Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **1**, January **2017**

VQ-based Image Embedding Scheme Using Adaptive Codeword Grouping Strategy

Chin-Feng Lee[a], Chi-Yao Weng*, and Kuan-Ting Li[b]

[a,b] Department of Information Management,
Chaoyang University of Technology, Taichung 41349, Taiwan, R.O.C.

*Department of Computer Science,
National Pingtung University, Pingtung 90003, Taiwan, R.O.C.
*Corresponding author:cyweng@mail.nptu.edu.tw

Received May, 2016; revised August, 2016

Abstract. *The adaptive embedding algorithms for VQ-compressed images are proposed in this paper. By creating and adjusting the predefined distance thresholds according to the required image quality, the proposed algorithms designed as the cluster of codeword grouping strategies. The cluster indicates a number of similar codewords together, a codeword in a group can be embedded as a sub-message while achieving high embedding capacity and preserving good image quality. Our research proposed three types of categorizing algorithms. Type 1 could put the most similar codewords into a group to increase the embedding capacity; however, it does require more time to go through the complex computations. Type 2 is then created for lower the complex computations, but the reconstructed image will most likely lose its original form due to dissimilar codeword replacement for embedding large amounts of hidden data. Type 3 is thus proposed to prevent the image distortion; we utilize the concept of average value to categorize the codebooks. Compared with the ACE method, Chang and Wu's scheme and Lin et al's scheme, the experimental results have shown that the embedding capacity of proposed scheme for test images is obviously superior, and the peak signal-to-noise ratio (PSNR) of image quality is higher in most cases.*

Keywords: VQ-compressed images; Grouping strategies; Lower complex computation; Embedding capacity.

1. **Introduction.** Computer networks do not provide the users complete privacy; packet data may be intercepted. Therefore, computer network security is becoming a prevailing issue in this field. To address this, various papers in cryptography and data hiding [1-7, 10-11, 13-18] have proposed for protecting the transmission of data over the Internet. Data hiding means that the secret messages are hidden in cover media, which are often in the format of images, videos, audios, etc. Generally, data hiding technique enhances the security of communication over the Web. Data compression such as vector quantization (VQ) [4] and side-match vector quantization [8] can provide a solution for this problem, because the sizes of compressed file are greatly reduced and are easily stored or transmitted over the Internet without occupying huge amounts of bandwidth.

Recently, many studies [1, 3, 5-8, 10-11, 13-18] of data hiding based on VQ-compressed images have proposed, which contribute to the reduction of transmission costs as well as the unawareness of hidden contents. VQ technique divides the original image in equal-sized blocks and recovers the image blocks using codewords. To embed a message into a

VQ-compressed image, an index of codeword is replaced by another index of codeword, which is assigned to embed the message.

For keeping the higher image quality, a codeword will not to be replaced for hiding a message while the codeword can not to find out the others similar cordword. Li and Li [10] organized similar codewords in a group. The number of codewords in a group must be exactly equal to a power of 2, that is to say, a group has 2^k codewords; therefore, each codeword in the group can embed a message of k bits. The embedding capacity of an image using Li and Li's scheme is limited if the number of codewords in a group does not meet to equal to a power of 2. The same fault might occurr with Chang and Wu [1] scheme in which two codewords in a group may carry the same sub-message while the number of codewords in that group is not equal to a power of 2. This might also incur more image degradation. Lin et al. [11] proposed an adaptive embedding scheme for VQ-compressed images, which allows the number of codewords in the group and each group is not equal to a power of 2. If the number of codewords in a group is between 2^k and 2^{k+1}, each codeword can carry a distinct sub-message of k or $k+1$ bits. This property makes Lin et al.'s [11] scheme surpasses previous schemes in the embedding capacity.

In this paper we designed codeword arrangement algorithms to cluster a number of similar codewords together, so that a codeword index in a group can carry a sub-message by replacing with another index that is designed to conceal the secret data of k or k *+1* bits and can achieve good image quality when the image has been reconstructed by VQ decoding. The rest of this paper is organized as follows: In Section 2, we review VQ coding and Lin et al.'s embedding scheme [11]. In Section 3, a VQ-based image embedding scheme is proposed on the basis of intra-group codeword homogeneity and the heterogeneity of codewords among groups. The experimental results are demonstrated in Section 4. Finally, we draw the conclusion in Section 5.

2. **Related works.**

2.1. **Vector Quantization (VQ).**

Vector Quantization (VQ) [4], which is one of the most popular image lossy compression methods, was proposed by Gray *et al.,* in 1984 [4]. VQ technique initially divides several $W \times H$ grayscale images into non-overlapping and equal-sized small image blocks, and each block is an l-dimensional vector of pixels equal to $q \times q$. A vector quantization (VQ) is a mapping from a set, $T^l = \{B^{(i)} | i$ is the index of the training vectors $\}$ in the l -dimensional Euclidean space into a finite set of N representative points. The aim of LBG algorithm [9] is to minimize the average pairwise distance as shown below in Eq. (1) between the training vector B and the representative codevector C by their squared Euclidean distance,

$$d(B,\ C) = \sum\nolimits_{i=1}^{\ell} \left(b_i - cw_i\right)^2 \tag{1}$$

where b_i and cw_i are the $i-th$ component of the vectors. Let CB denote the codebook, and the vector quantization $VQ : T^l \to CB$, where $CB = \{cw_1, cw_2, ..., cw_N\}$ and cw_i (where $i = 1, 2, ..., N$) are called the codewords.

For encoding an image, the sender seeks the closest codeword through the given codebook CB for the given input vector. Euclidean distance is an encoding function employed in the searching process to measure the distance between the two vectors. For each original image block X, the index i of the selected codeword is transmitted, and in decompression, the receiver uses the index i to find the mapping codeword in the codebook to retrieve the closest codeword in codebook for recovering that image block. The compression rate of

VQ technique is equal to $1/l$, thereby reducing the transmission bandwidth. Figure 1 illustrates the image compression based on the VQ coding technique, where each codeword cw_i is an l -dimensional vector $(l = 16)$.

