International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 2, No. 2, June 2009

Distortion-Free Steganography Mechanism with Compression to VQ Indices

Chin-Chen Chang[1,2], Pei-Yu Lin[2], and Hsin-Chi Wang[3]

[1]*Department of Information Engineering and Computer Science,*
Feng Chia University,100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan.
[2]*Department of Information Communication,*
Yuan Ze University,135 Yuan-Tung Rd., Chung-Li, 320, Taiwan.
[3]*Department of Computer Science,*
National Tsing-Hua University, Hsinchu, 30013, Taiwan.
[1]*ccc@cs.ccu.edu.tw,* [2]*pagelin3@gmail.com,* [2]*g9662530@oz.nthu.edu.tw*

Abstract

Directly embedding the secret data into the VQ-compressed domain is practical for reducing storage and transmittal bandwidth. Many steganography techniques are currently exploited for the VQ index table. However, embedding strategies extend the amount of compression needed for embedding secret data and distort the original quality of the VQ decompressed image. To satisfy the essentials of increasing the secret payload, preserve the VQ index table without loss, and reduce the compression rate, the proposed scheme utilizes search order coding to find the correlation between the VQ indices and neighboring indices. Due to the similarity of adjacent indices, the new scheme can effectively embed the secret stream into the index table and encode the embedded index table with fewer bits. The experimental results indicate that the new scheme can achieve superior outcomes of the compression rate and embedding efficiency than related steganography techniques.

Keywords: *Steganography Mechanism, VQ Indices, Distortion-Free*

1. Introduction

The vector quantization (VQ) technique has been widely used for compressing multimedia such as image and speech data [1, 2]. In the VQ compression system, an image can be compressed at a rate using a predefined codebook. Due to the rapid decompression process and simple implementation, VQ becomes a popular method for image compression. The main concept of the VQ technique is encoding each block of host image with the most similar codeword of a representative codebook. That is, each block is encoded using an index corresponding to the selected codeword. As a result, the image is quantized by a number of representative indices, called the index table.

Based on the advantages of VQ compression, a number of steganography approaches have focused on the field of VQ-based information hiding [3-6]. Steganography involves embedding the secret message into a cover medium to achieve concealment. In general, steganography approaches embed the secret message into the spatial domain [7, 8], frequency domain [9-11], and compressed domain of the cover medium. The embedding techniques in the spatial and frequency domains, however, are incapable of

resisting compression. The medium is often compressed before being transmitted over the Internet in order to reduce its capacity due to the bandwidth limitation. Hence, the secret data of the embedded image will be destroyed after compression. Consequently, how to directly embed the secret message into the compressed stream is a critical issue for steganography.

Nowadays, many steganography schemes within the VQ-compression domain have been proposed that involve hiding secret data into the VQ index table. These schemes focus on how to increase the secret data capacity and reduce the compression rate effectively. The embedding procedure, however, are irreversible [3-6, 12-16]. In other words, the VQ index table is distorted after secret embedding and incapable of being restored to the original one. That is, the image quality after embedding is degraded. In contrast to traditional schemes, reversibility allows the receiver to extract the secret message and then completely restore the covered object (i.e. the VQ index table) without loss. This allows the image to be reconstructed in its original quality.

Recently, many researchers have attempted to achieve reversibility for VQ-based information hiding [17-20]. In 2004, Chang et al. [17] proposed a lossless hiding scheme for VQ indices. They embed the secret message into the index table according to two rules. When the embedded secret bit equals 1, the search order coding (SOC) [21] rule is used to decode the index value. On the other hand, when the embedded secret bit equals 0, the VQ rule is used to decode the index value. This approach expands the compressed stream in case where the processing index can be compressed by SOC, but the embedded secret equals 0.

To achieve reversible embedding in the VQ-compressed domain, Yang et al. [18] utilizes similarity between the current index and its four neighboring indices to embed a secret message. They used a threshold to classify all indices into embeddable and unembeddable types. If the average of the four neighboring indices is smaller than a predetermined threshold, then the current index is classified as embeddable; otherwise, the current index is classified as unembeddable. The decompressed image has high quality approximating to that of the original VQ image. The proposed algorithm can fully recover the original MFCVQ-compressed indices but not the original VQ-compressed indices [18].

In contrast, Lin and Chang [19] used a pair of dissimilar codewords to embed a secret bit into each index. Their algorithm [19] is nearly reversible; however, it cannot completely restore the changed blocks to the original VQ-compressed indices. Chang et al. [20] used the left and upper neighboring indices to embed a secret bit and compress the index table. This scheme can obtain high embedding capacity as [18] and higher bit rates than these methods proposed in [18, 19]. The original VQ-compressed indices can be restored completely. However, the indices located in the first row and the first column of the index table cannot be used to embed secret data, which reduces embedding capacity [20].

Obviously, the tradeoff among the embeddable secret capacity, the compression rate, and the lossless VQ image is difficult to satisfy. To confirm these essentials, we propose a distortion-free steganography technique for the VQ-compression domain. Employing the concept of local indices similarity with SOC [21], the new algorithm can embed secret information into all indices and further compress the index table with a

satisfactory bit rate (bpp). Moreover, the proposed embedding algorithm is reversible. Thus, we can achieve a higher compression rate and improved embedding efficiency than these methods proposed in [17-20] and can restore the original VQ image completely.

