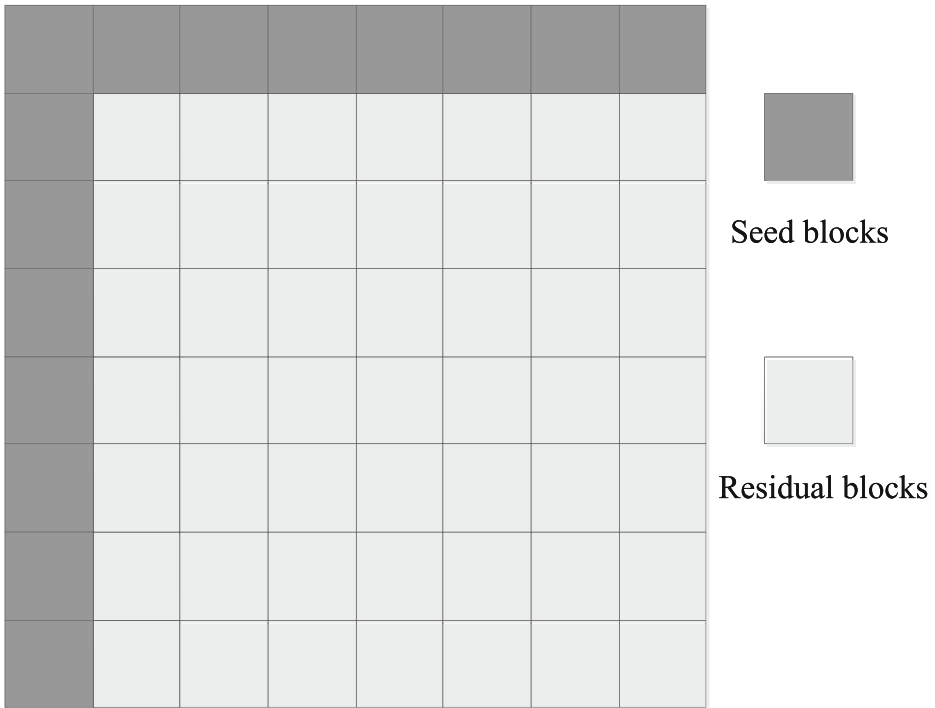
In ESMVQ strategy, the first row blocks and the first column blocks are encoded by conventional VQ as the seed blocks and the residual blocks are encoded in a raster scanning order using a state codebook or a complementary state codebook. Fig. 4 shows the diagram of seed blocks and residual blocks.

For each input vector  $X_i$ , ESMVQ first sorts the codewords according to the side-match distortion  $d_{sm}$  in an ascending order and selects the first  $(N_s - 1)$  codewords to construct a state codebook and uses the other codewords to construct a complementary state codebook for the input vector  $X_i$ .

Then ESMVQ finds the best-match codeword  $Y_{bm1}$  for vector  $X_i$  in SC and calculates the distortion  $dis_1$  between the input vector  $X_i$  and codeword  $Y_{bm1}$ . If the  $dis_1$  is smaller than the pre-defined threshold  $TH_{dis}$ , ESMVQ considers that the encoding quality of vector  $X_i$  is high



**Fig. 3** The state codebook and its complementary state codebook in a codebook



**Fig. 4** The diagram of seed blocks and residual blocks

enough and encodes vector  $X_i$  with the index of codeword  $Y_{bm1}$  in SC. Otherwise, ESMVQ goes to find the best-match codeword  $Y_{bm2}$  for vector  $X_i$  in SCc and calculates the distortion  $dis_2$  between input vector  $X_i$  and codeword  $Y_{bm2}$ . Then, ESMVQ compares these two distortions by eq. (5).

$$diff = dis_1 - dis_2 \quad (5)$$

If  $diff$  is smaller than the pre-defined threshold  $TH_{diff}$ , ESMVQ also considers that the codeword  $Y_{bm1}$  is the closest codeword for the vector  $X_i$  and encodes vector  $X_i$  with the index of codeword  $Y_{bm1}$  in SC. Otherwise, ESMVQ encodes vector  $X_i$  with the index of codeword  $Y_{bm2}$  in the complementary state codebook.

As the best-match codeword for vector  $X_i$  is either  $Y_{bm1}$  or  $Y_{bm2}$ , ESMVQ needs an indicator to indicate the best-match codeword is either  $Y_{bm1}$  or  $Y_{bm2}$ . ESMVQ defines the size of the state codebook is  $(Ns-1)$  and only selects the first  $(Ns-1)$  smallest-distortion codewords as its state codebook. It means that the index of codeword in the state codebook are in binary representation from 0 to  $(Ns-2)$  and the binary representation of  $(Ns-1)$  as an indicator to indicate the encoded index value is  $Y_{bm2}$ , then followed by the encoded index value  $Y_{bm2}$ . For example, if  $Ns$  is equal to 4 ( $\log_2 Ns=2$ ), the indicator will be  $(11)_2$  in binary format. If the best-match codeword for vector  $X_i$  is  $Y_{bm2}$ , ESMVQ first sends the indicator  $(11)_2$  and then sends the encoded index value of  $Y_{bm2}$  in SCc to the receiver. If the best-match codeword for vector  $X_i$  is  $Y_{bm1}$ , ESMVQ only sends the encoded index value of  $Y_{bm1}$  in SC to the receiver. Taking the advantage of the complementary state codebook to find  $Y_{bm2}$ , ESMVQ can achieve a much better image quality than conventional SMVQ at the low bit rate.



