INFORMATICA, 2014, Vol. 25, No. 4, 523–540

DOI: http://dx.doi.org/10.15388/Informatica.2014.27

A Reversible Data Hiding Scheme for SMVQ Indices

Chin-Chen CHANG[1,2*], Thai-Son NGUYEN[1,3]

[1]*Department of Information Engineering and Computer Science*
Feng Chia University, Taichung 40724, Taiwan, R.O.C
[2]*Department of Biomedical Imaging and Radiological Science*
China Medical University, Taichung 40402, Taiwan, R.O.C
[3]*Department of Information Technology, Tra Vinh University*
Tra Vinh Province, Vietnam
e-mail: alan3c@gmail.com, thaison@tvu.edu.vn

Received: November 2012; accepted: September 2014

Abstract. Reversible data hiding is a method that can guarantee that the cover image can be reconstructed correctly after the secret message has been extracted. Recently, some reversible data hiding schemes have concentrated on the VQ compression domain. In this paper, we present a new reversible data hiding scheme based on VQ and SMVQ techniques to enhance embedding capacity and compression rate. Experimental results show that our proposed scheme achieves higher embedding capacity and smaller average compression rate than some previous methods. Moreover, our proposed scheme maintains the high level of visual quality of the reconstructed image.

Key words: VQ, image compression, SMVQ, reversible data hiding, embedding capacity.

1. Introduction

With the rapid development of multimedia and the Internet, a massive amount of digital information (i.e., images, video, and audio) is transmitted daily over a public channel, the Internet. This means that transmitted data can be tampered easily or copied by malicious users. Therefore, the protection of transmitted data has become an issue of increasing concern. Generally, data protection techniques can be divided into two approaches: cryptography and data hiding. Cryptography transforms secret data into a meaningless form by using cryptographic algorithms such as DES (Davis, 1978) or RSA (Rivest *et al*., 1978). The meaningless form gives the transmitted data some protection, but it also may raise the suspicions of attackers. By contrast, data hiding hides secret data in meaningful cover media, such as images, videos, audios or texts, before it is transmitted over the Internet. The original nature of the cover media remains even through it carries the secret data such as the 'Hello Kitty' image. Such an image is called a 'stego-image'. This technique prevents attackers from suspecting that the stego-image carries a secret message. Therefore, it is

*Corresponding author.

more difficult for malicious attackers to distinguish whether a media object carries secret data. In other words, the security of the hidden data is guaranteed.

In the past ten years, data hiding has received significant interest due to its characteristics. Many data hiding schemes have been proposed for various goals. Traditional data hiding methods have been used to avoid attracting the unwanted attention of attackers. However, most of data hiding schemes cause distortions in the cover image after the secret data have been extracted, and such schemes are referred to as irreversible data hiding schemes (Wang *et al*., 2000, 2001; Thien and Lin, 2003; Zhang and Wang, 2005; Lee *et al*., 2010a). Reconstruction of the original cover image after the extraction of the secret data is very important in some fields, such as military and medical applications. For this reason, researchers have concentrated their attention on developing distortion-free data hiding techniques, and these are known as reversible data hiding techniques (Tian, 2003; Chang *et al*., 2007a, 2011, 2013; Chen and Huang, 2009; Yang and Lin, 2009; Lee *et al*., 2010b, 2011; Yang *et al*., 2011; Hong and Chen, 2011). In reversible data hiding, the cover image can be reconstructed exactly after the secret data are completely extracted. Typically, cover images that are utilized in data hiding can be classified into three domain types, i.e., the spatial domain, the frequency domain, and the compression domain. To embed secret data in the spatial domain (Tian, 2003; Thien and Lin, 2003; Hong and Chen, 2011), all pixel values of the cover image are modified directly. In the frequency domain (Iwata *et al*., 2004; Chang *et al*., 2007b), the cover image is transformed into coefficients, and these coefficients are shifted to hide the secret message. However, in these two techniques, the larger bandwidth space and storage space are required because of the increased size of cover media files after processing. Recently, to save the storage and bandwidth space of the embedded image, many data hiding schemes for compressed images have been proposed in the literature. The main reason is that the sizes of compressed images are much smaller than the sizes of the original images before and after data embedding. Various compression techniques, i.e., JPEG (Chang *et al*., 2002; Tseng and Chang, 2004), JPEG 2000 (Su and Kuo, 2003), block truncation coding (BTC) (Chang *et al*., 2008), and vector quantization (VQ) (Chang *et al*., 2007a, 2011, 2013, Lu *et al*., 2009; Chen and Huang, 2009; Yang and Lin, 2009; Lee *et al*., 2010a, 2010b; Yang *et al*., 2011), have been applied for embedding data to achieve both a good compression rate and good embedding capacity. Over the last five years, researchers have begun to embed secret information into compression domain. Some researchers have proposed reversible data hiding schemes based on vector quantization (VQ) and side-match vector quantization (SMVQ). Chang *et al*. (2007a) proposed a lossless data embedding scheme by classifying the VQ codebook into three groups. To embed secret data, only the VQ indices located in group with the highest referred counts are used, while the other two groups are used for image reconstruction. Then, Chen and Huang (2009) proposed a new hybrid lossless data hiding method based on combination of dynamic tree-coding scheme (DTCS) and modified search-order coding (MSOC), to embed the secret message into the VQ indices without generating any extra distortion. In the same year, Yang and Lin (2009) introduced a reversible data hiding scheme for VQ images to obtain high embedding capacity and compression rate. In Yang

