A Reversible Data Hiding Scheme Based on Side Match Vector Quantization

Chin-Chen Chang, *Fellow, IEEE*, Wei-Liang Tai, and Chia-Chen Lin

***Abstract*—Many researchers have studied reversible data hiding techniques in recent years and most have proposed reversible data hiding schemes that guarantee only that the original cover image can be reconstructed completely. Once the secret data are embedded in the compression domain and the receiver wants to store the cover image in a compression mode to save storage space, the receiver must extract the secret data, reconstruct the cover image, and compress the cover image again to generate compression codes. In this paper, we present a reversible data hiding scheme based on side match vector quantization (SMVQ) for digitally compressed images. With this scheme, the receiver only performs two steps to achieve the same goal: extract the secret data and reconstruct the original SMVQ compression codes. In terms of the size of the secret data, the visual quality, and the compression rate, experimental results show that the performance of our proposed scheme is better than those of other information hiding schemes for VQ-based and SMVQ-based compressed images. The experimental results further confirm the effectiveness and reversibility of the proposed scheme.**

***Index Terms*—Reversible data hiding, side match vector quantization (SMVQ), steganography.**

I. Introduction

REVERSIBLE data hiding, or lossless data hiding, hides secret data in a digital cover image in a reversible way. Relatively large amounts of secret data are embedded in a cover image so that the decoder can extract the hidden secret data and restore the stego-image to the original cover image. Although the distortion in the stego-image is imperceptible to the eye in traditional data hiding, its original content inevitably may be modified by hiding the secret data. Any change affects the information and access to the original cover image. Thus, reversible data hiding is the best way to overcome this defect.

In the past, much literature [1]–[4] has been published on a variety of data hiding techniques such as steganography and digital watermarking; however, most of these techniques damage the cover image during the data hiding procedure. Since reversible data hiding not only satisfies all the criteria for traditional data hiding but also successfully restores the cover image, reversible data hiding [5]–[13] has drawn much recent attention. Many scholars have proposed a variety of applications for reversible data hiding. In the opinion of Kamstra and Heijmans, reversible data hiding is a fragile technique in the sense that the embedded data will mostly be destroyed by small distortions of the image. This technique is ideal for applications, such as military or medical imaging, that involve the straight storage of metadata into an image where the loss of quality is always intolerable [9]. Li also points out that reversible data hiding can be used to design fragile watermarking, which emphasizes authenticity and integrity at the moment of authentication rather than the robustness and existence of the watermark in the cover image after the authentication phase [14].

The earliest reversible data hiding technique described in the literature was proposed by Barton in 1997 [5]. Barton's scheme compresses the bits to be influenced by the embedding operation, then hides the compressed data and the payload in the host image. During retrieval, the original bits are decompressed and used to restore the modified bits to reconstruct the original block. Reversible data hiding was later applied to the compression domain. Fridrich, *et al.* [6] presented an invertible watermarking scheme to authenticate digital images in the JPEG domain. They use an order-2 function, which is an inverse function, to modify the quantization table to enable lossless embedding of one bit per discret cosine transform (DCT) coefficient. Later, Xuan, *et al.* [12] proposed a high-capacity distortion-free data hiding technique based on the integer wavelet transform. This scheme uses histogram modification to embed secret data in the middle frequency of the wavelet domain. The scheme can be applied in JPEG2000 compressed images because JPEG2000 is based on the wavelet transform domain. In 2005, Yang, *et al.* [13] proposed a reversible watermarking technique that uses modified vector quantization (VQ). During the encoding phase, adjacent blocks are used to encode the current block, but additional flag bits are required. They modified standard VQ to achieve the reversibility property, but at the expense of degradation in visual quality and compression ratio.

Due to the limited bandwidth of networks, they can not keep up with the growing sizes of various multimedia files. Many well-accepted image compression algorithms have been proposed to counter this problem, such as VQ [15], side match VQ (SMVQ) [16], JPEG [17], JPEG2000 [18], and so on. One of the most commonly studied image compression techniques is VQ [15], which is an attractive choice because of its simplicity and cost-effective implementation. Indeed, a variety of VQ techniques have been successfully applied in real applications such as speech and image coding [10], [19]. VQ not only has faster encode/decode time and a simpler framework than JPEG/JPEG2000 but it also requires limited information during decoding, and those advantages cost VQ a little low compression ratio and visual quality. VQ works best in applications in which the decoder has only limited information and a fast execution time is required [20].

In VQ, assume that the finite set $Y = \{y_i | i = 0, 1, \ldots, n-1\}$ is the codebook sized n, where $y_i = (y_{i1}, y_{i2}, \ldots, y_{im})$ is the

Manuscript received April 26, 2005; revised January 19, 2006. This paper was recommended by Associate Editor E. Izquierdo.

C. C. Chang is with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan, R.O.C. (e-mail: ccc@cs.ccu.edu.tw).

C. C. Chang and W. L. Tai are with the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan, R.O.C. (e-mail: taiwl@cs.ccu.edu.tw).

C. C. Lin is with the Department of Computer Science and Information Management, Providence University, Taichung 433, Taiwan, R.O.C. (e-mail: mhlin3@pu.edu.tw).