### 3.2 The proposed data hiding method

#### 3.2.1 The preprocessing phase

During the preprocessing phase, the main task is to obtain the ESMVQ compressed index table T. The process of obtaining compressed cover image with ESMVQ compressed indices is given as follows:

Input: the original cover image  $I$  sized  $H \times W$  and the codebook  $Y$  sized  $N$ .  
Output: the ESMVQ index table T.

- Step 1 Image  $I$  is partitioned into non-overlapping blocks with size of  $4 \times 4$  pixels.
- Step 2 The first row blocks and the first column blocks are encoded by conventional VQ and put the  $\log_2(N)$ -bit VQ index to the index table T.
- Step 3 For the residual blocks, the state codebook  $N_s$  sized 4. Encode the block in a raster scanning order using the ESMVQ method and put the ESMVQ index to the index table T.
- Step 4 Output the ESMVQ index table T.

#### 3.2.2 The embedding phase

The secret data is embedded into the ESMVQ index table T obtained in the preprocessing phase. The secret data is embedded into the index encoded using the state codebook in our proposed method. For an index  $x$  encoded using the state codebook, assume its 2-bit binary representation is denoted as  $x_2$ ,  $s$ -bit secret data  $sd$  will be embedded into it using the index modification operation which defined as follow.

$$x'_2 = x_2 \ll s + sd \quad (6)$$

where  $\ll$  is the left shifting function. As the length of embedded index  $x$  is 2-bit, the length of final encoded index is  $m$ , where  $m$  is calculated as follow.

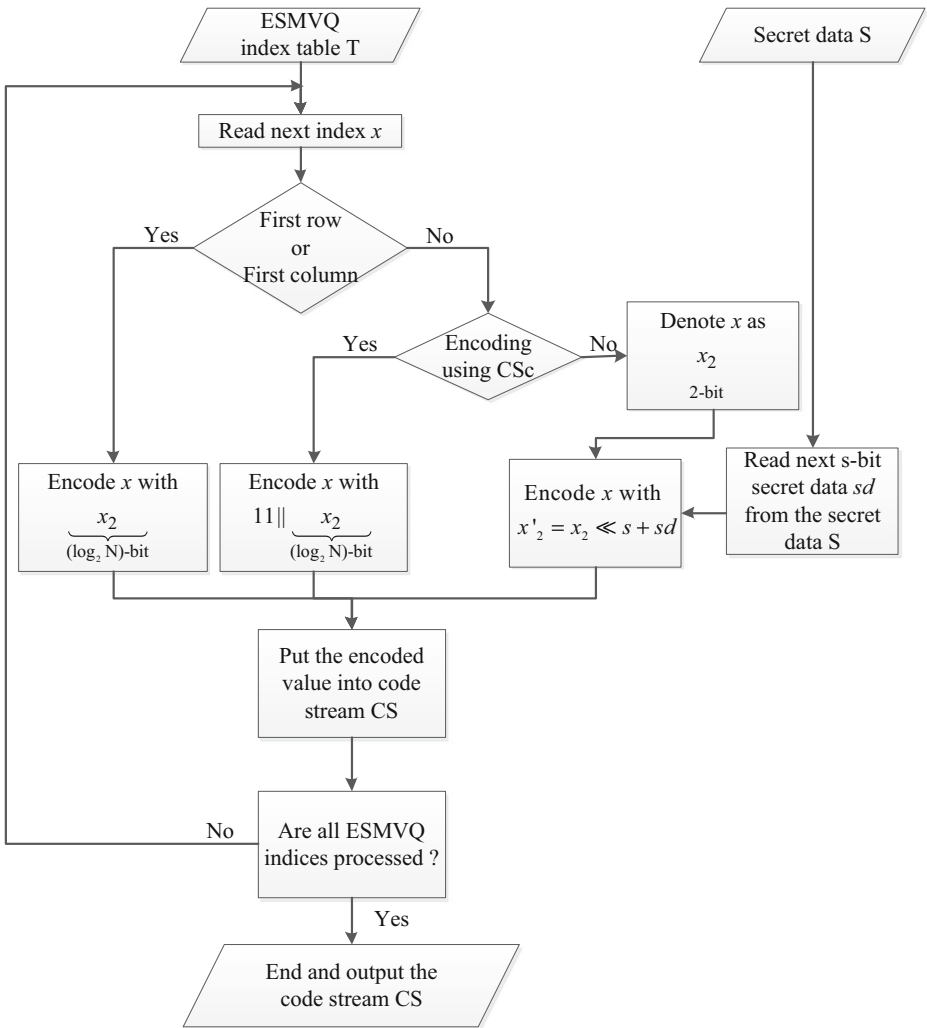
$$m = \begin{cases} 2 + s, & \text{embedded index} \\ 2 + \log_2(N), & \text{non-embedded index} \end{cases} \quad (7)$$

where  $N$  is the size of main codebook  $Y$ .

The embedding process is described as below and the flowchart for the embedding process is shown in Fig. 5.

Input: the ESMVQ index table T and the secret data S.  
Output: the code stream CS.

- Step 1 For the index  $x$  is in the first row and the first column, put its  $\log_2(N)$ -bit binary representation  $x_2$  to the code stream CS.
- Step 2 For the residual indices, Read the next index  $x$  from the ESMVQ index table T.
- Step 3 If the index  $x$  is encoded using the complementary state codebook, put  $11||x_2$  to the code stream CS, where  $x_2$  is  $\log_2(N)$ -bit binary representation of  $x$  and “||” denotes concatenation operation.



**Fig. 5** The flowchart for the embedding process

- Step 4 Otherwise, read next  $s$ -bit secret data  $sd$  from the secret data  $S$  and embed  $sd$  into the index  $x$  using the eq. (6). Put  $(2+s)$ -bit binary representation  $x'_2$  to the code stream  $CS$ .
- Step 5 Repeat the Step 1 to Step 4 until all ESMVQ indices are processed and output the code stream  $CS$ .

