Reversible Data Hiding Based on a Sorted VQ Index Table

Cheng-Hsing Yang, Yi-Cheng Lin, Sheng-Chang Wu, Min-Hao Wu
Department of Computer Science, National Pingtung University of Education,
{chyang, bm096114, bm097115, bm097116}@mail.npue.edu.tw

Abstract-*Recently, a reversible VQ-based data embedding scheme which emphasizes that the original VQ compressed codes can be recovered after data extracting was presented by Chang et al. In their scheme, a sorted VQ codebook is divided into* $2^{B-1} \times 3$ *clusters and indexes located in the front one-third clusters are used to embed secret data, where* B *denotes the size of secret data embedded into each VQ index. In this paper, a new reversible VQ-based hiding scheme is proposed. In our scheme, a sorted VQ codebook is divided into* 2^B *clusters and half of clusters are used to embed secret data. Strategies of priorities, indicators, and index exchanging are proposed to improve our scheme further. Under the same sorted VQ codebook, experimental results demonstrate that our data hiding algorithm has higher capacity and better compression rate.*

Keywords: Data hiding, Reversible embedding, Vector quantization, Data clustering.

1. Introduction

As multimedia and the Internet are popular, the problem of protecting transmitted media becomes more and more important. In order to enhance the safety of transmission, technologies based on data hiding [1] have attracted great attention. Data hiding usually embeds secret data into media, such as images and videos, for the purpose of secret transmission or copyright protection. In this paper, images are used as the embedded media. Images before and after data hiding are called cover images and stego-images, respectively.

In recent years, many hiding technologies have been developed based on the VQ (Vector Quantization) [2-4], and some of them have the character of reversibility [5-11]. The reversible data hiding based on VQ generally refers to owning the ability of extracting hidden data and recovering the images into original VQ coding or SMVQ coding.

According to the developed reversible data hiding technologies based on VQ, we put them into three categories by the characters of outputs as follows.

(1) Images as outputs:

After data hiding, some approaches are limited to producing images as outputs [5, 7]. Literature [5] presents a reversible data hiding scheme based on side match vector quantization (SMVQ). Another literature [7] presents a reversible information hiding scheme based on VQ.

(2) Legitimate VQ coding or SMVQ coding as outputs:

After data hiding, a formal VQ coding or SMVQ coding is created as outputs [6, 8]. Generally speaking, approaches in this category require more skills. Literature [6] proposes a reversible embedding scheme for VQ-compressed images that is based on side matching and relocation. Also, in literature [8], a reversible data-hiding scheme based on a modified side match vector quantization (SMVQ) technique is proposed.

(3) VQ coding or SMVQ coding with additional control messages as outputs:

Approaches in this category add control messages into the formal VQ coding or SMVQ coding as outputs [9-11]. Therefore, these approaches usually increase the lengths of coding results. Moreover, because the results are not general VQ codes or SMVQ codes, they are easy to cause attentions and expose the fact of data hiding.

In this paper, a new hiding approach of the third category is proposed. Our approaches can not only recovery the original VQ coding, but also flexibility adjusts the embedding capacity. Compared to Chang et al.'s method [11], our approach has larger capacity. The remainder of this paper is as follows. Chang et al.'s method is introduced in Section 2. Section 3 describes the details of our proposed scheme, and Section 4 shows some experimental results. Finally, some conclusions are given in Section 5.

2. Past Work by Chang et al.

In 2007, Chang et al. provided a VQ-based embedding method which can losslessly recover the VQ index table [11]. In their method, a codebook is partitioned into some clusters and some indexes of the codebook are reserved for acting as indicators. Data are embedded into the VQ index table by transferring index values from one cluster to another cluster and sometimes index values are led by indicators. For a traditional codebook Ψ with N codewords, the codebook is redesigned as codebook Ψ' with $N' = \left\lfloor \frac{N - 2^{B-1}}{2^{B-1} \times 3} \right\rfloor \times 2^{B-1} \times 3$ codewords, where B denotes the size of secret data embedded into each VQ index. Also, the surplus 2^{B-1} index values are used as the indicators. Some standard images have been used to train the codebook Ψ' in their paper and

codewords in Ψ' were sorted by the referred counts in descending order. Then, codebook Ψ' is partitioned into $2^{B-1}\times 3$ clusters with the same size $m=\left\lfloor \frac{N-2^{B-1}}{2^B} \right\rfloor$. The front one-third clusters with highest referred counts are used to embed secret data.