and Lin's scheme, the VQ codebook is sorted and divided into 2B groups, where B is the size of secret bits that are embedded into each VQ index, and half of the groups is used to hide secret data. To increase embedding capacity, Wang and Lu (2009) introduced a path-optional reversible data hiding scheme without generating any distortion based on VQ joint neighboring coding. Wang and Lu's average embedding rate was about 1.95 bpi, which outperforms than some previous schemes (Chen and Huang, 2009; Yang and Lin, 2009). However, the average compression rate of Wang and Lu's scheme was larger than 0.6 bpp, which is worse than that of previous schemes. Lee *et al*. (2010a) developed a new data hiding scheme, based on SMVQ prediction through classification codebooks, to improve the quality of stego image and the embedding capacity further. In Lee et al.'s scheme, attempts were made to embed secret data into edge blocks and non-sufficient smooth blocks. However, the embedding capacity of this scheme is still low, with an average embedding capacity of 16 308 bits or average embedding rate of about 1 bpi. Then, to increase the embedding capacity, Yang *et al*. (2011) used Huffman-code strategies to encode each index of the VQ index table, and the embedding capacity of this scheme was higher than previous schemes. Even so, the compression performance of Yang et al.'s scheme, an average of approximately 0.56 bpp, was quite a bit lower than that of Yang and Lin's scheme. Thus, Chang *et al*. (2011) designed a new reversible data embedding scheme for the VQ compression domain by using locally-adaptive coding in order to obtain better embedding capacity while maintaining a good compression rate and good visual quality of the reconstructed image. Experimental results produced by Chang et al.'s scheme indicated that both high embedding capacity and good visual quality of the reconstructed image were achieved, when compare with some previous schemes (Chen and Huang, 2009; Yang and Lin, 2009; Lee *et al*., 2010a; Yang *et al*., 2011). However, the compression rate of Chang et al.'s scheme was still lower than that of the traditional VQ technique. Therefore, in this paper, we have presented our new reversible data hiding scheme that was designed based on VQ and SMVQ to further improve embedding capacity and compression rate while keeping embedding distortion low. Experimental results confirmed that our proposed scheme obtains higher embedding capacity and higher embedding rate, while guaranteeing the same visual image quality as that provided of traditional VQ encoding. The rest of this paper is organized as follows. Section 2 provides a review of related works, such as vector quantization (Linde *et al*., 1980) and side-match vector quantization (Kim, 1992). In Section 3, the details of our proposed scheme are discussed. The experimental results and discussions are provided in the Section 4, and our conclusions are presented in Section 5.

2. Related Work

Our proposed scheme is based on vector quantization (VQ) (Linde *et al*., 1980) and side-match vector quantization (SMVQ) (Kim, 1992). Therefore, VQ encoding is demonstrated in Section 2.1, and the SMVQ algorithm is discussed in detail in Section 2.2.

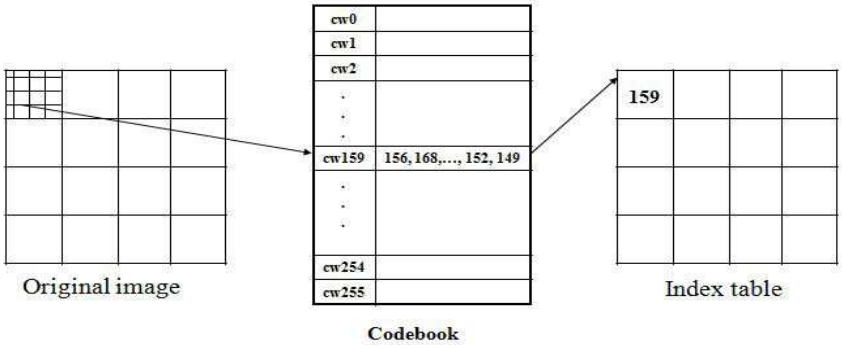

Fig. 1. Example of a VQ algorithm.

2.1. *Vector Quantization*

Figure 1 shows an example of the VQ encoding in which the size of the codebook and the vector dimension are set as 256 and 16, respectively.

In the encoding phase, image blocks with sizes of 4×4 are used as input. Then, the VQ encoder seeks to determine the best-matching codeword, which has the least-squared Euclidian distance to the input block by using Eq. (1). In this case, the codeword with value of 159 is the best match for the first block that is used as input. Then value 159 is sent to the index table as the compression code of the first block.

$$d(x, y_i) = \sum_{j=1}^{k} (x_j - y_{i,j})^2. \tag{1}$$

In the VQ decoding phase, the searching operation is performed to find the corresponding codeword from the codebook used in the encoding phase. Then, the codeword that is found becomes the reconstructed image block output.

2.2. *Side-Match Vector Quantization*

To further improve the compression performance by using the correlation of image blocks in the neighboring area, side-match vector quantization (SMVQ) (Kim, 1992) is used, which is a variation of the VQ algorithm. First, an original image I sized of $W \times H$ is divided into a non-overlapping block with the size of 4×4. It means that $P \times Q$ blocks are generated, where $P = W/4$ and $Q = H/4$. The image blocks in the first row and first column are denoted as seed blocks, and they are encoded by using the traditional VQ technique with super codebook Y sized N, $Y = \{y_i;\ i = 1, 2, \ldots, N\}$. The rest of the blocks, $(P \times Q - P - Q + 1)$ blocks, are defined as residual blocks, and they are used in SMVQ encoding. Each residual block is encoded with the assistance of upper, left neighboring block of the current processing block, as shown in Fig. 2.

To encode the residual block, the SMVQ encoding phase consists of four steps. First, the boundary value of block X is predicted using the values of neighboring values of

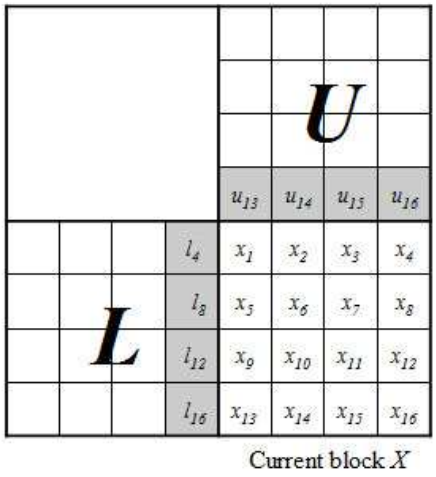

Fig. 2. Current processing block and its upper, left neighboring blocks.

block U and block L, such as $x_1 = (u_{13} + l_4)/2$ and $x_2 = u_{14}$, $x_3 = u_{15}$, $x_4 = u_{16}$, $x_5 = l_8$, $x_9 = l_{12}$, $x_{13} = l_{16}$, respectively. Then, the predicted value is used to look up codewords in super codebook Y to generate the corresponding state codebook. A state codebook contains M codewords that are selected from super codebook Y. These M codewords have the minimum side-match distortion, when compared with the gray areas in Fig. 2. Here, the minimum side-match distortion is calculated by using Eq. (1). After the state codebook has been generated, the current residual block is encoded by the best-match codeword of the current residual block, which is obtained by searching the state codebook instead of the super codebook. Notably, the state codebook contains M selected codewords, which is much smaller than the size of the super codebook. In addition, the state codebook is used to encode a residual block instead of the super codebook. That is, a residual block can be encoded with fewer bits than it could be with VQ, and SMVQ provides a better compression rate than VQ. However, the visual quality of compressed-image encoded by SMVQ will be lower than encoded by VQ. In other words, when compare with super codebook, the smaller the size of state codebook is, the lower visual quality of compressed image will be and vice versa. Equations (2), (3), and (4) were defined to compute the least side-match distortion. In these equations, y represents a codeword, and U and L represent the upper and left neighboring blocks of the current processing block, respectively. Equation (2) defines the upper distortion $UD(y)$ between the codeword y and the upper block U.