The rest of this paper is organized as follows. We present the new method in Section 2. In Section 3 we offer the experimental results to demonstrate the effectiveness of the new method. Finally, we make conclusions in Section 4.

2. Proposed Scheme

To improve the embedding capacity and compression rate in VQ-compressed indices, the new scheme utilizes local index similarity to further compress the index table and embed the secret information into all indices. For security, the secret data *S* is encrypted by using a pseudo-random number which is produced by a secret key *SK*. Only the authorized receiver can correctly decrypt the secret stream with *SK*. The details of the new scheme are presented in the following subsections.

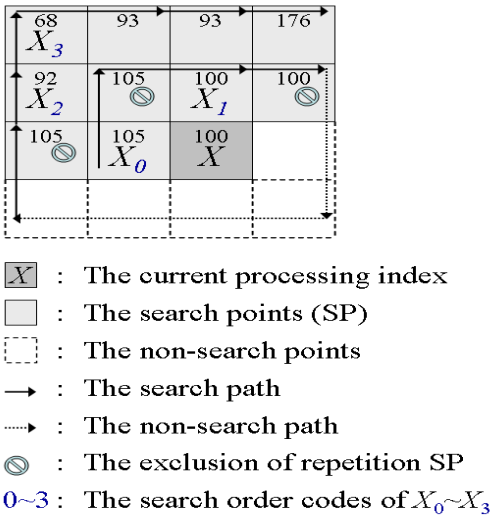


Figure 1. The index table and the embedding order

2.1. Distortion-Free Embedding Procedure

Given a VQ-compressed index table *T* and encrypted secret data *S*, the search order coding (SOC) mechanism [21] is utilized to locate the correlation between the processing index and the adjacent indices. Let *X* be the current processing index in *T* with length of *m* bits. That is, the codebook contains 2^m codewords. The decoder can learn *n* neighboring indices X_0, X_1, …, X_{n-1} of *X* using SOC. Here, *n* is the power of 2.

For instance, assume that the current processing index X=100 and n=4 as shown in Figure 1. All of the indices appearing in the predefined search path are called search points (SP); the non-search points are the indices that appear after *X* and cannot be applied to encode *X*. According to the search path, the adjacent indices of *X* are X_0, X_1, X_2, X_3 and the corresponding search order codes are 0, 1, 2, 3, respectively. It is obvious that the indices X_0, X_1, X_2 and X_3 in the search path have high correlation with *X*. Due to

this similarity, X can be classified into three different cases according to the following rules:

Index Classification Rules

***R1*:** If the value of X has the same value within n adjacent indices: X_0, X_1, …, X_{n-1}, then classify X into *Case 1* or *Case 2*, and set the finding bit f =1.

***R2*:** If the value of X has no match within n adjacent indices: X_0, X_1, …, X_{n-1}, then classify X into *Case 3*, and set the finding bit f =0.

It is obvious that X has high correlation with the adjacent indices. Hence, for *R1*, X can be compressed to further reduce the code stream of T. Nevertheless, for *R2*, the adjacent indices are completely different from X. The original index value of X must be recorded to prevent distortion of the VQ image quality. After classification, the proposed scheme embeds a secret bit s of S into X to achieve steganography in the compressed code.

Secret Embedding Phase

Let X_k be the adjacent index that has the same index value as X for *Case 1* and *Case 2*. Here, k is the search order code of X_k, $k \in \{0,1,\ldots,n-1\}$. Assume that $LSB(k)$ is the least significant bit (LSB) of binary representation of k. According to the classified cases of X, a secret bit s is embedded as follows:

Case 1: If $LSB(k) = s$, then set the hitting bit h =1 and concatenate with the secret bit s and the *indicator*, which is defined as

$$indicator = k \bmod (\frac{n}{2}). \tag{1}$$

Encode the current index X into binary form as $\{f \parallel h \parallel s \parallel (indicator)_2\}$. Here, $(indicator)_2$ denotes the binary representation of *indicator*, and again "||" denotes the concatenation operator.

Case 2: If $LSB(k) = \bar{s}$, then set the hitting bit h =0 and concatenate with the secret bit s and the *indicator*. Here, $\bar{s}$ denotes the complement value of s. Encode the current index X into binary form as $\{f \parallel h \parallel s \parallel (indicator)_2\}$.

Case 3: Encode the current index X into its binary form as $\{f \parallel s \parallel (X)_2\}$. Here, $(X)_2$ denotes the binary representation of index value X.

Table 1. The encoded stream of the proposed scheme

s	***Case 1***				***Case 2***				***Case 3***		
	f	h	s	*indicator*	f	h	s	*indicator*	f	s	indix value
0	1	1	0	$(k \bmod \frac{n}{2})_2$	1	0	0	$(k \bmod \frac{n}{2})_2$	0	0	$(X)_2$
1	1	1	1	$(k \bmod \frac{n}{2})_2$	1	0	1	$(k \bmod \frac{n}{2})_2$	0	1	$(X)_2$

Table 1 lists the encoded results for the cases with different secret *s*. By repeating the above steps, each index of *T* can be further compressed to reduce the compression rate and can be embedded into the secret stream to achieve steganography. The decoder then can distribute the encoded binary stream T' to the receiver.