### 3.2.3 The decoding and extracting phase

In the decoding and extracting phase, the receiver can extract the secret data and recover the original ESMVQ indices, then reconstruct the ESMVQ compressed image. The blocks in the first row and the first column are reconstructed first using the table look-up technique in the main codebook. As the first 2-bit is different for the embedded index and non-embedded

index, read the first 2-bit  $cs$ . If the  $cs$  equals to “11”, it means this is a non-embedded index. Otherwise, it means this is an embedded index. For the non-embedded index, the next  $\log_2(N)$ -bit  $cs$  is the original ESMVQ index. For the embedded index, the original ESMVQ index is  $cs$  and the next  $s$ -bit  $cs$  is secret data  $sd$ .

The extracting process is shown as follows:

Input: the code stream CS and the codebook  $Y$  sized  $N$ .

Output: the secret data S and the reconstructed ESMVQ compressed image.

- Step 1 For the first row blocks and the first column blocks, read the  $\log_2(N)$ -bit binary representation  $x_2$  from the code stream CS to reconstructed the block using the table look-up technique in the main codebook  $Y$ .
- Step 2 For the residual blocks, reconstruct them in a raster scanning order.
- Step 3 For the current block to be reconstructed, use its upper block U and left block L previously reconstructed to create its state codebook and its complementary state codebook .
- Step 4 Read the next 2-bit code stream  $cs$ .
- Step 5 If 2-bit code stream  $cs$  equals to “11”, then read the next  $\log_2(N)$ -bit code stream  $cs$  from the code stream CS. Reconstructed the block using the table look-up technique in SCc.
- Step 6 Otherwise,  $cs$  is the original ESMVQ index and the next  $s$ -bit  $cs$  is secret data  $sd$ . Put the secret data  $sd$  to the secret data S. Reconstructed the block using the table look-up technique in SC.
- Step 7 Repeat Step 4 to Step 6 until all code stream is processed. Output the secret data S and the reconstructed ESMVQ compressed image.

#### 4 Experimental results

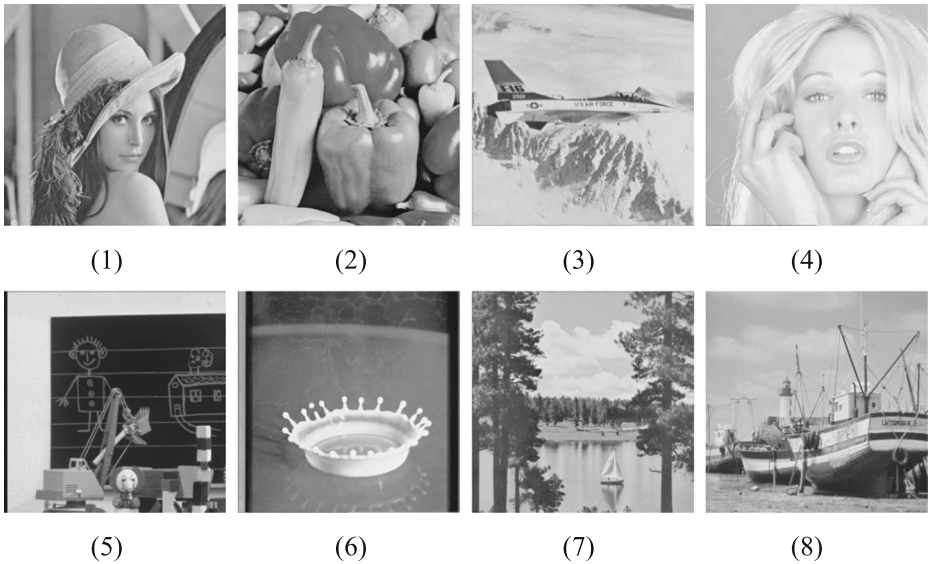
Experiments are performed to demonstrate the performance of our proposed method. In our experiments, eight typical grayscale images are used as the test images shown in Fig. 6. Each test image is  $512 \times 512$  pixels and each pixel has 256 Gy levels. Each test image is first partitioned into 16,384 non-overlapping blocks and the size of each block is  $4 \times 4$  pixels. We test our proposed method using the codebooks with the size of 128, 256, 512 and 1,024 respectively. A pseudorandom number generator is used to generate a series of secret data which consist of 0 and 1 in binary format in the experiments.

We use three important criterions as the compressed image quality, the bit rate and embedding capacity to evaluate the performance of our proposed method.

The first criterion is the compressed image quality which is used to estimate distortion between the original cover image and the compressed image. The quality of the stego-image is estimated by the peak signal-to-noise-ratio (PSNR). The PSNR is defined as follow.

$$MSE = \frac{1}{H \times W} \sum_{i=1}^M \sum_{j=1}^N (x_{i,j} - \hat{x}_{i,j})^2 \quad (8)$$

$$PSNR = 10 \times \log \left( \frac{255^2}{MSE} \right) \quad (9)$$



**Fig. 6** Eight test images: (1) Lena, (2) Peppers, (3) Airplane, (4) Tiffany, (5) Toys, (6) Splash, (7) Lake, (8) Boat

where  $H$  and  $W$  are the height and width of an image.  $x_{i,j}$  and  $\hat{x}_{i,j}$  respectively denote the pixel value of the original cover image and compressed image. Basically, a higher PSNR means a better quality of the compressed image.

