A Capacity-enhanced Reversible Data Hiding Scheme Based on SMVQ[1]

Shu-Fen Chiou[†], I-En Liao[†], and Min-Shiang Hwang[‡]

Department of Computer Science and Engineering[†]
National Chung Hsing University
250 Kuo Kuang Road, Taichung, Taiwan 402
Email: s9356055@cs.nchu.edu.tw, ieliao@nchu.edu.tw

Department of Management Information Systems[‡]
National Chung Hsing University
250 Kuo Kuang Road, Taichung, Taiwan 402
Email: mshwang@nchu.edu.tw

[1]Responsible for correspondence: Prof. Min-Shiang Hwang (Department of Management Information Systems, National Chung Hsing University, 250 Kuo Kuang Road, Taichung, Taiwan 402),
Email: mshwang@nchu.edu.tw

Abstract

In recent years, there are many reversible data hiding scheme proposed. Most schemes are guaranteed that the original cover image can be reconstructed completely. When the secret data are hidden in the compression image, the receiver need to extract the secret data, reconstruct the original cover image, and compress the image to save the space. In 2006, Chang et al. [5] proposed a reversible data hiding scheme based on SMVQ (Side Match Vector Quantization). Their method can extract the secret data and reconstruct the SMVQ-compressed cover image. In this paper, we proposed a capacity-enhanced reversible data hiding scheme. The hiding payload size of secret data of our proposed scheme is more than that of both Chang and Wu and Chang et al.'s schemes for VQ-based and SMVQ-based compressed images.

Keywords: Data hiding, reversible, VQ, SMVQ

1 Introduction

Data hiding is a technique that hides the secret data in the digital cover image [1, 4, 5, 6, 10, 12, 13, 15, 18, 19, 21]. The secret data means the information we want nobody to know but some specific people. However, the traditional data hiding has a disadvantage; in fact, the original cover image may be changed during the data hiding process. In order to restore the cover image when decoding the secret data, some researchers proposed the reversible data hiding.

The first reversible data hiding scheme was proposed by Barton in 1997 [2]. This method is to compress bits and hide the data and payload in the host image. After retrieval, the host image can be restored by the modified bits with the compressed bits and payload. After Barton's scheme, there are many researchers presenting various schemes [7, 20, 21].

Because of the limited bandwidth, the networks can not maintain heavy-size multimedia files. In order to resolve this problem, many compressed image data methods were proposed, such as VQ (Vector Quantization) [9], SMVQ (Side Match Vector Quantization) [11], JPEG [16], and JPEG 2000 [17], etc.

Vector Quantization (VQ) [8, 21, 4] is one of the image compression techniques. Using this method, first we need to train the codebooks. There are many training methods, e.g., LBG method [14]. Suppose the finite set $C = \{c_i \mid i = 0,1,...,n-1\}$ is the codebook size n, where $c_i = \{c_{i1}, c_{i2}, ..., c_{im}\}$ is the m-dimension codeword size. And we divide the image I into non-overlapping blocks. When the j-th block I_{X_j} wants to find the best codeword, usually we will calculate the Euclidean distance between I_{X_j} and each codeword in codebook C to find the shortest codeword c_k by the following equation:

$$ED(I_{X_j}, c_k) = \sqrt{\sum_{i=1}^{m} (I_{X_{ji}} - c_{ki})^2}\ , \text{ for } \ 0 \le k \le n-1.$$

The k is the index of the current block. When we decode the image I, I_{X_j} is decoded by C_k. This method is very simple and efficient, but it still has limitations. Because the block and its neighboring blocks are all independent, usually, there are visible boundaries between blocks. SMVQ can solve such problem.

In 2005, Chang and Wu proposed a hiding scheme using SMVQ and VQ [6]. They use two mapping tables, $list_{VQ}$ and $list_{SMVQ}$ to check whether it matches the secret data. For SMVQ, if it does, they will calculate the distance between the original block and the compressed block. If the distance is less than TH_{SMVQ}, the index in subcodebook will be stored in the compressed code; otherwise, it will use the VQ to hide the secret. If the distance is greater than TH_{VQ}, the block will not hide the secret data. In their method, the payload size of secret data is small, and the stego-image can not be restored by the compressed code. Moreover, they need to store the extra mapping tables for the sender and the receiver.