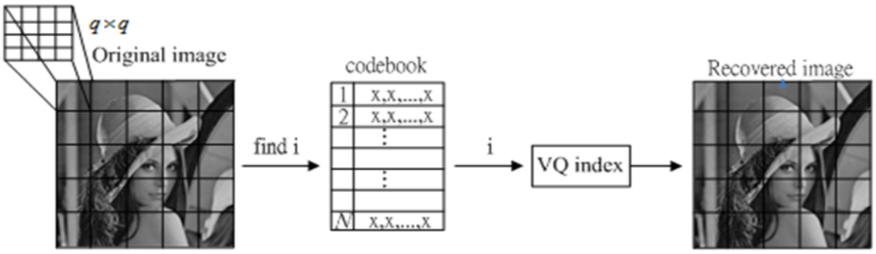
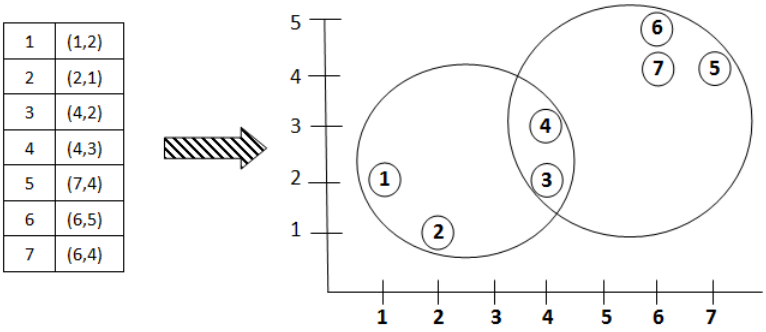


Figure 1. VQ encoding and decoding flowchart

2.2. **Lin et al.'s scheme of VQ coding.** In 2009, Lin *et al.* [11] proposed the VQ compression technique to cover the secret message during image embedding. In order to avoid the loss of reconstructed image quality after embedding the secret message, the grouping of codebooks is proposed to calculate the distance between each codeword. If this distance is smaller than the designated threshold TH, then these codewords would be grouped together. Lin *et al.,'*s method does not require each group to have a set of codewords of power of 2, say 2^k, so we could put in k or $k+1$ bits in each codeword, thus providing the flexibility while storing data.

Lin *et al.*[11] proposed three phases: grouping, embedding, and extracting. In the first phase, a new group of codewords is initiated by two most similar codewords that are not belong to any other group. Afterwards, for each codeword which does not belong to any other group, if it is similar to all of the codewords in a certain group, it will be put into that group.

In the second phase, each codeword in a group is assigned a certain sub-message whose length is determined by the number of codewords in that group. Since the number of codewords in the group is not always equal to a power of two, Lin *et al.*[11] adaptively encode each codeword by increasing one extra bit. Afterwards, an index of codeword is replaced by another index of codeword in the same group to embed the sub-message in the index table. In the final extracting phase, the encoded table is generated, and each of the indexes in the index table is matched to the code of its group in order to obtain the decompressed image.



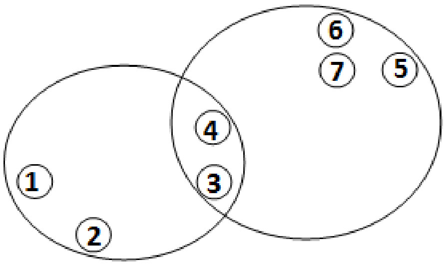
1	(1,2)
2	(2,1)
3	(4,2)
4	(4,3)
5	(7,4)
6	(6,5)
7	(6,4)

Figure 2. Grouping example of overlapped codewords

In the extracting phase, given a codeword and the number of codewords in the group to which the codeword belongs, the embedded sub-message can be extracted from each of the indexes in the index table so the decompressed image can be obtained. In Lin *et al.'s* proposed scheme [11], the more number of codewords in a group, the higher embedding capacity is therefore obtained. In Lin *et al.*'s [11] algorithm outperforms previous algorithms [1, 3] with regard to the embedding capacity and image fidelity.

3. **Proposed scheme.** In the proposed scheme, we followed the VQ-encode scheme and inspected the VQ-compression approach to gain the phenomenon that a good strategy of codewords grouping will keep a better performance in terms of hiding capacity and image quality. Following up this observation, we thus proposed three methods for codewords grouping. Suppose the grayscale host image of sized $W \times H$ were divided into non-overlapping $q \times q$ blocks where $q=4$. Similar to the scheme of Lin *et al.* [11], three phases displayed in the proposed scheme inclusive of codewords grouping, secret embedding, and message extracting. In the first phase, a new group of codewords is initiated by two most similar codewords that are independent. Then, each codeword will be assigned into groups according to one of the following codeword grouping strategies with adaptive threshold TH.

The first phase uses three codeword arrangement strategies to cluster codewords of a given codebook into groups such that the inter-group isolation (or heterogeneity among groups), and the intra-group homogeneity are preserved. The inter-group heterogeneity and intra-group homogeneity can reduce image distortion after embedding secret data into the codewords. A sub-message is embedded into a codeword index by replacing the codeword index with an index that is designated to represent the to-be-embedded sub-message M in non-overlapping blocks.



Group1 center : $X: \frac{1+2+4+4}{4} = 2.75$ $\quad Y: \frac{2+1+2+3}{4} = 2$

Group2 center : $X: \frac{4+4+7+6+6}{5} = 5.4$ $\quad Y: \frac{2+3+4+5+4}{5} = 3.6$

Figure 3. Calculation of group centers

In codeword arrangement strategy, it is expected that the embedding capacity of the secret message could increase while keeping the accepted image quality. Each strategy has its effect on either the embedding capacity or image quality. The first method provides a better grouping effect to achieve higher embedding capacity and greater quality. However, this method requires more time to group the codewords, the second method was thus designed to increase the speed of codewords grouping. This method does not further process the codewords that belong to multiple groups, instead they are being recognized as the same group; therefore, the grouping time would be reduced quite a bit but the image quality would be affected as well. The third method utilizes the concept of codeword's distance mean to group the codewords into the group with the smallest mean. Such method maintains not only the embedding ability of secret message, but the image quality after embedding as well. All the methods will be fully described in next subsection. For simplicity our scheme, some notations are defined as follows:

cw_i: The *i-th* codeword in a given codebook *CB*.