2.2. Retrieving and Restoration Procedure

With the secret key *SK* and the parameter *n*, the authorized receiver can retrieve the secret data and then reconstruct the original VQ image from the binary stream T'. First, the receiver must learn to which case of the current processing index belongs. If the retrieved prefix bit equals 0, it means that the current processing index has been classified to *Case 3*. The receiver thereby can learn the secret bit *s* in the next bit of T'. With the next *m* bits after *s*, the original VQ index value, $(X)_2$, can then be restored. In contrast, if the first retrieved prefix bit equals 1, then the current processing index belongs to *Case 1* or *Case 2*. Consequently, the receiver can extract the information from hitting bit *h* and the *indicator* to reconstruct the secret *s* and the original index. The details of the decoding procedure are shown as follows:

Step 1: Read a prefix bit as finding bit *f* from T'.

Step 2: If f =0, then the processing index is classified as *Case 3*; go to Step 3. Otherwise, go to Step 4.

Step 3: Learn the next bit as secret *s*. Meanwhile, the original VQ index *X* can be decoded by transforming the next *m* binary stream into decimal representation. Go to Step 10.

Step 4: Get the next bit as hitting bit *h*. If h =1, then classify the processing index as *Case 1*. Otherwise, classify as *Case 2*.

Step 5: Learn the next bit as the secret *s*.

Step 6: Read the next $n/2$ bits as the binary stream $(indicator)_2$.

Step 7: Generate the binary stream $(k\_bin)_2$ by concatenating the bits *indicator* and *s*:

$$(k\_bin)_2 = \begin{cases} (indicator \,\|\, s)_2, & \text{if } h = 1, \\ (indicator \,\|\, \bar{s})_2, & \text{if } h = 0. \end{cases} \tag{2}$$

Step 8: Transform the binary representation value $(k\_bin)_2$ into decimal value as *k*.

Step 9: Decode the original VQ index *X* by the *k*-th adjacent index X_k with search order coding.

Step 10: Repeat Step 1 until all bits in T' are processed.

According to the retrieved secret stream, the receiver can decrypt the secret data using a pseudo-random number generate with secret key *SK*. Meanwhile, the receiver can obtain the VQ-compressed index table *T* without loss.

3. Experimental Result

To demonstrate the performance of the new mechanism, different types of images used in the simulations are shown in Figure 2. The size of the grayscale images is set to 512×512 pixels, and each image is divided into non-overlapping blocks with 4×4

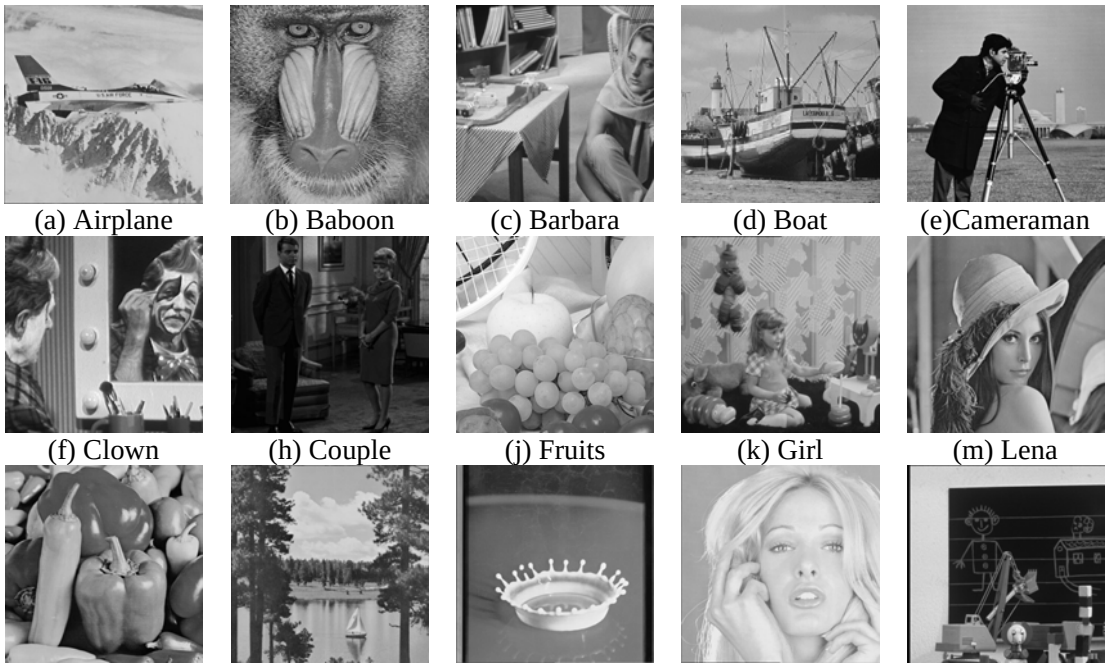
pixels. The VQ codebooks used in the experiments were generated using the well-known LBG algorithm [1], and the dimension of each codeword is 4×4. We set the length of the searched adjacent indices $n=4$. The image quality is evaluated by both the human eye and the peak signal–to-noise rate (*PSNR*). The *PSNR* formula is described as follows:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)\text{dB}. \tag{3}$$

The mean square error (*MSE*) of an image with $H\times W$ pixels is defined as

$$MSE = \frac{1}{H\times W}\sum_{i=1}^{H\times W}\left(p_i - p_i'\right)^2, \tag{4}$$

where p_i is the original pixel value and p_i' is the processed pixel value.