Digital Object Identifier 10.1109/TCSVT.2006.882380

ith m-dimension codeword. In brief, VQ can be defined as a mapping function Q from an m-dimension Euclidean space R^m to a finite subset $Y \subset R^m$

$$Q : R^m \rightarrow Y.$$

For the encoder, a desired mapping function Q is designed so that the Euclidean distance between the input block X_j and the mapped codeword $Q(X_j) = y_k$ is the shortest. The Euclidean distance, defined in Formula (1), is frequently used to measure the similarity between the input block and the mapped codeword

$$\mathrm{ED}(X_j, y_k) = \sqrt{\sum_{i=1}^{m\times m} (X_{ji} - y_{ki})^2}, \quad \text{for } 0 \le k \le n-1. \tag{1}$$

The decoder uses the transmitted code k as the index to extract the corresponding codeword y_k from the same codebook Y so that the input block X_j can be approximated. The key point in designing a perfect VQ scheme is to create a perfect codebook. The LBG algorithm, presented by Linde, *et al.* in 1980 [21], gives a good answer and is probably the most prominent codebook design algorithm.

Recent compression techniques are typically applied to conquer the bandwidth problem of transmission. Thus, compressed images are being substituted for images intended as cover media. In 2002, Jo and Kim [3] embedded the secret data in VQ compression codes. In their scheme, the main codebook Y is first divided into three subcodebooks so that $Y = \{SY_{-1}, SY_0, SY_1\}$. The grouping rule is that each pair of the nearest codewords is selected from the main codebook and classified into subcodebooks SY_0 and SY_1, respectively. The remaining unmatchable codewords are then grouped into subcodebook SY_{-1}.

During the hiding phase, the codeword Y_i is searched using VQ for the current block. The codeword Y_i is used to encode this current block if it belongs to the subcodebook SY_{-1}. If $Y_i \in SY_k$ and k equals to the secret data b_j, then the codeword Y_i is used to encode the current block. If $Y_i \in SY_k$ and k does not equal to the secret data b_j, then the codeword closest to Y_i from subcodebook SY_{b_j} is selected to encode this block. In the extracting phase, the group number of the subcodebook indicates the secret data bit b_j. For example, if the encoded codeword $Y_i \in SY_{-1}$, it denotes that there are no secret data in this block. If the encoded codeword $Y_i \in SY_0$ or $Y_i \in SY_1$, it denotes that the hidden secret bit is 0 or 1, respectively. Thus, in Jo and Kim's scheme [3], the secret data size and the visual quality of the stego-image depend on the clustering results of the codewords. The secret data size decreases if the encoded codewords in the subcodebook SY_{-1} are larger, and vice versa. On the other hand, their scheme yields better visual stego-image quality if the clustering result of similar codewords is appropriate, and vice versa.

Nevertheless, VQ has its limitations. For example, it typically causes visible boundaries between blocks because the current block is coded independently of its neighboring blocks. Kim proposed SMVQ in 1992 [16] to deal with this problem. In his invention, the blocking effect is reduced successfully by using local edge information and provides a better visual quality and compression ratio than VQ does. As a result, to make data hiding more convenient, some researchers have tried to hide secret data in cover images that have been compressed by VQ or SMVQ [1], [2], [13], [19].

Fig. 1. Highlight on the visible staircase effect around the shoulder.

Although the existing VQ-based or SMVQ-based data hiding schemes successfully make VQ- or SMVQ-compressed images to be the cover media, VQ or SMVQ compression codes are always damaged by the hidden secret data and they cannot be reconstructed completely after data extraction. Therefore, the damaged compression codes can no longer serve as cover media. As a result, when two people want to transmit secret data using a compression domain steganography scheme, both must generate their own VQ or SMVQ compression codes in advance. In this paper, we propose a reversible data hiding technique for the lossy image format SMVQ. The secret data are hidden in the SMVQ-compressed cover image in a way that prevents malicious attackers from detecting the existence of the secret data in the stego-image. This technique enables the communicating parties to share the cover media since the original SMVQ compression codes can be reconstructed completely after the hidden data are extracted. Moreover, the reconstructed SMVQ-compressed codes can be stored directly to save storage space and can be reused repeatedly for a variety of purposes.

The rest of this paper is organized as follows. In Section II, the background of SMVQ and the related SMVQ-based data hiding scheme are described. Our proposed reversible data hiding scheme, including the hiding phase, the extracting phase, and the reversing phase, are presented in Section III. Section IV covers our experimental results and discussions and demonstrates the superiority of our proposed scheme. Finally, the conclusion is provided in Section V.

II. Related Works

In this section, we first look at the side match vector quantization technique and compression results, as well as a related SMVQ-based steganographic scheme [2] since our proposed reversible data hiding scheme is based on lossy format SMVQ.

A. SMVQ

VQ takes advantage of the high degree of correlation between individual pixels within a block without considering the similarity between neighboring blocks. In VQ, each block is coded independently and therefore tends to cause visible boundaries between blocks. Fig. 1 shows a staircase effect arising from these visible boundaries. SMVQ [16] is designed to enhance the visual quality of VQ by reducing these visible boundaries.

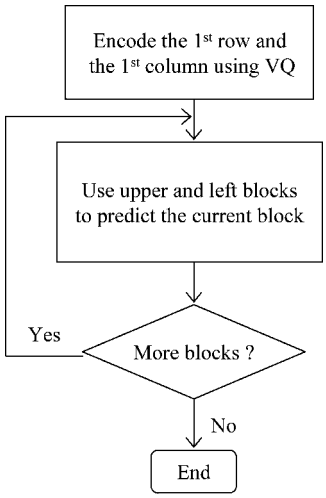

Fig. 2. Flowchart for SMVQ.