Inx_i: The *i-th* index indicating the index of the codeword cw_i is similar to block *i*.

$|G_g|$: The number of codeword groups, where g is the identification of group with similar codewords and $g \geq 0$.

$G^{-1}(Inx_i)$: The group which index Inx_i belongs.

$CB(cw_z^g)$: The codeword of code book *CB* which the *z-th* codeword of group g belongs.

$|G^{-1}(Inx_i)|$: The number of codewords in the group $G^{-1}(Inx_i)$,

k: The value of $= \lfloor \log_2 |G^{-1}(Inx_i)| \rfloor = \lfloor \log_2 n \rfloor$.

S: The secret message which is a bit-string concatenation to be embedded.

S_i:The *i-th* sub-message of k or $k+1$ bits, for *i = 0, 1, 2, ..., M+1* , where $M = (W \times H)/(q \times q\ )$.

b_k : A bit string of k bit to be embedded.

$d(b_k)$: The unsigned decimal of bitstream b_k.

TH: A fidelity controller used to assign a codeword to the group G_g such that the distance between the codeword and the mean of the group G_g is less than or equal to the threshold *TH.*

Inx_z : The index of the *z-th* codeword designated to represent the to-be-embedded sub-message in group G_g.

cw_z^g: The *z-th* codeword in group G_g.

$ME_g(z)$: The mean of the distance from the *z-th* codeword in group G_g to its neighboring codewords of the same group.

A. Codeword Grouping Phase

For each codeword cw_p, the Euclidean distances *Ed* between all pairs of codewords cw_p and cw_q are being calculated, then we compare it with the threshold TH. If the Euclidean distance $Ed(cw_p, cw_q) \leq TH$, these two codewords can be treated as neighbors to each other, and we tag assign cw_p and cw_q to the same group.

As we go through this grouping phase, the third codeword (cw_3) and the fourth codeword (cw_4) are overlapped between two groups, as shown in Figure 2.

Therefore, we need to put these overlapping codewords to the only one group that fits. However, the large distances between group members would bring distortion to the recovery images after sub-message has been extracted and VQ decoding is employed. In order to group the similar codewords, we proposed three methods that could find a better fit for these overlapping codewords.

The detailed steps are show below:

Method 1: Take each overlapped codewords into consider and assign each overlapped codeword to a group according to the shortest distance between the group center and the codeword, so each overlapped codeword will be assigned to only one group with the most closed neighbors in a group. The steps are:

Step 1: Calculate each group center.

Step 2: Calculate the Euclidean distance *Ed* between the overlapped codeword and the group center, and assign the codeword to the group with the shortest distance. Repeat Steps 1 and 2 until all codewords are assigned.

The first step focuses on calculating the distances between the overlapped codewords and the group centers and assigns the overlapped codeword to a group with the minimum distance between the group center and itself. Since the group center changes after a codeword assignment occurs, the method will recalculate the distances between the remainder of overlapped codewords with group centers. Although the first method makes the closet codewords cluster together and thus maintains good visual quality of the reconstructed

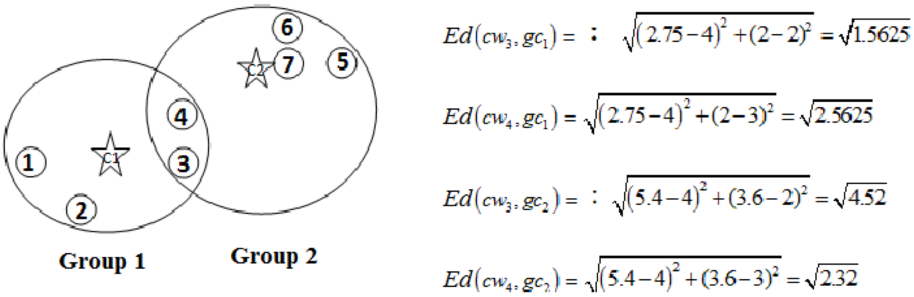


Figure 4. Calculation of distance between group center and overlapped codewords

image after VQ decoding, the time-consuming is heavy since the group center needs to be recalculated every time an overlapped codeword is assigned.

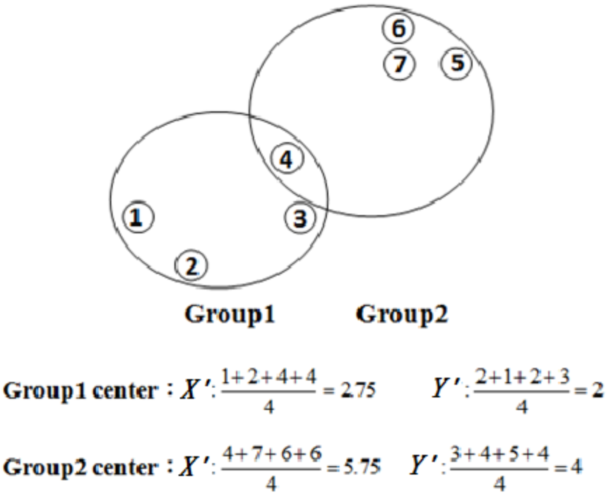


Figure 5. Calculation of new group centers

For example, assume that the first calculated the group centers of Groups 1 and 2 is shown in Figure 3. Then, we calculate the distance between the group center and the overlapped codeword, shown in Figure 4. Here, c_1 and c_2 indicate the group center in group 1 and group 2, respectively. Since the center of Group 1 and cw_3 have the closest distance e.g., $Ed(cw_3, gc_1)$ is minimum, cw_3 is assigned to Group 1.

After that, we need to recalculate the new group center for the first and second groups as displayed in Figure 5, and the new distances between the group centers and the overlapped codewords are shown in Figure 6. Again, the overlapped codewords are being assigned to wherever the shortest distance lies until all codewords are assigned, as shown in Figure 7, where $Ed(cw_4, gc_1) < Ed(cw_4, gc_2)$; therefore, cw_4 is assigned to the first group.