The second criterion is the bit rate (BR) which is used to evaluate the compression performance of the compression methods in the compression domain. The bit rate is to compare the length of the output code stream with the size of the original cover image. And it is defined as follow.

$$BR = \frac{\|I\|}{H \times W} \quad (10)$$

where  $\|I\|$  represents the total length of the compressed code stream and  $H$  and  $W$  are the height and width of the original cover image, respectively. Generally, the smaller the bit rate is the better the compression is.

The third criterion is the embedding capacity which is evaluated by the hiding capacity (HC) and the embedding rate (ER). The hiding capacity shows how many secret bits can be embedded into a cover image. The embedding rate is defined as follows:

$$ER = \frac{HC}{\|I\|} \quad (11)$$

The embedding rate shows that how many secret bits can be carried by each bit when the output code stream is transmitted. In general, larger hiding capacity and higher embedding rate indicate a better embedding performance.

Table 1 shows the compressed results of image Lena (without secret data) for different main codebook and state codebook among the conventional VQ, SMVQ ESMVQ. The main codebook sizes are 128, 256, 512 and 1,024. The size of state codebook  $N_s$  is 4, 8, 16 and 32 in SMVQ and ESMVQ respectively. In ESMVQ, there are two pre-defined thresholds of

$TH_{dis}$  and  $TH_{diff}$  which need to be confirmed. As the 30 dB is a typically acceptable value in lossy image compression and 30 dB is corresponding to the mean-square-error MSE of 65.025. The pre-defined threshold  $TH_{dis}$  which corresponds to the sum square error for a block sized  $4 \times 4$  is selected as  $16 \times 65.025$ . The pre-defined threshold  $TH_{dis}$  is the threshold to guarantee the compressed image quality. The other per-defined threshold  $TH_{diff}$  is to provide a tradeoff between the compressed image quality and the bit rate. The threshold  $TH_{diff}$  is denoted as the 10 % of the threshold  $TH_{dis}$ .

From Table 1, it can be found that our proposed ESMVQ which combines both the advantage of VQ and SMVQ has a good image quality while keeping a low bit rate. For example, the PSNR is 31.544 dB and the BR is 0.500 bpp in VQ and the PSNR is 31.338 dB and the BR is 0.262 bpp in ESMVQ when the main codebook size is 256. The PSNR is 28.184 dB with the BR of 0.253 bpp at  $N_s$  of 32 in SMVQ while the PSNR is 30.553 dB with the BR of 0.235 bpp at  $N_s$  of 4 in ESMVQ when the main codebook is 128. To achieve a high embedding capacity for the data hiding later, we set  $N_s$  equal to 4 for various main codebook sizes in our proposed method.

Table 2 gives the PSNR comparisons for different test images and main codebook sizes between VQ and ESMVQ when  $N_s$  is 4. From the Table 2, we can find that the average image quality difference between VQ and ESMVQ becomes larger as the main codebook size increases. The average image quality difference is from 0.067 dB to 0.494 dB. It shows that ESMVQ can keep a good image quality and the image quality of ESMVQ is close to that of VQ, especially in the case of smaller main codebook size.

Table 3, 4, 5, 6 show the results for various images when the main codebook size is 128, 256, 512 and 1,024 in our proposed method, respectively. The number  $s$  of the

**Table 1** The compressed results of image Lena (without secret data) for different main codebook and state codebook among VQ, SMVQ and ESMVQ

Main codebook (N)	State codebook ( $N_s$ )	VQ		SMVQ		ESMVQ	
		PSNR (dB)	BR (bpp)	PSNR (dB)	BR (bpp)	PSNR (dB)	BR (bpp)
128	4	30.637	0.438	23.186	0.130	30.553	0.235
	8			26.115	0.191	30.617	0.253
	16			28.184	0.253	30.632	0.285
	32			29.725	0.314	30.636	0.326
256	4	31.544	0.500	23.125	0.131	31.338	0.262
	8			26.163	0.192	31.474	0.278
	16			28.286	0.254	31.520	0.307
	32			29.783	0.325	31.540	0.344
512	4	32.427	0.563	22.673	0.132	32.013	0.284
	8			25.373	0.193	32.264	0.301
	16			27.612	0.255	32.364	0.331
	32			29.664	0.316	32.409	0.365
1,024	4	33.231	0.625	22.474	0.133	32.584	0.306
	8			25.172	0.190	32.914	0.326
	16			26.868	0.256	33.078	0.354
	32			29.203	0.317	33.160	0.386

**Table 2** PSNR comparisons for different test images and main codebook sizes between VQ and ESMVQ ( $N_s=4$ )

Images	128		256		512		1,024	
	VQ	ESMVQ	VQ	ESMVQ	VQ	ESMVQ	VQ	ESMVQ
Lena	30.637	30.553	31.544	31.338	32.427	32.103	33.231	32.584
Peppers	30.712	30.604	31.514	31.286	32.475	31.987	33.361	32.644
Airplane	30.141	30.098	31.207	31.111	32.002	31.740	32.896	32.523
Tiffany	31.087	31.020	31.586	31.483	33.028	32.580	33.841	33.071
Toys	29.994	29.964	31.074	31.025	32.297	32.139	33.255	32.929
Splash	30.725	30.581	31.658	31.366	32.752	32.269	33.467	32.768
Lake	28.256	28.226	29.024	28.973	29.651	29.513	30.327	30.112
Boat	28.686	28.661	29.653	29.587	30.065	29.932	30.717	30.509
Average	30.030	29.963	30.908	30.771	31.837	31.533	32.637	32.143

secret data for each compressed index belonging to the state codebook is defined as follows:

$$s = \log_2(N) - \log_2(N_s) - 1 \tag{12}$$

For example, when the main codebook size  $N$  equals 256 and  $N_s$  equals to 4,  $s$  is equal to 5. And it is for that the bit rate of the proposed method is close to that of VQ at the same main codebook size. For example, the average BR is 0.483 bpp in the proposed method which is close to 0.500 bpp in VQ when the main codebook size is 256.