In 2006, Chang et al. proposed a reversible scheme based on SMVQ [5]. The secret data are hidden in the SMVQ compressed cover image. And after extracting secret data, the original SMVQ compressed cover image can be completely restored by this method. And the reconstructed compressed data can be stored to save space and be reused for other purposes. In their method, the payload size is also small. And when they hide the data, they need to process the multiply operation.

In this paper, we proposed a capacity-enhanced reversible data hiding scheme based on SMVQ. The secret data are hidden in SMVQ compressed cover image in our scheme. In addition, we don't hide the secret data in the first row and the first column that are compressed by VQ. In each pixel in a residual block, we could hide a secret bit. Therefore, it could hide larger capacity of secret data using our scheme than that using Chang-Wu [6] and Chang et al.'s [5] schemes.

This paper is organized as follows: Section 2 introduces SMVQ and Chang et al.'s method. We propose our method in Section 3. Section 4 gives the experimental results and discussion. The final section contains our conclusions.

2 Related Works

2.1 SMVQ

In VQ, it considers pixels' correlation in one block. When using VQ to compress image, every block is independent. In fact, blocks just consider themselves without considering their neighboring blocks. This characteristic causes the visible boundaries between blocks. SMVQ [11] can reduce this disadvantage because when current block compresses, it considers neighboring blocks. It uses the above and the left-hide side of current block to predict this block. Following we briefly introduce the SMVQ scheme.

In the SMVQ scheme, we have two codebooks: one is the main codebook which is used to compress the first row and first column of blocks in the image. In Figure 1, supposing that the current block I_X will be encoded, we use the previously encoded upper-side block I_U and left-side I_L block of I_X to generate the subcodebook. After computing the distortion by the grey values and the codeword in the main codebook, we select N least side-match distortions as the subcodebook. Then, we compress current block by this subcodebook.

Next, we introduce how to generate the subcodebook as follows. First, we select the gray area in I_U and compute the upper-side distortion $I_{UD}(c)$ with all codewords c in the main codebook C by the following equation:

$$I_{UD}(c) = \sum_{j=1}^{4} (I_{U_{4j}} - c_{1j}).$$

Then, we compute the left-side distortion $I_{LD}(c)$ by the following equation:

$$I_{LD}(c) = \sum_{i=1}^{4} (I_{L_{i4}} - c_{i1}).$$

Finally, we compute side-match distortion of codeword c as follows:

$$SMD(c) = I_{UD}(c) + I_{LD}(c).$$

After compare with all codewords in the main codebook C, we pick up N codewords with least side-match distortion as the subcodebook $K = \{k_i \mid i = 0,1,...,N-1\}$. Here, N is a multiple of 2 and is smaller than the size of the main codebook. After choosing N-size subcodebook, we use the VQ method to decide the I_X's best codeword. This phase will be repeated until all residual blocks are encoded. When we want to decode the image, we first restore the blocks in the first row and the first column by VQ, and use these blocks to predict the subcodebook to reconstruct the other blocks.

2.2 Related SMVQ schemes

2.2.1 Chang and Wu's scheme [6]

They generate two mapping tables, $list_{SMVQ}$ and lis_{tVQ} by random seed. In the two tables, they contain half 0's and half 1's. The size of $list_{SMVQ}$ is equal to the size of the subcodebook, and the size of $list_{VQ}$ is the same as the size of the main codebook. First, the image is divided into the non-overlapping blocks, and VQ is used to encode the first row and the first column. Then, for each residual block, its corresponding subcodebook is created by SMVQ. Each residual block will hide one bit in the secret data.

In terms of one residual block, we first use SMVQ to hide the data. In the corresponding subcodebook, the three closest codewords will be found. In terms of the first closest codeword, we check the hiding bit and the index in $list_{SMVQ}$. If the hiding bit is matched with the index in $list_{SMVQ}$, the distance between the original block and the compressed block will be calculated. If the distance is less than TH_{SMVQ}, the index in subcodebook will be stored in compressed code; otherwise it will use the VQ to hide the secret. If the distance is greater than TH_{VQ}, the second and the third index will be checked. If all of them don't match the condition, the block will not hide the secret data.

