

Lossless VQ Indices Compression Based on the High Correlation of Adjacent Image Blocks

Zhi-Hui Wang¹, Hai-Rui Yang¹, Chin-Chen Chang^{2,3,*}, Gwoboa Horng⁴ and Ying-Hsuan Huang⁴

¹ School of Software, Dalian University of Technology

Dalian, Liaoning, China

[e-mail: wangzhihui1017@gmail.com, hr.dlut@gmail.com]

² Department of Information Engineering and Computer Science, Feng Chia University
Taichung 407, Taiwan

[e-mail: alan3c@gmail.com]

³ Department of Computer Science and Information Engineering, Asia University
Taichung 413, Taiwan

[e-mail: alan3c@gmail.com]

⁴ Department of Computer Science and Engineering, National Chung Hsing University
Taichung 402, Taiwan

[e-mail: gborng@cs.nchu.edu.tw, phd9807@cs.nchu.edu.tw]

*Corresponding author: Chin-Chen Chang

Received March 18, 2014; revised June 1, 2014; accepted June 25, 2014; published August 29, 2014

Abstract

Traditional vector quantization (VQ) schemes encode image blocks as VQ indices, in which there is significant similarity between the image block and the codeword of the VQ index. Thus, the method can compress an image and maintain good image quality. This paper proposes a novel lossless VQ indices compression algorithm to further compress the VQ index table. Our scheme exploits the high correlation of adjacent image blocks to search for the same VQ index with the current encoding index from the neighboring indices. To increase compression efficiency, codewords in the codebook are sorted according to the degree of similarity of adjacent VQ indices to generate a state codebook to find the same index with the current encoding index. Note that the repetition indices both on the search path and in the state codebooks are excluded to increase the possibility for matching the current encoding index. Experimental results illustrated the superiority of our scheme over other compression schemes in the index domain.

Keywords: Lossless compression, vector quantization, compression efficiency

1. Introduction

Vector quantization (VQ) [1] is a popular lossy image compression algorithm, which is used extensively in the fields of data transmission and storage due to its high compression efficiency and decoding simplicity. Before encoding a digital image, the VQ encoder divides several images into some $n \times n$ blocks and trains a codebook consisting of the most representative codewords, which are calculated by using an LBG algorithm [1]. In the encoding phase, each block of the input image is individually replaced by the index of the closest codewords in the codebook. In the decoding phase, the VQ indices can be directly mapped to the corresponding codewords, thereby restructuring a similar image.

When the size of the image block is small, the similarity between the current image block and its neighboring blocks is high. Thus, the current VQ index can be found easily from neighboring VQ indices. Based on this characteristic, if the i^{th} neighboring VQ index is the same as the current VQ index, then the ID number i is used to represent the current VQ index, thereby reducing the length of the index. In the decoding phase, the original VQ index can be derived by mapping the i^{th} neighboring VQ index. Finite-state VQ (FSVQ) methods [2-3] encode the current indices by using the state codebook of previously-processed indices, where the size of the state codebook is much smaller than that of the original VQ codebook. Thus, the bit rate of this method is lower than that of the traditional VQ compression method. However, this method incurs image distortion. In 1990, Nasrabadi and King [4] proposed an address VQ (AVQ) compression method, which uses the correlation of four adjacent blocks and constructs an address codebook. This method doesn't result in extra image distortion and significantly decreases the bit rate. However, the method has a high computational cost.

In 1996, Hsieh and Tsai [5] proposed a search-order coding (SOC) compression method to further encode the VQ index, thereby achieving better compression performance. The SOC method searches the same VQ index from previously-processed indices and encodes the current VQ index by the search-order code. The SOC scheme improves compression efficiency and is simple to use.

In 2009, Chang *et al.* [6] proposed a further improvement in the bit rate by using principle component analysis (PCA) to sort the codewords in the codebook to minimize the distance between neighboring indices. This new scheme is superior to both the SOC and VQ methods, but it increases computational cost. In 2013, Wang *et al.* [7] proposed an adjoining state codebook mapping (ASCM) scheme that can map the VQ index to the state codebooks of two adjacent image blocks, thereby reducing the length of the VQ index.

In this paper, we propose an SOC-based encoding scheme to further compress the VQ index table without extra image distortion. This method first searches the same VQ index from previously-processed indices, and the indices on the search path are relatively similar to the current processing index even though they are not the same. Furthermore, the state codebooks of these indices on the search path are generated, and they consist of indices of the most similar codewords in the codebook. Therefore, there is a high probability of finding the same VQ index with the current processing index from the indices of the state codebook. In addition, the repetition VQ indices on the search path and in the state codebooks are excluded to increase the possibility of finding the same index. The proposed method searches the same index from the neighboring indices and from the state codebooks of indices on the search path, thereby fully utilizing the correlation of neighboring image blocks and achieving the purpose of the VQ index compression.

The rest of this paper is organized as follows. Section 2 reviews three compression algorithms, i.e., VQ, SOC, and ASCM. The proposed scheme is presented in Section 3. Section 4 shows the experimental results and compares them with the SOC method, Chang *et al.*'s scheme, and the ASCM method. Finally, our conclusions are presented in Section 5.

2. Related Work

In this section, we introduce three compression algorithms, i.e., VQ, SOC, and ASCM, in Subsections 2.1, 2.2, and 2.3, respectively.

2.1 VQ Algorithm

The VQ encoding method, which was proposed by Linde *et al.* [1] in 1980, is a popular lossy image compression algorithm. **Fig. 1** illustrates the process of VQ encoding and decoding. In this method, an image to be encoded is first divided into several non-overlapping $n \times n$ blocks, where n is a pre-determined value. The greater the size of the image block, the lower the compression rate becomes. In the compression phase, the VQ encoder maps each image block to the most similar codeword in a pre-trained codebook and outputs the index of the codeword as the compression code. The VQ decoder can construct an image by mapping the VQ indices to their corresponding codewords in the same codebook. The restructured image is very similar to the original image because it consists of the most similar codewords.