$$UD(y) = \sum_{j=13}^{16} (u_j - y_j). \tag{2}$$

Similarly, the left distortion $LD(y)$ of the codeword y and the left block L can be computed by Eq. (3).

$$LD(y) = \sum_{i=1}^{4} (l_{4\times i} - y_{4\times i}). \tag{3}$$

Finally, the side match distortion of the codeword y is calculated according to Eq. (4).

$$SMD(y) = UD(y) + LD(y). \tag{4}$$

In the SMVQ decoding phase, first, the indices in the first row and first column are decoded by VQ with the super codebook as was done in the encoding phase. To reconstruct residual blocks, the previously decoded upper and left neighboring blocks are used to generate a state codebook that contains M codewords with the minimum side-match distortion for the current block. Then, the state codebook is searched fully to find the mapping codeword required for the received index to recover its block. After all received indices have been processed; the reconstruction of the original image is obtained.

3. The Proposed Scheme

In this section, a new reversible data hiding scheme is introduced based on VQ and SMVQ techniques. To provide a better explanation, some notations are provided, such as a grayscale image I is denoted as the cover image with the size of $W \times H$, and the traditional codebook CB is equal to $\{C_0, C_1, \ldots, C_{n-1}\}$. The secret image is defined as $S = (s_1, s_2, \ldots, s_r)$, where $s_i = \{0, 1\}$ and $0 \leqslant i \leqslant r$. Our proposed scheme consists of two phases, the data-embedding phase and the data-extracting phase. These two phases are discussed in Sections 3.1 and 3.2, respectively.

3.1. *Data Embedding Phase*

Initially, cover image I is encoded to generate index table IT by using the traditional VQ algorithm. Then, the SMVQ technique can be used to change index table IT, transforming it into index table IT'. To do this, index table IT will be reconstructed completely, and the size of the state codebook is set to be the same size as the codebook. For each reconstructed image block, the codebook is sorted by using the least side-match distortion between the current reconstructed image block and the codewords in the codebook, as presented in Eq. (1). Then, to generate the state codebook for the current reconstructed image block, all codewords in the sorted codebook are chosen. Finally, the current image block is encoded by the best-match transformed index found in the state codebook. The flowchart in Fig. 3 shows the process involved in generating the transformed index table IT'.

Figure 4 presents the histogram of transformed index table IT' of the image "Lena", when the size of the codebook is 256. It is apparent that the most high-frequency indices of the image "Lena" are distributed around the value of 0. Figure 5 shows the distribution of the high-frequency indices in different test images. Obviously, most of the high-frequency, transformed indices are smaller than 16. Therefore, this study exploits the above-mentioned, dominant feature of the transformed index table to enhance embedding capacity and compression performance. Compared with some previous schemes (Chang *et al*., 2011; Lee *et al*., 2010a; Yang *et al*., 2011; Wang and Lu, 2009) in the VQ compression domain, our proposed scheme achieves both a better embedding capacity and a higher compression rate. Figure 6 presents the flowchart of the data embedding phase.

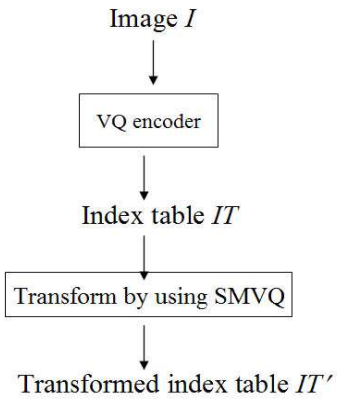

Fig. 3. Flowchart showing the generation of transformed index table IT'.

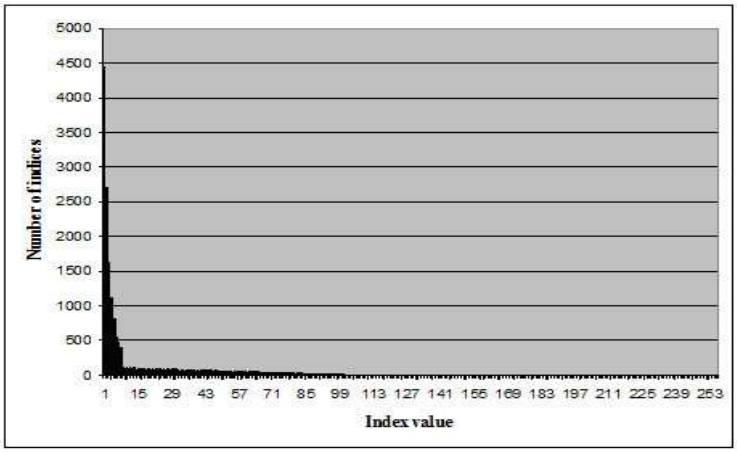

Fig. 4. Histogram of transformed index table of the image "Lena".

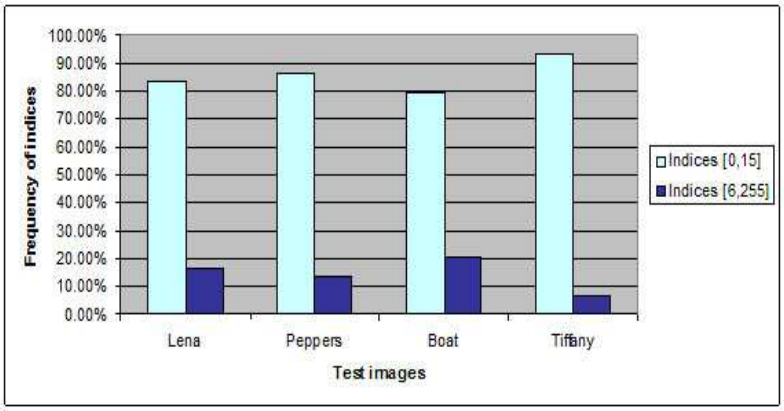

Fig. 5. Distribution of high-frequency indices in different test images.