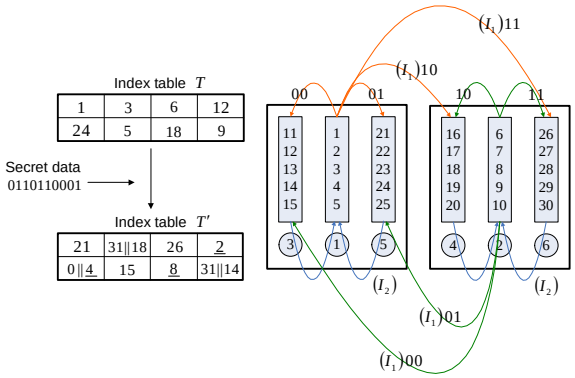


Figure 1. Example of Change et al.'s method for embedding two-bit secret data.

Figure 1 shows an example of Chang et al.'s proposed method for embedding two-bit secret data into each index value in $cluster_1$ or $cluster_2$ ($B=2$). The original codebook Ψ consists of 32 codewords ($N=32$) and the new sorted codebook Ψ' has 30 codewords ($N'=30$). The new codebook Ψ' is partitioned into six clusters, each of them has the same number of codewords ($m=5$). In the case that 2-bit secret data is embedded into each index, two indicators are used. Indicator I_2, valued 0, is carried ahead for an index belonging to $cluster_5$ or $cluster_6$. The other indicator I_1, valued 31, is carried ahead when an index in $cluster_1$ is embedded 2-bit secret data $(10)_2$ or $(11)_2$ and an index in $cluster_2$ is embedded 2-bit secret data $(00)_2$ or $(01)_2$.

For example, the second index 3 in the left table belongs to $cluster_1$, and the 2-bit secret data is $(10)_2$. Therefore, index 3 is transformed into index 18 in $cluster_4$, and indicator 31 is added in front of the index 18. In Figure 1, the underlined value is the one with no secret data embedded. In this case, to represent an index value, indicator 0, or indicator 31 needs $\lceil \log_2 N \rceil$ bits, except that the length of the index value following indicator 0 is $\lceil \log_2 2m \rceil$ bits.

3. Our Proposed Method

In this section, a new reversible hiding method based on VQ is proposed. Compared to Chang et al.'s method which uses the front one-third of codebook Ψ' to embed secret data, our method uses the front half of codebook Ψ' to embed secret data for the purpose of raising embedding capacity. Firstly, some drawbacks of Chang et al.'s method and their improvements are pointed out. Then, our proposed method is introduced by showing its data embedding procedure. Finally, the strategies of using indicators flexibly and fully are discussed, and a skill of exchanging indexes in the sorted codebook is proposed to improve embedding results further.

3.1. Simple improvements for Chang et al.'s method

In this subsection, some drawbacks of Chang et al.'s method are pointed out and improved. In Figure 2, the left diagram shows part of transformation strategies of Chang et al.'s for $B=2$, where the embedded data and used indicator I_1 are depicted. In Chang et al.'s method, indexes in $cluster_1$ and $cluster_2$ are transferred to $cluster_4$ when the embedded data are $(10)_2$, and indicator I_1 is used in the transfer from $cluster_1$ to $cluster_4$. However, the indexes in $cluster_1$ containing higher referred counts should own higher priorities. Indicator I_1 added to indexes of $cluster_1$ will create more overhead than added to indexes of $cluster_2$. Therefore, indicator I_1 should be assigned to the transfer from $cluster_2$ to $cluster_4$ as shown in the right diagram of Figure 2. Similarly, as shown in the bottom of Figure 2, assigning indicator I_1 to the transfer from $cluster_2$ to $cluster_6$ is better than assigning to the transfer from $cluster_1$ to $cluster_6$.