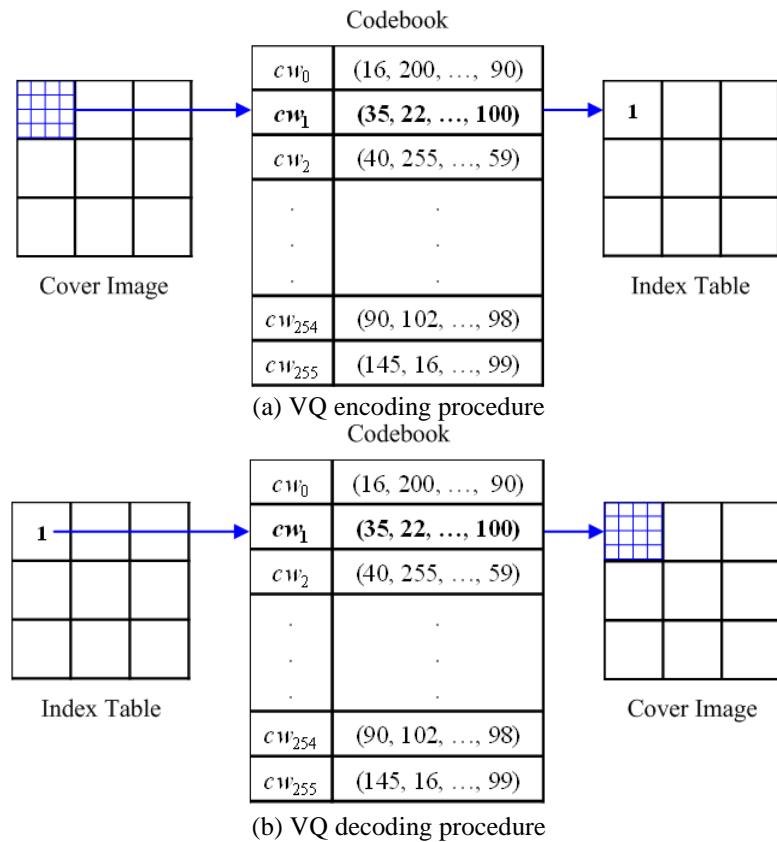


Fig. 1. Flowchart of VQ method

2.2 SOC Algorithm

The SOC algorithm, which was proposed by Hsieh and Tsai in 1996 [5], is an image compression technique that reduces the bit rate of the VQ index table without causing any distortion. The SOC method searches the same index with the current encoding index from the neighboring indices and replaces the current VQ index with the corresponding search-order code. Thus, efficiently searching for the same VQ index from neighboring indices and generating the short SOC code has become a significant issue for the SOC algorithm.

The SOC algorithm processes the VQ index table in a raster scan order, i.e., from left to right and top to bottom. The current encoding index denotes a search center (SC), and the starting search point (SSP) is pre-determined. In **Fig. 2**, the black dot represents the SC, and 1, 2, 3, and 4 denote different SSPs. The encoder attempts to find a match index of SC. **Fig. 3** shows two different search orders, i.e., $\text{SSP} = 1$ and $\text{SSP} = 4$. Since VQ indices are encoded in a raster scan order, the solid boundary squares encoded before SC are chosen as search points (SPs), and the dotted boundary squares encoded after SC are not considered on a search path as non-search points. The encoder will search all SPs that appear on the search path, i.e., 1st level, 2nd level, ..., until an SP is matched or all of the SPs are excluded. If an SP is matched, then the search-order code is sent to the decoder. Otherwise, if all SPs are not matched with the SC, then the original index value (OIV) of SC is transmitted to the decoder. In addition, an indicator bit is added into the front of the SOC and OIV to discriminate between them.

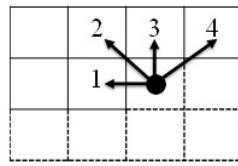


Fig. 2. Search directions

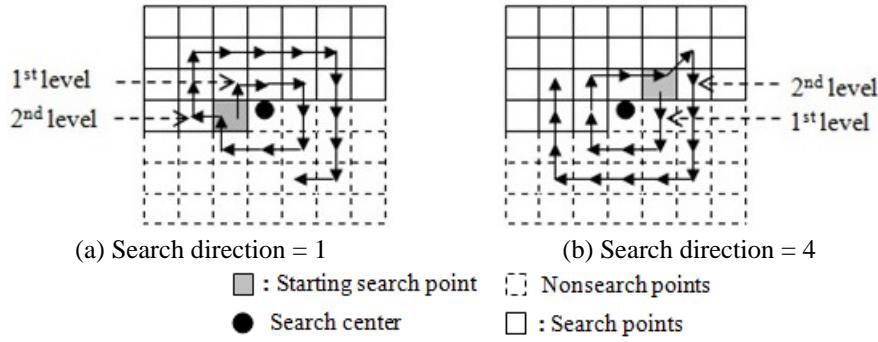


Fig. 3. Search path

The SOC algorithm excludes the repetition points to increase coding efficiency during the encoding phase. A repetition point represents an SP that has the same index value as the previous SP on the search path. For example, in **Fig. 4**, the index of SC is 16, and $\text{SSP} = 1$. The repetition points are located at the coordinates (2, 2), (3, 2), and (1, 3) because their VQ index values are all 18. These repetition points cause redundant searches and thereby lower the coding efficiency. Therefore, if an SP is a repetition point, it will be neglected by the encoder. The exclusion of repetition points increases the possibility of finding an SP with the same index value as SC because the bit number n of the search-order codes is limited. If we let the bit number n to be 2, three SPs, located at the coordinates (3, 2), (2, 4), and (2, 1), will be

searched. Since the index value of the 3rd SP is matched with the SC, the search-order code for the SC is “10.”

