Reversible Data Hiding Base on VQ and Halftoning Technique

Kuo-Ming Hung
Dep. of Information Management
Kainan University
Taoyuan, Taiwan
hkming@mail.knu.edu.tw

Wen-Kai Su
Dep. of Information Management
Kainan University
Taoyuan, Taiwan
esshowtime2000@hotmail.com

Ting-Wen Chen
Dep. of Information Management
Kainan University
Taoyuan, Taiwan
alucardoom@hotmail.com

Li-Ming Chen
Dep. of Information Management
Kainan University
Taoyuan, Taiwan
ms0071799@hotmail.com

Abstract—**Data hiding conceals the existence of secret information while cryptography protects the content of messages. A reversible data hiding is an algorithm, which can recover the original image losslessly from the stego-image after the hidden data have been extracted. In this paper, we propose a method that combines reversible data hiding, halftoning and vector quantization (VQ) technique to embed a grayscale image in other image. Experimental results show the proposed method can payload the larger grayscale secret image while keeping a high visual quality in stego-image and extracting secret image.**

Keywords-halftoning; inverse halftone; reversible data hiding; histogram modification; vector quantization (VQ)

I. INTRODUCTION

Data hiding is a technique that plays an important role in information security. It is referred to as a process of embedding information into a cover object. Its aims at embedding imperceptible confidential information data in cover media such as static images, audios, videos, 3D meshes, and so on.

Although data hiding which cannot perceive between stego-image and cover image by human visual used to protecting copyright, the cover image still be harmed in processing. In the past several years, reversible data hiding approaches are proposed by researchers. The reversibility means not only embedding data but also original image can be precisely recovered in the extracting stage, and thus the applications are used in where medical, military, and some special units they needs.

Nowadays, many of algorithms are proposed for reversi-ble data hiding can be classified into hiding secret data using divide of between nearly pixels [1] and hiding in shifting histogram that produced from gray image [7] two types. For the higher capacity and quality after embedding, the method be improved by many researchers continually [2-6].

The reversible data hiding of histogram was proposed for the first time by Ni et al. in 2006 [7]. After their method proposed, the improving for reversible data hiding of histogram is proposed by changing the histogram which from original gray image onto difference of nearly pixels. Then Kim et al. [9] proposed the method which using embedding level (EL) to increase capacity in 2009. The next year, Luo et al. [8] proposed their algorithm improving the capacity again, and increasing the quality of embedding image at the same time. However, the data can be embedded is binary one or smaller gray image in cover image. For this problem, we will propose a method to improve it.

In this paper, we combine the reversible data hiding of histogram with halftoning, inverse halftone and vector quantization (VQ) to embed gray image. First we compressed gray image into binary image by halftoning. Then for the better quality, we coding the table which train for inverse halftone by VQ, and hiding in cover image simultaneously. Making the gray image which is extracted is more similar to original one.

The rest section of this paper is organized as follows. Sections 2 to 4 are review the reversible data hiding scheme proposed by Luo et al. [8], the halftoning, and VQ individually. Section 5 extensively describes the method our proposed. Experimental results are shown in Section 6. Finally, conclusions are given in Section 7.

II. REVERSIBLE DATA HIDING BASED ON BLOCK MEDIAN PRESERVATION

In this section, we review the principle the method of Luo et al. [8] proposed briefly. Using subsample makes cover image into more equal sub-images. Then find the best pixel which least by subtracting with others in corresponding posi-tion of pixels. Comparing with Kim et al.'s method [9], it gains the better capacity and quality in stego-image.

A. Embedding Method

In embedding process, first subsample size W×H original image into many u×v sub-images. Next find the pixel which value is middle after the pixels be sorted by Eq.(1), and subtract with others to get the difference pixels d_2, d_3, and d_4 where between -255 and 255 in correspond position. Then embed secret data with EL from setting to 0 by Eq.(2) and Eq(3). When EL is 0 in this method, the data embedding with different processes for keeping the meddle value as TABLE I.