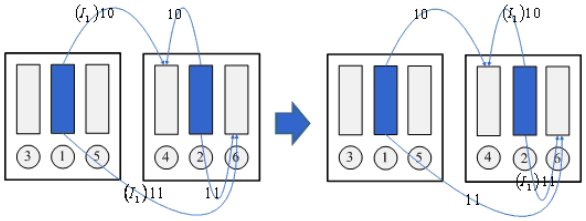


Figure 2. Change the indicator for Chang et al.'s method of B = 2.

Table 1 shows some common codebook sizes N and their corresponding codebook sizes N' for $B=1$ and $B=2$. Another drawback of Chang et al.'s method is that they do not utilize indicators fully. When $B=1$ and $N\in\{128,512\}$, there is one index value unused. Similarly, when $B=2$ and $N\in\{256,1024\}$, there are two index values unused. However, these unused index values can be used as extra indicators to reduce code length further. Figure 3 shows an example of using an extra indicator I_2 for $B=1$. If $N'=126$ codewords are divided into $2^{1-1}\times 3=3$ clusters, each cluster has $m=42$ codewords. In Change et al.'s method, only indicator I_1 is used. Therefore, it needs $\lceil \log_2 m \rceil=6$ bits to

represent the index value following indicator I_1. However, it needs only $\left\lceil \log_2 \frac{m}{2} \right\rceil = 5$ bits if both indicators I_1 and I_2 are used like the strategy shown in Figure 3.

Table 1. Some common codebook sizes N and their N' : (a) B = 1and 1 indicator; (b) B = 2 and 2 indicators.

(a)

N	128	256	512	1024
N'	126	255	510	1023

(b)

N	128	256	512	1024
N'	126	252	510	1020

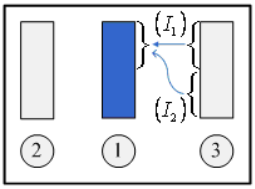


Figure 3. Indicator is used in full for B = 1.

3.2. Our proposed method and the data embedding procedure

Given a codebook Ψ with size N, a sorted codebook Ψ' with size $N' = \left\lfloor \frac{N - 2^{B-1}}{2^B} \right\rfloor \times 2^B$ is created and is divided into 2^B clusters, where B denotes the number of secret bits embedded into each VQ index. All clusters have the same size $m = \left\lfloor \frac{N - 2^{B-1}}{2^B} \right\rfloor$. In our approach, half of indexes in Ψ' will be used to embed secret data and at least 2^{B-1} indexes are used as indicators. For an image H, each block H_i is encoded into index T_i by traditional VQ coding. Then, T_i is transformed into T_i', where T_i' is the corresponding index in some cluster and sometimes includes an indicator. In order to describe our approach easily, a function $trans_j(T_i)$ is used to represent the transformation where index T_i is transformed into the corresponding index in $cluster_j$. Also, if an indicator is carried ahead, $trans_j(T_i)$ is encoded.

Our method and Chang et al.'s method for embedding 2-bit secret data into a block are shown in Figure 4, where our method is in the right side. In our method, 2-bit secret data is embedded when T_i is in $cluster_1$ or $cluster_2$. If T_i is in $cluster_1$, T_i is transferred to $cluster_1$, $cluster_2$, $cluster_3$, and $cluster_4$ when embedded 2-bit secret data is $(00)_2$, $(01)_2$, $(10)_2$, and $(11)_2$, respectively. The above operations will set T_i' to be $trans_1(T_i)$, $trans_2(T_i)$, $trans_3(T_i)$, and $trans_4(T_i)$, respectively. The operations of T_i in $cluster_2$ are similar to T_i in $cluster_1$, expect that indicator I_1 is additionally carried ahead. If T_i is in $cluster_3$ or $cluster_4$, no secret data is embedded and T_i is transferred to the cluster of itself with an indicator I_2. Here, the encoded size of $trans_3(T_i)$ or $trans_4(T_i)$ is $\lceil \log_2 2m \rceil$.

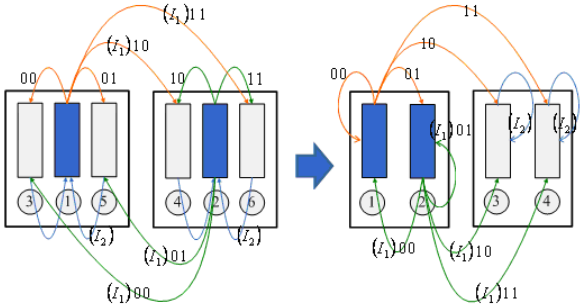


Figure 4. Our proposed method for B = 2 compared with Chang et al.'s method.