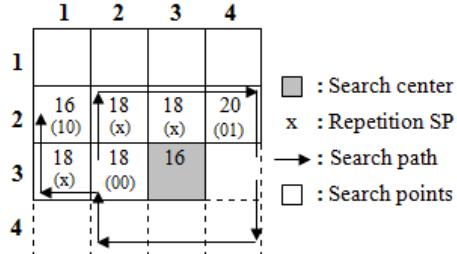


Fig. 4. Example of excluding repetition points

In the decoding phase, the indicator bit positioned in the front of the compression codes is used to distinguish between the SOC index and the OIV. Then, the decoder performs the searching procedure with the same parameters SSP and n that were used in the encoding phase to generate the search-order codes and extract the corresponding VQ index. After recovering the VQ index table, the image can be reconstructed by the traditional VQ decoder.

2.3 ASCM Algorithm

In 2013, Wang *et al.* [7] proposed an ASCM compression scheme that encodes the VQ index table by utilizing the high similarity of neighboring image blocks. High similarity means that the index of a block to encode may appear in the state codebooks of neighboring image blocks. The codewords of the original VQ codebook are sorted according to their degree of similarity to the upper block, and the indices of the first n codewords of the sorted codebook are put into the state codebook of the upper block (SCUB). The state codebook of the left block (SCLB) is generated using the same procedure that was used to generate the SCUB. Note that the SCLB excludes some codewords that appear in the SCUB.

Fig. 5 shows an encoding example of the ASCM method. Assume that the VQ index of block A is 73, the VQ index of upper block B is 65, and the VQ index of block C is 80. First, SCUB is generated, which consists of four VQ indices, i.e., 65, 70, 75, and 81. Then, SCLB is generated, and it also is composed of four indices, i.e., 80, 81, 70, and 67. However, VQ indices 81 and 70 appear in SCUB and SCLB, thus SCLB excludes them and adds two indices of the closest codewords, i.e., 69 and 73. Consequently, SCLB is composed of 80, 67, 69, and 73. Since the VQ index of the encoding block A is mapped to the 3rd index of SCLB, the output compression codes $(111)_2$ are sent to the decoder. The first bit “1” denotes that block A is encoded by the ASCM method. The second bit “1” indicates that the VQ index of block A can be mapped to SCLB. The last two bits “11” indicate that the block A is mapped to the 3rd index of SCLB.

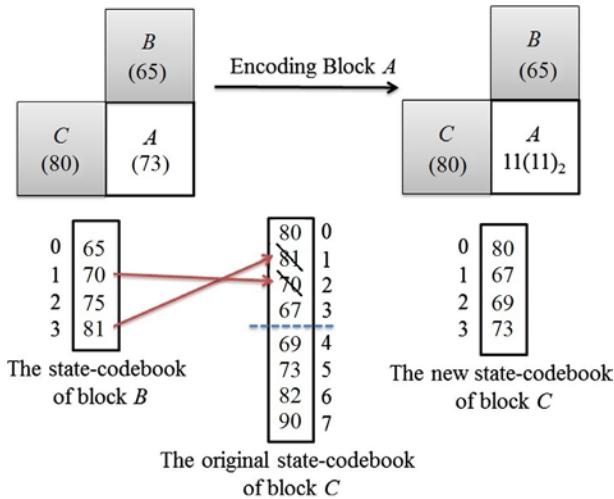


Fig. 5. Example of the ASCM method's generation of state codebooks

3. Proposed Scheme

The SOC algorithm just searches the same VQ index with the current encoding index from adjacent indices, while the ASCM algorithm utilizes state codebooks of the upper and the left image blocks of the current encoding image block to compress VQ indices. Both the SOC and ASCM algorithms utilize the high correlation of neighboring indices to further improve compression efficiency. To make full use of the correlation of adjacent indices, we combined the SOC algorithm and the ASCM algorithm to further decrease the bit rate. In our scheme, the SOC algorithm is first performed and the SOC indicator bit “0” plus the SOC index are sent to the decoder if any SP is matched with the current encoding index. Otherwise, state codebooks of indices on the search path are generated, and the state codebook indicator bits “10” plus the state codebook index (SCI) are sent to the decoder if the current encoding index is matched in any state codebook. Finally, if the current encoding index cannot be encoded by the above two compression methods, the non-compressible indicator bits “11” plus the original index value (OIV) are sent to the decoder. By searching the same index in the neighboring indices and in the state codebooks of the indices on the search path, the proposed scheme achieves a higher possibility of finding the same index, thus decreasing the bit rate.

Our proposed method processes VQ indices of the index table in a raster scan order. All indices are encoded by three kinds of compression codes, that is, SOC index, SOC index plus SCI, and OIV. To encode a VQ index, our proposed scheme first executes the searching procedure of the SOC algorithm to find the same index value. If any SP is matched with the current encoding index, the SOC indicator bit “0” plus the SOC index are output. Even if no SP is matched, the indices of SPs on the search path are similar to the encoding index. Therefore, we generate state codebooks of these indices on the search path and try to find the same index with the current encoding index in these state codebooks. The sizes of the state codebooks are much smaller than that of the original codebook generated by the traditional VQ algorithm, thus using the index in the state codebook can achieve higher compression performance. Note that codewords in these state codebooks are sorted by their degree of similarity to the corresponding index. Once the current processing index appears in the state codebook, the indicator bits “10” and the SOC index plus the SCI are output. In addition, a lookup table of all Euclidean distances between each two codewords in the original VQ

codebook is generated in advance, and state codebooks can be generated in real time. If the index is not matched on the search path and in the state codebooks, the non-compressible indicator “11” and its OIV are output. The indicator bit should be added to the front of the compression codes of each index so the decoder can distinguish them. Subsections 3.1 and 3.2 introduce the proposed encoding and decoding algorithms in detail, respectively.