2.2.2 Chang et al.'s scheme [5]

There are three phases in their scheme: first is the preprocessing phase, second is the hiding phase, and the last one is the extracting and reversing phase. We introduce the phases in this subsection. First, we define some symbols: the main codebook $C = \{c_i \mid i = 0,1,...,n-1\}$, the subcodebook $K = \{k_i \mid i = 0,1,...,N-1\}$, the SMVQ compressed cover image M, subindices $S = \{s_j \mid j = 0,1,...,r\}$ and the secret data $B = \{b_l \mid l = 0,1,...,r\}$, where $b_l = \{0,1\}$, and $0 \le l \le r$.

1. Preprocessing phase: In this phase, the secret data are encrypted by some known methods and are compressed. After this preprocessing, the cover image will be encoded by SMVQ for hiding the secret data.

2. Hiding phase: In this phase, the secret data B are hidden in the blocks without blocks in the first row and the first column. These blocks are called residual blocks. For every residual block, we compute the Euclidean distance between the block and the subcodebook to get its closest codeword k_x from its subcodebook K. If $b_l = 0$, then k_x is the content values of the block in the stego-image. Else, we find another codeword k_y from K which has the shortest Euclidean distance between itself and k_x. And calculate the approximate codeword, which becomes the values in this block, by the following equation:

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

   We repeat the above processes until we generate the whole stego-image.

3. Extracting and reversing phase: When the receiver receives this stego-image, the receiver could extract and reverse the image. First, this image is divided into the non-overlapping blocks, and the blocks in the first row and the first column are decompressed by VQ. For each residual block, we generate the sub- codebook K by its upper and left blocks, and we calculate the closest codeword k_x. For current block I_{X_i}, we compute the Euclidean distance $ED(I_{X_i}, k_x)$. If the value is equal to 0, then secret data b_l will be 0. The k_x is used to restore the current block I_{X_i}. If the value is not equal to 0, then secret data b_l will be 1. The k_x is also used to reconstruct the current block.

In Chang et al.'s method, each block hides one bit, and if the secret bit is equal to 1, the approximate block is computed by Equation (1). This quality of stego-image will be lower. In this paper, we propose a novel method based on SMVQ. Our scheme hides more secret data in SMVQ compressed image, and can get better quality in stego-image.

3 The Proposed Method

Similar to the Chang et al.'s method, we first encrypt the hidden data and compress these. For the cover image, we divide it into the non-overlapping blocks and compress them by

SMVQ. Besides the blocks in the first row and the first column, we hide the secret data in the residual blocks. There are two phases in our method. We describe these in the following subsections. The symbols of the main codebook, the SMVQ compressed cover image, and the subindices, are the same as those in Chang et al.'s scheme. We divide the secret data B into $B = \{b_l \mid l = 0,1,...,r\}$ where $b_l = \{b_{l1}, b_{l2}, ..., b_{lm}\}$.

3.1 Hiding Phase

There are the following steps in this phase:

1. The compressed cover image M is divided into $\sqrt{m} \times \sqrt{m}$ non-overlapping blocks. Because the blocks in the first row and the first column are compressed with VQ, we don't hide the secret data in these blocks. We could hide the secret data B in the residual blocks.

2. For each block, we generate the corresponding subcodebook $K=\{k_i|i=0,1,\ldots,N\text{-}1\}$, where $k_i = \{k_{i1}, k_{i2}, ..., k_{im}\}$, from the main codebook C by using the upper and left-side blocks. And we use the subindex s_i to find the corresponding codeword k_x from K.

