A Novel Reversible Data Hiding Scheme for VQ-Compressed Images Using Index Set Construction Strategy

Chuan Qin[1], Chin-Chen Chang[2, 3] and Yen-Chang Chen[2]

[1] School of Optical-Electrical and Computer Engineering,
University of Shanghai for Science and Technology, Shanghai 200093, China
[e-mail: qin@usst.edu.cn]

[2] Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan
[e-mail: alan3c@gmail.com]

[3] Department of Computer Science and Information Engineering,
Asia University, Taichung 41354, Taiwan
[e-mail: alan3c@gmail.com]
*Corresponding author: Chin-Chen Chang

Received May 3, 2013; revised July 2, 2013; accepted July 31, 2013; published August 30, 2013

Abstract

In this paper, we propose a novel reversible data hiding scheme in the index tables of the vector quantization (VQ) compressed images based on index set construction strategy. On the sender side, three index sets are constructed, in which the first set and the second set include the indices with greater and less occurrence numbers in the given VQ index table, respectively. The index values in the index table belonging to the second set are added with prefixes from the third set to eliminate the collision with the two derived mapping sets of the first set, and this operation of adding prefixes has data hiding capability additionally. The main data embedding procedure can be achieved easily by mapping the index values in the first set to the corresponding values in the two derived mapping sets. The same three index sets reconstructed on the receiver side ensure the correctness of secret data extraction and the lossless recovery of index table. Experimental results demonstrate the effectiveness of the proposed scheme.

Keywords: Reversible data hiding, vector quantization, index table, set construction

This research was supported by the Shanghai Municipal Natural Science Foundation (13ZR1428400).

http://dx.doi.org/10.3837/tiis.2013.08.016

1. Introduction

Data hiding is an effective technique for privacy protection and covert communication. As a branch of data hiding [1], reversible data hiding has attracted the interest of many researchers in recent years [2-11]. The concept of reversibility means that the original cover data can be recovered losslessly from its stego version after the embedded secret information has been extracted. In addition to reversibility, hiding capacity and the quality of the stego data are also the main factors that are used to evaluate reversible data hiding schemes. The commonly-used cover medium for data hiding include text, audio, images, and videos. Since digital images are used extensively in Internet communications nowadays, a large number of reversible data hiding schemes have been reported in recent years, and they are based on various image forms, such as gray images [4], color images [5], compressed images [6], and encrypted images [7].

Digital images are usually compressed before transmission to save storage space and communication bandwidth. Therefore, the readily-available compressed codes of digital images can be utilized as the cover data for data hiding. As a well-known data compression algorithm, vector quantization (VQ) can also be used for digital images [12-14]. During the VQ compression process, first, the image is divided into k non-overlapped blocks, each of which consists of n^2 pixels. A VQ codebook that includes Q codewords is established and it is shared by the sender and the receiver. The length of each codeword is n^2. After using Euclidean distance to calculate the similarity between the codewords and each image block, the index of the codeword that has the smallest Euclidean distance with the block is recorded in the VQ index table. Therefore, the output of VQ compression for a given gray image is one VQ index table consisting of k indices. Because only the index of the codeword is stored for each block and each index requires $\log_2 Q$ bits for binary representation, the compression ratio of VQ for the whole image is $(8 \times n^2) / \log_2 Q$. During the decompression process, according to the indices in the VQ index table, the corresponding codewords in the codebook can be used easily to generate the decompressed image. Due to the representativeness of VQ codewords, a high compression ratio can be obtained, and satisfactory visual quality of the decompressed image can be achieved [15]. In this work, we mainly discuss reversible data hiding in the image compressed codes of VQ, i.e., VQ index table.

In the past few years, many schemes have been reported about the techniques of reversible data hiding using VQ index tables [16-24]. An adaptive data hiding method for VQ compressed images was proposed by Du and Hsu in [18] that can vary the embedding process according to the amount of hidden data. In this method, the VQ codebook was divided into two or more sub-codebooks, and the best match in one of the sub-codebooks was determined to embed secret data. However, the hiding capacity of this method is not very high, so, in order to increase the hiding capacity, a VQ-based data hiding scheme that used a codeword-clustering technique was presented in [19]. The secret data were embedded into the VQ index table by a codeword-order-cycle permutation. By this technique, more possibilities and greater flexibility are available to improve the performance of the scheme. Chang *et al*. clustered the VQ codebook into three groups and used the VQ indices in the high-frequency cluster to hide secret data [20]. The other two clusters were used for the recovery that would be conducted later. Because the clustering technique was involved in the embedding processes for [19-20], the computational complexity of these two schemes was relatively high. Inspired by [18-19], Lin *et al*. adjusted the pre-determined distance threshold according to the required hiding capacity and arranged a number of similar codewords in one group to embed the secret sub-message [21]. A reversible data hiding method for VQ-compressed images based on locally-adaptive coding was proposed in [22]. In this method, the index table was compressed in a block-by-block manner, and the secret message was embedded into the VQ indices simultaneously. In [23], Yang and Lin sorted the VQ codebook using the referred counts and then divided the codebook into several clusters. Half of these clusters were used to embed the secret bits. This method can embed more data than [20], but it is more time-consuming. In order to increase hiding capacity further, Yang and Lin replaced the traditional trace for processing the index table with a fractal Hilbert curve, allowing the compression rate to be improved during data embedding by following the curve to process the index table [24].