3.1 Encoding algorithm

Input: A VQ index table T ; codebook C , which consists of M codewords; the number of search-order codes N_1 ; and state codebook size N_2

Output: Compression codes

Step 1: Read a VQ index X from the VQ index table T in raster scan order.

Step 2: Perform the searching procedure of the SOC algorithm, with each SP corresponding to a search-order code N , $N \in [0, N_1 - 1]$. If an SP is matched with the current encoding index X , then output compression codes $0||(N)_2$, where $(N)_2$ consists of $\log_2(N_1)$ bits, and go to Step 6; otherwise, go to Step 3. The compression code consists of $\log_2(N_1 + 2)$ binary bits, where the \log_2 bit is used to represent the indicator “0”. Consequently, the number of bits of the compression code is $\log_2(M - N_1 - 2)$ bits less than that of the original VQ index.

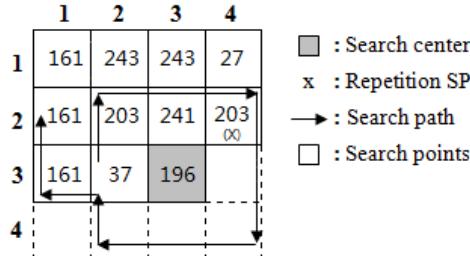
Step 3: For each SP, sort M codewords according to the degree of similarity to the VQ index X .

Step 4: Generate a state codebook consisting of the first N_2 indices of codewords in the sorted codebook. If a codeword with state codebook index SCI is matched with the current encoding index X , then output compression codes $10||(N)_2||(SCI)_2$, where $(SCI)_2$ consists of $\log_2(N_2)$ bits, and go to Step 6; otherwise, go to Step 5. The compression code requires $\log_2(N_1 + N_2 + 4)$ bits for binary representation, where the $\log_2 4$ bits are used to represent the indicator “10”. Therefore, the number of bits of the compression code is $\log_2(M - N_1 - N_2 - 4)$ bits less than that of the original VQ index.

Step 5: The VQ index X cannot be compressed, thus output the non-compressible indicator bits “11” and the original VQ index, i.e., $11||(X)_2$, where $(X)_2$ consists of $\log_2(M)$ bits; continue with Step 6. Since the compression code consists of the two indicator bits and the original VQ index, the number of bits of the compression code is 2 bits more than that of the original VQ index.

Step 6: Repeat Steps 1 to 6 until all VQ indices in the VQ index table T have been processed.

Fig. 6 shows an example of our encoding algorithm. Let the codebook size, the number of search-order codes, and the state codebook size be 256, 4, and 4, respectively. The first encoding index $X = 196$ is at position (3, 3), and its four SPs are (3, 2), (2, 2), (2, 3), and (3, 1) with index values of 37, 203, 241, and 161, respectively. Since no SP is matched, the four state codebooks shown in **Table 1** are generated to determine whether any codeword is matched or not. Obviously, the 4th codeword of the 3rd SP’s state codebook is matched with the current processing index $X = 196$. Therefore, the output compression codes are “101011,” which is two bits less than the original VQ index consisting of eight bits. The first two bits “10” indicate that the current encoding index X is encoded by the state codebook index. The following two bits “10” denote that the index X is mapped to the state codebook of the 3rd SP. The last two bits “11” mean that the VQ index of the fourth index of the corresponding state codebook are the same as the current encoding index X .

**Fig. 6.** Example of generating the state codebook**Table 1.** Four state codebooks of all searching points

(a) State codebook indices and their corresponding VQ indices in the first state codebook generated by VQ index “37”

State codebook index	VQ index
--	37
00	109
01	25
10	195
11	226

(b) State codebook indices and their corresponding VQ indices in the second state codebook generated by VQ index “203”

State codebook index	VQ index
--	203
00	149
--	241
--	25
--	37
01	204
10	67
11	137

(c) State codebook indices and their corresponding VQ indices in the third state codebook generated by VQ index “241”

State codebook index	VQ index
--	241
--	137
--	149
--	203
00	155
01	243
10	99
--	25
--	67
11	196

- (d) State codebook indices and their corresponding VQ indices in the fourth state codebook generated by VQ index “161”

State codebook index	VQ index
--	161
00	120
01	246
10	80
11	105

3.2 Decoding algorithm

Input: Compression codes CC ; codebook C , which consists of M codewords; the number of search-order codes $N1$; and state codebook size $N2$

Output: Original VQ index table

Step 1: Read one bit $b1$ from the compression codes CC . If $b1$ is equal to “0,” it means that the following $\log_2(N1)$ bits are search-order codes. In order to recover the original VQ index X , the $\log_2(N1)$ bits are transformed into the decimal value N , and the searching procedure of the SOC algorithm is performed to find the index value of the N^{th} SP on the search path; if that is the case, go to Step 4; otherwise, go to Step 2.

Step 2: If the extracted bit $b1$ is equal to 1, then read the next bit $b2$ from the compression codes CC . If two bits $\{b1, b2\}$ are equal to “10,” then it means that the following $\log_2(N1+N2)$ bits are compression codes. The $\log_2(N1+N2)$ bits are transformed into two decimal values N and SCI. Then, the SCIth index of the state codebook of the N^{th} index on the search path is extracted to obtain the original VQ index X . After performing the above procedure, go to Step 4; otherwise, go to Step 3.