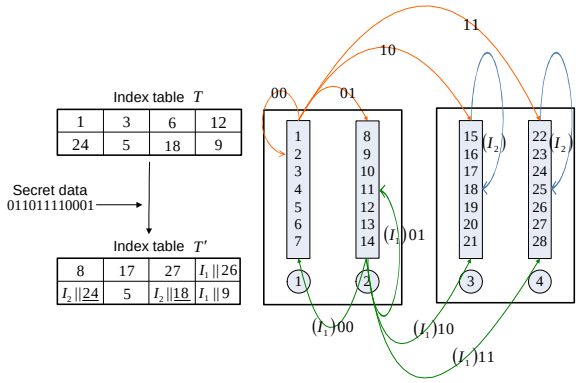


Figure 5. Example of our proposed method for embedding two-bit secret data.

Figure 5 shows an example of our proposed method for embedding two-bit secret data into each index belonging to $cluster_1$ or $cluster_2$ ($B = 2$). This example has the same VQ codebook and index table as the example shown in Figure 1. The original VQ codebook Ψ consists of 32 codewords ($N = 32$) and the new sorted codebook Ψ' has 28 codewords ($N' = 28$). The new codebook Ψ' is partitioned into four clusters, each of them has the same number of codewords ($m = 7$). For simplicity, only indicators I_1 and I_2 are used in this example despite the fact that four indexes can be used as indicators. Indicator I_1 is carried ahead when an index in $cluster_2$ is embedded secret data. Indicator I_2 is carried ahead for an index belonging to $cluster_3$ or $cluster_4$. Due to our

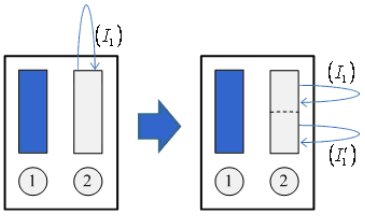
proposed method using the front half of the sorted codebook Ψ' to embed secret data, more indexes can be used for embedding secret data. For example, the forth index 12 in Figure 1 is not embedded secret data, but in Figure 5 is embedded the 2-bit secret data $(11)_2$. Therefore, under the same codebook and index table, our proposed method can embed more secret data.

In Figure 5, the underlined value is the one with no secret data embedded. The above two-bit embedding example shows that the length of the index value following indicator I_1 is $\lceil \log_2 4m \rceil = 5$ bits and the length of the index value following indicator I_2 is $\lceil \log_2 2m \rceil = 4$ bits.

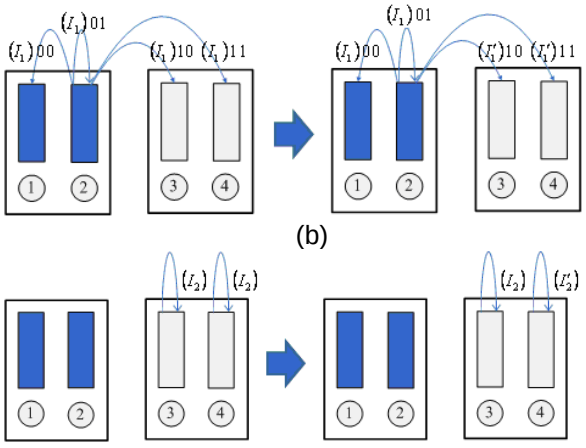
3.3. Strategies of using indicators flexibly and fully

As mentioned in Subsection 3.1, extra indicators can be used to reduce the coding length. Therefore, in our approach, all unused index values are used as extra indicators. That is, our approach uses indicators fully. Figure 6 shows some examples of using an extra indicator to reduce the coding length. In Figure 6 (a), an extra indicator I_1' is used in the transforming rule of $cluster_2$ when $B=1$. Then, the length of $trans_2(T_i)$ is reduced from $\lceil \log_2 m \rceil$ bits to $\left\lceil \log_2 \frac{m}{2} \right\rceil$ bits, where m is the size of a cluster. When $B=2$, Figure 6 (b) shows the case that an extra indicator I_1' is used in the transforming rule of $cluster_2$. Similarly, the length of $trans_j(T_i)$, $j=1,2,3,4$, is reduced from $\lceil \log_2 4m \rceil$ bits to $\lceil \log_2 2m \rceil$ bits. Finally, the case in Figure 6 (c) reduces the length of $trans_3(T_i)$ and $trans_4(T_i)$ from $\lceil \log_2 2m \rceil$ bits to $\lceil \log_2 m \rceil$ bits.