Method 2: The method 1 spends more time to recalculate the new group center, resulting in heavy time-consuming. To minimize time-consuming process, the second strategy of codeword grouping is to make the overlapped codewords as a new group. Following up the same example depicted in Figure 2, the method 2 can assign that cw_3 and cw_4 are grouped together to form a new group (Group 3); however, the grouping effect of the new group would not be decent, where the resultant of codewords grouping is illustrated in Figure 8. Although cw_3 and cw_4 are grouped together, if they are different from each other, the image could be distorted after the image recovery using VQ decoding.

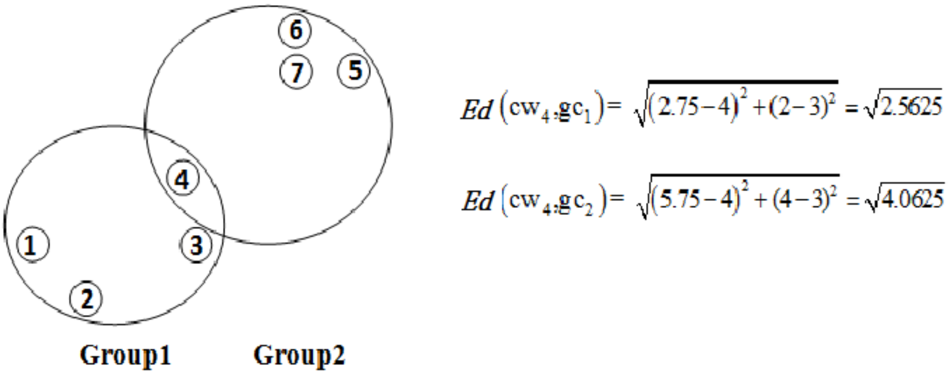


Figure 6. Calculation of the distance of the codewords

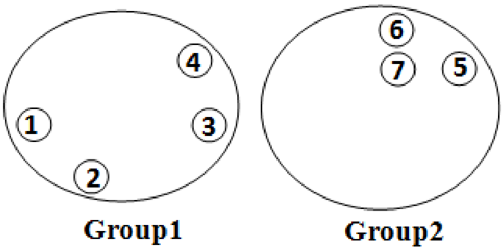


Figure 7. Completion of assignment for Method 1

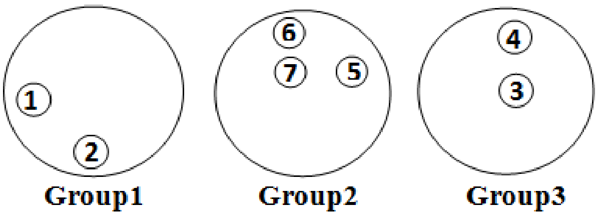


Figure 8. Completion of group assignment for Method 2

Method 3: To reduce the member differences in a group, the codewords are assigned to a group with the smallest mean. For each set of codewords belonging to distinct groups, calculate the mean of the distance from the z-th codeword in group G_g, to its neighboring codewords of the same group. We assign the set of codewords to the group whose distance mean is the smallest. The distance mean $ME_g(z)$ from the z-th codeword in group G_g to its neighboring codewords of the same group is defined in Eq. (2) below.

$$ME_g(z) = {}^{\sum_{j=0,j\neq z}^{n-1} Ed(cw_z^g, cw_j^g)}/_{|G_g|} \tag{2}$$

The codeword grouping step considers both the intra-group homogeneity as well as the inter-group heterogeneity. A smaller threshold brings fewer codewords in a group, and the codewords in the group are more similar; thereby reducing the image distortion and presenting a better visual quality. However, the smaller the threshold is set, the shorter the secret message is obtained. The experimental results in Section 4 demonstrated that the codeword group strategies provide good performance in the next phase of secret embedding.

B. Secret embedding phase

A sub-message is embedded into a codeword index by changing the codeword index to an index which is designated to represent the sub-message to be embedded.

In the proposed scheme, two algorithms are created for concealing the secret message. In the first algorithm, each codeword in G_g can carry a sub-message b_k of k bits or a sub-message b_{k+1} of *k+1* bits if the group size n of G_g is greater than one, where the value of $k = \lfloor \log_2 |G^{-1}(Inx_i)| \rfloor = \lfloor \log_2 n \rfloor$. Initially, we encoded the gray scale image by VQ technique to obtain a corresponding VQ-compressed image. Afterwards, we scanned the VQ-compressed image (also called index table), row by row from left to right, top to bottom, and exploited the following two steps to perform the secret embedding.

Step 1: For each index Inx_i related to block *i,* if the size of group $G^{-1}(Inx_i)$ which the index Inx_i belongs to is greater than one, perform Step 2; otherwise, leave index Inx_i unchanged.

Step 2: Compute $k = \lfloor \log_2 |G^{-1}(Inx_i)| \rfloor = \lfloor \log_2 n \rfloor$, the unsigned decimal value $d(b_k)$ of k bitstring, and the unsigned decimal value $d(b_{k+1})$ of *k+1* bitstring.

Step 2 continues to check the following situations. If the size of the group which index Inx_i belongs to is exactly equal to a power of 2, e.g., $n = 2^k$, we then extract the sub-message b_k of k bits from binary secret message s as the embedded secret s_i; the new index related to the codeword $CB^{-1}(cw_{d(bk)})$ of codebook CB is used to replace the original VQ index Inx_i. Otherwise, when $2^k < n \leq 2^{k+1}$, first check the soon-to-be embedded sub-message b_{k+1} of $k + 1$ bits. There are three cases for the to-be embedded secret Si as follows:

Case 1: If the condition $d(b_{k+1}) > n - 1$ is true, then, extract the sub-message b_k of k bits from binary secret message S as the embedded secret S_i; the new index related to the codeword $CB^{-1}(cw_{d(bk)})$ of codebook CB is used to replace the original VQ index Inx_i.