(a) Airplane (b) Baboon (c) Barbara (d) Boat (e)Cameraman
(f) Clown (h) Couple (j) Fruits (k) Girl (m) Lena
(n) Peppers (o) Sailboat (p) Splash (q) Tiffany (r) Toys

Figure 2. Test images

The proposed scheme utilizes the local index similarity to further improve the compression rate of the VQ index table (i.e. *Case 1* and *Case 2*). Accordingly, the frequencies of *Case 1* and *Case 2* influence the outcome of the compression rate. To analyze the correlation between the processing index and adjacent indices, Table 2 illustrates the distribution of the *Case 1*, *Case 2*, and *Case 3* for different codebook sizes. On average, there are 73% indices (i.e., *Case 1* and *Case 2*) that have the same values as the four neighborhoods for the codebook with 128 codewords. For *Case 3*, only 27% of the indices differed from the adjacent indices.

According to the results with 256 codewords, the sum of the percentages in *Case 1* and *Case 2* is nearly 60.2% and that in *Case 3* is 39.8%. The correlation between indices with the neighborhoods is still high. For 512 codewords, the sum of the percentages in *Case 1* and *Case 2* is approximate half. Thus, for larger codebooks, search order coding can achieve a satisfactory relationship. This high correlation can be used to enhance the compression rate and embed secret data effectively.

Table 2. PSNR values and the percentage of cases with various codebook sizes

Images	Codebook size											
	128 codewords				256 codewords				512 codewords			
	PSNR	Percentage (%)			PSNR	Percentage (%)			PSNR	Percentage (%)		
		Case 1	*Case 2*	*Case 3*		*Case 1*	*Case 2*	*Case 3*		*Case 1*	*Case 2*	*Case 3*
Airplane	29.33	59.8	15.2	25.0	30.58	45.1	18.8	36.0	31.58	29.6	15.2	55.1
Baboon	23.68	19.6	12.9	67.5	24.10	10.8	7.6	81.6	24.43	6.8	4.3	88.9
Barbara	25.01	41.2	17.9	40.9	25.80	29.9	16.6	53.4	26.39	23.3	13.8	62.8
Boat	28.22	54.4	17.8	27.7	29.39	45.1	17.3	37.5	30.16	36.3	16.1	47.7
Cameraman	26.71	57.8	15.5	26.7	28.54	50.2	17.1	32.8	29.27	43.5	16.8	39.7
Clown	28.71	45.4	22.2	32.4	30.23	32.3	19.8	47.9	30.81	25.2	15.0	59.8
Couple	26.14	78.8	13.8	7.3	29.47	62.4	18.2	19.4	29.90	56.4	15.8	27.9
Fruits	26.07	57.4	20.3	22.4	27.24	44.9	20.3	34.8	27.63	34.4	15.2	50.4
Girl	29.81	41.5	24.4	34.1	31.17	29.7	19.4	50.9	31.97	21.6	14.1	64.3
Lena	29.61	45.3	25.8	29.0	31.37	30.6	24.1	45.3	32.25	21.4	18.0	60.7
Pepper	28.99	48.6	22.6	28.8	30.73	35.6	20.6	43.8	31.41	26.8	15.1	58.1
Sailboat	27.48	49.6	18.0	32.5	28.62	33.2	19.6	47.2	29.25	23.7	14.7	61.6
Splash	29.05	63.8	24.7	11.5	30.96	52.5	27.1	20.5	31.62	45.9	24.0	30.2
Tiffany	28.52	67.9	24.3	7.9	30.32	51.6	22.9	25.4	30.73	44.1	19.0	36.9
Toys	27.57	71.9	16.3	11.8	29.92	60.9	18.2	20.9	31.16	57.0	17.7	25.4
Average	**27.89**	**53.5**	**19.5**	**27.0**	**29.23**	**41.0**	**19.2**	**39.8**	**30.17**	**33.1**	**15.6**	**51.3**

To evaluate the performance of steganography in VQ-compressed indices, the major concerns are the *PSNR* of the decompressed image, compression rate, embedding capacity, and embedding efficiency [17-20]. For *PSNR*, the new scheme can embed the secret information into the VQ-compressed indices without distortion of the original indices. Thus, the quality of the decompressed image is same as the original VQ image.

The compression rate, also called the bit rate (*BR*), is defined by the following equation:

$$BR = \frac{\text{The number of bits in a compressed stream}}{\text{The number of pixels in a VQ image}} \text{ (bpp).} \tag{5}$$

In general, the bit rate (*BR*) for the original VQ technique with size 256 codebook is given by

$$BR = \frac{(512/4)\times(512/4)\times 8}{512\times 512} = 0.5 \text{ (bpp),}$$

where $(512/4)\times(512/4)$ is the total number of blocks, and each block has to be decoded by 8 bits in the VQ index. A lower *BR* value is helpful to reduce data storage

and bandwidth. An efficient steganography technique must reduce the *BR* value as much as possible after embedding the secret information into the VQ-compression domain (i.e. less than 0.5).