There are the two most popular algorithms for the enhancement of the performance of VQ, i.e., search-order coding (SOC) and side match vector quantization (SMVQ). SOC algorithm was proposed by Hsieh and Tsai for use in compressing the VQ index table further, thereby achieving better performance of the bit rate by searching nearby identical image blocks in a spiral path [25]. Some steganographic schemes also have been proposed for embedding secret data into the compressed codes of SOC [26-27]. Several investigations of hiding secret data by SMVQ were reported in [28-32]. In 2010, Chen and Chang proposed an SMVQ-based secret-hiding scheme using adaptive index [29]. The weighted, squared Euclidean distance (WSED) was used to increase the probability of SMVQ for a high embedding rate, but this scheme was not reversible. In order to make secret data imperceptible to potential interceptors, Shie and Jiang embedded secret data in the SMVQ compressed codes of the image by using a partially sorted codebook [30], and the original SMVQ-compressed image can be recovered on the receiver side.

Two main issues for the above-mentioned VQ-related schemes require further study and improvement, i.e., 1) the hiding capacities of some schemes are not very satisfactory and 2) some schemes have relatively high computational complexity in the embedding and extraction procedures even though they use some specialized techniques, such as clustering and fractals, to improve their hiding capacities [19-20, 23-24]. Therefore, in this work, we focus on enhancing the hiding capacity compared to previously-reported schemes and attempt to achieve reversible embedding and extraction with lower computational complexity. As a result, an effective reversible data hiding scheme for VQ-compressed images is proposed. The procedures of embedding and extracting data are both conducted in the VQ index tables of the images. On the sender side, for a given VQ index table of an image, three index sets are constructed, and the first and second index sets include the indices with greater and less occurrence numbers in the index table. The index values belonging to the second set are added with the prefixes from the third set to eliminate index collisions, and this prefix adding operation also has data hiding capability. The main data embedding procedure can be achieved easily by mapping the index values in the first set to the corresponding values in the two derived mapping sets. The compact side information that marks the chosen indices in the first index set should be shared by the receiver to guarantee the correct data extraction and lossless recovery of index table.

The rest of the paper is organized as follows. Section 2 describes our proposed reversible data hiding scheme using index set construction strategy. Experimental results and comparisons are given in Section 3, and Section 4 concludes the paper.

2. Proposed Scheme

In the proposed scheme, all index values in the VQ index table of an image are constructed into three sets. The first constructed set is only utilized for secret data embedding with the assistance of the second set, and the third constructed set is mainly used for solving the possible collision between the first and the second index set. The construction procedure of VQ index sets is simple, and the following embedding and extraction procedures can also be implemented with very low computational complexity. Generally, for the index table of one VQ-compressed image, if the occurrence numbers of more than half of the VQ index values are zero, the optimal performance of the hiding capacity can be achieved. The flowchart of the data embedding procedure is shown in **Fig. 1**.

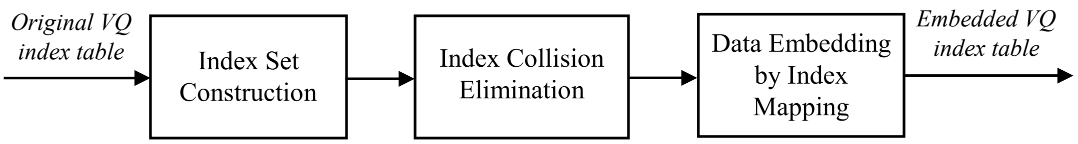

Fig. 1. Flowchart of the data embedding procedure

2.1 Index Set Construction

In the proposed scheme, the sender and the receiver both have the same VQ codebook Ψ with Q codewords, and each codeword length is n^2. Denote the original uncompressed image sized $M \times N$ as $\mathbf{I}$, and divide it into the non-overlapping $n \times n$ blocks. For simplicity, we assume that M and N can be divided by n with no reminder. Denote all k divided blocks in raster-scanning order as $\mathbf{B}_{i,j}$, where $k = M \times N / n^2$, $i = 1, 2, \ldots, M / n$, and $j = 1, 2, \ldots, N / n$.

After VQ compression, the produced index table for the image $\mathbf{I}$ is denoted as $\mathbf{T}$, and all k index values in $\mathbf{T}$ for k blocks belong to $\{0, 1, \ldots, Q-1\}$. We first sort the Q kinds of indices, i.e., $\{0, 1, \ldots, Q-1\}$, in the descending order according to their occurrence numbers in $\mathbf{T}$. Then, the P indices among all Q kinds of indices with greater occurrence numbers, i.e., $\{V_1, V_2, \ldots, V_P\}$, are chosen to generate the first index set, i.e., $\mathbf{\Omega}_1$. Suppose that the indices in $\mathbf{\Omega}_1$ are arranged in the ascending order. The remaining $Q-P$ indices are constructed as the second index set $\mathbf{\Omega}_2$, i.e., $\{V_{P+1}, V_{P+2}, \ldots, V_Q\}$. Each index value in $\mathbf{T}$ must belong to either $\mathbf{\Omega}_1$ or $\mathbf{\Omega}_2$. Denote the corresponding occurrence numbers in $\mathbf{T}$ of the indices in $\mathbf{\Omega}_1 = \{V_1, V_2, \ldots, V_P\}$ and $\mathbf{\Omega}_2 = \{V_{P+1}, V_{P+2}, \ldots, V_Q\}$ as $\{x_1, x_2, \ldots, x_P\}$ and $\{x_{P+1}, x_{P+2}, \ldots, x_Q\}$, respectively, where the minimum of $\{x_1, x_2, \ldots, x_P\}$ must be no smaller than the maximum of $\{x_{P+1}, x_{P+2}, \ldots, x_Q\}$. The third index set $\mathbf{\Omega}_3$ is constructed by L indices, i.e., $\{Q-L, Q-L+1, \ldots, Q-1\}$. Note that the values of Q, P, and L should satisfy the following relationship:

$$P = \lfloor (Q-L)/2 \rfloor, \quad \text{subject to } \min(x_1, x_2, \ldots, x_P) \geq \max(x_{P+1}, x_{P+2}, \ldots, x_Q), \quad (1)$$

where $2 \leq L \leq (Q-2)$ and L is the powers of 2, and $\lfloor \cdot \rfloor$ is the function to obtain the nearest integer in the direction of negative infinity. The third index set $\mathbf{\Omega}_3$ may have some of the same indices with $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$. In order to mark the P chosen indices, i.e., $\{V_1, V_2, \ldots, V_P\}$, in $\mathbf{\Omega}_1$, a vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_Q]$ with Q binary components is created, in which $\mu_{V_i} = 1$ ($i = 1, 2, \ldots, P$) and the other $Q-P$ components in $\boldsymbol{\mu}$ are all zero. This vector $\boldsymbol{\mu}$ should be transmitted to the receiver as the side information after being compressed by entropy encoding. With the assistance of the vector $\boldsymbol{\mu}$, the receiver can easily know which P indices are chosen to produce $\mathbf{\Omega}_1$ by the sender.

2.2 Index Collision Elimination

In the proposed scheme, to enhance security, the secret bits for embedding are scrambled by a secret key initially. Before using the first index set $\mathbf{\Omega}_1$ for data embedding, $\mathbf{\Omega}_1$ is mapped into the other two index sets, i.e., $\mathbf{\Omega}_1'$ and $\mathbf{\Omega}_1''$. For each element V_i (i = 1, 2, …, P) in $\mathbf{\Omega}_1$, the corresponding two mapping elements in $\mathbf{\Omega}_1'$ and $\mathbf{\Omega}_1''$ are $i - 1$ and $i + P - 1$, respectively. Therefore, the P indices in the set $\mathbf{\Omega}_1'$ are $\{0, 1, \ldots, P - 1\}$, and the set $\mathbf{\Omega}_1''$ includes $\{P, P + 1, \ldots, 2P - 1\}$. We can clearly find from Eq. (1) that the two index sets, i.e., $\mathbf{\Omega}_1'$ and $\mathbf{\Omega}_1''$, have no elements in common with the set $\mathbf{\Omega}_3$. However, the index set $\mathbf{\Omega}_2$ may have some same elements with $\mathbf{\Omega}_1'$ and $\mathbf{\Omega}_1''$, respectively. Thus, in order to avoid the future index collision between the two mapping sets of $\mathbf{\Omega}_1$, i.e., $\mathbf{\Omega}_1'$ and $\mathbf{\Omega}_1''$, with $\mathbf{\Omega}_2$ caused by data embedding, the operation of index collision elimination should be conducted.

The index table $\mathbf{T}$ is traversed in the raster-scanning order. For each scanned index value X in $\mathbf{T}$, if X belongs to the second index set $\mathbf{\Omega}_2$, i.e., $\{V_{P+1}, V_{P+2}, \ldots, V_Q\}$, X should be added a prefix χ to differentiate with the possible same index in $\mathbf{\Omega}_1'$ and $\mathbf{\Omega}_1''$ for collision elimination, and the added prefix is chosen from the L elements in $\mathbf{\Omega}_3$, i.e., $\{Q - L, Q - L + 1, \ldots, Q - 1\}$, see Eq. (2).

$$X \Rightarrow \chi \parallel X, \quad X \in \mathbf{\Omega}_2 \text{ and } \chi \in \mathbf{\Omega}_3, \tag{2}$$

where $\Rightarrow$ denotes that the index value in its left part is changed into its right part in $\mathbf{T}$. In order to increase the efficiency of the index collision elimination, we also realize the function of data embedding in this operation: the added prefix χ from $\mathbf{\Omega}_3$ for each scanned index value X belonging to $\mathbf{\Omega}_2$ depends on the current $\log_2 L$ secret bits for embedding. In other words, if the decimal form of the current $\log_2 L$ secret bits for embedding is d ($0 \le d \le L - 1$), the $(d + 1)$th element, i.e., $Q - L + d$, in $\mathbf{\Omega}_3$ is used as the prefix χ of the index value X. For example, if Q = 256, L = 4, and the current two embedding bits is "11", the 4th element, i.e., 255, is chosen from $\mathbf{\Omega}_3$ = {252, 253, 254, 255} and is used as the prefix of the index value X.

After traversing the index table $\mathbf{T}$, all index values in $\mathbf{T}$ belonging to $\mathbf{\Omega}_2 = \{V_{P+1}, V_{P+2}, \ldots, V_Q\}$ are added with the prefixes, which can be distinguished with the indices in $\mathbf{\Omega}_1'$ and $\mathbf{\Omega}_1''$. Because the occurrence numbers for the $Q - P$ indices $\{V_{P+1}, V_{P+2}, \ldots, V_Q\}$ in $\mathbf{T}$ is $x_{P+1}, x_{P+2}, \ldots, x_Q$, thus, the total number Z of the added prefixes is:

$$Z = \sum_{j=P+1}^{Q} x_j \,. \tag{3}$$

The added prefixes may lead to the enlargement of the index table $\mathbf{T}$, but the occurrence numbers for $\{V_{P+1}, V_{P+2}, \ldots, V_Q\}$, i.e., $x_{P+1}, x_{P+2}, \ldots, x_Q$, are usually smaller or close to zero. Therefore, the enlargement of $\mathbf{T}$ must not be significant. Furthermore, this index collision elimination operation can achieve the additional capability of data hiding. Each original index value in $\mathbf{T}$ belonging to $\mathbf{\Omega}_2$ is embedded with $\log_2 L$ secret bits by adding the prefix. Thus, the hiding capacity C_2 of all index values belonging to $\mathbf{\Omega}_2$ is:

$$C_2 = Z \times \log_2 L = \sum_{j=P+1}^{Q} x_j \times \log_2 L. \tag{4}$$

After the procedures of index set construction and index collision elimination, the main data embedding procedure by the two mapping index sets of $\mathbf{\Omega}_1$, i.e., $\mathbf{\Omega}_1{}'$ and $\mathbf{\Omega}_1{}''$, is conducted, which is described in the next subsection.

2.3 Embedding Procedure

Based on the three constructed index sets, i.e., $\mathbf{\Omega}_1$, $\mathbf{\Omega}_2$, and $\mathbf{\Omega}_3$, the secret bits can be easily embedded into the index table $\mathbf{T}$. The detailed steps are as follows:

Step 1: Traverse the index table $\mathbf{T}$ in the raster-scanning order to search the index values belonging to the first index set $\mathbf{\Omega}_1 = \{V_1, V_2, \ldots, V_P\}$.

Step 2: For each scanned index value Y in $\mathbf{T}$, if Y belongs to the first index set $\mathbf{\Omega}_1$, read one secret bit s sequentially from the secret information for embedding.

Step 3: Suppose that the searched index value Y is equal to V_i ($i = 1, 2, \ldots, P$). If the current embedding secret bit $s = 0$, Y is mapped to the ith element in the set $\mathbf{\Omega}_1{}' = \{0, 1, \ldots, P-1\}$, i.e., $i-1$; otherwise, if $s = 1$, Y is mapped to the ith element in the set $\mathbf{\Omega}_1{}'' = \{P, P+1, \ldots, 2P-1\}$, i.e., $i + P - 1$, see Eq. (5).

$$\begin{array}{ll} Y \Rightarrow i-1 & \text{if } s=0 \\ Y \Rightarrow i+P-1 & \text{if } s=1 \end{array} \quad \text{subject to } Y = V_i \text{ and } i = 1, 2, \ldots, P. \tag{5}$$

Step 4: Substitute the current scanned index value Y belonging to the index set $\mathbf{\Omega}_1$, i.e., V_i, with its mapped value in Eq. (5). Then, go back to *Step 1* for the iterative implementation until all index values in $\mathbf{T}$ belonging to $\mathbf{\Omega}_1$ are traversed.

After the embedding procedure finishes, all original index values in $\mathbf{T}$ belonging to $\mathbf{\Omega}_1$ are conducted index mapping operations, and each of them is embedded with one secret bit. Thus, the hiding capacity C_1 of all index values in $\mathbf{T}$ belonging to $\mathbf{\Omega}_1$ is the summation for $x_1, x_2, \ldots, x_P$. The total hiding capacity C of the proposed scheme can be calculated by Eq. (6):

$$C = C_1 + C_2 = \sum_{i=1}^{P} x_i + \sum_{j=P+1}^{Q} x_j \times \log_2 L. \tag{6}$$

Note that, if $x_{P+1}, x_{P+2}, \ldots, x_Q$ are all equal to 0, it means that the occurrence numbers in $\mathbf{T}$ of all indices belonging to $\mathbf{\Omega}_2$ are all zero, thus, no index collision will appear, and the prefix adding for index collision elimination is not necessary. Furthermore, in this case, the enlargement of the index table will not appear, and the hiding capacity C of our scheme is exactly equal to C_1 ($C_2 = 0$). The embedded index table $\mathbf{T}'$ and the side information of the compressed vector $\boldsymbol{\mu}$ are transmitted to the receiver side for secret information extraction and index table recovery.

2.4 Extraction and Recovery Procedure

The flowchart of the extraction and recovery procedure is shown in **Fig. 2**. Three index sets, i.e., $\mathbf{\Omega}_1$, $\mathbf{\Omega}_2$, and $\mathbf{\Omega}_3$, should be first reconstructed at the receiver side before conducting secret data extraction and index table recovery. According to the locations of the ones in the decompressed binary vector $\boldsymbol{\mu}$, the receiver can know which P indices are chosen in the sender side to form the set $\mathbf{\Omega}_1 = \{V_1, V_2, \ldots, V_P\}$, and its two derived mapping sets, i.e., $\mathbf{\Omega}_1{}'$ and $\mathbf{\Omega}_1{}''$, can also be obtained accurately. After the set $\mathbf{\Omega}_1$ is reconstructed, $\mathbf{\Omega}_2$ can be determined easily by the remaining $Q - P$ indices, i.e., $\{V_{P+1}, V_{P+2}, \ldots, V_Q\}$. The set $\mathbf{\Omega}_3$ includes L fixed indices, i.e., $\{Q - L, Q - L + 1, \ldots, Q - 1\}$.