Remention that codebook Ψ' divided into 2^B clusters has $N' = \left\lfloor \frac{N-2^{B-1}}{2^B} \right\rfloor \times 2^B$ codewords and each cluster has $m = \left\lfloor \frac{N-2^{B-1}}{2^B} \right\rfloor$ codewords. At least 2^{B-1} indicators are needed in our approach. Although the number of indicators can be flexibly adjusted in order to reduce the coding length further, it will reduce the number of codewords in codebook Ψ' and result in a poor image quality. Therefore, our approach doesn't consider the strategy of flexibly adjusting the number of indicators.



(a)



(b)

(c)

Figure 6. Some cases of using extra indicators. (a) Using extra indicator I_1' for $cluster_2$ when B = 1. (b) Using extra indicator I_1' when B = 2. (c) Using extra indicator I_2' when B = 2.

3.4. Strategies of exchanging indexes in the sorted codebook

In this section, we proposed the strategies of exchanging indexes in the sorted codebook. Note that the sorted codebook Ψ' is predesigned from some training images and is fixed for all cover images. Therefore, for any processed cover image H, it would occur the phenomenon that some lower referred count codewords are in the front half codebook and some higher referred count codewords are in the behind half codebook. In our approach, it is better that higher referred count codewords are located in the front half codebook. So, we exchange the lower referred count codewords in the front half codebook with the higher referred count codewords in the behind half codebook. The exchanging process is silently included in the whole embedding procedure, where the sorted codebook Ψ' is not physically modified and the overhead keeping the exchanging information is embedded too. By the strategies of exchanging indexes in the sorted codebook, we not only increase the capacity, but also decrease the total transformed bits.

Figure 7 shows an exchanging example, where the size of sorted codebook Ψ' is assumed to be 256 for simple explanation. Firstly, the codeword with the minimum referred count 5 is codeword C_0 in the front half of codebook and the codeword with the maximum referred count 162 is codeword C_{130} in the behind half of codebook. Then, codeword C_0 is exchanged with codeword C_{130}. Ideally, this exchange increases the capacity $162-5=157$ bits. Also, the overhead of recording the exchanging information $(0,130)$ is 14 bits

($\left\lceil \log_2 \frac{N'}{2} \right\rceil + \left\lceil \log_2 \frac{N'}{2} \right\rceil = 7+7=14$). Similarly, (C_2, C_{154}) is the next exchange. The numbers of exchanges also need to be recorded. These exchanges are worth if the increased capacity is larger enough than overhead. In our approach, the overhead is embedded using codebook Ψ' firstly. Then, a dummy codebook Ψ'', which is the exchanging result of codebook Ψ', is used to embed secret data W into the unprocessed blocks of cover image H.

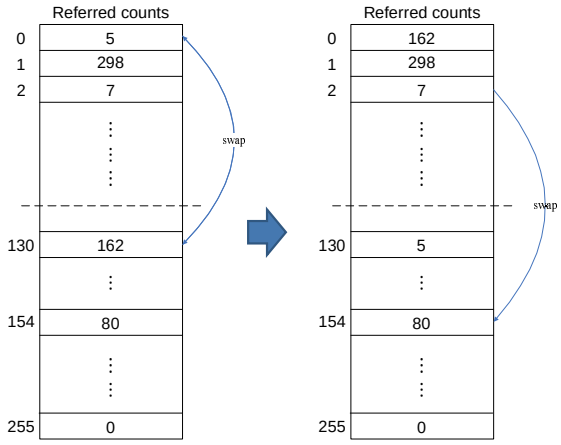


Figure 7. Exchanging indexes in codebook according to referred counts.