The embedding efficiency *EE* can be calculated as

$$EE = \frac{\text{The number of secret bits}}{\text{The number of bits in a compressed stream}}. \quad (6)$$

The larger value of *EE* indicates that the steganography technique can achieve superior performance for embedding secret data with a reduced compressed stream.

Table 3. Performance evaluation of the proposed scheme with various codebook sizes

Images	Secret capacity (bits)	Codebook size								
		128 codevectors			256 codevectors			512 codevectors		
		compressed stream	*BR*	*EE*	compressed stream	*BR*	*EE*	compressed stream	*BR*	*EE*
Airplane	16384	86011	0.328	0.191	100972	0.385	0.162	128774	0.491	0.127
Baboon	16384	120841	0.461	0.136	145744	0.556	0.113	167512	0.639	0.100
Barbara	16384	99001	0.378	0.166	118042	0.450	0.139	137573	0.525	0.119
Boat	16384	88241	0.337	0.186	102412	0.361	0.160	120199	0.458	0.136
Cameraman	16384	87421	0.333	0.188	97762	0.373	0.168	111092	0.424	0.148
Clown	16384	92071	0.351	0.178	112576	0.429	0.146	134143	0.511	0.122
Couple	16384	71546	0.273	0.229	84598	0.323	0.194	97477	0.372	0.168
Fruits	16384	83856	0.319	0.195	99742	0.380	0.164	123363	0.470	0.133
Girl	16384	93446	0.356	0.175	115606	0.441	0.142	139232	0.531	0.118
Lena	16384	89261	0.341	0.184	110074	0.419	0.149	135102	0.515	0.121
Pepper	16384	89101	0.340	0.184	108616	0.414	0.151	132134	0.504	0.124
Sailboat	16384	92146	0.352	0.178	111922	0.427	0.146	136131	0.519	0.120
Splash	16384	74971	0.286	0.219	85648	0.326	0.191	100130	0.382	0.164
Tiffany	16384	71986	0.275	0.228	90544	0.345	0.181	107900	0.412	0.152
Toys	16384	75231	0.287	0.218	86128	0.328	0.190	94621	0.361	0.173
Average	**16384**	**87675**	**0.334**	**0.190**	**104692**	**0.397**	**0.160**	**124359**	**0.474**	**0.135**

Table 3 lists the outcomes of the proposed scheme for various test images and codebook sizes. Since the new scheme can embed the secret bit into each index, the embedding capacity has $(512/4)\times(512/4)$ bits. On average, the compressed stream after embedding is approximately 87675 bits for the codebook size with 128. The values of *BR* and *EE* are 0.334 and 0.19, respectively. It is obvious that the proposed scheme can effectively embed secret data and compress the VQ index table. For larger codebook sizes, the similarity between indices is reduced. Nevertheless, the *BR* values for codebooks size 256 and 512 can be reduced to 0.397 and 0.474 bpp, respectively, which is less than 0.5 bpp of the original VQ technique.

Table 4 illustrates the functionality comparisons between well-known steganography techniques [17-20] and the proposed mechanism. The original quality of the VQ-compression technique is shown in the second column of Table 4. From the *PSNR* of the decompressed image, it is clear that the techniques [18, 19] distort the index table for the sake of embedding secret information into the VQ-compressed domain. The related methods [17, 20] and the new scheme can achieve lossless image results.

Compared with the previous schemes, the new scheme has the larger capacity and the less compressed stream.

Table 4. Functionality comparisons among related steganography mechanisms and the proposed scheme

Images	Methods						
	VQ (0.5 bpp)	Chang et al. [17]			Yang et al. [18]		
	PSNR	PSNR	capacity (bits)	compressed stream (bits)	PSNR	capacity (bits)	compressed stream (bits)
Airplane	30.58	30.58	16384	107008	28.36	8590	95661
Baboon	24.10	24.10	16384	157416	24.06	1006	141165
Barbara	25.80	25.80	16384	168152	24.95	3987	123279
Boat	29.39	29.39	16384	112376	27.87	6988	105273
Cameraman	28.54	28.54	16384	106704	27.90	6582	107709
Clown	30.23	30.23	16384	121208	28.45	5103	116583
Couple	29.47	29.47	16384	84400	28.73	5370	114981
Fruits	27.24	27.24	16384	107472	26.64	6615	107511
Girl	31.17	31.17	16384	126840	29.31	4402	120789
Lena	31.37	31.37	16384	121184	29.66	7490	102261
Pepper	30.73	30.73	16384	115416	28.72	6967	105399
Sailboat	28.62	28.62	16384	120152	27.75	4537	119979
Splash	30.96	30.96	16384	90632	29.00	9966	87405
Tiffany	30.32	30.32	16384	95056	29.66	7009	105147
Toys	29.92	29.92	16384	90120	25.41	8084	98697
Averaged	*29.23*	*29.23*	*16384*	*114942*	*27.76*	*6180*	*110123*