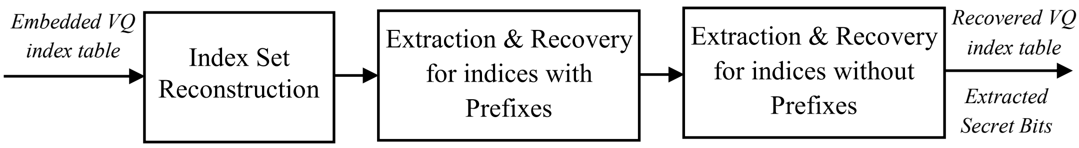

Fig. 2. Flowchart of the data extraction and recovery procedure

After $\mathbf{\Omega}_1$, $\mathbf{\Omega}_2$, and $\mathbf{\Omega}_3$ are reconstructed, secret data extraction and index table recovery can be conducted on the received embedded index table $\mathbf{T}'$. $\mathbf{T}'$ can be divided into two parts: the index values with prefixes and the index values without prefixes. Both of them are carried with secret bits and require to be recovered. The detailed steps are as follows:

Step 1: Traverse the embedded index table $\mathbf{T}'$ in the raster-scanning order to search the index values with prefixes. Denote the current searched index value with the prefix as $\chi \| X$. Note that the prefix χ and the index value X must belong to $\mathbf{\Omega}_3$ and $\mathbf{\Omega}_2$, respectively.

Step 2: Retrieve the prefix χ of the index value X. Since χ belongs to $\mathbf{\Omega}_3 = \{Q - L, Q - L + 1, \ldots, Q - 1\}$, we suppose that χ is equal to $Q - L + d$ $(0 \le d \le L - 1)$. The embedded $\log_2 L$ secret bits s_r ($r = 1, 2, \ldots, \log_2 L$) in $\chi \| X$ can be extracted using Eq. (7).

$$s_r = \left\lfloor \frac{d}{2^{r-1}} \right\rfloor \bmod 2, \quad r = 1, 2, \ldots, \log_2 L. \tag{7}$$

Step 3: Go back to *Step 1* and *Step 2* for the iterative implementation until all Z index values with prefixes in $\mathbf{T}'$ are processed. After that, C_2 secret bits embedded using the index values belonging to $\mathbf{\Omega}_2$ and the added prefixes belonging to $\mathbf{\Omega}_3$ can be extracted, and then, all prefixes are removed to recover all original index values belonging to $\mathbf{\Omega}_2$.

Step 4: Traverse $\mathbf{T}'$ in the raster-scanning order to search the index values without prefixes. Denote the current searched index value without the prefix as Y. Note that the index value Y must belong to either $\mathbf{\Omega}_1' = \{0, 1, \ldots, P - 1\}$ or $\mathbf{\Omega}_1'' = \{P, P + 1, \ldots, 2P - 1\}$.

Step 5: If Y belongs to $\mathbf{\Omega}_1'$, the embedded secret bit "0" is extracted; otherwise, if Y belongs to $\mathbf{\Omega}_1''$, the embedded secret bit "1" is extracted. If Y is equal to $i - 1$ or $i + P - 1$ ($i = 1, 2, \ldots, P$), the index value Y is recovered to V_i that belongs to $\mathbf{\Omega}_1$.

Step 6: Go back to *Step 4* and *Step 5* for the iterative implementation until all index values without prefixes in $\mathbf{T}'$ are processed. After that, C_1 secret bits embedded using the index values belonging to $\mathbf{\Omega}_1$ can be extracted and all original index values belonging to $\mathbf{\Omega}_1$ are recovered.

After the above steps are completed, the receiver can extract all $C_1 + C_2$ embedded secret bits that can then be reversed to the original version by the same scrambling key shared with the sender, and because all original index values belonging to $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are recovered, the original index table $\mathbf{T}$ can be reversed losslessly.

3. Experimental Results and Comparisons

Experiments were conducted on a group of gray-level images with different sizes to verify the effectiveness of the proposed scheme. In our experiment, the size of the divided non-overlapping image blocks for VQ compression was 4×4, i.e., $n = 4$. Thus, the length of each codeword in the used VQ codebooks was 16. We utilized MATLAB 7 to pseudo-randomly generate the binary sequences of different lengths that were used as the secret bits for embedding. All experiments were implemented on a computer with a 3.00 GHz AMD Phenom II processor, 2.00 GB memory, and Windows 7 operating system.

We first present an example to illustrate the detailed procedures of the proposed scheme. Suppose that the size of the used VQ codebook, i.e., Q, was equal to 8, and its indices are {0, 1, 2, 3, 4, 5, 6, 7}. The original index table for a given 16×16 example image is shown in **Fig. 3** (a). The numbers of the indices in the first index set $\boldsymbol{\Omega}_1$ and the third index set $\boldsymbol{\Omega}_3$, i.e., P and L, are set to 3 and 2, respectively, which satisfy the relationship in Eq. (1). The three indices with greater occurrence numbers in the index table of **Fig. 3** (a), i.e., {2, 4, 5} are constructed as $\boldsymbol{\Omega}_1$, and $\boldsymbol{\Omega}_2$ includes the other five indices, i.e., {0, 1, 3, 6, 7} with less occurrence numbers. The occurrence numbers of the indices of $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$ are (3, 4, 3) and (2, 2, 1, 0, 1), respectively. The vector $\boldsymbol{\mu}$ is [0, 0, 1, 0, 1, 1, 0, 0], which means the three indices {2, 4, 5} are chosen to form $\boldsymbol{\Omega}_1$. The third set $\boldsymbol{\Omega}_3$ is constructed by the two indices {6, 7}, which are used as the prefixes for index collision elimination and also represent the embedding bits "0" and "1", individually. Suppose that the secret bits for embedding are "1001010100010101…". Six index values in the index table of **Fig. 3** (a) belonging to $\boldsymbol{\Omega}_2$ = {0, 1, 3, 6, 7} are first added with prefixes from $\boldsymbol{\Omega}_3$ in the raster-scanning order, as shown in **Fig. 3** (b), and the six secret bits "100101" are embedded by the prefixes 6 or 7. The two derived sets of $\boldsymbol{\Omega}_1$, i.e., $\boldsymbol{\Omega}_1'$ and $\boldsymbol{\Omega}_1''$ are {0, 1, 2} and {3, 4, 5}, respectively. Then, the remaining ten index values in the index table belonging to $\boldsymbol{\Omega}_1$ = {2, 4, 5} are mapped into the corresponding indices in $\boldsymbol{\Omega}_1'$ and $\boldsymbol{\Omega}_1''$ according to the following ten secret bits "0100010101" for embedding, and the final embedded index table is produced, see **Fig. 3** (c). The index mappings from $\boldsymbol{\Omega}_1$ to $\boldsymbol{\Omega}_1'$ and $\boldsymbol{\Omega}_1''$ are illustrated in **Fig. 4**. The receiver can first extract the embedded bits "100101" from the six prefixes and removes these prefixes to recover the index values belonging to $\boldsymbol{\Omega}_2$. Then, according to the mapping relationship in **Fig. 4**, the secret bits "0100010101" embedded in the ten index values belonging to $\boldsymbol{\Omega}_1'$ and $\boldsymbol{\Omega}_1''$ can be extracted and these ten index values can be also recovered to the corresponding original index values belonging to $\boldsymbol{\Omega}_1$.