For different main codebook sizes, the average BR of our proposed method is smaller than that of VQ. But the BR of some images in our proposed method is not smaller than that of VQ. For example, the average BR is 0.549 bpp in our proposed method which is smaller than 0.563 bpp in VQ. The BR of image Lake is 0.580 bpp which is larger than 0.563 bpp in VQ and the BR of image Boat is 0.563 bpp which is equal to 0.563 bpp in VQ.

The embedding capacity and the image quality increase as the main codebook size increases. For example, the average HC is 72,068 bits at the main codebook size of 512 which is larger than 61,329 bits at the main codebook size of 256 and the average ER is 0.503 at the main codebook size of 512 which is larger than 0.486 at the main codebook size of 256. The average PSNR is 31.533 dB at the main codebook size of 512 which is larger than 30.771 dB at the main codebook size of 256.

In Table 7–8, we compare our proposed method with Chang et al.’s method [5] and Huang et al.’s method [8]. In the two methods of Chang et al.’s [5] and Huang et al. [8], the fixed main codebook is 256 and the fixed state codebook is 128. Therefore, the main codebook size of 256 is used in our comparison experiments. Table 7 shows the comparison results for various

**Table 3** The results for different test images in our proposed method ( $N=128$ )

Image	Lena	Peppers	Airplane	Tiffany	Toys	Splash	Lake	Boat	Average
PSNR (dB)	30.553	30.604	30.098	31.020	29.964	30.581	28.226	28.661	29.963
BR (bpp)	0.421	0.416	0.417	0.402	0.411	0.398	0.437	0.428	0.416
HC (bit)	48,760	50,524	50,032	55,404	52,404	56,808	43,312	46,484	50,466
ER	0.442	0.463	0.457	0.526	0.487	0.544	0.378	0.415	0.464

**Table 4** The results for different test images in our proposed method ( $N=256$ )

Image	Lena	Peppers	Airplane	Tiffany	Toys	Splash	Lake	Boat	Average
PSNR (dB)	31.338	31.286	31.111	31.483	31.025	31.366	28.973	29.587	30.771
BR (bpp)	0.488	0.482	0.483	0.469	0.476	0.463	0.505	0.497	0.483
HC (bit)	59,150	61,785	61,010	67,545	64,485	70,015	51,395	55,250	61,329
ER	0.463	0.489	0.481	0.550	0.517	0.577	0.388	0.424	0.486

**Table 5** The results for different test images in our proposed method ( $N=512$ )

Image	Lena	Peppers	Airplane	Tiffany	Toys	Splash	Lake	Boat	Average
PSNR (dB)	32.103	31.987	31.740	32.580	32.139	32.269	29.513	29.932	31.533
BR (bpp)	0.552	0.545	0.548	0.537	0.540	0.527	0.580	0.563	0.549
HC (bit)	70,134	73,476	72,156	78,030	76,488	83,082	59,046	64,134	72,068
ER	0.485	0.514	0.502	0.555	0.541	0.601	0.389	0.434	0.503

test images between Chang et al.'s method [5] and our proposed method. Table 8 compares the results for various test images between Huang et al.'s method [8] and our proposed method. The left and middle of Table 8 show the results of Huang et al.'s method with 1-bit and 2-bit of secret data embedded into each block. The right of Table 8 shows the results of our proposed method.

In Chang et al.'s method [5], 1-bit secret data is embedded into each block. There are two data hiding methods which embed 1-bit or 2-bit secret data into each block in Huang et al.'s method [8], respectively. In our proposed method, 5-bit secret data is embedded into each compressed index belonging to the state codebook. Therefore, our proposed method has a larger HC and a higher ER than that of these two methods. Our proposed ESMVQ has good image quality because it combines the advantage of VQ and SMVQ. From the Table 7 and Table 8, it is clear that the average PSNR of our proposed method is 30.771 dB which is higher than 30.550 dB in Chang et al.'s method and higher than 30.581 dB and 30.439 dB in Huang et al.'s method.

From the Table 7 and Table 8, it is clearly found that our proposed method has a larger HC and a higher ER than that of these two methods. Compared with the method of Chang et al., the average HC of our proposed method is approximately 3.5 times as high as that of Chang et al.'s method. Similarly, the average ER of our proposed method is approximately 3.8 times as high as that of Chang et al.'s method. Compared with the method of Huang et al., which embed 1-bit secret data into each block, the average HC of our proposed method is approximately 3.5 times as high as that of Huang et al.'s method. Similarly, the average ER of our

**Table 6** The results for different test images in our proposed method ( $N=1,024$ )

Image	Lena	Peppers	Airplane	Tiffany	Toys	Splash	Lake	Boat	Average
PSNR (dB)	32.584	32.644	32.523	33.071	32.929	32.768	30.112	30.509	32.143
BR (bpp)	0.615	0.608	0.612	0.601	0.602	0.589	0.639	0.628	0.612
HC (bit)	81,123	85,512	83,286	89,810	89,355	97,545	66,892	73,409	83,367
ER	0.503	0.537	0.519	0.570	0.566	0.632	0.400	0.446	0.522