Images	Methods								
	Lin and Chang [19]			Chang et al. [20]			Ours		
	PSNR	capacity (bits)	compressed stream (bits)	PSNR	capacity (bits)	compressed stream (bits)	PSNR	capacity (bits)	compressed stream (bits)
Airplane	30.58	16129	133736	30.58	16129	138576	30.58	16384	100972
Baboon	24.10	16129	138928	24.10	16129	146320	24.10	16384	145744
Barbara	25.75	16129	137360	25.80	16129	147008	25.80	16384	118042
Boat	29.36	16129	134512	29.38	16129	138640	29.39	16384	102412
Cameraman	26.51	16129	134752	28.54	16129	139224	28.54	16384	97762
Clown	29.96	16129	133856	30.23	16129	146496	30.23	16384	112576
Couple	25.47	16129	131256	29.47	16129	130392	29.47	16384	84598
Fruits	26.53	16129	133096	27.24	16129	133592	27.24	16384	99742
Girl	30.87	16129	132152	31.17	16129	136992	31.17	16384	115606
Lena	31.37	16129	133304	31.37	16129	140888	31.37	16384	110074
Pepper	29.73	16129	133160	30.73	16129	138560	30.73	16384	108616
Sailboat	28.62	16129	134856	28.62	16129	142136	28.62	16384	111922
Splash	29.74	16129	131840	30.96	16129	132776	30.96	16384	85648
Tiffany	29.25	16129	131272	30.32	16129	130000	30.32	16384	90544
Toys	28.04	16129	134176	29.92	16129	135176	29.92	16384	86128
Averaged	*28.39*	*16129*	*133884*	*29.23*	*16129*	*138452*	*29.23*	*16384*	*104692*

Figures 3 through 5 compare related techniques with the new scheme in terms of compressed code stream, bit rate (*BR*), and embedding efficiency (*EE*), respectively. Figure 3 shows the compressed stream of the compressed index table. The result indicates that the proposed process can effectively compress the index table. The *BR*

values of Figure 4 indicate that the new scheme can achieve satisfactory results for most of the test images.

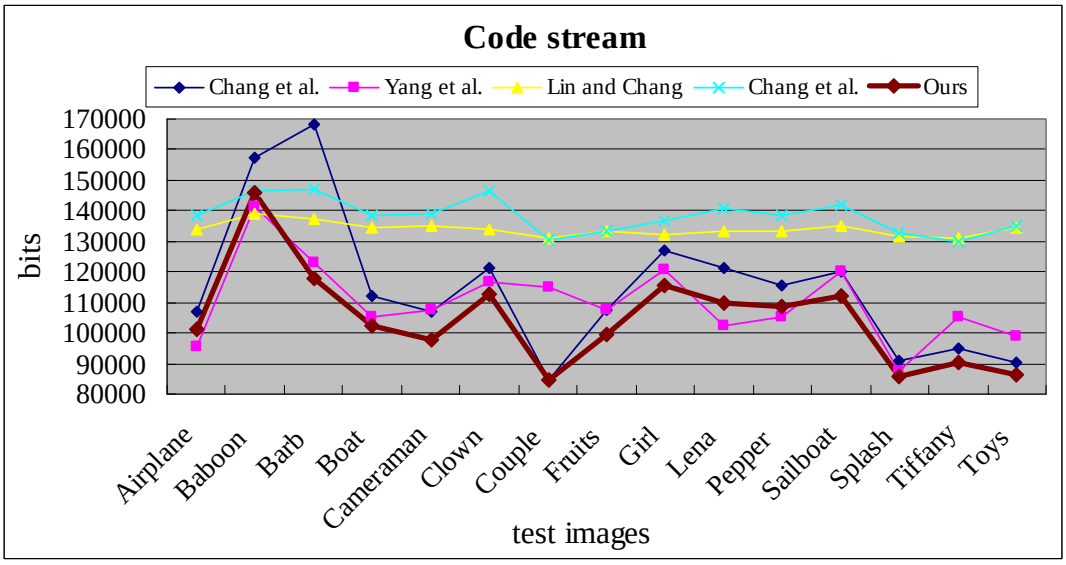


Figure 3. The compressed stream of related schemes and the proposed scheme, codebook=256