Case 2: If the condition $2^k \leq d(b_{k+1}) \leq n - 1$ is real, then, extract the sub-message b_{k+1} of $k + 1$ bits from binary secret message S as the embedded secret S_i; the new index related to the codeword $CB^{-1}(cw_{d(bk)})$ of codebook CB is used to replace the original VQ index Inx_i.

Case 3: If the condition $d(b_{k+1}) < 2^k$ is authentic, extract the sub-message b_k of k bits from binary secret message S as the embedded secret S_i; the new index related to the codeword $CB^{-1}(cw_{d(bk)})$ of codebook CB is used to replace the original VQ index Inx_i.

With regards to those unusable codewords (also called isolation codewords) which had not originally been assigned to any group, the second algorithm of the proposed scheme exploits garbage collection to recycle these isolation codewords into a group such that they can be useful to carry more important messages, thus enhances the embedding capacity. The recycling concept is shown in the experimental results later.

C. Extracting message phase

Given the codebook *CB* and the threshold *TH*, the decoder can perform the same process of grouping codewords just as the encoder's classification of the codewords, and the grouping results will also be the same as those obtained by the encoder. Through the following process, the decoder can then extract the embedded sub-messages from the stego-index table and use the codeword to recover the image blocks to 0 for each stego-index (Inx'_i) in the stego-index table corresponding to the *i-th* image block:

Step 1: Compute $k = \lfloor \log_2 |G^{-1}(Inx_i)| \rfloor = \lfloor \log_2 n \rfloor$.

Step 2: Assume that group G_g is the group which the stego-index Inx'_i corresponding to the i-th image block belongs, and the codewords are ordered according to their serial numbers *z's* in group G_g.

Step 3: If the size of the group which index Inx'_i belongs to is exactly equal to a power of 2, e.g., $n = 2^k$, then we have the extracted sub-message b_k of k bits where $d(b_k) = z - 1$.

Otherwise when $2^k < n \leq 2^{k+1}$, there are three cases for the soon-to-be extracted secret S_i according the ranking z of the group $G^{-1}(Inx_i)$ which the codeword with index Inx'_i belongs to.

Case 1.$z > n - 1$, the embedded secret S_i is the binary format of the serial number Z of *k+1* bits.

Case 2. $2^k \leq z \leq n - 1$, the embedded secret S_i is the binary format of the serial number z of $k + 1$ bits.

Case 3. $z < 2^k$, the embedded secret S_i is the binary format of the serial number z of k bits.

Step 4. The *i-th* image block is recovered by the codeword of codebook *CB* related to the index $Inx'i$.

Step 5. Repeat the above steps until all stego-indices are decoded. The secret message can be obtained by concatenating all sub-messages and the final decompressed image can be presented.

The proposed algorithm also recycles the isolation codewords into a group such that the enhanced scheme can benefit the embedding capacity with slight image distortion and large threshold values in some cases. The performance was evaluated by the number of binary bits to be embedded as well as the peak signal-to-noise ratio (PSNR) values, which are shown in the experimental results of the next section.

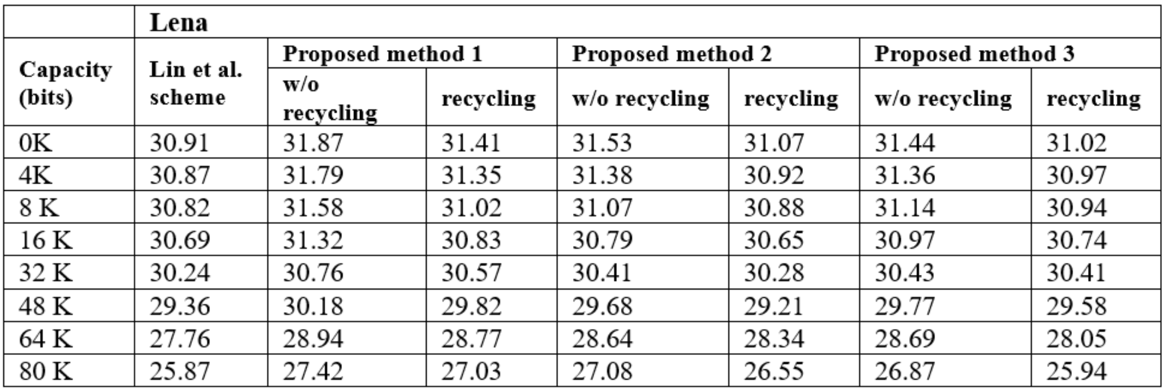
4. **Experimental results.** In the experiment, we used five gray images, such as Lena, F16, Boat, and Pepper as test images, each with a size of 512×512 pixels. Test images are divided into 16,384 non-overlapping blocks of 4×4 pixels for VQ encoding. A codebook of size 512 was generated using the LBG algorithm [9]. The image quality was evaluated using the peak signal-to-noise ratio (PSNR) and the embedded secret data were pure binary messages. The distortion threshold *TH* is considered in the following experiments.

Table 1. Comparison of PSNR values and embedding capacity (in bits) for Lin *et al's* and the proposed schemes inclusive of recycling isolation codewords, with various thresholds (*TH*)

	Lin *et al.* scheme		**Proposed scheme**			**Proposed scheme + recycling**		
TH	PSNR	Capacity (A)	PSNR	Capacity (B)	Gains (B- A)/A	PSNR	Capacity (C)	Gains (C- A)/A
10	30.86	4418	31.29	6668	50.93%	31.07	6724	52.20%
15	30.69	15790	30.88	18525	17.32%	30.71	18741	18.69%
20	30.45	25626	30.62	29180	13.87%	30.46	29491	15.08%
25	30.18	33815	30.39	37719	11.55%	30.18	38149	12.82%
30	29.89	39119	30.17	45236	15.64%	29.94	46470	18.79%
40	29.21	50280	29.54	57462	14.28%	29.58	58003	15.36%
50	28.49	58067	28.72	63108	8.68%	28.64	63508	9.37%
60	27.9	62919	27.91	69025	9.70%	27.74	69401	10.30%
70	27.36	67576	27.39	75943	12.38%	27.22	76648	13.42%
80	26.59	74503	26.74	81557	9.47%	26.28	82037	10.11%
90	26.09	78843	26.27	86871	10.18%	26.01	87482	10.96%
92	25.87	80051	25.91	92425	15.46%	25.61	93028	16.21%

In Table 1, the comparison of embedding capacity (in bits) for Lin *et al.*'s [11] (Column A) and the proposed scheme (Column B) with various thresholds (*TH*), have shown that the proposed scheme markedly outperformed Lin *et al.*'s [11] in terms of embedding capacity. Moreover, Table 1 shows that the proposed scheme is better than Lin *et al.*'s [11] in image fidelity as well.