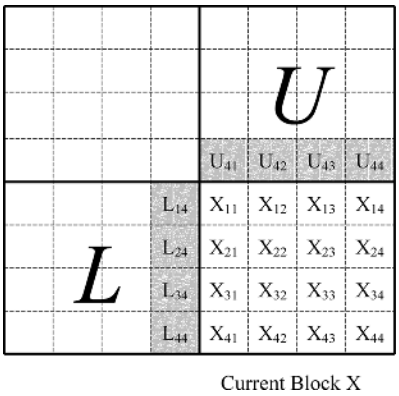

Fig. 3. Upper (U) and left (L) blocks used to generate the subcodebook.

It employs the previous coded blocks, which are above and on the left-hand side of the current block, to help predict the current block so that the visible boundaries can be reduced. Fig. 2 shows the SMVQ flowchart.

In SMVQ, a main codebook is required to encode the blocks in the first row and first column, and a subcodebook is required to encode the rest of the blocks. The subcodebook is a subset of the main codebook. SMVQ is based on the concept that the pixels of the top row in the current block are correlated closely with those of the bottom row in the upper block, and the pixels of the first column in the current block are correlated closely with those of the right column in the left block. The gray areas in Fig. 3 represent the upper and left blocks. These gray regions are used to choose codewords from the main codebook to create a subcodebook.

The blocks in the first row and first column are encoded using VQ. During this process, the main codebook is fully searched to find the best representative codeword to replace the original blocks. The blocks of the first row and first column must be encoded accurately since these blocks are used to predict future blocks as described in the following paragraphs. If an error occurs in this encoding step, it propagates throughout the entire image.

In the subcodebook generation procedure, the upper and left blocks previously encoded are used to generate the subcodebook for the current block. The subcodebook consists of the N codewords, is selected from the main codebook having the least side-match distortion when compared with the gray areas. The current block is encoded using VQ compress scheme with the subcodebook of size N. This process is repeated for each block of the original image until all the blocks are encoded. This approach requires that only the codewords in the subcodebook need to be searched, rather than all codewords in the main codebook, and the size of the subcodebook is much smaller than that of the main codebook. Hence, the advantage of SMVQ is its significant saving in the number of bits required to encode blocks.

Assume that y represents a codeword, and U and L represent the upper and left blocks around it, respectively. We define the upper distortion $\mathrm{UD}(y)$ between the codeword y and the upper block U by

$$\mathrm{UD}(y) = \sum_{j=1}^{4} (U_{4j} - y_{1j}). \tag{2}$$

Similarly, the left distortion $\mathrm{LD}(y)$ between the codeword y and the left block L is computed as

$$\mathrm{LD}(y) = \sum_{i=1}^{4} (L_{i4} - y_{i1}). \tag{3}$$

The side-match distortion of the codeword y is defined as

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

To decode a block X, the previously encoded upper and left blocks are used to predict the subcodebook with the least side-match distortion for the current block X. The generated subcodebook is then searched to find the corresponding codeword to approximate the current block. Thus, SMVQ saves significant bits-per-pixel (bpp) without a significant reduction in peak-signal-to-noise ratio (PSNR). The PSNR is applied to measure the degree of distortion, and the PSNR between the stego-image and the cover image can be calculated by

$$\mathrm{PSNR} = 10 \times \log\left(\frac{255^2}{\mathrm{MSE}}\right). \tag{5}$$

Assume that C is the cover image, S is the stego-image, and C_i and S_i denote the pixel values of H and S, respectively. Here, MSE is defined as

$$\mathrm{MSE} = \frac{1}{m \times n} \sum_{i=1}^{m \times n} (S_i - C_i)^2 \tag{6}$$

where $m \times n$ is the image size of H and S. The image quality is better if its PSNR value is higher. Figs. 4 and 5 show the results of VQ and SMVQ working on the test image "Lena" at the same bit rates. Compared with VQ, SMVQ succeeds in reducing the

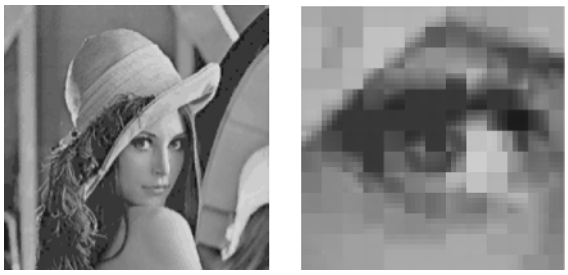

Fig. 4. "Lena" encoded at a bit-rate of 0.4375 by VQ (PSNR: 29.1069).

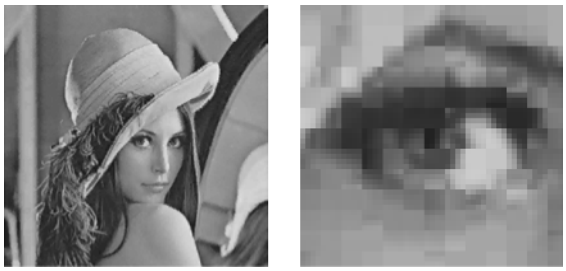

Fig. 5. "Lena" encoded at a bit-rate of 0.4375 by SMVQ (PSNR: 31.0487).