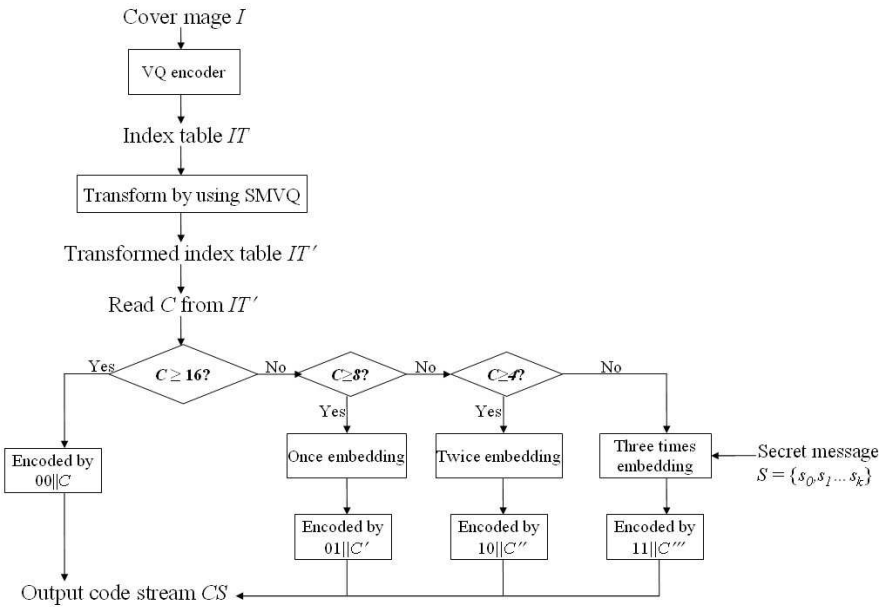

Fig. 6. Flowchart of data embedding phase.

The proposed data embedding phase can be divided into seven steps, which are provided in detail below.

Data embedding algorithm

Input: Grayscale cover image I, super codebook CB, secret message S

Output: Code stream CS in binary form

1. Encode cover image I by applying VQ algorithm to get index table IT.
2. Change index table IT to transformed index table IT' by using the SMVQ method.
3. Read transformed index C from table IT'.
4. If $C \geqslant 16$, C is encoded by indicator $00\|C_2$. Then, $00\|C_2$ is used as output to code stream CS, where C_2 is the 8 bits of binary information of C and $\|$ is defined as the concatenation operation.
5. If C is smaller than 16, the following steps are used to embed the secret bits.
   5.1. If $C \geqslant 8$, encode index C by indicator "01" and the 5-bit binary representation of C', where C' is computed by using Eq. (5) and is used to conceal one secret bit s_i. Then, compression code of C is sent to output code stream CS.

$$C' = 2^j \times C + s, \tag{5}$$

   where j is the number of secret bits are embedded into C, and s is decimal value of j secret bits to be embedded.

   5.2. Otherwise (i.e., $C < 8$), if $C \geqslant 4$, transformed index C is encoded by indicator "10" and the 5-bit binary representation of C'. Equation (5) is used to calculate

C' with $j = 2$, and embed two secret bits, s_i and s_{i+1}, into transformed index C. Here, secret data s is decimal value of two secret bits, s_i and s_{i+1}. For example, when two secret bits, s_i and s_{i+1} are 1 and 1, respectively, the value of s is equals to $(10)_2 = 3$. Finally, compression code of C is concatenated into code stream CS.

5.3. Otherwise (i.e., $C < 4$), three secret bits, s_i, s_{i+1}, and s_{i+2}, are read and embedded into transformed index C. Then, transformed index C is encoded by indicator "11" and followed by 5-bit of C', where C' is also calculated by using Eq. (5) with $j = 3$, and secret data s is decimal value of three secret bits, s_i, s_{i+1}, and s_{i+2}. Then, the compression code is sent to output code stream CS.

6. Repeat Step 3 through Step 5 until all transformed indices in transformed index table IT' have been processed completely.
7. Output the code stream CS.

After the data embedding phase has been processed completely, we can obtain the output code stream CS, which, then, is sent to the receiver. To enhance the security of the embedded secret data, modern cryptographic algorithms, such as DES (Davis, 1978) and RSA (Rivest *et al*., 1978), are also used to encrypt the secret data in advance of the data embedding phase to enhance the security in our scheme. An illustration is provided in Table 1 to clarify the explanation of our data embedding phase. Assume that the transformed index table is {20, 15, 14, 6, 0, 2, 1, 8}, and that secret message S is {11001011100011}. The first transformed index value is $C = 20$, which is larger than 16, so no secret bit is embedded. Therefore, the output compression code of C is "00‖00010100," which is sent to code stream CS. Here, "00010100" is 8 bits of binary information of transformed index C. The second transformed index, $C = 15$ is larger than 8, and secret bit $s_i = 1$ is read from secret message S. According to Step 5(a), the second transformed index is encoded by indicator "01," followed with the 5-bit binary representation of $C' = 2^1 \times 15 + 1 = 31$. Then, compression code "01‖11111" is concatenated into code stream CS. Additional details about the example are provided in Table 1. When all transformed indices in the transformed index have been encoded completely, code stream CS is obtained, and the sender sends CS to the receiver for reverse processing. In our data embedding phase, when the transformed index value is smaller than 16, a secret bit is embedded, and fewer bits are required to encode this transformed index. This allows our scheme to provide a better compression rate. In addition, as shown in Fig. 5, most of the high-frequency transform indices are smaller than 16. It means that our scheme achieves the goal of being able to embed more secret data. Therefore, our proposed scheme has a higher embedding capacity than other schemes.

3.2. *Data Extracting Phase*

In this Subsection, we show our data extracting phase, which is used to extract secret message S and to restore the original transformed index table IT'. Fig. 7 provides the flowchart of our data extracting phase.

The data extracting algorithm is described below.