Step 3: If the two extracted bits $\{b1, b2\}$ are equal to “11,” it means that the following $\log_2(M)$ bits are transformed into a decimal value to obtain the original VQ index X .

Step 4: Repeat Steps 1 to 4 until all of the VQ indices have been recovered.

4. Experiment Results

We compared our proposed scheme with SOC scheme [5], Chang *et al.*’s scheme [6], and ASCM scheme [7]. Fig. 7 shows the nine typical grayscale images that were used as test images. The VQ codebook was generated by the LBG algorithm [1]. The bit rate (BR) was used to evaluate the compression efficiency, and BR was calculated using the following equation:

$$\text{BR} = \frac{\text{size of compression codes}}{\text{size of cover image}} (\text{bpp}). \quad (1)$$

The lower the BR, the shorter the compression codes are, denoting better compression efficiency. The BR value of an effective compression method should be smaller than 1. If the BR value is equal to or greater than 1, it means that the image cannot be compressed. We performed several experiments to test the compression performance of the proposed scheme with different sizes of VQ codebooks and with different sizes of image blocks.



Fig. 7. Nine test images

In our first experiment, the sizes of the image block and the codebook were set to be 4×4 and 256, respectively. For these conditions, the BR value of the VQ compression method was 0.5 bpp. **Fig. 8** shows the lowest BR values of the proposed scheme, SOC scheme [5], Chang *et al.*'s scheme [6], and ASCM scheme [7], where, for each scheme, optimal parameters were used for the different test images. The results showed that the proposed scheme achieved better compression efficiency than any of the other schemes. This is because the proposed method can efficiently compress the original VQ indices as the search-order codes. Also, the uncompressible VQ indices, which are not encoded as search-order codes, are further compressed by the state codebook. Consequently, the number of compressed indices of the proposed method is more than that of the three related compression methods [5-7], confirming that the BR value of the proposed method is lower than that of the related methods.

In the above experiment, we found that the parameters that each scheme used to achieve the lowest bit rates for the different test images were very close. Therefore, we set the same parameter for different test images to test the compression performance of our scheme. **Fig. 9** shows the BR values of the proposed scheme and the three related schemes for each test image with the same parameter. In SOC scheme, n and SSP were set to be 2 and 1, which means that the search-order code was composed of two bits, and the starting search point was 1. In addition, two parameters in Chang *et al.*'s scheme, i and j , were set to be 2 and 2, which implies that the bit length of the state codebook index is $i + j = 4$. Simultaneously, the SCI size of ASCM scheme was set to be 1. In our proposed scheme, the number of search-order codes $N1$ and the state codebook size $N2$ were set to be 4. The proposed method produces shorter compression codes than the three related schemes. In addition, the performance of our proposed scheme with the same parameter was similar to that with the optimal parameter.

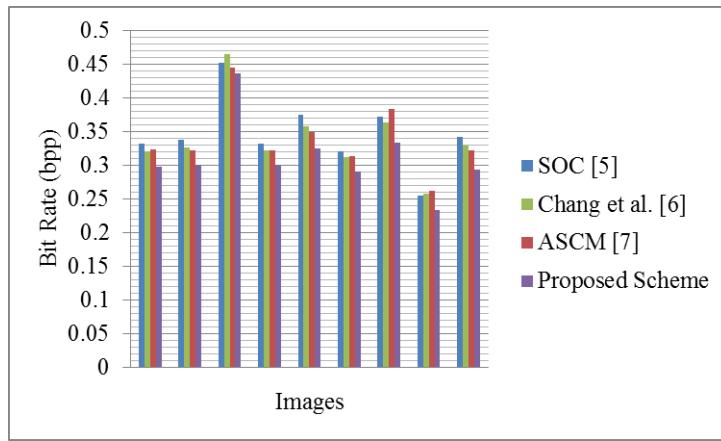


Fig. 8. Bit rates of the proposed scheme and the three related schemes [5-7], with the VQ codebook size of 256 and optimal parameters for different test images

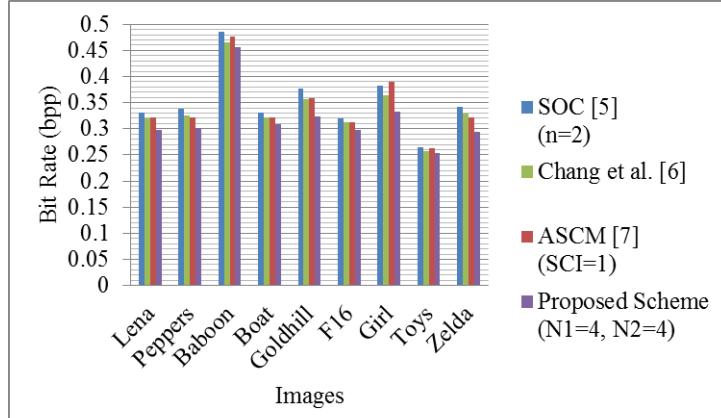


Fig. 9. Bit rates of the proposed scheme and the three related schemes [5-7], with the VQ codebook size of 256 and the same parameters for each test image

The second experiment compared our scheme and the three related schemes with the codebook sized 512 and an image block sized 4×4 . The BR value of the VQ encoding method

is 0.5625 bpp for the condition. In this experiment, we compared the BR values of each scheme with the optimal parameters and the same parameters, where the optimal parameters were obtained from vast experimental results. **Fig. 10** shows that our proposed scheme achieved a lower BR value than the other schemes, irrespective of the parameters. In addition, our BR values were much smaller than 0.5625 bpp, which means that our compression method significantly outperformed the traditional VQ compression method. This is because most VQ indices can be compressed as an SOC index or SCI by our method, thereby decreasing the bit rate.