TABLE I
ENCODE "LENA" USING SMVQ FOR A RANGE OF MAIN CODEBOOK SIZES AND SUBCODEBOOK SIZES

Sub-codebook Size	Main Codebook Size					
	128		256		512	
	PSNR	bpp	*PSNR*	bpp	*PSNR*	bpp
16	26.74	0.252	26.73	0.253	25.90	0.254
32	28.41	0.314	28.65	0.315	28.02	0.316
64	29.22	0.375	30.09	0.377	29.78	0.377
128			31.05	0.438	31.31	0.439
256					31.37	0.501

number of bits required to encode the image and improves the visual quality. Table I lists the PSNRs and bpps for SMVQ at various main codebook sizes and various subcodebook sizes.

B. Related SMVQ-Based Steganographic Scheme

In 2005, Chang and Wu [2] proposed a steganographic scheme to hide secret data using SMVQ and VQ. In their scheme, a random seed is generated and used to generate two mapping tables, $\mathrm{list}_{\mathrm{SMVQ}}$ and $\mathrm{list}_{\mathrm{VQ}}$, that contain half 0's and half 1's. Mapping table $\mathrm{list}_{\mathrm{SMVQ}}$ is the same size as the subcodebook and mapping table $\mathrm{list}_{\mathrm{VQ}}$ is the same sizes as the main codebook. Thereafter, VQ is used to encode the first row and the first column of the cover image. For the residual blocks, the corresponding subcodebooks are created using SMVQ. The codewords in each subcodebook are sorted in advance according to the similarity between the codewords and the corresponding blocks.

Because each subcodebook is sorted in advance, the codewords are checked sequentially until the value in mapping table $\mathrm{list}_{\mathrm{SMVQ}}$ that corresponds to the index value of the checked codeword is the same as the secret bit. However, if the Euclidean distance between the checked codeword and the current block exceeds the given threshold $\mathrm{TH}_{\mathrm{SMVQ}}$, VQ rather than SMVQ is used to hide secret data in the block.

The encoding steps using VQ are almost the same as the steps mentioned above. Since the main codebook is sorted in advance, the codewords are checked sequentially until the value in the mapping table $\mathrm{list}_{\mathrm{VQ}}$ for the index value of the checked codeword is the same as the secret bit. However, once the Euclidean distance between the checked codeword and the current block exceeds the given threshold $\mathrm{TH}_{\mathrm{VQ}}$, no secret is hidden in this block.

Depending on secret data size, visual quality, and compression rate, the performance of Chang and Wu's scheme is better than the hiding scheme described by Jo and Kim [19]. Nevertheless, the Chang and Wu's scheme requires additional information to indicate that the current block is encoded using VQ or SMVQ, and the additional indicators will yield a low compression rate. Furthermore, their scheme does not achieve reversibility. To conquer this weakness, we propose a reversible data hiding scheme for SMVQ-based compressed images that does not require the use of additional indicators.

III. PROPOSED REVERSIBLE DATA HIDING SCHEME

According to the previous literature, if a sender tries to hide secret data in the compression domain, the compression codes must be modified to hide the secret data, and the modifications may distort the compressed image. In addition, the original compression codes cannot be reconstructed or stored for later use as long as the secret data are hidden in the compression codes. That means the receiver could not use the received compression code as a carrier after extracting the hidden data. That condition is ineffective for communicating parties. To ensure that the original compression codes can be recovered directly during the extracting phase and stored for later use, we wanted to modify the codeword selection strategy of SMVQ and develop a novel data hiding scheme with the property of reversibility. To achieve this goal, we broke our proposed scheme into three phases: the preprocessing phase, the hiding phase, and the extracting and reversing phase. Those phases are described in greater detail in the following paragraphs.

A. Preprocessing Phase

Secret data must be preprocessed for security reasons. Encrypting the hidden data prevents them from being illegally accessed or unscrambled. Some existing encryption techniques, such as DES,[1] RSA [22], and others can be used to encrypt hidden data. Secret data also can be compressed in advance using lossless compression techniques to reduce the amount of hidden data and increase the visual quality of stego-image, and thus deceive potential grabbers.

After preprocessing in our proposed scheme, the cover image is encoded using SMVQ and the SMVQ compressed image is created. The preprocessed secret data are then hidden in the SMVQ compressed image using our proposed hiding scheme. The stego-image is thus generated, ready for transmission to a receiver. Receivers can extract the secret data from the stego-image using our proposed extracting and reversing scheme. In addition, receivers can restore the original SMVQ compressed

[1]DES Encryption Standard (DES), National Bureau of Standards (U.S.), Federal Information Processing Standards Publication 46, National Technical Information Service, Springfield, VA, Apr. 1997.