Table 1
Example of data embedding phase

Input transformed index	Category	Secret bit is embedded	Index value after embedding	Output compression code of transformed index	Applied steps
				Indicator – the remain	
20	No embedding	–	–	00 – 00010100	Step 4
15	Once embedding	1	$2^1 \times 15 + 1 = 31$	01 – 11111	Step 5.1
14	Once embedding	1	$2^1 \times 14 + 1 = 29$	01 – 11101	Step 5.1
6	Twice embedding	00	$2^2 \times 6 + 0 = 24$	10 – 11000	Step 5.2
0	Three times embedding	101	$2^3 \times 0 + 5 = 5$	11 – 00101	Step 5.3
2	Three times embedding	110	$2^3 \times 2 + 6 = 22$	11 – 10110	Step 5.3
1	Three times embedding	001	$2^3 \times 1 + 1 = 9$	11 – 01001	Step 5.3
8	Once embedding	1	$2^1 \times 8 + 1 = 17$	01 – 10001	Step 5.1

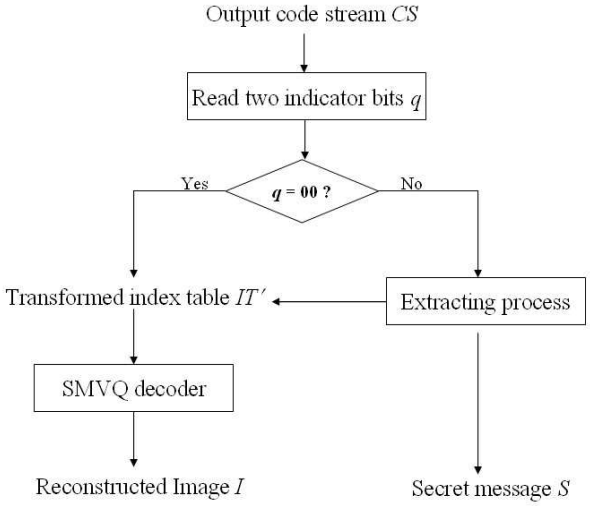

Fig. 7. Flowchart of data extracting phase.

Data extracting algorithm
Input: Code stream *CS* in binary form
Output: Reconstructed image *I*, secret message *S*

1. Secret message S and transformed index table IT' are set as empty.
2. Read two indicator bits, q, from code stream CS.
3. If $q = 00$, then the next 8 bits are read and changed into decimal value C, which is the recovered transformed index. Then, C is sent to transformed index table IT'.
4. If $q = 01$, read the consecutive 5 bits and convert them to the decimal value as C'. Then, transformed index C is restored by using Eq. (6) and the secret data s are extracted by using Eq. (7). Then, the reconstructed transformed index C is inserted to transform index table IT', and secret data s is converted into one bit s_i and con-

Table 2
Example of data extracting phase.

Indicator q	Binary next bits are read	Extracted secret data and secret bits	Reconstructed transformed index	Applied steps
00	00010100	–	20	Step 3
01	11111	$1 = (1)_2$	15	Step 4
01	11101	$1 = (1)_2$	14	Step 4
10	11000	$0 = (00)_2$	6	Step 5
11	00101	$5 = (101)_2$	0	Step 6
11	10110	$6 = (110)_2$	2	Step 6
11	01001	$1 = (001)_2$	1	Step 6
01	10001	$1 = (1)_2$	8	Step 4

catenated into secret message S.

$$C = C'/2^j, \tag{6}$$

$$s = C' \bmod 2^j, \tag{7}$$

where j is the number of bits which are extracted from C'

5. If $q = 01$, the subsequent 5 bits are read from code stream CS and changed into decimal value as C'. Then, the original transformed index value C is reconstructed, and secret data s are extracted by using Eqs. (6) and (7), respectively. The original transformed index C is concatenated into code stream CS, while the two secret bits, s_i and s_{i+1}, are obtained by representing secret data s into two bits and output to secret message S.
6. if $q = 11$, read the next 5 bits and convert them into decimal value as C'. The original transformed index C is reconstructed by using Eq. (6). Then, the secret data s are extracted by using Eq. (7). The reconstructed transformed index value C is sent to transformed index table IT'. Finally, the three extracted secret bits s_i, s_{i+1}, and s_{i+2} are obtained by representing secret data s into three bits and concatenated into secret message S.
7. Repeat Step 2 through Step 6 until all bits in code stream CS have been processed.

Once the data extracting phase is finished completely, secret message S is extracted correctly. In addition, the original transformed index table IT' is also reconstructed exactly. Then, the transformed index table IT' is used to recover original image I by using the SMVQ algorithm. Therefore, the process of re-transforming index table IT' to index table IT is no longer necessary. Table 2 provides a detailed description of the data extracting phase. Assume that the code stream CS is received from the sender. According to our data extracting process, it is obvious that the secret message S, {11001011100011}, is extracted exactly, while the transformed index table IT', {20, 15, 14, 6, 0, 2, 1, 8}, is recovered correctly, as shown in Table 2. In the first row of Table 2, two secret bits $q =$ "00" are read. This means that no secret bit is extracted. Then, according to Step 3, the next 8 bits, "00010100," are read and converted to the decimal value of 20. Thus, the reconstructed transformed index value C is 20. Then, this transformed index is output to

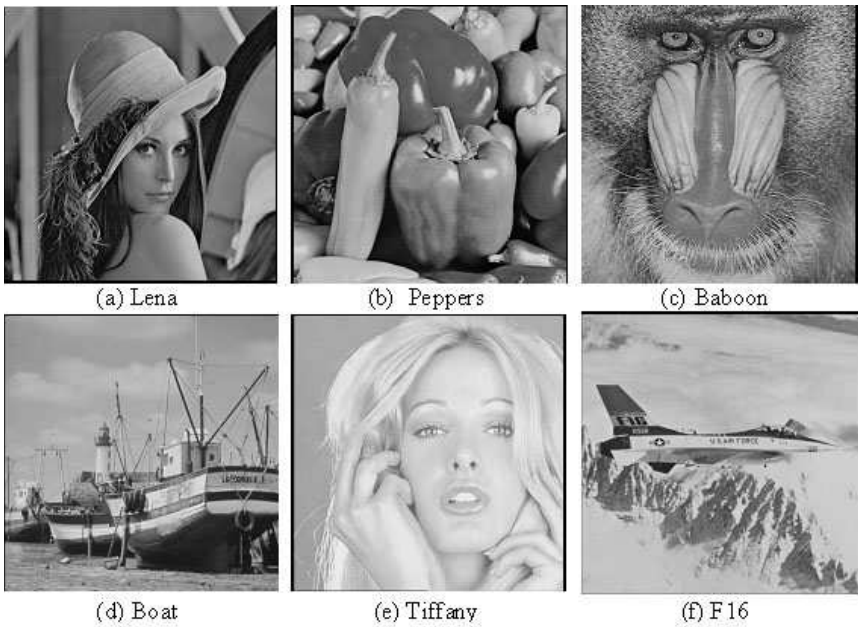

Fig. 8. Flowchart of data extracting phase.

the transformed index table IT'. In the second row of Table 2, two indicator bits q are "01." Hence, the consecutive 5 bits, "11111," are read and changed to decimal value as $C' = 31$. Then, according to Step 4, the secret data, $s = 1$, is extracted and the original transformed index is restored as $C = 15$. Then, index C is sent to transformed index table IT', and one bit s_i is reconstructed by representing s into one bit as $(1)_2$. For the sixth row in Table 2, the indicator q equals "11," which are read from code stream CS. Then the next 5 bits are read as "10110," which are transformed to decimal value as $C' = 22$. Therefore, according to Step 6, the reconstructed transform index value is $C = 2$, which is sent to the transformed index table IT'. The secret data s are extracted as 6. Then, three secret bits, s_i, s_{i+1}, and s_{i+2}, are reconstructed as $(110)_2$. The data extracting phase is complete when all bits in code stream CS are processed completely.