Re-subsample the stego-image after plus the middle value in each corresponding pixels when embedding is complete.

$$S_m(\mathrm{i,j}) = p_{\left\lfloor \frac{\mathrm{u\times v}+1}{2} \right\rfloor}(\mathrm{i,j}) \tag{1}$$

$$d_k'(i,j) = \begin{cases} d_k(i,j) + EL + 1 & if\ d_k(i,j) > EL \\ d_k(i,j) - EL - 1 & if\ d_k(i,j) < -EL \\ d_k(i,j) & otherwise \end{cases} \quad (2)$$

$$d_k''(i,j) = \begin{cases} d_k'(i,j) + EL + w & if\ d_k'(i,j) = EL \\ d_k'(i,j) - EL - w & if\ d_k'(i,j) = -EL \end{cases} \quad (3)$$

TABLE I BLOCK CLASSIFICATION BASED ON MEDIAN [8]

Block Type	Definition
Type 1	n_0=1
Type 2	$n_0 \geqq 2, n_l = n_r$
Type 3	$n_0 \geqq 2, n_l < n_r$
Type 4	$n_0 \geqq 2, n_l > n_r$

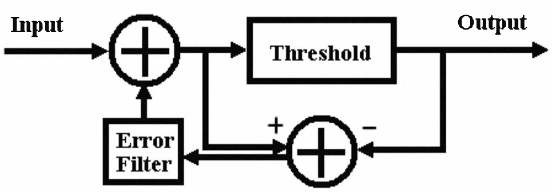


Fig.1. The flow chart of Error Diffusion

B. Extracting and Reversing Method

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. HALFTONING AND INVERSE HALFTONING

Halftoning use the feature of spatial integration that makes non-continuous tone images is watched like continuous tone one in human visual [10] is used in our lives widely, such as magazine, news paper, printer, monitor, fax machine etc.

A. Error Diffusion Algorithm

Error Diffusion (ED) has two main elements in its algorithm. One element is error filter which deciding the weight and position when it diffuse. The sum of weight is 1 that is saving the feature of original image in error filter. The other element is threshold. It always is set half of its color tone. The flow chart of ED is show as Fig.1. First, compare the input pixel with threshold. If equal or greater than threshold, output the 255. Or output 0. Next add the difference which between original and be changed to around with the weight of filter. When processing the next pixel, it is going to use the one which after adding.

In 1976 [11], Floyd and Steinberg proposed the method that made ED into 2-array. The filters they proposed show as Fig.2. X is the pixel that is changing from continuous tone to halftone. The difference X compared with threshold is going to add to around with the weight of filter.

	O	7/16
3/16	5/16	1/16

(a)

1/16	5/16	3/16
7/16	O	

(b)

Fig.2 Error filters by Floyd and Steinberg propose

x	x	x	x
x	x	x	x
x	x	O	x
x	x	x	x

Fig.3 RECT Table of LIH

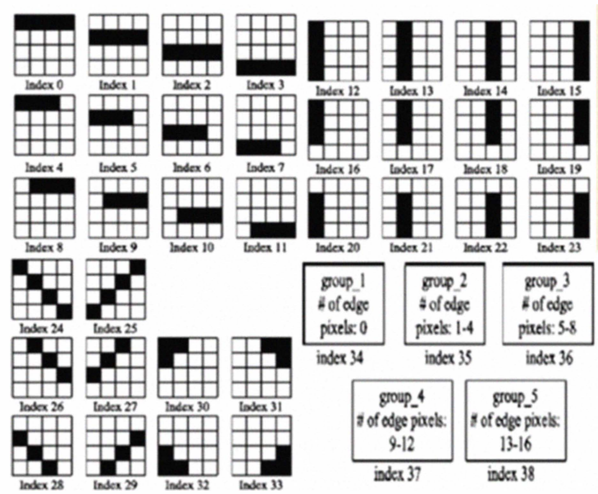
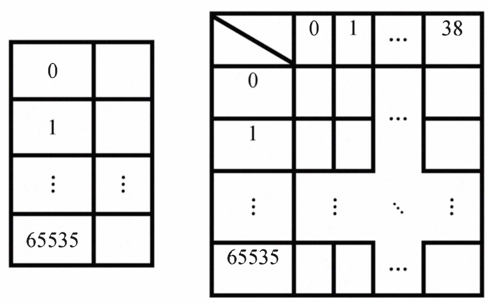


Fig.4 39 edge indexes by Chung and Wu [11]



(a) (b)

Fig.5 The difference between LIH and ELIH

B. Edge-based LUT Inverse Halftoning

After Inverse Halftone which Look-up Table (LUT) is proposed, many researchers whose want to increase quality improve it continually. In 2005, Chung and Wu [12] propose the method that considers the edge to improve the quality of LUT algorithm. In their method, they use RECT table which show as Fig.3 is proposed by Look-up Table inverse halftone (LIH) [13]. Then they proposed 39 edge models that appear possibly as Fig.4.