3. For each residual block, we hide the bit in every pixel in this block. In the other words, we hide n bits in one residual block. For each secret bit b_{l_p}, $0 \le p \le m$, if b_{l_p} is equal to 0, the pixel value k'_{xp} of the block in the stego-image will be the same as k_{xp}. Else, in default, k'_{xp} is equal to $k_{xp} + 1$. After comparing all bits in b_l, we restore k'_x into the stego-image. In this step, we need to consider whether k'_x is the same as the other codeword in the subcodebook K. If it is, we may extract the wrong secret data and reconstruct the wrong block in the next phase. Thus, we need to check the block when we change values in this block. We use Algorithm 1 to check whether $k'_x \stackrel{?}{=} k_y$, where k_y denotes a codeword in K. If it does, we change the values of the pixels which hiding data is 1. We use $k'_x = k_x + 1$ or $k'_x = k_x - 1$ to get 2^a-1 results and check whether one of these results is or not equal to other codewords k_y in K. For example, m = 16 and b_{l1} = 1111000000000000. In default, we compute $k'_{x0} = k'_{x0} + 1$, $k'_{x1} = k'_{x1} + 1$, $k'_{x2} = k'_{x2} + 1$, and $k'_{x3} = k'_{x3} + 1$. $k'_{x4} - k'_{x15}$ are the same as $k_{x4} - k_{x15}$. If k'_x is equal to k_y, we compute $k'_{x0} = k_{x0} - 1$. We then check whether k'_x is or not equal to k_y. If it is, we will compute other results. The worst case is that all of the results collided with other codewords in the subcodebook K. When the worst case happened, we don't hide secret data in this block and hide the block index in the stego-image.

4. Repeating the Steps 2-3 until the whole stego-image is generated.

Algorithm 1 Hiding process in one block

for each block after hide the secret data
$a = 0$
if $k'_x = k_y$, where k_y is one codeword in the K **then**
  for i=0;i++;$i \le m$ **do**
    if $k'_{xi} = k_{xi} + 1$ **then**
      Compute $a = a + 1$
    end if
  end for
end if
for j=0;j++;$j \le 2^a$-1 **do**
  Compute k'_x that $k'_{xi} = k_{xi} + 1$ or $k'_{xi} = k_{xi} - 1$ when k_{xi} is hide 1
  if k'_x != k_y **then**
    quit
  else
    continue
  end if
end for
if $k'_x = k_y$ **then**
% k'_x still equals k_y
We do not hide the secret data in this block and hide the block index in the image.
end if
end for

3.2 Extracting and Reversing Phase

When the receiver receives this stego-image, he/she extracts the secret data andvreverses the SMVQ-compressed cover image as follows.

1. The image is divided into $\sqrt{m} \times \sqrt{m}$ non-overlapping blocks. The blocks in the first row and the first column are compressed by using VQ and generate their corresponding indexes.
2. For every residual block, we generate the subcodebook $K = \{k_i \mid i = 0,1,..., N-1\}$ using its upper and left-side blocks. And find k_x whose Euclidean distance is the shortest among the current block I_{X_i}.
3. For the current block I_{X_i}, we compare the k_x and I_{X_i}. If $I_{X_{i1}}$ is equal to the k_{x1}, then the secret bit will be 0, else the secret bit will be 1. After we compare all pixels in

this block, we can get the B_i bits secret data. Then we reconstruct the block by using k_x.

4. Repeating the Step 2-3 until the whole stego-image is extracted. Finally, we can get the whole secret data and reverse the SMVQ-compressed cover image.

4 Experimental Results and Discussion

In this section, we describe the experimental results and discuss them to show our proposed method is efficient. In our experiments, we use five standard 512 × 512 gray-level images as follows: “Lena”, “F16”, “Boat”, “Peppers”, and “Baboon”. And we compress these images by the main codebook of 256 codewords and the subcodebook of 128 codewords. We use a random generator to generate the secret data. Table 1 is the relative PSNR (with/without hidden secret data) for the cover images after SMVQ-compressed. The *PSNR* is computed as follows:

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

$$\mathrm{MSE} = \frac{1}{m \times n}\sum_{i=1}^{m \times n}(M_i - I_i)^2$$

where $m \times n$ is the image size of the cover image M and the stego-image T.

Since the without hidden data images are compressed by SMVQ, the without hidden data images are lossly, and the PSNR are lower than original images. By our experiment, we hide the secret data according to the codeword. The pixel values with hidden secret data images are changed. After performing SMVQ, with hidden secret data images may lead to get better PSNR in some cases.