4. Experimental Results

In the experiment, six gray images, i.e., Lena, Peppers, Baboon, Boat, Tiffany, and F16, were tested, as shown in Fig. 8. The size of each test image was 512×512. To utilize VQ and SMVQ encoding, the test image was divided into non-overlapping blocks that were 4×4 pixels in size. A codebook with size 256 was used, which was trained by using the well-known VQ algorithm (Linde *et al.*, 1980) with the six test images mentioned above. In this experiment, Secret message S was in binary form, i.e., 0 and 1, and it is created by using a pseudo-random number generator. To maintain the high level of security, se-

cret message S is encoded in advance using well-known encryption techniques, i.e., DES (Davis, 1978) and RSA (Rivest *et al*., 1978).

To estimate the performance of our scheme compare to that of some previous schemes, we defined four measurements, i.e., embedding capacity (*EC*), compression rate (*CR*), visual quality of reconstructed image, and Pearson's correlation coefficient (*PCC*), are defined in Eqs. (8)–(12). The first measurement is embedding capacity (*EC*), which is introduced to describe how many secret bits can be hidden into one cover image, and defined in Eq. (8). The *EC* parameter is used to compare the embedding capacity of our scheme with that of some previous schemes.

$$EC = \sum_{i=1}^{3} N_i \times i, \tag{8}$$

where N_i is the number of indices are used to embed i bits. The second measurement was the compression rate, known as *CR*, which is defined in Eq. (9). Here, the size of output code stream *CS*, $\|CS\|$, was compared with that of the original cover image I. In the compression domain, a small value of *CR* indicates a good compression. Equation (9) is shown as follows:

$$CR = \frac{\|CS\|}{H \times W} \text{ (bpp)}, \tag{9}$$

where $\|CS\|$ is the size of the output code stream, and H and W are the height and width, respectively, of cover image I. The unit of this parameter is bits per pixel (bpp), which the number of bits are used to encode one pixel. Then, to evaluate the visual quality of the reconstructed image, the peak signal-to-noise ratio (*PSNR*), is defined as shown in Eq. (10).

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right), \tag{10}$$

where the mean square error (MSE) for a $W \times H$ gray-level image is defined as follows.

$$MSE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(I_{ij} - I'_{ij} \right)^2, \tag{11}$$

where W and H are the dimensions of the images, and I_{ij} and I'_{ij} are the pixel values of the cover image and the reconstructed image, respectively. Obviously, in Eq. (10), there is an inverse relationship between *MSE* and *PSNR*. This means that the lower the value of *MSE* becomes, the higher the value of *PSNR* becomes. In principle, higher values of *PSNR* are desirable, because it means that the ratio of the signal to the noise is higher. In this case, by "signal", we mean the original cover image, and, by "noise", we mean the error in the reconstructed image. In the experiment, our proposed scheme and some previous schemes we use to conduct reversible data embedding. In other words, the visual quality of

Table 3
Distribution of transformed indices of different test images in our scheme.

Cases	Lena	Percentage	Peppers	Percentage	Tiffany	Percentage
Three times embedding	9834	60.0	10515	64.2	12630	68.5
Twice embedding	2265	13.8	2041	12.4	1721	8.3
Once embedding	1546	9.5	1587	9.7	994	7.2
No embedding	2739	16.7	2241	13.7	1039	16.0

the recovered images in these schemes is exactly the same, except for Lee *et al*. (2010a), which is an irreversible data hiding scheme. To carefully estimate the visual quality of the reconstructed image, the fourth parameter, Pearson's correlation coefficient, *PCC*, is defined to re-evaluate the visual quality of the reconstructed image. This coefficient is defined in Eq. (12). In principle, when the value of *PCC* is approximately 1, a strong correlation is indicated. Conversely, a value of '0' indicates that the correlation is either weak or non-existent.

$$PCC = \frac{\sum XY - \frac{(\quad X)(\quad Y)}{n}}{\sqrt{\left(\sum X^2 - \frac{(\quad X)^2}{n}\right)\left(\sum Y^2 - \frac{(\quad Y)^2}{n}\right)}}, \tag{12}$$

where both X and Y are the pixel values of the cover image and the reconstructed image, respectively. Table 4 shows the embedding capacity results of our proposed scheme compared to that of four other schemes (i.e., Wang and Lu, 2009; Lee *et al*., 2010a; Chang *et al*., 2011; Yang *et al*., 2011), when a codebook size of 256 is used. In our scheme, which has the highest frequency of transform indices, these indices are used to hide secret information. In addition, for different test images, more than 60% of the indices in the transformed index table can embed two or three bits into one index, as shown in Table 3. Thus, the average embedding capacity of our scheme is 34, 421 bits, which outperforms than that of all other four schemes. For comparison, the average embedding capacities of the other four schemes are (1) Wang and Lu (2009) (32 004 *bits*), (2) Chang *et al*. (2011) (22 311 *bits*), (3) Yang *et al*. (2011) (21 811 *bits*), and (4) Lee *et al*. (2010a) (16 308 *bits*). Also, as shown in Table 4, among the five schemes, our proposed scheme achieves the largest embedding capacity every case except for the "Baboon" image. In this image, the embedding capacity of Wang and Lu (2009) and Chang *et al*. (2011) exceeded our result. However, when compare from aspect of pure capacity, our scheme is the largest in all cases. To clearly prove excellent performance of our scheme, Table 3 provides the distribution of the transformed indices of different test images in our scheme. Taking the image "Lena" for example, 9, 834 indices can be embedded three secret bits, which occupy 60% of the transformed indices in the transformed index table. For another example, there are more than 64% of the transformed indices of the transformed index table of the image "Peppers", which are used to embed three secret bits. It is easy to see that our scheme offers better embedding rates in smooth images than in complex images. Considering the smooth image "Tiffany" and the complex image "Baboon" for example, it is clear that the embedding capacity of the image "Tiffany" is greater than that of the image "Baboon".

Table 4
Embedding capacity and pure capacity for our scheme and four other schemes.