In these edge models, index 0 to index 11 are horizon edge, index 12 to index 23 are vertical edge, index 24 to index 29 are oblique edge, index 30 to index 33 are corner edge, and index 34 to index 38 are irregular edge.

Chung and Wu used it with Canny edge detection [15] to find the edge of images.

For training an Edge-based Look-up Table (ELIH), step 1 they used the method of LIH to make a Look-up Table between continuous tone and halftone. Next produce the Canny edge images from LIH inverse halftone images by halftoning images that original images transform in step 2. Final step, use original, halftone, Canny edge images to train an Edge-based Look-up Table which the method similarity with LIH. Then an Edge-based Look-up Table of ELIH method can be obtained after training.

It has a problem that is the same with LIH when ELIH method processing. The RECT Table on a model which never happened in training, transforming of this pixel will fail. For this problem, Chung and Wu use the table which is produced by LIH. If it neither happened, using the method which LIH is.

According the experiment of ELIH algorithm, it increases the quality from comparing with LIH algorithm successfully. Nevertheless, the other problem is proposed themselves. LIH algorithm show as Fig.5 (a), it has only 2^{16} to memory the Look-up Table. But the Look-up Table which memory it needs is 39×2^{16} in ELIH algorithm, show as Fig.5 (b). It makes the algorithm that be finishing needs huge computing.

IV. VECTOR QUANTIZATION (VQ)

Vector Quantization is a method which is lossy compression. For fewer stores in images, videos and transports, obtains the lower data rate and rebuilds the signal that has some loss. Vector Quantization is proposed first time by Y. Linde, A. Buzo, and M. Gray in 1980 [16]. This method produces codebook that combining with each representative vectors which call code word symbolically by data training. The size and domain of codebook decide the rate that compress. The generating, optimization, encoding, and decoding are included in codebook of VQ encoding.

First, select original codebook that results in the problem of local optimization after optimization is important in generating.

Next, Linde-Buzo-Gray (LBG) [17] was used to maximize codebook often. Last, divide image to equal size in non-overlapping block. The size of each blocks are the same with coding dimension and regard as vectors. The codebook is combined with code words that are a set of representative sample. The flow chart of coding book is show as Fig.6.

Fig.7 is the flow chart of decoding. When decoding, it only uses the result which is produced in coding to search the sets of index. Then find the corresponding code words that according to original sort in codebook and recover image data. After all of indexes process, the decoding is finished.

V. PROPOSED METHOD

In this section, we proposed a reversible data hiding technique combine with halftoning, inverse halftone and VQ. In embedding, first use halftoning to compress the image from grayscale to halftone. Next, compute the difference between original image and one which inversed by LIH. Employing the VQ compress the difference, and embed it with secret data. Then we can recover the host image better when extracting the secret data by the difference. The flow craft of embedding and extracting show as Fig.8 and Fig.9.

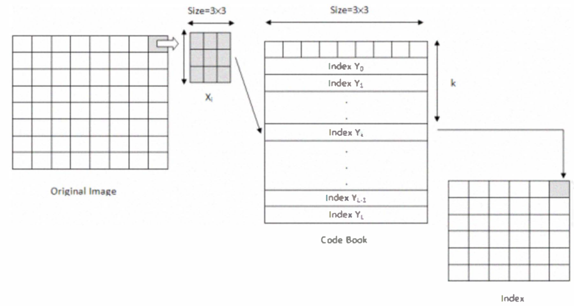


Fig.6 The flow chart of coding (Code words 3×3，codebook Size=L)

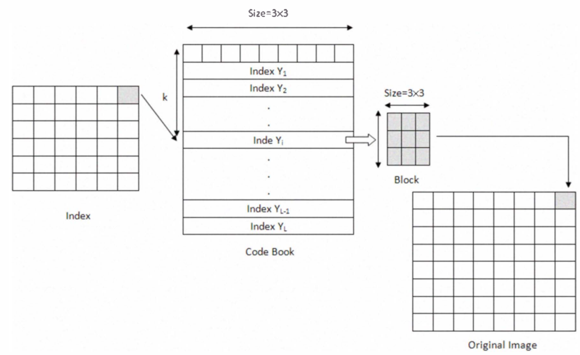


Fig.7 The flow of decoding (Code words 3×3，codebook Size=L)

A. Embedding Method

In this paper, we employ the method which [8] proposed to embed. One of embedding data is the difference which compressed, the other is secret data. The step of embedding as following:

Suppose W is grayscale secret image which size is $W_w \times W_h$. Firstly, make W from continuous tone to halftone WH with employing the Error Diffusion (ED) [10].