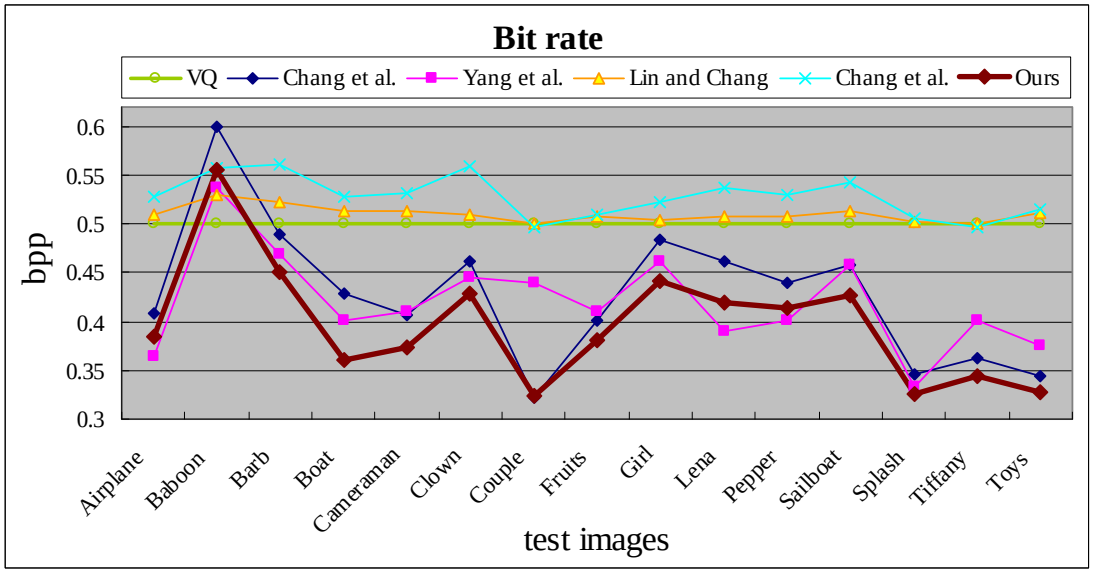


Figure 4. The bit rate (*BR*) results of related schemes and the proposed scheme, codebook=256

In Yang et al.'s method [18], the derived compressed stream and *BR* have better values than that of the proposed scheme for certain test images. This is due to the fact that [18] embeds fewer bits in the index table (6180 on average). However, the proposed scheme has nearly 2.65 times the embedding capacity than [18] and preserves the lossless VQ image. Figure 5 displays the results of embedding efficiency *EE* with a

codebook size of 256. The new process achieved better results than related techniques [17-20] for various test images. The results show the new scheme is practical for steganography in the VQ-compression domain and can preserve the VQ quality without loss.

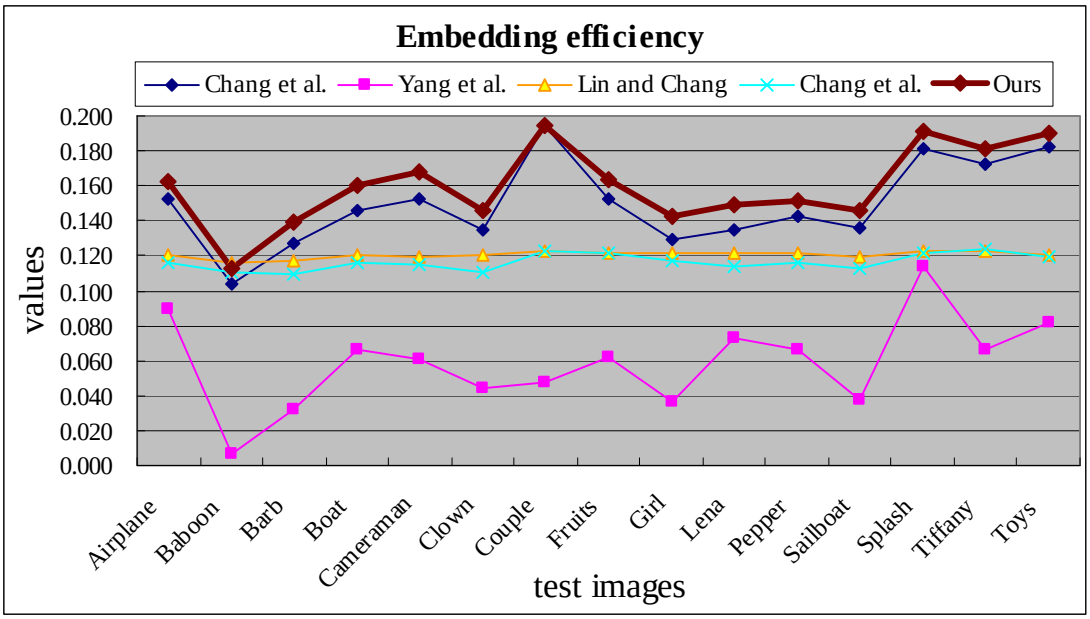


Figure 5. The embedding efficiency (*EE*) results of related schemes and the proposed scheme, codebook=256

4. Conclusions

The steganography of VQ-compression domain is practical for advanced secret information embedding and reducing the compression stream over the Internet. The new scheme utilizes the concept of search order coding to embed secret information into the VQ index table. According to the high correlation among indices, we can effectively embed the secret into the processing index; meanwhile, the bit rate of index table can be reduced (i.e. *Case 1* and *Case 2*). Considering the essentials of decompressed PSNR, bit rate, secret capacity, and embedding efficiency overall, the new mechanism achieves the higher image quality, lower *BR*, and better *EE* values.