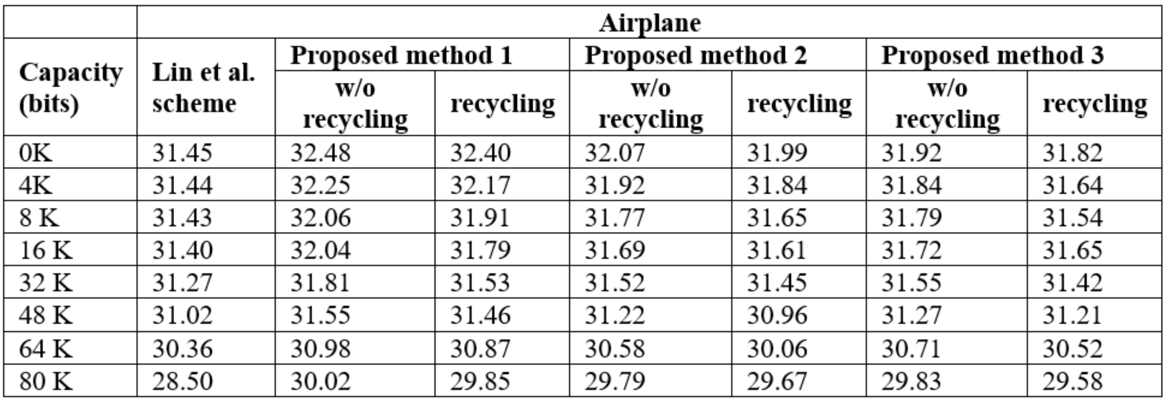
Table 2. Comparison of PSNR values in Lin *et al*'s and our proposed method for embedding image Lena with capacity ranging from 0 Kbit to 80 K bits (*TH=25*)



	Lena						
Capacity (bits)	Lin et al. scheme	Proposed method 1		Proposed method 2		Proposed method 3	
		w/o recycling	recycling	w/o recycling	recycling	w/o recycling	recycling
0K	30.91	31.87	31.41	31.53	31.07	31.44	31.02
4K	30.87	31.79	31.35	31.38	30.92	31.36	30.97
8 K	30.82	31.58	31.02	31.07	30.88	31.14	30.94
16 K	30.69	31.32	30.83	30.79	30.65	30.97	30.74
32 K	30.24	30.76	30.57	30.41	30.28	30.43	30.41
48 K	29.36	30.18	29.82	29.68	29.21	29.77	29.58
64 K	27.76	28.94	28.77	28.64	28.34	28.69	28.05
80 K	25.87	27.42	27.03	27.08	26.55	26.87	25.94

With regard to those unusable codewords (also called isolation codewords), have not been originally assigned to any group. The proposed scheme exploits garbage collection to recycle these isolation codewords into a group so that they may be usable carriers of sub-messages. The recycling concept has worked and the experimental results in Table 1 proved that the codeword recycling can improve the amount of secret bits as well as the PSNR values of image quality. Generally speaking, PSNR values greater than 30 dB are considered to be in good quality and acceptable to most image applications. From Table 1, the results of PSNR values for both schemes of Lin *et al.* and ours can preserve good image quality in human visual system when the threshold is set as 25.

Table 3. Comparison of PSNR values in Lin *et al*'s and our proposed method for embedding image Airplane with capacity ranging from 0 Kbit to 80 K bits (*TH=25*)



	Airplane						
Capacity (bits)	Lin et al. scheme	Proposed method 1		Proposed method 2		Proposed method 3	
		w/o recycling	recycling	w/o recycling	recycling	w/o recycling	recycling
0K	31.45	32.48	32.40	32.07	31.99	31.92	31.82
4K	31.44	32.25	32.17	31.92	31.84	31.84	31.64
8 K	31.43	32.06	31.91	31.77	31.65	31.79	31.54
16 K	31.40	32.04	31.79	31.69	31.61	31.72	31.65
32 K	31.27	31.81	31.53	31.52	31.45	31.55	31.42
48 K	31.02	31.55	31.46	31.22	30.96	31.27	31.21
64 K	30.36	30.98	30.87	30.58	30.06	30.71	30.52
80 K	28.50	30.02	29.85	29.79	29.67	29.83	29.58

In our experimental designs, the embedding capacity of secret message ranges from 0 K bits to 80 K bits with a threshold value set at 25. The PSNR values are compared with that of the scheme of Lin *et al.* using the same embedding capacity for various test images. Tables 2 and 3 provide the comparison results of Lin *et al*'s and our proposed methods for the smooth images Lena and Airplane, in terms of PSNR values, and Table 4 and 5 provides the comparison results for the complex images Boat and Peppers. The results in the first rows of Tables 2, 3, 4, and 5, indicate the proposed scheme gets greater PSNR values compared with the scheme of Lin *et al*'s when no embedding capacity (0 K) is considered.