Secondly, compute the difference which loss in tone transform WD between W and WI that is WH which inverse halftone by ELIH [14] as Eq. (4).

After computing, we employ VQ [16] to make WV by code WD.

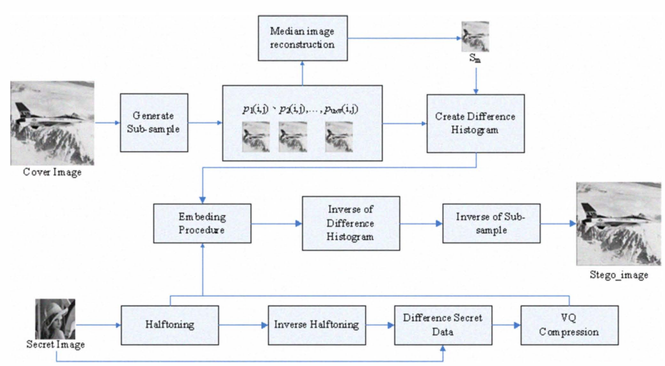


Fig.8 Digital image embedding

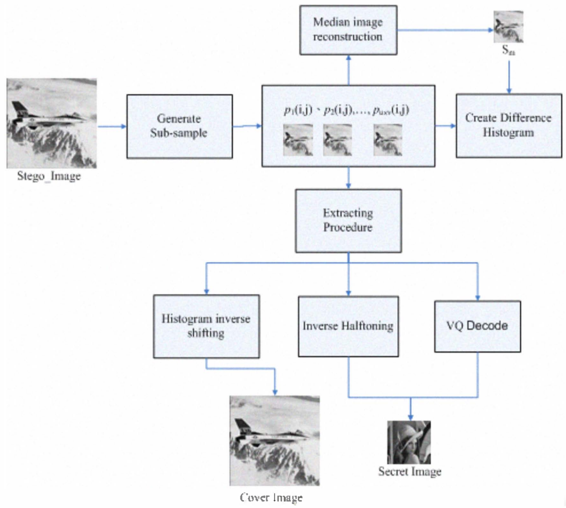


Fig.9 Digital image extraction and Image recovery

$$\mathrm{WD(i,j) = WI(i,j) - W(i,j)} \tag{4}$$

Thirdly, subsample the cover image I which size is $I_w \times I_h$ to $u \times v$ blocks, and each size is $\lfloor \frac{I_w}{u} \rfloor \times \lfloor \frac{I_h}{v} \rfloor$. Then find the middle of $S_{\lfloor \frac{I_w}{u} \rfloor \times \lfloor \frac{I_h}{v} \rfloor}(i,j)$ set S_m after sorting by Eq. (5).

Forth, employ S_m to get the difference of histogram by Eq. (6), show as Fig.10 (a). When histogram is got, as Fig.10 (b) shows, shift the data as Eq. (7) by EL.

Finally, If $EL > 0$, embed the data by Eq. (8), show as Fig10 (c) and (d).

Then to keep S_m is middle, as TABLE I shows, the method has four types to process this way. Where n_l, n_0 and n_r are the number of value that smaller, equal and bigger than middle. The Fig.11 is show as four types processing, and q is the numbers that equal with S_m.

Type 1: This type cannot embed because S_m is middle. Show as Eq. (9).

Type 2: When $n_l = n_r$ and $n_0 \geq 2$, as Eq. (10) shows, embed the data by Eq. (11).

Type 3: When $n_0 \geq 2$ and $n_r > n_l \geq 0$, as Eq. (12) shows, embed the data by Eq. (13).

Type 4: When $n_0 \geq 2$ and $n_l > n_r \geq 0$, as Eq. (14) shows, embed the data by Eq. (15).