4	5	2	4
4	2	2	3
0	5	5	1
1	4	0	7

(a) original index table

4	5	2	4
4	2	2	7||3
6||0	5	5	6||1
7||1	4	6||0	7||7

(b) after collision elimination

1	5	0	1
1	3	0	7||3
6||0	5	2	6||1
7||1	4	6||0	7||7

(c) Embedded index table

Fig. 3. An example of the proposed scheme

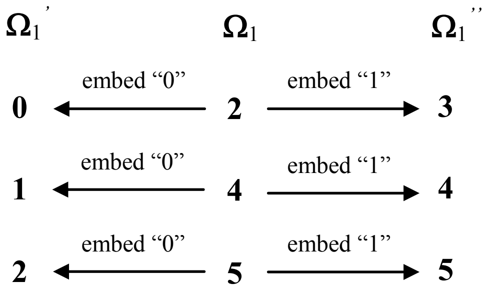

Fig. 4. Illustration of the index mappings from $\mathbf{\Omega}_1$ to $\mathbf{\Omega}_1'$ and $\mathbf{\Omega}_1''$

The six standard, 512 × 512 images that were used for testing, including *Airplane*, *Goldhill*, *Lena*, *Peppers*, *Toys*, and *Zelda*, are shown in **Fig. 5**. We first conducted VQ compression for these six images, and the size of the used codebook, i.e., Q, was equal to 256. The numbers of the indices in the first index set $\mathbf{\Omega}_1$ and the third index set $\mathbf{\Omega}_3$, i.e., P and L, are set to 126 and 4, respectively. During the experiments, we denote the length of the side information $\boldsymbol{\mu}$ after compression as C_e. Because the total number of the embedded secret bits is C, the pure hiding capacity C_p can be defined as:

$$C_p = C - C_e. \tag{8}$$

(a) Airplane (b) Goldhill (c) Lena

(d) Peppers (e) Toys (f) Zelda

Fig. 5. Six standard test images

Table 1 presents the results of the proposed scheme for the six images in **Fig. 5**. The second row in **Table 1** gives the total occurrence numbers of the index values belonging to $\mathbf{\Omega}_1$ in the index table for each image, i.e., the summation for $x_1, x_2, \ldots, x_P$, and the third row gives the total occurrence numbers of the index values belonging to $\mathbf{\Omega}_2$ in each index table, i.e., the summation for $x_{P+1}, x_{P+2}, \ldots, x_Q$. We can observe that, for the most natural images, the summation for $x_{P+1}, x_{P+2}, \ldots, x_Q$ is significantly smaller than the summation for $x_1, x_2, \ldots, x_P$, thus, the enlargement of the index table due to index collision elimination must not be significant. The last two rows are the total hiding capacity C and the pure hiding capacity C_p for each image. Since the VQ index table of the image sized 512×512 has 16384 index values, we can find from **Table 1** that, in the proposed scheme, each index value can carry more than one secret bit averagely. We also evaluated the execution time of the proposed scheme. **Table 2** presents the execution time consumed by the embedding procedure and the extraction and recovery procedure of our scheme for the six test images in **Fig. 5**. It can be seen from **Table 2** that the execution time of the proposed scheme for each image is less than 1.5 seconds averagely, which demonstrates our scheme has satisfactory efficiency and practicability in the real-time applications.