Table 4. Comparison of PSNR values in Lin *et al*'s and our proposed method for embedding image Boat with capacity ranging from 0 Kbit to 80 K bits (*TH=25*)

	Boat						
Capacity (bits)	Lin et al. scheme	Proposed method 1		Proposed method 2		Proposed method 3	
		w/o recycling	recycling	w/o recycling	recycling	w/o recycling	recycling
0K	31.87	32.59	32.46	32.11	31.95	32.07	31.94
4K	31.84	32.37	32.31	31.98	31.81	31.96	31.90
8 K	31.78	32.27	32.09	31.82	31.67	31.85	31.78
16 K	31.60	32.04	31.92	31.57	31.33	31.69	31.61
32 K	31.03	31.68	31.43	30.92	30.75	31.17	30.98
48 K	30.16	31.05	30.73	30.03	29.84	30.47	30.07
64 K	28.47	30.44	30.07	29.62	29.39	29.94	28.29
80 K	26.16	28.57	28.15	27.55	27.81	27.83	26.83

Moreover, Tables 2, 3, 4 and 5 show that, for a given amount of hidden secret data ranging from 4 K bits to 80 K bits, the PSNR values of the proposed scheme all exceed those of the scheme of Lin *et al.*, meaning that the proposed scheme outperforms in terms of image quality. This is obvious, because we embed the secret bits into codeword indices within a group which has been organized in a condensed manner; namely, both the intra-group homogeneity of codewords and the heterogeneity among groups are preserved by the codeword grouping strategies of proposed scheme which keep good visual qualities for the reconstructed images after VQ decoding.

Table 5. Comparison of PSNR values in Lin *et al*'s and our proposed method for embedding image Peppers with capacity ranging from 0 Kbit to 80 K bits (*TH=25*)

	Peppers						
Capacity (bits)	Lin et al. scheme	Proposed method 1		Proposed method 2		Proposed method 3	
		w/o recycling	recycling	w/o recycling	recycling	w/o recycling	recycling
0K	29.81	30.48	30.41	30.05	29.97	30.13	30.09
4K	29.80	30.32	30.28	29.88	29.81	30.06	29.97
8 K	29.78	30.17	30.05	29.76	29.62	29.99	29.94
16 K	29.73	30.08	29.92	29.61	29.48	29.87	29.85
32 K	29.59	29.81	29.72	29.43	29.25	29.51	29.40
48 K	29.10	29.54	29.36	29.13	28.96	29.27	29.13
64 K	27.71	28.49	28.22	27.74	27.46	27.92	27.86
80 K	25.70	26.71	26.51	25.92	25.67	26.03	25.97

The more embedding capacity needs to be hidden, the more execution time is required. Method 1 put the most similar codewords into a group to increase the embedding capacity and image quality; however, it does require more time to go through the complex computations.

Method 2 is then proposed to lower the complex computations, but the reconstructed image will most likely lose its original form. The third method utilizes the average value to categorize the codebooks, such that the more similar codewords will be clustered together to provide high embedding capacity and good image quality for each group. The execution time for Method 2 is shortest. The time comparisons are shown in Figure 9 which used

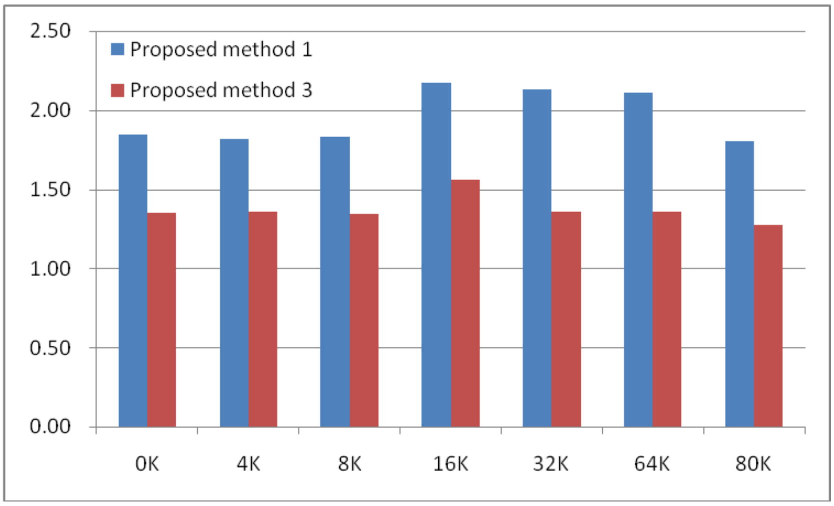


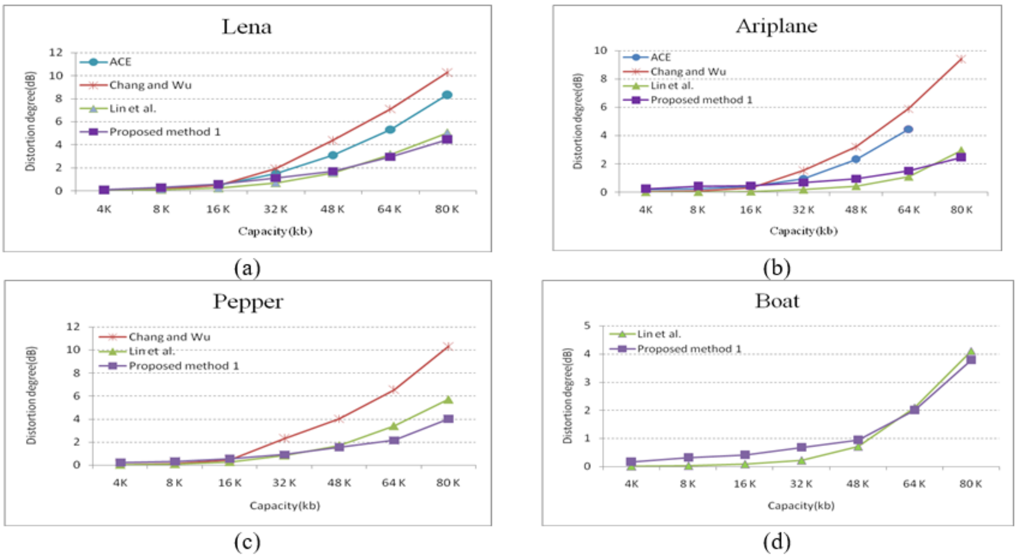
FIGURE 9. Execution time of Method 1/Method 2 and Method 3/Method2 for different capacity

the execution time of Method 2 as the basis. The results in Figure 9 show that the first method spends more time which is around two times than that of the second method; the execution time of third method is around 1.4 times than that of Method 2.