Embedding procedure is as follows**:**

Input**:** Cover image H, secret data W, codebook Ψ'

Output**:** Transformed index table T'

Step1: Encode cover image H by sorted codebook Ψ' to produce index table T.

Step2: Exchange some indexes in codebook Ψ' to form a dummy codebook Ψ'' and get the overhead.

Step3: Embed the overhead into index table T by the sorted codebook Ψ'.

Step4: Continue to embed secret data W into index table T by the dummy codebook Ψ'' to produce transformed index table T'.

Extraction procedure is as follows**:**

Input**:** Transformed index table T', codebook Ψ'

Output**:** Recovered index table T, secret data W

Step1: Extract the overhead from transformed index table T' by the sorted codebook Ψ'.

Step2: Use the overhead to transform the sorted codebook Ψ' into the dummy codebook Ψ''.

Step3: Continue to extract secret data W from index table T' by the dummy codebook Ψ'', and recover index table T from index table T'.

4. Simulation and Experimental Results

In this section, some experimental results are demonstrated to show the capacity and VQ rate of Chang et al.'s method and our proposed method. Before starting the test experiments, four kinds of codebooks with sizes of 128, 256, 512, and 1024 codewords were all acquired by the LBG training algorithm, and each codeword was a 16-dimensional vector. The five standard images, 'Lena', 'Jet', 'Boat', 'Sailboat', and 'Toys', involved in the above-mentioned training process are called inside images. To be applied to the experiments of our proposed method, these four codebooks were trained again to generate the appropriate numbers of codewords, such as "124, 252, 508, 1020" for $B=2$, and were sorted according to the referred counts of codewords on the prior five inside images. These derived codebooks were used in the following experiments.

Six popular 512×512 images, Lena, Jet, Toys, Pepper, GoldHill and Zelda, are used as the testing images. By using Chang et al.'s method and our proposed method on these six testing images, Table 2 shows the embedding capacity and VQ rate for $B=2$. Due to the codebook divided into six clusters in Chang et al.'s method for $B=2$, $cluster_1$ and $cluster_2$ have 50% probability to add a $\lceil \log_2 N' \rceil$-bit indicator. Due to the codebook divided into four clusters in our proposed method, $cluster_1$, which contains higher referred count indexes, doesn't add any indicator. As shown in Table 2, our proposed method for $B=2$ has lower VQ rate.

Table 3 shows the experimental results for strategies of exchanging indexes in the sorted codebook. When sorted codebook Ψ'' is divided into four clusters for $B=2$, each index in $cluster_1$ needs $\lceil \log_2 4m \rceil$ bits and each index in $cluster_2$ needs $\lceil \log_2 N \rceil + \lceil \log_2 2m \rceil$ bits, both of them can embed a 2-bit secret data. Also, each index in $cluster_3$ or $cluster_4$ needs $\lceil \log_2 N \rceil + \lceil \log_2 m \rceil$ bits and cannot embed any secret data. Due to exchanging some lower referred count codewords in front half of codebook Ψ' with some higher referred count codewords in behind half of codebook Ψ', indexes in $cluster_3$ or $cluster_4$ may be exchanged with those in $cluster_1$ or $cluster_2$. Hence, our exchange method raises up the embedding capacity, but has the probability of increasing or decreasing the VQ rate as shown in Table 3.

5. Conclusions

Under the same sorted codebook Ψ', this paper presented the strategies of using indicators flexibly and fully and proposed strategies of exchanging indexes in the sorted codebook. Experimental results reveal that our proposed method by using front half of sorted codebook Ψ' to embed data can raise the embedding capacity than that of Chang et al.'s. Also, the strategies of exchanging indexes in the sorted codebook can enhance the sorted codebook Ψ' to even more raise the embedding capacity, especially in outside images.