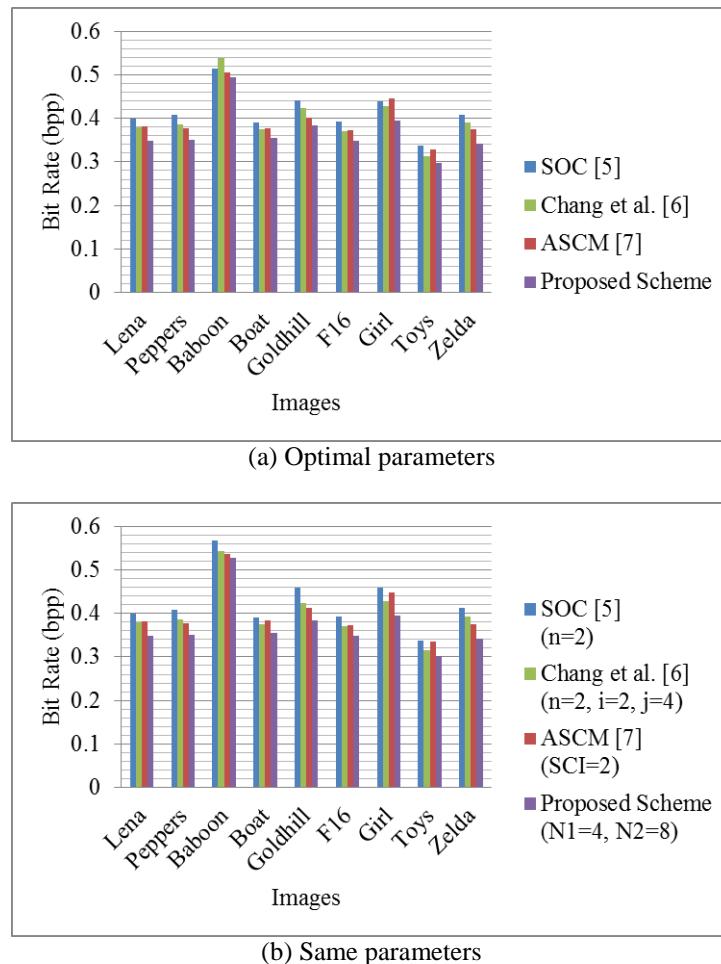


Fig. 10. Bit rate values of the proposed scheme and the three related schemes [5-7], in the condition of a VQ codebook sized 512

In the third experiment, the size of the image block was set to be 8×8 and 16×16 , and the codebook's size remain unchanged, i.e., $N1 = 512$. In the cases of two different block sizes, the BR values of the traditional VQ compression method were 0.1406 and 0.0352 bpp, respectively. **Fig. 11** shows the BR values of the proposed scheme and the three related schemes with the optimal parameters and the same parameter, where the image block size is 8×8 . The BR values in **Fig. 11** are lower than those in **Fig. 12**, in which the block size is 16×16 . This is because the large block size caused a larger difference between the current encoding VQ index and its adjacent indices, thereby decreasing the possibility of finding the same VQ

index from neighboring indices. This drawback decreased the number of the compressible VQ indices, thereby increasing the BR value.

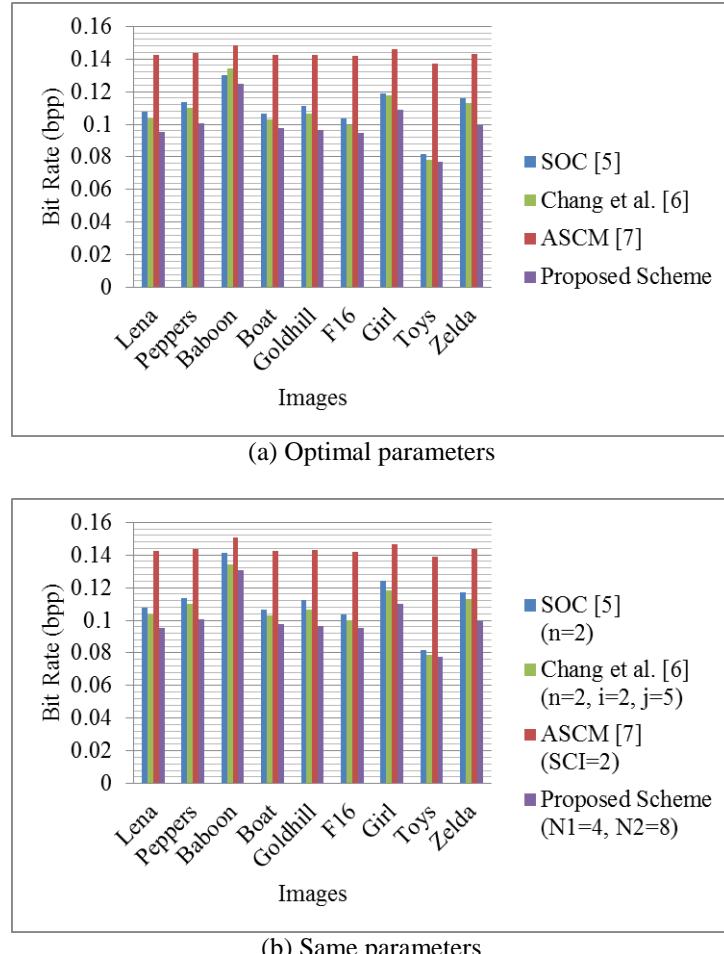
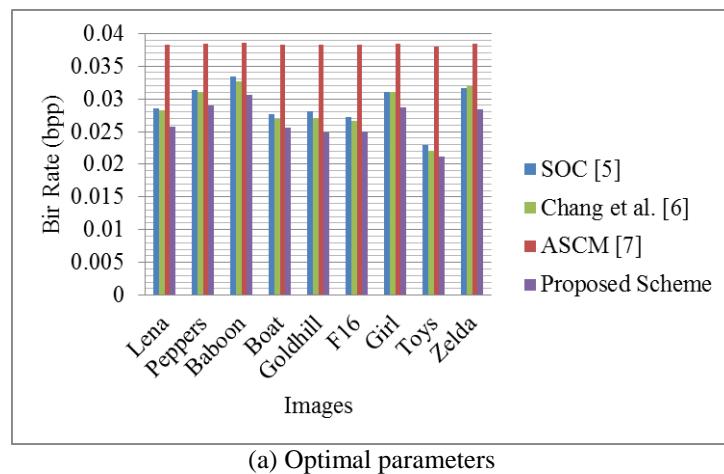
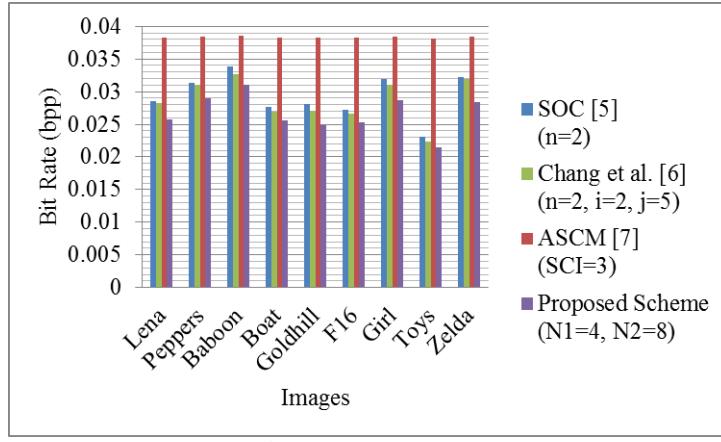