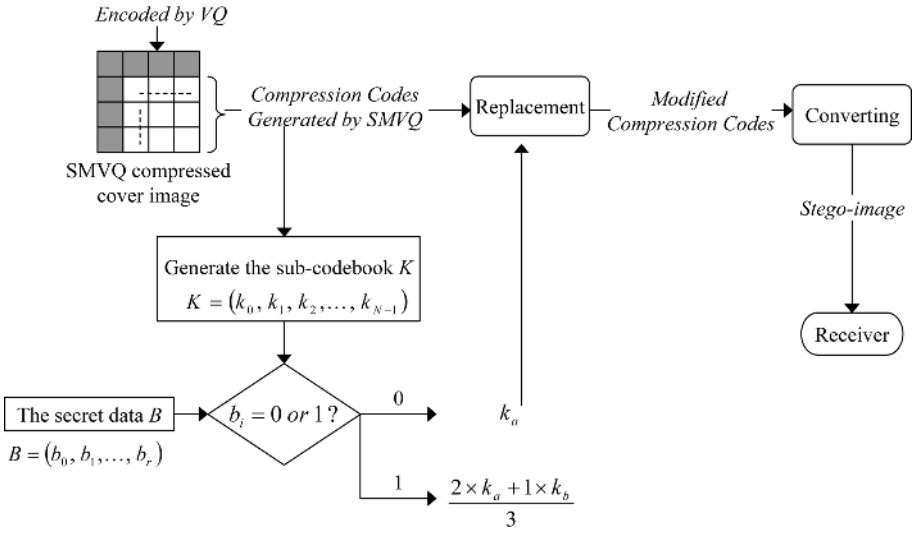

Fig. 6. Flowchart showing the steps in the hiding phase.

TABLE II
PSNRs FOR SMVQ COMPRESSED COVER IMAGES WITH/WITHOUT HIDDEN DATA

Images	At the same bitrate (0.438 bpp)	*PSNRs* (dB)
Lena	With hidden data	30.7746
	Without hidden data	31.0487
F16	With hidden data	29.9363
	Without hidden data	30.2280
Boat	With hidden data	28.7894
	Without hidden data	29.0675
Peppers	With hidden data	29.0675
	Without hidden data	30.1798
Baboon	With hidden data	22.6614
	Without hidden data	22.8004

(a) (b)

Fig. 7. "Lena." (a) Original compressed image (PSNR: 31.0487 dB). (b) Stego-image (PSNR: 30.7746 dB).

TABLE III
HIDING RESULTS OF "LENA" FOR VARIOUS MAIN CODEBOOK SIZES AND SUBCODEBOOK SIZES

Sub-codebook Size	Main Codebook Size					
	128		256		512	
	PSNR	bpp	*PSNR*	bpp	*PSNR*	bpp
16	26.37	0.252	26.54	0.253	25.73	0.254
32	27.92	0.314	28.42	0.315	27.81	0.316
64	28.67	0.375	29.85	0.377	29.52	0.377
128			30.77	0.438	31.06	0.439
256					31.21	0.501

codes completely. The reconstructed SMVQ compressed codes can be stored directly to save storage space. The hiding, extracting, and reversing phases are described in detail in the following subsections.

B. Hiding Phase

To better explain this phase, we define the symbols to be used in hiding phase as follows: the main codebook $Y = (y_0, y_1, \ldots, y_{n-1})$, the SMVQ compressed cover image C, the subindices $D = (d_0, d_1, \ldots, d_r)$, and the secret data $B = (b_0, b_1, \ldots, b_r)$, where $b_i = \{0, 1\}$ and $0 \leq i \leq r$. The hiding phase consists of the following steps.

1) The SMVQ-compressed cover image C is divided into nonoverlapping blocks. Because the blocks in the first row and first column are encoded by VQ, the secret data B are hidden in the residual blocks.
2) For each residual block, the upper and left encoded blocks in C are used to generate the subcodebook $K = (k_0, k_1, \ldots, k_{N-1})$, where k_i is the ith codeword and $i = 0, 1, \ldots, N-1$. The subindex d_i is used to find the corresponding codeword k_a from the subcodebook K.
3) If the secret bit b_i is equal to 0, then the codeword k_a becomes the content of the stego-image.
4) If the secret bit b_i is equal to 1, then we search the codeword k_b from the subcodebook K so that the codeword k_b is the closest to the codeword k_a. The approximate codeword becomes the content of the stego-image and is defined as

$$\text{Approximate codeword} = \left\lfloor \frac{2 \times k_a + 1 \times k_b}{3} \right\rfloor. \quad (7)$$

5) Steps 2–4 are repeated until the whole stego-image is generated.

In our proposed reversible data hiding scheme, the modified SMVQ compression codes are converted into a stego-image. The stego-image must be transmitted without extra messages

TABLE IV
COMPARISONS AMONG THE PROPOSED SCHEME, JO AND KIM'S SCHEME, AND CHANG AND WU'S SCHEME

Images	Jo and Kim's scheme			Chang and Wu's scheme			Proposed scheme		
	Payload size (bits)	*PSNR* (dB)	Bit Rate (bpp)	Payload size (bits)	*PSNR* (dB)	Bit Rate (bpp)	Payload size (bits)	*PSNR* (dB)	Bit Rate (bpp)
Lena	14930	28.19	0.5	13487	29.25	0.47	16129	30.78	0.44
F16	14318	28.58	0.5	13914	29.15	0.45	16129	29.94	0.44
Boat	13902	27.66	0.5	13246	28.12	0.46	16129	28.79	0.44
Peppers	14992	28.74	0.5	13984	29.07	0.45	16129	30.18	0.44
Baboon	12462	21.54	0.5	8794	22.43	0.61	16129	22.66	0.44

being required to achieve reversibility. Fig. 6 is a flowchart of the proposed hiding procedure.