The distortion degree, caused by embedding data, was used to measure the performance of an embedding algorithm and was calculated by following Eq. (3).

$$Distortion\ degree = PSNR_o - PSNR_t \tag{3}$$

where $PSNR_t$ was the PSNR value when a message of t bits was embedded in an image, and $PSNR_o$ is PSNR value using VQ-encoded. A smaller value of distortion degree implies that the reconstructed image is more similar to the original image.



(a) (b)

(c) (d)

FIGURE 10. Comparison of distortion degrees between the ACE method, Chang and Wu's scheme, Lin *et al.*,s scheme and our algorithm: (a) Lena and (b) Airplane (c) Peppers (d) Boat.

In Figure 10 (a), experiments of the distortion degrees are performed on the ACE method [3], Chang and Wu's scheme [1], Lin *et al.*'s scheme [11], and our first method respectively for embedding a message ranging from 4 K to 80 K bits into images Lena, Airplane, Peppers and Boat. We can see that the image distortion of method 1 is significantly smaller than that of the other three. For example, the image distortion of image "Lena" at the embedding capacity of 80 K bits for the ACE method [3], Chang and Wu's scheme [1], Lin *et al.*'s scheme [11], and the proposed Method 1 are 31.24 - 22.91 = 8.33 dB, 32.24 - 21.97 = 10.27 dB, 30.91-25.87 = 5.04 dB, and 31.87 - 27.42 = 4.45 dB, respectively. This means that the image quality of our proposed Method 1 is more similar to its original form when compared with the other three algorithms. In addition, the visual quality of an image using our algorithm is much better than that of using the other three when embedding a message of 80 K bits. We can obtain the following two results: (1) the image distortion of Chang and Wu's scheme is twice of ours; the PSNR values of ours and Lin *et al.'s* scheme is of approximately equal quality. (2) When the embedding capacity reaches to 64K, the image distortion degree is less than that of Lin *et al.'s* scheme.

5. **Conclusions.** In this paper, we proposed three algorithms to group codewords within a given codebook for embedding secret messages into VQ-based images. The strategies of codeword rearrangement consider that the codewords within a group should be as homogeneous as possible, while maintaining heterogeneity between groups. In our proposed scheme, make the more number of similar codewords into a group, the higher the embedding capacity is thus obtained. By means of recycling all isolated codewords and grouping them together to increase the embedding capacity, the extended version of the proposed scheme also surpasses those of ACE method, Chang and Wu's scheme and Lin *et al.,'s* scheme in both the embedding capacity and the fidelity of image quality. .

REFERENCES

[1] C. C. Chang and W. C. Wu, Hiding secret data adaptively in vector quantization index tables, *IEEE Transaction on Vision, Image and Signal Processing,* vol. 153, no. 5, pp. 589-597, 2006.

[2] C. K. Chan and L. M. Cheng, Hiding data in images by simple LSB substitution, *Pattern Recognition*, vol. 37, no. 3, pp. 469-474, 2004.

[3] W. C. Du and W. J. Hsu, Adaptive data hiding based on VQ compressed images, *IEEE Transaction on Vision, Image and Signal Processing*, vol. 150, no. 4, pp. 233V238, 2003.

[4] M. R. Gray, Vector quantization, *IEEE Acoustics, Speech and Signal Processing Magazine*, vol. 1, no. 2, pp. 4-29, 1984.

[5] Y. C. Hu, Grey-level image hiding scheme based on vector quantization, *Electronics Letters*, vol. 39, no. 2, pp. 202-203, 2003.

[6] Y. C. Hu, High-capacity image hiding scheme based on vector quantization, *Pattern Recognition*, vol. 39, no. 9, pp. 1715-1724, 2006.

[7] M. Jo and H. Kim, A digital image watermarking scheme based on vector quantization, *IEICE Transactions on Information and Systems*, vol. E85-D, no. 6, pp. 1054-1056, 2002.

[8] T. Kim, Side match and overlap match vector quantizers for images, *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 170-185, 1992.

[9] Y. Linde, A. Buzo, and R. M. Gray, An algorithm for vector quantization design, *IEEE Transactions on Communication*, vol. 28, no. 1, pp. 84-95, 1980.

[10] Y. Li and C. T. Li, Steganographic scheme for VQ compressed images using progressive exponential clustering, *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance, IEEE Computer Society*, Sydney, Australia, 2006.

[11] C. C. Lin, S. C Chen., and N. L. Hsueh, Adaptive embedding techniques for VQ-compressed images, *Information Sciences*, vol. 179, pp. 140-149, 2009.

[12] Y. C. Lin and S. C. Tai, A fast Linde-Buzo-Gray algorithm in image vector quantization, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 3, pp. 432-435, 1998.

[13] W. J Wang., C. T. Huang, and S. J Wang., Reversible data hiding using neighboring block in VQ-compressed images, *Proceedings of the Third International Conference on Ubiquitous Information Management and Communication (ICUIMC-2010)*, Suwon, Korea, Jan. 2010.

[14] K. Duc and R. Sterling, A reversible steganography scheme for VQ indices based on joint neighboring coding, *Expert Systems with Application,* vol. 42, no. 2, pp. 713-722, 2015.

[15] C. C. Lin, X. L. Liu, S. M. Yuan, Reversible data hiding for VQ-compressed images based on search-order coding and state-codebook mapping, *Information Sciences*, vol. 293, no. 1, pp. 314-326, 2015.

[16] Z. Pan, X. Ma, and X. Deng, New reversible full-embeddable information hiding method for vector quantisation indices based on locally adaptive complete coding list, *IET Image Processing*, vol. 9, no. 1, pp. 22-30, 2015.

[17] J. D. Lee, Y. H. Chiou, and J. M. Guo, Information hiding based on block match coding for vector quantization-compressed Images, *IEEE Systems Journal*, vol. 8, no. 3, pp. 737-748, 2014.

[18] W. J. Wang, C. T. Huang, and S. J. Wang, VQ Applications in steganographic data hiding upon multimedia Images, *IEEE Systems Journal*, vol. 5, no. 4, pp. 528-537, 2014.