**Table 7** Comparison results for different test images between Chang et al.'s method [5] and our proposed method

Image	Chang et al.' method [5]				Proposed method			
	PSNR (dB)	BR (bpp)	HC (bit)	ER	PSNR (dB)	BR (bpp)	HC (bit)	ER
Lena	31.281	0.438	16,129	0.140	31.338	0.488	59,150	0.463
Peppers	31.248	0.438	16,129	0.140	31.286	0.482	61,785	0.489
Airplane	30.902	0.438	16,129	0.140	31.111	0.483	61,010	0.481
Tiffany	31.317	0.438	16,129	0.140	31.483	0.469	67,545	0.550
Toys	30.383	0.438	16,129	0.140	31.025	0.476	64,485	0.517
Splash	31.306	0.438	16,129	0.140	31.366	0.463	70,015	0.577
Lake	28.708	0.438	16,129	0.140	28.973	0.505	51,395	0.388
Boat	29.258	0.438	16,129	0.140	29.587	0.497	55,250	0.424
Average	30.550	0.438	16,129	0.140	30.771	0.483	61,329	0.486

proposed method is approximately 3.8 times as high as that of Huang et al.'s method. Compared with the method of Huang et al., which embed 2-bit secret data into each block, the average HC of our proposed method is approximately 1.7 times as high as that of Huang et al.'s method. Similarly, the average ER of our proposed method is approximately 1.9 times as high as that of Huang et al.'s method. Therefore, our proposed method has a better performance in image quality, hiding capacity and embedding rate compared with the Chang et al.'s method and Huang et al.'s method.

In order to further prove that our proposed method has a better performance compared with Chang et al.'s method [5] and Huang et al.'s method [8], another comparison experiment is conducted in which we embed the same size of secret data as Chang et al.'s method [5] and Huang et al.'s method [8]. In this experiment, the secret data is not embedded any more if a certain size of secret data has been embedded. The experiment results are shown in Table 9 and Table 10 as below. In Table 9, we set the size of secret data to 16,129 bits which is the same as

**Table 8** Comparison results for different test images between Huang et al.'s method [8] and our proposed method

Image	Huang et al.' method [8] (1-secret bit per block)				Huang et al.' method [8] (2-secret bit per block)				Proposed method			
	PSNR (dB)	BR (bpp)	HC (bit)	ER	PSNR (dB)	BR (bpp)	HC (bit)	ER	PSNR (dB)	BR (bpp)	HC (bit)	ER
Lena	31.286	0.438	16,129	0.140	31.153	0.438	32,258	0.281	31.338	0.488	59,150	0.463
Peppers	31.261	0.438	16,129	0.140	31.109	0.438	32,258	0.281	31.286	0.482	61,785	0.489
Airplane	30.898	0.438	16,129	0.140	30.784	0.438	32,258	0.281	31.111	0.483	61,010	0.481
Tiffany	31.311	0.438	16,129	0.140	31.153	0.438	32,258	0.281	31.483	0.469	67,545	0.550
Toys	30.394	0.438	16,129	0.140	30.208	0.438	32,258	0.281	31.025	0.476	64,485	0.517
Splash	31.302	0.438	16,129	0.140	31.124	0.438	32,258	0.140	31.366	0.463	70,015	0.577
Lake	28.818	0.438	16,129	0.140	28.716	0.438	32,258	0.281	28.973	0.505	51,395	0.388
Boat	29.378	0.438	16,129	0.140	29.267	0.438	32,258	0.281	29.587	0.497	55,250	0.424
Average	30.581	0.438	16,129	0.140	30.439	0.438	32,258	0.281	30.771	0.483	61,329	0.486



**Table 9** Comparison results for different test images between Chang et al.’s method and our proposed method under the same hiding capacity

Image	Chang et al.’ method [5]				Proposed method			
	PSNR (dB)	BR (bpp)	HC (bit)	ER	PSNR (dB)	BR (bpp)	HC bit)	ER
Lena	31.281	0.438	16,129	0.140	31.338	0.324	16,129	0.190
Peppers	31.248	0.438	16,129	0.140	31.286	0.308	16,129	0.200
Airplane	30.902	0.438	16,129	0.140	31.111	0.312	16,129	0.197
Tiffany	31.317	0.438	16,129	0.140	31.483	0.273	16,129	0.225
Toys	30.383	0.438	16,129	0.140	31.025	0.292	16,129	0.211
Splash	31.306	0.438	16,129	0.140	31.366	0.257	16,129	0.239
Lake	28.708	0.438	16,129	0.140	28.973	0.370	16,129	0.166
Boat	29.258	0.438	16,129	0.140	29.587	0.348	16,129	0.177
Average	30.550	0.438	16,129	0.140	30.771	0.311	16,129	0.201

that of Chang et al.’s method [5]. In Table 2, we set the size of secret data to 32,258 bits which is the same as that of Huang et al.’s method with 2-bit of secret data embedded into each block.

In Table 9, it is clear that our proposed method has a better average image quality (30.771 dB) which is higher than that of Chang et al.’s method (30.550 dB). The average bit rate of our proposed method is 0.311 bpp which is much lower than that of Chang et al.’s method (0.438 bpp). Our proposed method also has a higher average embedding rate (0.201) compared with that of Chang et al.’s method (0.140).

In Table 10, it can be clearly found that the best average image quality is our proposed method (30.771 dB), followed by the 1-bit embedding method (30.581 dB) and the 2-bit embedding method (30.439 dB) in Huang et al.’s method [8]. The average bit rate of our proposed method is 0.372 bpp which is lower than that of Huang et al.’s method (0.438 bpp). The highest average embedding capacity is our proposed method (0.334), followed by the 2-bit embedding method (0.281) and the 1-bit embedding method (0.140) in Huang et al.’s method [8].