C. Extracting and Reversing Phase

Once the stego-image is received, the receiver can extract the secret data without having to refer to the original cover image. The steps for extracting and reversing follow.

1) The stego-image is divided into nonoverlapping blocks. The first row and first column blocks are encoded using VQ and the indexes are generated.
2) For each residual block, the previously reconstructed upper and left blocks are used to generate a subcodebook $K = (k_0, k_1, \ldots, k_{N-1})$, where k_i is the ith codeword and $i = 0, 1, \ldots, N-1$. The codeword k_a is selected from the subcodebook K such that the Euclidean distance between the current block X_i and the codeword k_a is the shortest.
   If the Euclidean distance $\mathrm{ED}(X_i, k_a)$ is equal to 0, then the secret bit $b_i = 0$. The index a of the codeword k_a is outputted to restore the original state.
   If the Euclidean distance $\mathrm{ED}(X_i, k_a)$ does not equal 0, then the secret bit $b_i = 1$. The index a of the codeword k_a is outputted to restore the original state.
   Steps 2–4 are repeated until all the secret data are extracted and all the original indexes are generated.

After all five steps in the extracting and reversing phase have been performed, the secret data $B = (b_0, b_1, \ldots, b_r)$ can be accurately extracted and the output indexes should equal the original SMVQ-compressed codes.

The reconstructed compressed codes can now be stored directly to save storage space and can be reused repeatedly for a variety of applications.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we show that experimental results confirm our proposed reversible data hiding scheme, then demonstrate the effectiveness and the feasibility of our scheme. Our experiments use five 512×512 gray-level test images: "Lena," "F16," "Boat," "Peppers," and "Baboon." These standard gray-level images are compressed as the cover images using SMVQ with a main codebook of 256 codewords and a subcodebook of 128 codewords. The secret data is a randomly generated bitstream. The relative PSNRs for our SMVQ-compressed cover images (without hidden data) are shown in Table II.

Hiding the secret data in the compressed information certainly creates larger distortions in stego-images. However, our proposed reversible data hiding scheme shows its ability to hide the secret data in a low bit rate (0.438 bpp) compressed cover image that achieves a very high hiding capacity and keeps the distortion low. Fig. 7 illustrates the hiding results using 16 K bits of secret data. As Fig. 7 shows, our proposed scheme keeps the hiding distortion low and achieves very high visual quality. Table II shows the PSNRs with and without hidden data at the same bit rate (0.438 bpp) for various images. The table also shows that the average PSNR of the five stego-images is about 29 dB. In the best case, the PSNR is still 30.7746, which is close to that of the SMVQ-compressed cover images.

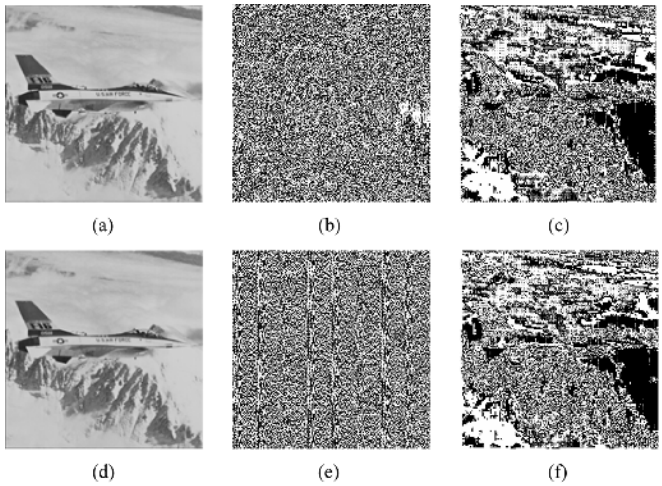

Fig. 8. Example "F16" for visual attack using enhancing LSBs. (a) Original "F16." (b) Enhanced LSBs of (a). (c) Enhanced LSBs of reconstructed "F16" from its SMVQ indices. (d) "F16" with secret data. (e) Enhanced LSBs of (c). (f) Enhanced LSBs of stego-image of "F16."

Table III gives additional hiding results for the "Lena" image at a range of sizes for the main codebook and the subcodebook. Compared with Table I, the average PSNR of "Lena" in Table III is still 29.0686, which is quite close to that of SMVQ compressed "Lena" without the secret data embedded but with the same compression rate and a main codebook of 512 codewords. All this shows that our proposed scheme not only guarantees that the receiver can accurately extract the hidden secret data, but also that the SMVQ-compressed codes can be recovered and reused after the secret data are extracted.

Table IV compares the proposed scheme, Jo and Kim's scheme, and Chang and Wu's scheme by size of secret data, visual quality, and compression rate. In Jo and Kim's scheme, the threshold value is set to 50 for classifying codewords into three subcodebooks during the experiment. In Chang and Wu's scheme, $\mathrm{TH}_{\mathrm{SMVQ}} = 30$, $\mathrm{TH}_{\mathrm{VQ}} = 50$, and a subcodebook of