Table 1. Results of hiding capacity for the images in Fig. 5 under P = 126 and L = 4

Images	Airplane	Goldhill	Lena	Peppers	Toys	Zelda
$\Sigma(x_1, x_2, \ldots, x_P)$	15789	15961	16058	15706	15888	16383
$\Sigma(x_{P+1}, x_{P+2}, \ldots, x_Q)$	595	423	326	678	496	1
C	16979	16807	16710	17062	16880	16385
C_p	16955	16784	16689	17040	16857	16363

Table 2. Execution time of the proposed scheme for the images in Fig. 5 (unit: second)

Execution Time	Airplane	Goldhill	Lena	Peppers	Toys	Zelda
Embedding	1.320720	1.341069	1.300354	1.464001	1.306655	1.147342
Extraction & Recovery	0.095777	0.095252	0.092906	0.100524	0.095910	0.085565
Total	1.416497	1.436321	1.393260	1.564525	1.402565	1.232907

We compared the hiding capacity performance of our scheme with four recently reported schemes, i.e., Chang *et al*.'s scheme [20], Chang *et al*.'s scheme [22], Yang and Lin's scheme [23], and Yang and Lin's scheme [24]. During the comparisons, besides the codebook with 256 codewords (Q = 256), we also used the codebook with 512 codewords (Q = 512) for evaluation. Note that the number of the indices in the third index set $\mathbf{\Omega}_3$, i.e., L, is set to 4 for both two cases, i.e., Q = 256 and Q = 512. Thus, the corresponding two numbers of the indices in the first index set $\mathbf{\Omega}_1$, i.e., P, can be calculated by Eq. (1), i.e., 126 and 254. **Tables 3-4** present the comparison results of hiding capacity of the five schemes for the six standard images in **Fig. 5**. Due to the good performance of the index set construction strategy, it can be seen from **Tables 3-4** that the proposed scheme can achieve higher pure hiding capacity than the other schemes in [20, 22-24]. Furthermore, in the proposed scheme, because the data embedding and extraction procedures can be efficiently implemented by the simple operation of index mapping, the computational complexity of the proposed scheme is also lower than the other schemes using more complex operations, such as clustering and fractal [19-20, 23-24].

Table 3. Comparison results of hiding capacity under the codebook size Q = 256

Schemes	Pure Hiding Capacity C_p (bits)					
	Airplane	Goldhill	Lena	Peppers	Toys	Zelda
Chang *et al.*'s Scheme [20]	13163	13209	12016	13248	14316	12012
Chang *et al.*'s Scheme [22]	16222	16224	16219	16223	16225	16248
Yang and Lin's Scheme [23]	15457	14402	15405	15668	15961	15735
Yang and Lin's Scheme [24]	16222	16226	16219	16223	16225	16248
Proposed Scheme	16955	16784	16689	17040	16857	16363

Table 4. Comparison results of hiding capacity under the codebook size Q = 512

Schemes	Pure Hiding Capacity C_p (bits)					
	Airplane	Goldhill	Lena	Peppers	Toys	Zelda
Chang *et al.*'s Scheme [20]	12304	12527	11313	11296	13730	10753
Chang *et al.*'s Scheme [22]	16130	16135	16133	16130	16138	16180
Yang and Lin's Scheme [23]	14980	13258	14990	15302	15634	15321
Yang and Lin's Scheme [24]	16130	16132	16133	16130	16138	16180
Proposed Scheme	16836	16618	16614	16935	16702	16357

5. Conclusion

In this work, we propose a reversible data hiding scheme for VQ compressed images using index set construction strategy. The VQ indices with greater and less occurrence numbers in the index table are utilized to construct the first set and the second set, respectively. In order to avoid index collision, the index values belonging to the second set are added with the prefixes from the third set. Since the occurrence numbers of the index values belonging to the second set are smaller, the degree of the index table enlargement caused by the prefix adding is not significant. Furthermore, the added prefixes for index collision elimination can also achieve the capability of secret data embedding. On the other hand, by using the greater number of index values belonging to the first set, larger amounts of secret data can be embedded efficiently through mapping the elements in the first set into those in its two corresponding derived sets. Therefore, all index values in the index table contribute to the total hiding capacity of the proposed scheme. The same three index sets can be reconstructed on the receiver side, which guarantees the correctness of secret data extraction and the lossless recovery of index table. Experimental results show that the proposed scheme has better performance of the hiding capacity than the recent reported schemes.

References

[1] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information hiding — a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062-1078, 1999. Article (CrossRef Link).

[2] Z. C. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, 2006. Article (CrossRef Link).

[3] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, 2003. Article (CrossRef Link).

[4] S. W Weng, S. C. Chu, N. Cai and R. X. Zhan, "Invariability of mean value based reversible watermarking," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 2, pp. 90-98, 2013. Article (CrossRef Link).

[5] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147-1156, 2004. Article (CrossRef Link).

[6] C. C. Chang, C. C. Lin, C. S. Tseng and W. L. Tai, "Reversible hiding in DCT-based compressed images," *Information Sciences*, vol. 177, no. 13, pp. 2768-2786, 2007. Article (CrossRef Link).

[7] X. P. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255-258, 2011. Article (CrossRef Link).

[8] W. L. Tai, C. M. Yeh and C. C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906-910, 2009. Article (CrossRef Link).

[9] C. F. Lee, H. L. Chen and H. K. Tso, "Embedding capacity raising in reversible data hiding based on prediction of difference expansion," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1864-1872, 2010. Article (CrossRef Link).

[10] W. Hong and T. S. Chen, "Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism," *Journal of Visual Communication*

and Image Representation, vol. 22, no. 2, pp. 131-140, 2011. Article (CrossRef Link).

[11] C. Qin, C. C. Chang and L. T. Liao, "An adaptive prediction-error expansion oriented reversible information hiding scheme," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2166-2172, 2012. Article (CrossRef Link).

[12] N. M. Nasrabadi and R. King, "Image coding using vector quantization: a review," *IEEE Transactions on Communications*, vol. 36, no. 8, pp. 957-971, 1988. Article (CrossRef Link).

[13] C. C. Chang and W. C. Wu, "Fast planar-oriented ripple search algorithm for hyperspace VQ codebook," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1538-1547, 2007. Article (CrossRef Link).