Images	Capacity	Proposed	Wang and Lu	Lee et al.	Chang et al.	Wang et al.
Lena	Embedding capacity	35 578	32 004	18 085	22 461	20 294
	Pure capacities	35 578	5900	16 774	20 702	20 294
Peppers	Embedding capacity	37 214	32 004	15 905	22 175	22 565
	Pure capacities	37 214	9064	14 594	18 596	22 565
Baboon	Embedding capacity	18 658	32 004	5313	19 517	17 563
	Pure capacities	15 079	−9921	−1240	4985	8069
Boat	Embedding capacity	35 192	32 004	24 505	22 547	22 487
	Pure capacities	35 192	−1483	21 883	22 547	22 487
Tiffany	Embedding capacity	42 326	32 004	16 265	24 449	23 784
	Pure capacities	42 326	9783	14 954	24 449	23 784
F16	Embedding capacity	37 557	32 004	17 775	22 419	24 174
	Pure capacities	37 557	3790	17 775	22 419	24 174
Average	Embedding capacity	34 421	32 004	16 308	22 261	21 811
	Pure capacities	33 824	2856	14 123	18 950	20 229

Table 5
Comparison of compression rates of our scheme and four other schemes.

Schemes	Lena	Peppers	Baboon	Boat	Tiffany	F16	Average
Proposed	0.47	0.46	0.53	0.48	0.45	0.47	0.48
Wang & Lu	0.60	0.59	0.69	0.64	0.51	0.61	0.61
Lee et al.	0.51	0.51	0.55	0.52	0.51	0.50	0.52
Chang et al.	0.51	0.53	0.61	0.49	0.39	0.49	0.50
Yang et al.	0.50	0.47	0.54	0.47	0.45	0.48	0.49

This is because, in our proposed scheme, we embed secret information by depending on the most high-frequency transformed index, which is smaller than 16. However, in the "Baboon" image, there are only approximately 52% transformed indices that are used for embedding. In contrast, more than 83% of the transformed indices are used to embed secret bits in the image "Tiffany".

Table 5 presents the comparison of the compression rate of our scheme and that of four other schemes. As seen in Table 5, our scheme has the best compression rate performance, i.e., 0.48 *bpp*, followed by Yang et al.'s scheme (0.49 *bpp*), Chang et al.'s scheme (0.50 *bpp*), and Lee et al.'s scheme (0.52 *bpp*). It can be observed that Wang and Lu's scheme (0.61 *bpp*) is the worst one among the five schemes. Our scheme outperforms than all four of other schemes. This is because our proposed scheme is based on the high-frequency transformed indices, which are smaller than 16 to embed secret data. Therefore, once a transformed index is found to be smaller than 16, the secret bit is embedded. Then, this transformed index is encoded by only 7 bits, which include 2 indicator bits and 5 bits of the binary format of the transformed index. This means that the more transformed indices smaller than 16 that are found, the better compression rate will be. Nevertheless, in the four other schemes, secret data are embedded by using the correlation of neighboring indices. Therefore, indicator bits are required to support the extraction of secret data and

Table 6
Experimental results of PSNR and PCC values of our scheme.

Categories	Techniques	Lena	Peppers	Baboon	Boat	Tiffany	F16
PSNR	VQ	30.3	29.8	24.1	29.2	30.3	30.8
	Proposed	30.3	29.8	24.1	29.2	30.3	30.8
PCC	VQ	0.99	0.99	0.92	0.98	0.95	0.98
	Proposed	0.99	0.99	0.92	0.98	0.95	0.98

Table 7
Comparison of the execution time (in seconds) for the five schemes.

Phases	Proposed	Wang and Lu	Lee et al.	Chang et al.	Wang et al.
Data embedding phase	5.21	6.43	14.14	11.73	10.64
Data extracting phase	0.56	2.44	2.06	1.12	1.96

the reconstruction of the original indices in the data extracting phase. Thus, in most cases, more than 8 bits are required to encode one index.

Table 6 provides the experimental results of *PSNR* values and *PCC* values obtained for our proposed scheme. Note that compared with the conventional VQ technique, the *PSNR* of our proposed scheme is the same as that of conventional VQ encoding. Moreover, to further re-estimate the visual quality of the reconstructed image in our scheme, the Pearson's correlation coefficient value is computed. Again, it is obvious that Table 6 confirms that our scheme achieves good visual quality of the reconstructed image. This is because all *PCC* values for different images in our scheme are higher than 0.9, which is also similar to that of the traditional VQ technique when codebook of size 256 is used.

Table 7 shows the average execution times required by our proposed scheme and four other schemes in both the data embedding phase and the data extracting phase. It is clear to see that our scheme is the best among the five schemes. The execution time for both phases for our scheme is only 5.77 s, which outperforms the other four schemes, i.e., Wang and Lu's scheme (8.87 s), Yang et al.'s scheme (12.60 s), Chang et al.'s scheme (12.85 s), and Lee et al.'s scheme (16.20 s). Lee et al.'s scheme is the worst one, because it spends too much time for classifying block type and finding the suitable state codebook of the current index in order to encode the current index. Hence, our scheme saves more time among the five schemes. Moreover, the data extracting phase of our scheme is the fastest one among the five schemes.

5. Conclusions

In this paper, a new but simple reversible data hiding approach, based on VQ and SMVQ compression, was introduced for gray images. In our scheme, there are more than 50% indices of transformed index table, which can be used to embed two or three secret bits. Therefore, in embedding capacity and pure capacity, our scheme is the best one among the five schemes tested, with an average embedding capacity of more than 34 000 bits. In addition, from the aspect of compression performance, our scheme has an average compression

rate of 0.48 bpp, which also outperforms the other four schemes, i.e., Yang et al.'s scheme (0.49 bpp), Chang et al.'s scheme (0.50 bpp) Lee et al.'s scheme (0.52 bpp), and Wang and Lu's scheme (0.61 bpp). Moreover, in both the data embedding phase and the data extracting phase, our scheme has the shortest execution times of the five schemes. Our scheme takes only 5.77 s for both phases, which is much faster than the execution times of the other four schemes. Therefore, we can conclude that our proposed scheme is feasible for use in transmitting secure, multimedia information.

References

Chang, C.C., Chen, T.C., Chung, L.Z. (2002). A steganographic method based upon JPEG and quantization table modification. *Information Sciences*, 141(1), 123–138.

Chang, C.C., Wu, W.C., Hu, Y.C. (2007a). Lossless recovery of a VQ index table with embedded secret data. *Journal of Visual Communication and Image Representation*, 18(3), 207–216.