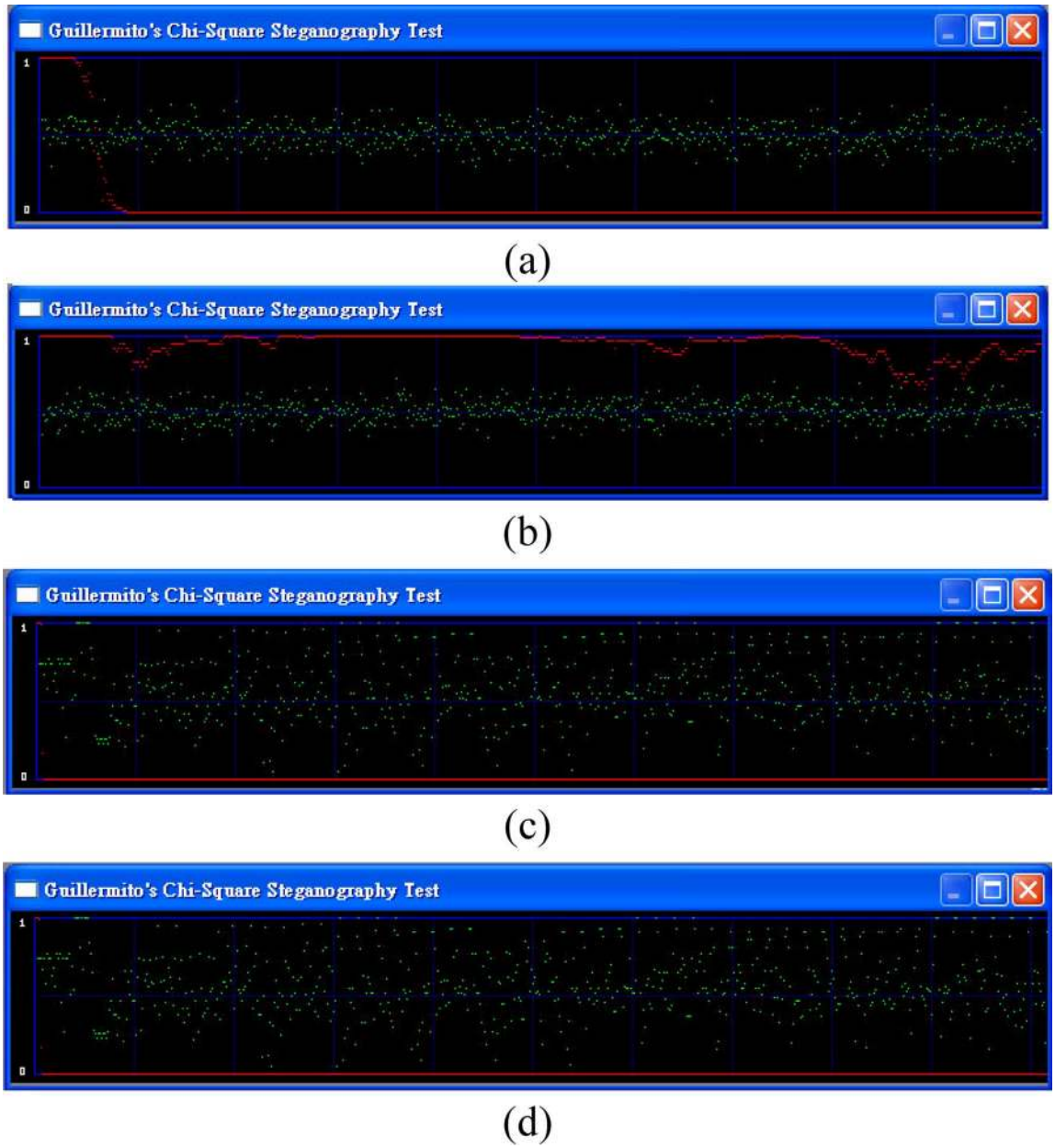

Fig. 9. Example of "F16" for statistical attack using Chi-square analysis. (a) Chi-square result of "F16." (b) Chi-square result of "F16" with secret data in the 1-LSB of each pixel. (c) Chi-square result of reconstructed "F16" from its SMVQ indices. (d) Chi-square result of stego-image of "F16." (Color version available online at: http://ieeexplore.ieee.org.)

size 16 is used to maintain a lower bit rate. The main codebook with a size of 256 is adopted in all schemes for this comparison. As the table shows, Jo and Kim's scheme can hide larger amount of secret data than Chang and Wu's scheme. However, the Jo and Kim scheme cannot achieve reversibility, nor can it support a higher visual quality or a lower compression rate than the others. Conversely, our hiding capacity is larger and the visual quality is still better than other schemes. In addition, our scheme can preserve the high compression rate because additional indicators are not needed. Thus, our compression rate is superior to the schemes of both Chang and Wu and Jo and Kim. Most important, our proposed scheme maintains the high hiding capacity and compression rate while achieving reversibility.

This superior performance enables the receiver to use the reconstructed compression codes as cover media, hide secret data and send back the SMVQ stego-image to make communication between parties more efficient. Our proposed scheme also can serve as reversible fragile watermarking for the SMVQ-compression codes, just as in Yang *et al.*'s scheme.

Compared with an LSB-based reversible hiding scheme with an encrypted bitstream, our proposed scheme is better able to resist visual and statistical attack. Visual and statistical attacks enable attackers to discover whether an LSB-based stego-image contains secret data, primarily because the hidden data are embedded in the least significant bits of each pixel in a cover image. Whether the secret data is randomly generated or encrypted by a modern encryption technique, there is an almost fifty percent

probability for each bit to be 0 or 1. Enhancing the least significant bits of an LSB-based stego-image reveals several regular patterns once the secret data are inside, as Fig. 8(e) shows. Our proposed scheme, on the other hand, hides secret data by modifying the compression codes rather than by directly modifying the least significant bits of each pixel in a cover image. Therefore, no regular patterns appear in our stego-image, as can be seen in Fig. 8(f).