Fig. 11. Bit rate values of the proposed scheme and the three related schemes [5-7] by the VQ codebook sized 512 and the image block sized 8×8





(b) Same parameters

Fig. 12. Bit rate values of the proposed scheme and the three related schemes [5-7] by the VQ codebook sized 512 and the image block sized 16×16

Although the compression efficiency of the proposed method is greater than that of the three compression methods [5-7], the computational cost of the proposed method is more than that of the other three compression methods, as shown in **Fig. 13**. This is because the proposed method combines two related methods [5, 7] to enhance the probability of finding the same VQ index, which led to more computational cost. However, the maximum computation time of the proposed method does not exceed three minutes. Moreover, as the block size increases, the computation time of the proposed method decreases, as shown in **Fig. 14**. This is because the number of VQ indices with the larger image blocks is less than that with smaller image blocks. Consequently, the proposed method takes a smaller amount of time to compress the few VQ indices with large image blocks.

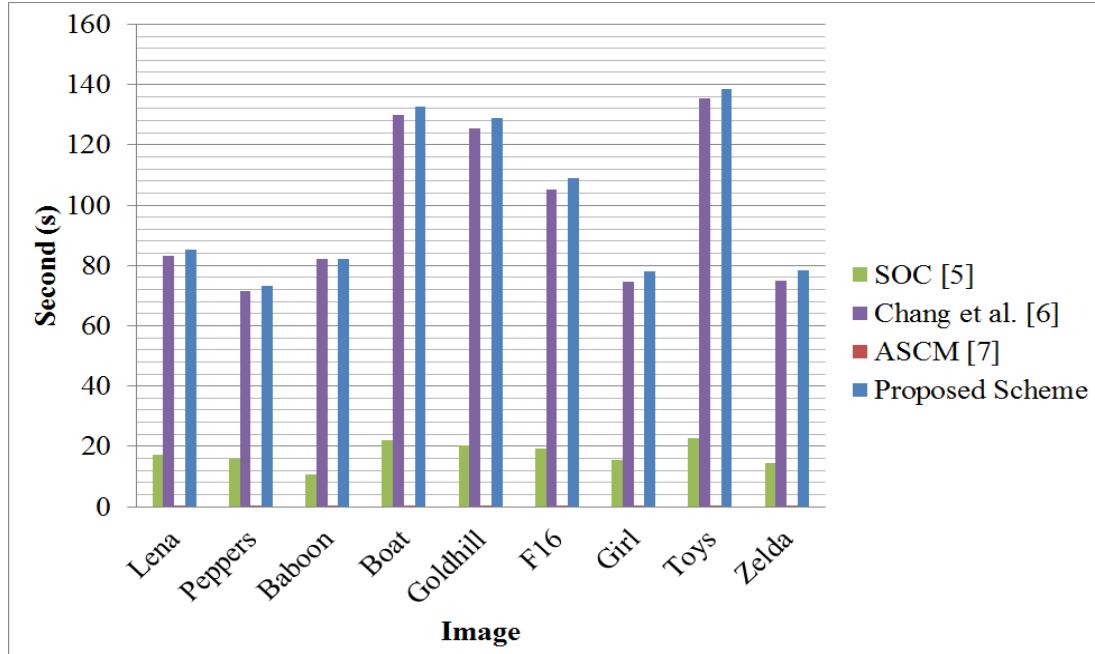


Fig. 13. Comparison of the computation time of the proposed scheme with those of three related schemes [5-7]