References

[1] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," IEEE Transactions on Communications, vol. 28, no. 1, 1980, pp. 84-95.
[2] R. M. Gray, "Vector quantization," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 1, no. 1, 1984, pp. 4-29.
[3] Z. M. Lu; D. G. Xu and S. H. Sun, "Multipurpose image watermarking algorithm based on multistage vector quantization," IEEE Transactions on Image Processing, vol. 14, no. 6, 2005, pp. 822-831.
[4] C. C. Chang and W. C. Wu, "Hiding secret data adaptively in vector quantization index tables," IEE Proceedings Vision, Image, and Signal Processing, vol. 153, no. 5, 2006, pp. 589-597.
[5] Y. P. Hsieh, C. C. Chang and L. J. Liu, "A two-codebook combination and three-phase block matching based image-hiding scheme with high embedding capacity," Pattern Recognition, vol. 41, no. 10, 2008, pp. 3104-3113.

[6] C. C. Lin, S. C. Chen and N. L. Hsueh, "Adaptive embedding techniques for VQ-compressed images," Information Sciences, vol. 179, no. 1-2, 2009, pp. 140-149.
[7] C. C. Chang and P. Y. Lin, "Adaptive watermark mechanism for rightful ownership protection," Journal of System and Software, vol. 81, no. 7, 2008, pp. 1118-1129.
[8] C. C. Lee, H. C. Wu, C. S. Tsai and Y. P. Chu, "Adaptive lossless steganographic scheme with centralized difference expansion," Pattern Recognition, vol. 41, no. 6, 2008, pp. 2097-2106.
[9] I. J. Cox, J. Kilian, F. T. Leighton and T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Transactions on Image Processing, vol. 6, no. 12, 1997, pp. 1673-1687.
[10] A. A. Reddy and B. N. Chatterji, "A new wavelet based logo-watermarking scheme," Pattern Recognition Letters, vol. 26, no. 7, 2005, pp. 1019-1027.
[11] C. C. Chang, P. Y. Lin and J. S. Yeh, "Preserving robustness and removability for digital watermarks using subsampling and difference correlation," Information Sciences, vol. 179, no. 13, 2009, pp. 2283-2293.
[12] Z. M. Lu and S. H. Sun, "Digital image watermarking technique based on vector quantization," Electronics Letters, vol. 36, no. 4, 2000, pp. 303-305.
[13] M. Jo and K. D. Kim, "A digital image watermarking scheme based on vector quantization," IEICE Transactions on Information and Systems, vol. E85-D, no. 6, 2002, pp. 1054-1056.
[14] H. C. Huang, F. H. Wang and J. S. Pan, "A VQ-based robust multi-watermarking algorithm," IEICE Transactions on Fundamentals, vol. E85-A, no. 7, 2002, pp. 1719-1726.
[15] W. C. Du and W. J. Hsu, "Adaptive data hiding based on VQ compressed image," IEE Proceedings on Vision, Image, and Signal Processing, vol. 150, no. 4, 2003, pp. 233-238.
[16] H. C. Wu and C. C. Chang, "Embedding invisible watermarks into digital image based on side-match vector quantization," Fundamenta Informaticae, vol. 63, no. 1, 2004, pp. 89-106.
[17] C. C. Chang, G. M. Chen and M. H. Lin, "Information hiding based on search-order coding for VQ indices," Pattern Recognition Letters, vol. 25, no. 11, 2004, pp. 1253-1261.
[18] B. Yang, Z. M. Lu and S. H. Sun, "Reversible watermarking in the VQ-compressed domain," Proceedings of the Fifth IASTED International Conference on Visualization, Imaging, and Image Processing, Benidorm, Spain, 2005, pp. 298-303.
[19] C. Y. Lin and C. C. Chang, "Hiding data in VQ-compressed images using dissimilar pairs," Journal of Computers, vol. 17, no. 2, 2006, pp. 3-10.
[20] C. C. Chang, T. D. Dieu and W. C. Wu, "A lossless data embedding technique by joint neighboring coding," Pattern Recognition, vol. 42, no. 7, 2008, pp. 1597-1603.
[21] C. H. Hsieh and J. C. Tsai, "Lossless compression of VQ index with search-order coding," IEEE Transactions on Image Processing, vol. 5, no. 11, 1996, pp. 1579-1582.

Authors

Chin-Chen Chang received his BS degree in applied mathematics in 1977 and his MS degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. During the academic years of 1980-1983, he was on the faculty of the Department of Computer Engineering at the National Chiao Tung University. From 1983-1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From 1989 to 2004, he has worked as a professor in the Institute of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan. Since 2005, he has worked as a professor in the Department of Information Engineering and Computer Science at Feng Chia University, Taichung, Taiwan. Dr. Chang is a Fellow of IEEE, a Fellow of IEE and a member of the Chinese Language Computer Society, the Chinese Institute of Engineers of the Republic of China, and the Phi Tau Phi Society of the Republic of China. His research interests include computer cryptography, data engineering, and image compression.

Pei-Yu Lin received the MS degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan in 2004. She received her Ph.D. degree in computer science and information engineering in 2009 from National Chung Cheng University, Chiayi, Taiwan. Since 2009, she has worked as an assistant professor in the Department of Information communication at Yuan Ze University, Taiwan. Her current research interests include digital watermarking, image protection, data mining, and information security.

Hsin-Chi Wang received the B.S. degree in Computer Science and Information Engineering from the Tam Kang University, Taiwan, in 2007. Currently, she is M.S. student in National Tsing Hua University, Taiwan. Her current research interests include image processing, information hiding, and secret sharing.