[14] C. H. Lin and C. Y. Yang, "Multipurpose watermarking based on blind vector quantization (BVQ)," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 3, pp. 236-246, 2011. Article (CrossRef Link).

[15] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992. Article (CrossRef Link).

[16] W. J. Wang, C. T. Huang and S. J. Wang, "VQ applications in steganographic data hiding upon multimedia images," *IEEE Systems Journal*, vol. 5, no. 4, 528-537, 2011. Article (CrossRef Link).

[17] Y. C. Hu, "High-capacity image hiding scheme based on vector quantization," *Pattern Recognition*, vol. 39, no. 9, pp. 1715-1724, 2006. Article (CrossRef Link).

[18] W. C. Du and W. J. Hsu, "Adaptive data hiding based on VQ compressed images," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 150, no. 4, pp. 233-238, 2003. Article (CrossRef Link).

[19] C. C. Chang and W. C. Wu, "Hiding secret data adaptively in vector quantisation index tables," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 153, no. 5, pp. 589-597, 2006. Article (CrossRef Link).

[20] C. C. Chang, W. C. Wu and Y. C. Hu, "Lossless recovery of a VQ index table with embedded secret data," *Journal of Visual Communication and Image Representation*, vol. 18, no. 3, pp. 207-216, 2007. Article (CrossRef Link).

[21] C. C. Lin, S. C. Chen and N. L. Hsueh, "Adaptive embedding techniques for VQ-compressed images," *Information Sciences*, vol. 179, no. 1, pp. 140-149, 2009. Article (CrossRef Link).

[22] C. C. Chang, T. D. Kieu and Y. C. Chou, "Reversible information hiding for VQ indices based on locally adaptive coding," *Journal of Visual Communication and Image Representation*, vol. 20, no. 1, pp. 57-64, 2009. Article (CrossRef Link).

[23] C. H. Yang and Y. C. Lin, "Reversible data hiding of a VQ index table based on referred counts," *Journal of Visual Communication and Image Representation*, vol. 20, no. 6, pp. 399-407, 2009. Article (CrossRef Link).

[24] C. H. Yang and Y. C. Lin, "Fractal curves to improve the reversible data embedding for VQ-indexes based on locally adaptive coding," *Journal of Visual Communication and Image Representation*, vol. 21, no. 4, pp. 334-342, 2010. Article (CrossRef Link).

[25] 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, 1996. Article (CrossRef Link).

[26] C. C. Lee, W. H. Ku and S. Y. Huang, "A new steganographic scheme based on vector quantisation and search-order coding," *IET Image Processing*, vol. 3, no. 4, pp. 243-248, 2009. Article (CrossRef Link).

[27] S. C. Shie and S. D. Lin, "Data hiding based on compressed VQ indices of images," *Computer Standards Interfaces*, vol. 31, no. 6, pp. 1143-1149, 2009. Article (CrossRef Link).

[28] C. C. Chang, W. L. Tai and C. C. Lin, "A reversible data hiding scheme based on side match vector quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1301-1308, 2006. Article (CrossRef Link).

[29] C. C. Chen and C. C. Chang, "High capacity SMVQ-based hiding scheme using adaptive index," *Signal Processing*, vol. 90, no. 7, pp. 2141-2149, 2010. Article (CrossRef Link).

[30] S. C. Shie and J. H. Jiang, "Reversible and high-payload image steganographic scheme based on side-match vector quantization," *Signal Processing*, vol. 92, no. 9, pp. 2332–2338, 2012. Article (CrossRef Link).

[31] C. F. Lee, H. L. Chen and S. H. Lai, "An adaptive data hiding scheme with high embedding capacity and visual image quality based on SMVQ prediction through classification codebooks," *Image and Vision Computing*, vol. 28, no. 8, pp. 1293-1302, 2010. Article (CrossRef Link).

[32] L. S. Chen and J. C. Lin, "Steganography scheme based on side match vector quantization," *Optical Engineering*, vol. 49, no. 3, pp. 0370081-0370087, 2010. Article (CrossRef Link).

Chuan Qin received the B.S. and M.S. degrees in electronic engineering from Hefei University of Technology, Anhui, China, in 2002 and 2005, respectively, and the Ph.D. degree in signal and information processing from Shanghai University, Shanghai, China, in 2008. Since 2008, he has been with the faculty of the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, where he is currently a Lecturer. He also has been with Feng Chia University at Taiwan as a Postdoctoral Researcher from July 2010 to June 2012. His research interests include image processing and multimedia security.

Chin-Chen Chang received his B.S. degree in applied mathematics in 1977 and the M.S. 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. During the academic years of 1980-1983, he was on the faculty of the Department of Computer Engineering at the National Chiao Tung University. From 1983-1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From August 1989 to July 1992, he was the head of, and a professor in, the Institute of Computer Science and Information Engineering at the National Chung Cheng University, Chiayi, Taiwan. From August 1992 to July 1995, he was the dean of the college of Engineering at the same university. From August 1995 to October 1997, he was the provost at the National Chung Cheng University. From September 1996 to October 1997, Dr. Chang was the Acting President at the National Chung Cheng University. From July 1998 to June 2000, he was the director of Advisory Office of the Ministry of Education of Taiwan. From 2002 to 2005, he was a Chair Professor of National Chung Cheng University. 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, data structures, computer cryptography and image processing. He is a fellow of the IEEE.

Yen-Chang Chen received the B.S. degree in information management from Minghsin University of Science and Technology, Taiwan, in 2011. He is currently pursuing the M.S. degree in computer science and engineering from Feng Chia University. Taiwan. He research interests include data hiding and image compression.