Figure 2 is our results after hiding the secret data. The images in Figure 2(a) are the original images, in Figure 2(b) are the images after compression, and in Figure 2(c) are the results after hiding the secret data.

Table 2 is a comparison among the Chang and Wu's scheme, Chang et al.'s scheme, and the proposed scheme. In Chang and Wu's scheme, the TH_{SMVQ} = 30, TH_{VQ} = 100, and the size of the subcodebook is equal to 16. Because we hide the bit in each pixel, we can get better payload size than those in both two schemes. And we also hold better visible quality of stego-images.

Figure 3 shows the results for SMVQ-compressed Lena in (a), Chang and Wu's method in (b), the Chang et al.'s method in (c), and the proposed method in (d).With eyes, (b), (c), and (d) can have high quality.

Table 3 is a comparison among the 1-bit LSB method [3]. Comparison the payload size, the LSB method is better than the proposed scheme because we don't hide the secret data in the first row and the first column that are compressed by VQ. Comparison with PSNR, the 1-bit LSB is also better than the proposed method because in the LSB method, secret data are embedded in spatial domain, but in the proposed scheme, we hide the data

in the images compressed by SMVQ.

5 Conclusion

Many reversible data hiding schemes are guaranteed that they can extract the secret data and reconstruct the original cover image. Due to the limitation of network bandwidth, most images are compression formatted by VQ, SMVQ, JPEG, JPEG 2000, and so on. For SMVQ hiding, Chang and Wu proposed a steganographic scheme in 2005. In order to extract the secret data and reconstruct the compressed images completely, Chang et al. [5] proposed a reversible data hiding scheme based on SMVQ in 2006. Their method can extract the secret data and reconstruct lossless the SMVQ-compressed cover image. In this paper, we propose an enhancement of reversible data hiding scheme to improve the size of secret data and visual quality. The experimental results show that the performance of our proposed scheme is better than both Chang and Wu and Chang et al.'s schemes for VQ-based and SMVQ-based compressed images.

References

[1] P. K. Amin, N. Liu, and K. P. Subbalakshmi: '*Statistical attack resilient data hiding*', International Journal of Network Security, 2007, **5**, 1, 112-120.

[2] J. M. Barton: 'Method and apparatus for embedding authentication information within digital data', Report, U. S. Patent 5 646997, 1997.

[3] C. K. Chan and L. M. Cheng: 'Hiding data in images by simple lsb substitution', *Pattern Recognition*, 2004, **37**, 469-474.

[4] C. C. Chang, D. Kieu, and Y. C. Chou: 'Reversible information hiding for vq indices based on locally adaptive coding', *Journal of Visual Communication and Image Representation*, 2009, **20**, 1, 57-64.

[5] 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*, 2006, **16**, 10, 1301-1308.

[6] C. C. Chang and W. C. Wu: 'A steganographic method for hiding secret data using side match vector quantization', *IEICE Transactions on Information and Systems*, 2005, **E88-D**, 9, 2159-2167.

[7] J. B. Feng, I. C. Lin, C. S. Tsai, and Y. P. Chu: 'Reversible watermarking: Current status and key issues', *International Journal of Network Security, 2006, 2, 3, 161-170.*

[8] A. Gersho and R. M. Gray: 'Vector Quantization and Signal Compression', 1992, Kluwer Academic Publishers.

[9] R. M. Gray: 'Vector quantization', *IEEE ASSP Magazine*, 4-29, 1984.

[10] J. Y. Hsiao, J. F. Chen, and J. M. Chang: 'An adaptive reversible informa- tion hiding method based on search-order coding for vq-compressed images', *Imaging Science Journal*, 2009, **57**, 1, 37-45.

[11] T. Kim: 'Side match and overlap match vector quantizers for images', *IEEE Transactions on Image Processing*, 1992, **1**, 2, 170-185.

[12] C. F. Lee, K. N. Chen, and C. C. Chang: 'A new data hiding strategy with restricted region protection', *Imaging Science Journal*, 2009, **57**, 5, 235- 249.