Table 2. Results of the 2-bit hiding by using different initial codebook sizes 128 and 256

Images		Codebook sizes: 128 (VQ: 0.4375 bpp) 256 (VQ: 0.5 bpp)							
		126 (Chang et al.'s method)		124 (Our method)		252 (Chang et al.'s method)		252 (Our method)	
		Capacity (bits)	Rate (bpp)	Capacity (bits)	Rate (bpp)	Capacity (bits)	Rate (bpp)	Capacity (bits)	Rate (bpp)
Inside images	Lena	27164	0.58	29596	0.52	23884	0.67	26980	0.65
	Jet	27684	0.58	29964	0.51	26170	0.67	28840	0.59
	Toys	30202	0.59	30838	0.47	28606	0.68	30692	0.57
Outside images	Pepper	28372	0.58	29468	0.51	26284	0.67	28818	0.61
	GoldHill	21928	0.59	25302	0.58	19420	0.68	23594	0.70
	Zelda	27258	0.58	29772	0.53	23754	0.67	27304	0.65
average		27101	0.58	29157	0.52	24686	0.67	27705	0.63

Table 3. Results of the 2-bit hiding for the strategy of exchanging indexes in codebook Ψ'

Images		Codebook sizes: 128 (VQ: 0.4375 bpp) 256 (VQ: 0.5 bpp)							
		124 (Our method)		124 (Our swap method)		252 (Our method)		252 (Our swap method)	
		Capacity (bits)	Rate (bpp)	Increasing (bits)	Rate (bpp)	Capacity (bits)	Rate (bpp)	Increasing (bits)	Rate (bpp)
Inside images	Lena	29596	0.52	1010	0.52	26980	0.65	2574	0.64
	Jet	29964	0.51	726	0.50	28840	0.59	1782	0.59
	Toys	30838	0.47	982	0.47	30692	0.57	804	0.57
Outside images	Pepper	29468	0.51	1240	0.50	28818	0.61	1088	0.61
	GoldHill	25302	0.58	4278	0.58	23594	0.70	4694	0.69
	Zelda	29772	0.53	2062	0.52	27304	0.65	4344	0.62
average		29157	0.52	1716	0.52	27705	0.63	2548	0.62

Acknowledgments

This research was partially supported by the National Science Council of the Republic of China under the Grants NCS 97-2221-E-153-001 and the TWISC@NCKU, National Science Council under the Grants NSC 97-2219-E-006 -003.

References

[1] A.P. Fabien, R.J. Anderson, and M.G. Kuhn, "Information hiding – a survey," Proceedings of the IEEE Special Issue on Protection of Multimedia Content, vol. 87, no. 7, pp. 1062-1078, 1999.

[2] H.C. Huang, F.H. Wang, and J.S. Pan, "Efficient and robust watermarking algorithm with vector quantisation," IEE Electronics Letters, vol. 37, no. 13, pp. 826-828, 2001.

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

[4] H.C. Wu and C.C. Chang, "A novel image watermarking scheme based on the vector quantization technique," Computers & Security, vol. 24, pp. 460-471, 2005.

[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, vol. 16, no. 10, Oct. 2006.

[6] C.C. Chang and C.Y. Lin, "Reversible steganography for VQ-compressed images using side matching and relocation," IEEE Transactions on Information Forensics And Security, vol. 1, no. 4, pp. 493-501, Dec. 2006.

[7] C.C. Chang and W.C. Wu, "A reversible information hiding scheme based on vector quantization," in Proceedings of Knowledge-Based Intelligent Information and Engineering Systems (KES 05), Melbourne, Australia, pp. 1101-1107, Sept. 2005.

[8] C.C. Chang, W.L. Tai, and M.H. Lin, "A reversible data hiding scheme with modified side match vector quantization," in: Proceedings of IEEE 19th International Conference on Advanced Information Networking and Applications, Taipei, Taiwan, pp. 947-952, 2005.

[9] C.C. Chang and C.Y. Lin, "Reversible steganographic method using SMVQ approach based on declustering," Information Sciences, vol. 177, no. 8, pp. 1796-1805, 2007.

[10] C.C. Chang and T.C. Lu, "Reversible index-domain information hiding scheme based on side-match vector quantization," J. of Sys. and Soft., vol. 79, no. 8, pp. 1120-1129, Sept. 2006.

[11] C.C. Chang, W.C. Wu, and Y.C. Hu, "Lossless recovery of a VQ index table with embedded secret data," Journal of Visual Communication and Image Representation, vol. 18, no. 3, pp. 207-216, June 2007.