Chang, C.C., Lin, C.C., Tseng, C.S., Tai, W.L. (2007b). Reversible hiding in DCT-based compressed images. *Information Sciences*, 13, 2768–2786.

Chang, C.C., Lin, C.Y., Fan, Y.H. (2008). Lossless data hiding for color images based on block truncation coding. *Pattern Recognition*, 41(7), 2347–2357.

Chang, C.C., Nguyen, T.S., Lin, C.C. (2011). A reversible data hiding scheme for VQ indices using locally adaptive coding. *Journal of Visual Communication and Image Representation*, 22(7), 664–672.

Chang, C.C., Nguyen, T.S., Lin, C.C. (2013). A novel VQ-based reversible data hiding scheme by using hybrid encoding strategies. *The Journal of Systems and Software*, 86(2), 389–402.

Chen, W.J., Huang, W.T. (2009). VQ indices compression and information hiding using hybrid lossless index coding. *Digital Signal Processing*, 19(3), 30–36.

Davis, R.M. (1978). The data encryption standard in perspective.*IEEE Communications Magazine*, 16(6), 5–9.

Hong, W., Chen, T.S. (2011). Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism. *Journal of Visual Communication and Image Representation*, 22, 131–140.

Iwata, M., Miyake, K., Shiozaki, A. (2004). Digital steganography utilizing features of JPEG images. *IEICE Transactions on Fundamentals*, E87-A(4), 929–936.

Kim, T. (1992). Side match and overlap match vector quantizers for images. *IEEE Transactions on Image Processing*, 1(4), 170–185.

Lee, C.F., Chen, H.L., Lai, S.H. (2010a). An adaptive data hiding scheme with high embedding capacity and visual image quality based on SMVQ prediction through classification codebooks. *Image Vision Computing*, 28(8), 1293–1302.

Lee, J.D., Chiou, Y.H., Guo, J.M. (2010b). Reversible data hiding on histogram modification of SMVQ indices. *IEEE Transactions on Information Forensics and Security*, 5(4), 638–648.

Linde, Y., Buzo, A., Gray, R.M. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1), 84–95.

Lu, Z.M., Wang, J.X., Liu, B.B. (2009). An improved lossless data hiding scheme based on image VQ-index residual value coding. *Journal of Systems and Software*, 82, 1016–1024.

Luo, H., Yu, F.X., Chen, H., Huang, Z.L., Li, H., Wang, P.H. (2011). Reversible data hiding based on block median preservation. *Information Sciences*, 181, 308–328.

Rivest, R., Sharmir, A., Adlenan, L. (1978). A method for obtaining digital signature and public key cryptosystems. *Communications of the ACM*, 21(2), 120–126.

Su, P.C., Kuo, C.C. (2003). Steganography in JPEG2000 compressed images. *IEEE Transactions on Consumer Electronics*, 49(4), 824–832.

Thien, C.C., Lin, J.C. (2003). A simple and high hiding capacity method for hiding digit-by-digit data in images based on modulus function. *Pattern Recognition*, 36, 2875–2881.

Tian, J. (2003). Reversible data hiding using difference expansion.*IEEE Transactions on Circuits and Systems for Video Technology*, 13(8), 890–896.

Tseng, H.W., Chang, C.C. (2004). High capacity data hiding in JPEG-compressed images. *Informatica*, 15(1), 127–142.

Wang, J.X., Lu, Z.M. (2009). A path optional lossless data hiding scheme based on VQ joint neighboring coding. *Information Sciences*, 179, 3332–3348.

Wang, R.Z., Lin, C.F., Lin, J.C. (2000). Hiding data in images by optimal moderately significant-bit replacement. *IEEE Electronic Letters*, 36, 2069–2070.

Wang, R.Z., Lin, C.F., Lin, J.C. (2001). Image hiding by optimal LSB substitution and generic algorithm. *Pattern Recognition*, 34, 671–683.

Yang, C.H., Lin, Y.C. (2009). Reversible data hiding of a VQ index table based on referred counts. *Journal of Visual Communication and Image Representation*, 20(6), 399–407.

Yang, C.H., Wu, S.C., Huang, S.C., Lin, Y.K. (2011). Huffman-code strategies to improve MFCVQ-based reversible data hiding for VQ indexes. *The Journal of Systems and Software*, 84(3), 388–396.

Zhang, X., Wang, S. (2005). Steganography using multiple-base notational system and human vision sensitivity. *IEEE Signal Processing Letters*, 12, 67–70.

C.-C. Chang received the BS degree in applied mathematics and the MS degree in computer and decision sciences from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1977 and 1979, respectively. He received the PhD degree in computer engineering from National Chiao Tung University, Hsinchu, in 1982. From July 1998 to June 2000, he was Director of the Advisory Office, Ministry of Education, R.O.C. From 2002 to 2005, he was a Chair Professor at National Chung Cheng University. From February 2005, he has been a Chair Professor at Feng Chia Univeristy. In addition, he was severed as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression, and data structures.

T.-S. Nguyen received the bachelor's degree in information technology from the Open University, HCM city, Vietnam, in 2005. From December 2006, he has been lecturer of Tra Vinh University, Tra Vinh, Vietnam. In 2011, he received MS degree in computer sciences from Feng Chia University, TaiChung, Taiwan. He is currently pursuing the PhD degree with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. His current research interests include data hiding, image and signal processing, multimedia security, information security.

Grįžtamoji duomenų maskavimo schema panaudojant SMVQ rodiklius

Chin-Chen CHANG, Thai-Son NGUYEN

Grįžtamasis duomenų maskavimas yra suprantamas kaip metodas ar jų grupė kuris garantuoja, kad skaitmeninis vaizdas gali būti korektiškai atstatytas, po įterptos, slaptos žinutės vaizde atkodavimo. Pastaruoju metu didelis dėmesys susijęs su grįžtamuoju duomenų maskavimu yra nukreiptas į VQ principais grindžiamus duomenų suspaudimo metodus. Šiame straipsnyje autoriai pristato naują grįžtamąjį duomenų maskavimo metodą grindžiamą VQ ir SMVQ principais, kuris leidžia padidinti įterpiamą žinutę. Eksperimentų rezultatai rodo, jog autorių siūloma schema užtikrina didesnį žinučių talpumą ir užtikrina mažesnį vidutinį kompresijos santykį. Tai pat pristatoma schema išlaiko didesnį atstatyto skaitmeninio vaizdo detalumą.