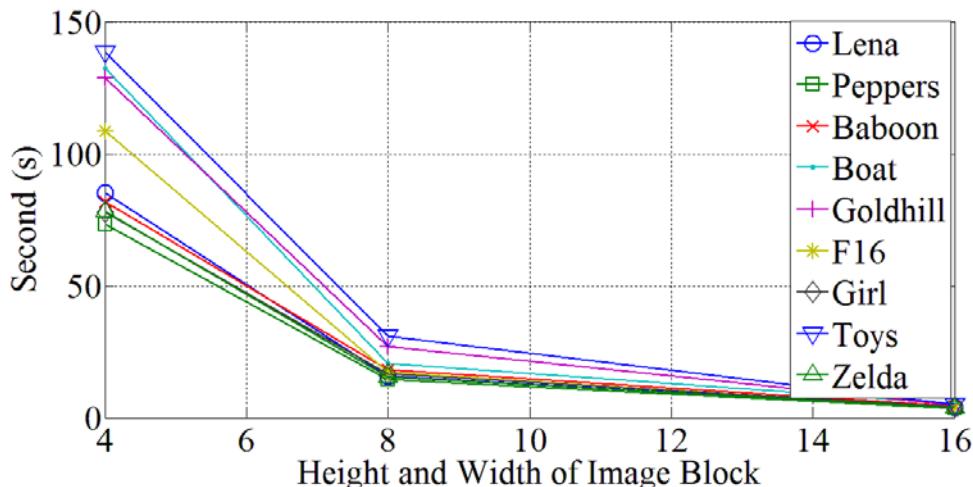


Fig. 14. Computation time of the proposed scheme with different block sizes

5. Conclusions

In this paper, we proposed a lossless VQ index compression scheme that exploits the characteristic of the high similarity of neighboring image blocks to compress VQ indices. Our scheme first uses the SOC algorithm to search the same index with the current encoding VQ index from neighboring indices. If the current encoding VQ index cannot be compressed by the SOC algorithm, then state codebooks consisting of indices of the closest codewords of indices on the search path are used to compress the VQ indices. Thus, most VQ indices can be compressed by our method, thereby decreasing the bit rate. The experimental results demonstrated that our scheme had better compression performance than the VQ scheme, the SOC scheme, the ASCM scheme, and Chang *et al.*'s scheme. In the future, we will try to embed authentication information [8] or secret data [9, 10] into the remaining spaces, thereby achieving the purposes of identification and privacy communication.

References

- [1] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no.1, pp. 84-95, January, 1980.
[Article \(CrossRef Link\)](#).
- [2] J. Foster, R. M. Gray, and M. O. Dunham, "Finite-state vector quantization for waveform coding," *IEEE Transactions on Information Theory*, vol. 31, no. 3, pp. 348-359, May, 1985.
[Article \(CrossRef Link\)](#).
- [3] C. H. Hsieh and J. S. Shue, "Frame adaptive finite-state vector quantization for image sequence coding," *Signal Processing: Image Communication*, vol. 7, no. 1, pp. 13-26, March, 1995.
[Article \(CrossRef Link\)](#).
- [4] N. M. Nasrabadi and Y. Feng, "Image compression using address-vector quantization," *IEEE Transactions on Communications*, vol. 38, no. 12, pp. 2166-2173, December, 1990.
[Article \(CrossRef Link\)](#).
- [5] C. H. Hsieh and J. C. Tsai, "Lossless compression of VQ index with search-order coding," *IEEE Transactions on Image Processing*, vol. 5, no. 11, pp. 1579-1582, November, 1996.
[Article \(CrossRef Link\)](#).

- [6] C. C. Chang, G. C. Chen, and C. C. Lin, "Lossless compression schemes of vector quantization indices using state codebook," *Journal of Software*, vol. 4, no. 4, pp. 274-282, June, 2009. [Article \(CrossRef Link\)](#).
- [7] W. J. Wang, C. T. Huang, C. M. Liu, P. C. Su, and S. J. Wang, "Data embedding for vector quantization image processing on the basis of adjoining state-codebook mapping," *Information Sciences*, vol. 246, pp. 69-82, October, 2013. [Article \(CrossRef Link\)](#).
- [8] C. Qin, C. C. Chang, and T. J. Hsu, "Effective fragile watermarking for image authentication with high-quality recovery capability," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 11, pp. 2941-2956, 2014. [Article \(CrossRef Link\)](#).
- [9] C. Qin, C. C. Chang, and Y. C. Chen, "A novel reversible data hiding scheme for VQ-compressed images using index set construction strategy," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 8, pp. 2027-2041, 2013. [Article \(CrossRef Link\)](#).
- [10] J. Chen, T. S. Chen, W. Hong, G. Horng, H. Y. Wu, and C. W. Shiu, "A new reference pixel prediction for reversible data hiding with reduced location map," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 3, pp. 1105-1118, 2014. [Article \(CrossRef Link\)](#).



Zhi-Hui Wang received the B.S. degree in software engineering in 2004 from the North Eastern University, Shenyang, China. She received her M.S. degree in software engineering in 2007 and the Ph.D. degree in software and theory of computer in 2010, both from the Dalian University of Technology, Dalian, China. Since November 2011, she has been an assistant professor of Dalian University of Technology. Her current research interests include information hiding and image processing.



Hai-Rui Yang received the B.S. degree in software engineering in 2012 from the Dalian University of Technology, Dalian, China. Since September 2012, he has been studying for his M.S. degree in software engineering in Dalian University of Technology, Dalian, China. His current research interests include information hiding and image processing.



Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.



Gwoboa Horng received the B.S. degree in Electrical Engineering from National Taiwan University in 1981, and the M.S. and Ph.D. degrees from the University of Southern California in 1987 and 1992, respectively, all in Computer Science. Since 1992, he has been on the faculty of the Institute of Computer Science at National Chung-Hsing University, Taichung, Taiwan. His current research interests include artificial intelligence, computer security, cryptography, and symbolic computation.



Ying-Hsuan Huang received the M.S. degree in Information Management from Chaoyang University of Technology, Taiwan. He is currently pursuing the Ph.D. degree in Computer Science and Engineering from National Chung Hsing University. His research interests include data hiding, secret sharing and image processing.