[13] Z. Liao, Y. Huang, and C. Li: 'Research on data hiding capacity', *International Journal of Network Security*, 2007, **5**, 2, 140-144.
[14] Y. Linde, A. Buzo, and R. M. Gray: 'An algorithm for vector quantizer design', *IEEE Transactions on Communication*, 1980, **28**, 1, 84-95.
[15] Y. Liu, X. Sun, I. J. Cox, and H. Wang: 'Natural language information hiding based on Chinese mathematical expression', *International Journal of Network Security*, 2009, **8**, 1, 10-15.
[16] W. B. Pennebaker and J. L. Mitchell: 'The JPEG Still Image Data Compression Standard', 1993, Kluwer Academic Publishers.
[17] D. S. Taubman and M. W. Marcellin: 'JPEG2000: Image Compression Fundamentals Standards and Practice', 2002, Kluwer Academic Publishers.
[18] C. M. Wang, N. Wu, C. S. Tsai, and M. S. Hwang: 'A high quality stegano- graphic method with pixel-value differencing and modulus function', *Journal of Systems and Software*, 2008, **81**, 150-158.
[19] N. I Wu and M. S. Hwang: 'Data hiding: Current status and key issues', *International Journal of Network Security*, 2007, **4**, 1, 1-9.
[20] S. Xinag and J. Huang: 'Analysis of quantization-based audio watermarking to D/A and A/D conversions'**,** *International Journal of Network Security*, 2006, **3**, 3, 230-238.
[21] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni, and W. Su: 'Distortionless data hiding based on integer wavelet transform', *Electronic Letters*, 2002, **38**, 25, 1646-1648.

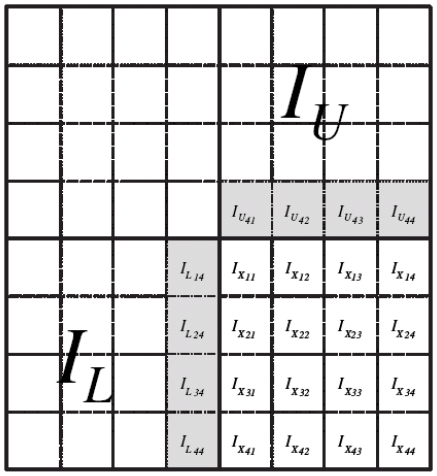

Figure 1: Using upper block I_U and left-side block I_L to generate subcodebook

Table 1: The relative PSNRs (with/without hidden secret data) for the cover images (at the same 0.44bpp)

Images	*PSNRs*(dB) without hidden	*PSNRs*(dB) with hidden
Lena	31.0741	31.3436
baboon	22.6943	22.6884
boat	28.4740	28.4481
peppers	27.6886	27.6042
F16	28.3786	28.3950

Table 2: Comparison among the proposed scheme, Chang and Wu's scheme, and Chang et al.'s scheme

Images	Chang and Wu's scheme		Chang et al's scheme		The proposed scheme	
	Payload size bits	*PSNR*(dB)	Payload size bits	*PSNR*(dB)	Payload size bits	*PSNR*(dB)
Lena	15950	30.4468	16129	30.7883	258064	31.3436
baboon	12650	22.5630	16129	22.5746	258064	22.6884
boat	15619	28.0097	16129	28.2303	258064	28.4481
peppers	15297	27.2916	16129	27.4717	258064	27.6042
F16	15503	27.8472	16129	28.0276	258064	28.3950

(a) original images

(b) compression images

(c) stego images

Figure 2: Our results

Table 3: Comparison between the 1-bit LSB and the proposed schemes

Images	1-bit LSB scheme		The proposed scheme	
	Payload size bits	*PSNR*(dB)	Payload size bits	*PSNR*(dB)
Lena	262144	51.1227	258064	31.3436
baboon	262144	51.1365	258064	22.6884
boat	262144	51.1344	258064	28.4481
peppers	262144	51.1303	258064	27.6042
F16	262144	51.1372	258064	28.3950

(a) The SMVQ-compressed Lena

(b) The Chang and Wu's method

(c) The Chang et al.'s scheme

(d) The proposed scheme

Figure 3: Example of Lena