**Table 10** Comparison results for different test images between Huang et al.’s method and our proposed method under the same hiding capacity

Image	Huang et al.’ method [8] (1-secret bit per block)				Huang et al.’ method [8] (2-secret bit per block)				Proposed method			
	PSNR (dB)	BR (bpp)	HC (bit)	ER	PSNR (dB)	BR (bpp)	HC (bit)	ER	PSNR (dB)	BR (bpp)	HC (bit)	ER
Lena	31.286	0.438	16,129	0.140	31.153	0.438	32,258	0.281	31.338	0.385	32,258	0.319
Peppers	31.261	0.438	16,129	0.140	31.109	0.438	32,258	0.281	31.286	0.369	32,258	0.333
Airplane	30.898	0.438	16,129	0.140	30.784	0.438	32,258	0.281	31.111	0.373	32,258	0.330
Tiffany	31.311	0.438	16,129	0.140	31.153	0.438	32,258	0.281	31.483	0.334	32,258	0.368
Toys	30.394	0.438	16,129	0.140	30.208	0.438	32,258	0.281	31.025	0.353	32,258	0.349
Splash	31.302	0.438	16,129	0.140	31.124	0.438	32,258	0.140	31.366	0.319	32,258	0.386
Lake	28.818	0.438	16,129	0.140	28.716	0.438	32,258	0.281	28.973	0.432	32,258	0.285
Boat	29.378	0.438	16,129	0.140	29.267	0.438	32,258	0.281	29.587	0.409	32,258	0.301
Average	30.581	0.438	16,129	0.140	30.439	0.438	32,258	0.281	30.771	0.372	32,258	0.334

**Table 11** Comparison results for different test images among Lee et al.’s method [11], Wang et al.’s method [19] and our proposed method

Image	Lee et al.’ method [11]			Wang et al.’s method [19]			Proposed method		
	BR (bpp)	HC (bit)	ER	BR (bpp)	HC (bit)	ER	BR (bpp)	HC (bit)	ER
Lena	0.589	69,840	0.452	0.577	50,684	0.335	0.552	70,134	0.485
Peppers	0.592	72,360	0.466	0.576	51,709	0.343	0.545	73,476	0.514
Airplane	0.600	71,863	0.457	0.576	50,932	0.337	0.548	72,516	0.502
Tiffany	0.551	65,290	0.452	0.572	55,632	0.371	0.537	78,030	0.555
Toys	0.578	71,166	0.470	0.571	56,208	0.375	0.540	76,488	0.541
Splash	0.573	74,826	0.499	0.567	60,948	0.410	0.527	83,082	0.601
Lake	0.581	59,046	0.388	0.583	43,860	0.287	0.580	59,046	0.389
Boat	0.576	52,284	0.346	0.578	49,260	0.325	0.563	64,134	0.434
Average	0.580	67,084	0.441	0.575	52,404	0.348	0.549	72,068	0.503

The experimental results in Table 9 and Table 10 can further demonstrate that our proposed method has a better image quality, a lower bit rate and a higher embedding rate compared with Chang et al.’s method [5] and Huang et al.’s method [8] at the same hiding capacity.

To further verify the superiority of our proposed method, we have conducted another two experiments to compare our proposed method with Lee et al.’s method [11] and Wang et al.’s method [19] in term of bit rate, hiding capacity and embedding rate. The main codebook is selected as 512 which is the same as that in Wang et al.’s method [19]. The first one is that the experiment is conducted using the 8 images shown in Fig. 6. The experimental results are listed in Table 11. The second one is that the experiment is conducted using 96 8-bit gray levels images sized  $512 \times 512$  and 68 8-bit gray levels images sized  $256 \times 256$  from CVG-UGR Image database [6] and the experimental results are listed in Table 12.

In Table 11, it is clearly found that the best average bit rate is our proposed method (0.549 bpp), followed by Wang et al.’s method [19] (0.575 bpp) and next is Lee et al.’s method [11] (0.580 bpp). Also the largest average hiding capacity is our proposed method (72,068 bits), followed by Lee et al.’s method [11] (67,084 bits) and next is Wang et al.’s method [19] (52,404 bits). From Table 11, it is also clear that our proposed method has the best embedding rate (0.503) which is higher than that of Lee et al.’s method [11] (0.441) and Wang et al.’s method [19] (0.348). In Table 12, it is also clear that our proposed method has a lower bit rate, a higher hiding capacity and a higher embedding rate compared with Lee et al.’s method [11] and Wang et al.’s method [19] in the  $512 \times 512$  resolution and  $256 \times 256$  resolution. The experimental results in Table 11 and Table 12 demonstrate that our proposed method has a

**Table 12** Comparison average results for different image resolution among Lee et al.’s method [11], Wang et al.’s method [19] and our proposed method

Image resolution	Lee et al.’ method [11]			Wang et al.’s method [19]			Proposed method		
	BR (bpp)	HC (bit)	ER	BR (bpp)	HC (bit)	ER	BR (bpp)	HC (bit)	ER
$512 \times 512$	0.600	57,736	0.370	0.585	42,001	0.276	0.569	60,896	0.412
$256 \times 256$	0.594	15,088	0.390	0.582	11,251	0.297	0.562	15,918	0.437

better performance in the bit rate, hiding capacity and embedding rate compared with the Lee et al.'s method [11] and Wang et al.'s method [19].

## 5 Conclusions

In this paper, a novel reversible data hiding method based on the enhanced side-match vector quantization is proposed to increase hiding capacity of secret data in images as well as to keep a high image quality. In our proposed method, we embed the secret data into the ESMVQ-compressed images. Because we encode the original cover image with a very small state codebook size of 4, the bit rate of compressed image is low and we can embed a large amount of secret data into the compressed index of image. The high compressed image quality is guaranteed by introducing the complementary state codebook at the same time. The experimental results show that the proposed reversible data hiding method has a large embedding capacity and a good image quality. The experimental results also show that our proposed method has a larger hiding capacity, higher embedding rate and better image quality than that of Chang et al.'s method [5] and Huang et al.'s method [8]. Additionally, our proposed method also has a better performance in terms of bit rate, hiding capacity and embedding rate compared with Lee et al.'s method [11] and Wang et al.'s method [19].