To further prove that our proposed scheme withstands the Chi-square attack, we use a Chi-square steganography test program provided by Guillermito[2] to perform steganography analyses. Fig. 9 shows the test results. In Fig. 9(b), the red curve is the result of the Chi-square test. It is close to one, so the probability for a random embedded message is high. The second output is green curve that presents the average value of the LSBs. In Fig. 9(b), the green curve stays at about 0.5, which means a random message is embedded. Note that in Fig. 9(d), the green average of LSBs is very variable and the Chi-square red output is flat at zero all along the picture. In other words, nothing is hidden in our stego-image.

V. Conclusion

Hiding data in SMVQ-compressed codes originally caused a large distortion in stego-images because SMVQ is a low bit-rate compression scheme. To maintain the advantages of SMVQ and make sure the original compression indexes can be successfully reconstructed after secret data are extracted, we hide the secret data in the compressed cover image and achieve the property of reversibility. The procedures for hiding and extracting and reversing are straightforward. Being reversible, the original compressed codes can be completely reconstructed after hidden secret data extraction, and the original compressed codes can be stored directly and used repeatedly. In addition, the proposed scheme can simply hide or extract the secret data and restore the SMVQ-compressed codes without complex computations. The hidden secret data can also be extracted from the stego-image without referencing the original compressed cover image. In terms of secret data size, visual quality, and compression rate, the performance of our proposed scheme is superior to that of other VQ or SMVQ-based reversible hiding schemes.

[2]Chi-square Steganography Test Program. [Online]. Available: http://www.guillermito2.net/stegano/tools/index.html

References

[1] C.-C. Chang, G.-M. Chen, and M.-H. Lin, "Information hiding based on search-order coding for VQ indices," *Pattern Recognit. Lett.*, vol. 25, pp. 1253–1261, 2004.

[2] C.-C. Chang and W.-C. Wu, "A steganographic method for hiding secret data using side match vector quantization," *IEICE Trans. Inf. Syst.*, vol. E88-D, no. 9, pp. 2159–2167, 2005.

[3] M. Jo and H. D. Kim, "A digital image watermarking scheme based on vector quantization," *IEICE Trans. Inf. Syst.*, vol. E85-D, no. 6, pp. 1054–1056, Jun. 2002.

[4] P.-Y. Tsai, M.-H. Lin, and C.-C. Chang, "An adaptive steganographic scheme for color image," *Fund. Inf.*, vol. 62, no. 3-4, pp. 275–289, Aug. 2004.

[5] J. M. Barton, "Method and apparatus for embedding authentication information within digital data," U.S. Patent 5 646 997, Jul. 8, 1997.

[6] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication watermark for JPEG images," in *Proc. ITCC* , Las Vegas, NV, Apr. 2001, pp. 223–227.

[7] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding—New paradigm in digital watermarking," *J. Appl. Signal Process.*, vol. 2002, no. 2, pp. 185–196, Feb. 2002.

[8] M. Goljan, J. Fridrich, and R. Du, "Distortion-free data embedding for images," in *Proc. 4th Int. Workshop Inf. Hiding*, 2001, vol. 2137, pp. 27–41.

[9] L. Kamstra and H. J. A. M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2082–2090, Dec. 2005.

[10] N. M. Nasrabadi and R. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, vol. 36, no. 8, pp. 957–971, Aug. 1988.

[11] J. Tian, "Reversible data embedding using difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[12] G. Xuan, J. Zhu, J. Chen, Y.-Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," *Electron. Lett.*, vol. 38, no. 25, pp. 1646–1648, Dec. 2002.

[13] B. Yang, Z.-M. Lu, and S.-H. Sun, "Reversible watermarking in the VQ-compressed domain," in *Proc. 5th VIIP*, Benidorm, Spain, Sep. 2005, pp. 298–303.

[14] C. T. Li, "Reversible watermarking scheme with image-independent embedding capacity," *Proc. Vis. Image Signal Process.*, vol. 152, no. 6, pp. 779–785, Dec. 2005.

[15] R. M. Gray, "Vector quantization," *IEEE Acoust., Speech, Signal Process.*, vol. 1, pp. 4–29, 1984.

[16] T. Kim, "Side match and overlap match vector quantizers for images," *IEEE Trans. Image Process.*, vol. 1, no. 4, pp. 170–185, Apr. 1992.

[17] W. B. Pennebaker and J. L. Mitchell, *The JPEG Still Image Data Compression Standard*. New York: Reinhold, 1993.

[18] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals Standards and Practice*. Norwell, MA: Kluwer, 2002.

[19] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.

[20] Z.-N. Li and M. S. Drew, *Fundamentals of Multimedia*. Englewood Cliffs, NJ: Prentice-Hall, Oct. 2003.

[21] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[22] R. L. Rivest, A. Shamir, and L. Adleman, "A scheme for obtaining digital signatures and public-key cryptosystems," *Commun. Assoc. Comput. Mach.*, vol. 21, no. 2, pp. 120–126, Feb. 1978.