**Acknowledgments** This work is supported in part by Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20130201110071), Project Supported by Key Science and Technology Program of Shaanxi Province (Grant No. 2012GY2-30) and Open Project Program of the State Key Lab of SKL (Grant No. KFKT2013B05), Nanjing University.

## References

1. Celik MU, Sharma G, Tekalp AM, Saber E (2002) Reversible data hiding. *Proceedings of IEEE International Conference on Image Processing 2:II-157-II-160*
2. Chan CK, Cheng LM (2004) Hiding data in images by simple LSB substitution. *Pattern Recogn 37(3):469–474*
3. Chang CC, Kieu TD, Chou YC (2009) Reversible information hiding for VQ indices based on locally adaptive coding. *J Vis Commun Image Represent 20(1):57–64*
4. Chang CC, Nguyen TS, Lin CC (2013) A novel VQ-based reversible data hiding scheme by using hybrid encoding strategies. *J Syst Softw 86(2):389–402*
5. Chang CC, Tai WL, Lin CC (2006) A reversible data hiding scheme based on side match vector quantization. *IEEE Trans Circ Syst Vi Technol 16(10):1301–1308*
6. CVG-UGR Image database [Online]. Available: <http://decsai.ugr.es/cvg/dbimágenes/index.php>
7. Gray RM (1984) Vector quantization. *IEEE ASSP Mag 1(2):4–29*
8. Huang CT, Wang WJ, Yang CH, Wang SJ (2013) A scheme of reversible information hiding based on SMVQ. *The Image Sci J 61(2):195–203*
9. Kim T (1992) Side match and overlap match vector quantizers for images. *IEEE Trans Image Process 1(2):170–185*
10. Lan TH, Tewfik AH (2006) A novel high-capacity data-embedding system. *IEEE Trans Image Process 15(8):2431–2440*
11. Lee JD, Chiou YH, Guo JM (2010) Reversible data hiding based on histogram modification of SMVQ indices. *IEEE Trans Inf Forensic Secur 5(4):638–648*
12. Nasrabadi NM, King RA (1988) Image coding using vector quantization: a review. *IEEE Trans Commun 36(8):957–971*
13. Ni ZC, Shi YQ, Ansari N, Su W (2006) Reversible data hiding. *IEEE Trans Circ Syst Vi Technol 16(3):354–362*

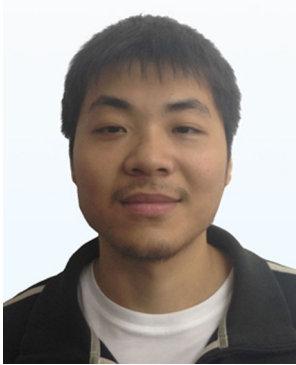
14. Pan ZB, Ma XX, Deng XM, Hu S (2013) Low bit-rate information hiding method based on search-order-coding technique. *J Syst Softw* 86(11):2863–2869
15. Petitcolas FAP, Anderson RJ, Kuhn MG (1999) Information hiding—a survey. *Proc IEEE* 87(7):1062–1078
16. Tai WL, Yeh CM, Chang CC (2009) Reversible data hiding based on histogram modification of pixel differences. *IEEE Trans Circ Syst Vi Technol* 19(6):906–910
17. Tsai P (2009) Histogram-based reversible data hiding for vector quantisation-compressed images. *IET Image Process* 3(2):100–114
18. Tsang PWM, Tsang WH (1996) Preservation of interlaced patterns in encoding video signals using side match vector quantization. *IEEE Trans Consum Electron* 42(1):112–120
19. Wang WJ, Huang CT, Liu CM, Su PC, Wang SJ (2013) Data embedding for vector quantization image processing on the basis of adjoining state-codebook mapping. *Inf Sci* 246:69–82
20. Wang WJ, Huang CT, Wang SJ (2011) VQ applications in steganographic data hiding upon multimedia images. *IEEE Syst J* 5(4):528–537
21. Wang JX, Lu ZM (2009) A path optional lossless data hiding scheme based on VQ joint neighboring coding. *Inf Sci* 179(19):3332–3348



**Xiaoxiao Ma** received the B.S. degree from Xi'an Jiaotong University, Xi'an, P.R. China, in 2012. He is currently working towards the Master degree in School of Electronic and Information Engineering at Xi'an Jiaotong University, Xi'an, P.R. China. His research interests include information hiding and image processing.



**Zhibin Pan** received the B.S. degree in Information and Telecommunication Engineering in 1985 and the M.S. degree in Automation Science and Technology in 1988 from Xi'an Jiaotong University, P.R. China, respectively. He received the Ph.D. degree in Electrical Engineering in 2000 from Tohoku University, Japan. He is a professor in the Department of Information and Telecommunication Engineering, Xi'an Jiaotong University, P.R. China. His current research interests include image compression, multimedia security and object recognition.



**Sen Hu** received the B.S. degree from Northwestern Polytechnical University, Xi'an, P.R. China, in 2010. He is currently working towards the Master degree in School of Electronic and Information Engineering at Xi'an Jiaotong University, Xi'an, P.R. China. His research interests include information hiding and image processing.



**Lingfei Wang** received the B.S. degree from Xi'an Jiaotong University, Xi'an, P.R. China, in 2011. He is currently working towards the Doctor degree in School of Electronic and Information Engineering at Xi'an Jiaotong University, Xi'an, P.R. China. His research interests include information hiding and image processing.