After embedding, make difference of histogram sum with their S_m, and re-subsample to finish the stego-image.

$$S_m(i,j) = p_{\lfloor \frac{u\times v+1}{2} \rfloor}(i,j) \tag{5}$$

$$dk(i,j)=(pk(i,j)-Sm(i,j)) \tag{6}$$

$$d_k'(i,j) = \begin{cases} d_k(i,j) + EL + 1 & \text{if } d_k(i,j) > EL \\ d_k(i,j) - EL - 1 & \text{if } d_k(i,j) < -EL \\ d_k(i,j) & \text{otherwise} \end{cases} \tag{7}$$

$$d_k''(i,j) = \begin{cases} d_k'(i,j) + EL + data & \text{if } d_k'(i,j) = EL \\ d_k'(i,j) - EL - data & \text{if } d_k'(i,j) = -EL \end{cases} \tag{8}$$

$$\begin{cases} n_l = \left\lfloor \frac{u\times v}{2} \right\rfloor - 1, n_0 = 1, n_r = \left\lfloor \frac{u\times v}{2} \right\rfloor & \text{if } (u\times v) \bmod 2 = 0 \\ n_l = n_r = \left\lfloor \frac{u\times v}{2} \right\rfloor, n_0 = 1 & \text{if } (u\times v) \bmod 2 = 0 \end{cases} \tag{9}$$

$$\begin{cases} n_0 \geq 2 \\ 0 \leq n_r = n_l \leq \left\lfloor \frac{(u\times v) - n_0}{2} \right\rfloor \end{cases} \tag{10}$$

$$d_k''(i,j) = \begin{cases} d_k'(i,j) + (-1)^{q+1} & \text{if } d_k'(i,j) = S_m(i,j), data = 1 \\ d_k'(i,j) & \text{if } d_k'(i,j) = S_m(i,j), data = 0 \end{cases} \tag{11}$$

$$\begin{cases} n_0 \geq 2 \\ 0 \leq n_l < n_r \leq (u\times v) - n_0 \end{cases} \tag{12}$$

$$d_k''(i,j) = \begin{cases} d_k'(i,j) - 1 & \text{if } d_k'(i,j) = S_m(i,j), data = 1, q < n_r - n_l \\ d_k'(i,j) - (-1)^{q+1} & \text{if } d_k'(i,j) = S_m(i,j), data = 1, q \geq n_r - n_l \\ d_k'(i,j) & \text{if } d_k'(i,j) = S_m(i,j), data = 0 \end{cases} \tag{13}$$

$$\begin{cases} n_0 \geq 2 \\ 0 \leq n_r < n_l \leq (u\times v) - n_0 \end{cases} \tag{14}$$

$$d_k''(i,j) = \begin{cases} d_k'(i,j) - 1 & \text{if } d_k'(i,j) = S_m(i,j), data = 1, q \leq n_l - n_r \\ d_k'(i,j) - (-1)^{q+1} & \text{if } d_k'(i,j) = S_m(i,j), data = 1, q > n_l - n_r \\ d_k'(i,j) & \text{if } d_k'(i,j) = S_m(i,j), data = 0 \end{cases} \tag{15}$$

B. Extracting and recovering Method

In extracting and recovering, first subsample the stego-image I' to $u \times v$ blocks. After, employ Eq. (16) to find the S_m'' which middle of them.

Next, compute with S_m'' to get the difference of histogram, then scan it to extract and recover the secret data and cover image by Eq. (17-20).

When secret data WH and WV are extracted, employ the methods that [12] and [16] proposed to inverse WH into secret image WI' and decode WV into the loss in tone transform WD'.

Finally, process WI' and WD' to get the grayscale secret image W' by Eq. (21).

$$S_m''(i,j) = p''_{\lfloor \frac{u\times v+1}{2} \rfloor}(i,j) \tag{16}$$

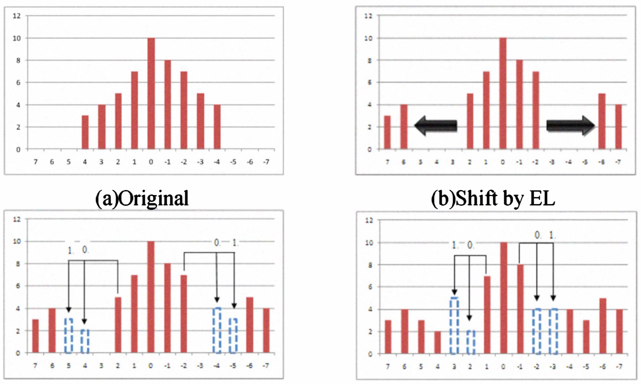
$$data = \begin{cases} 1 & \text{if } d_k''(i,j) \in \{-1,1\} \\ 0 & \text{if } d_k''(i,j) = 0 \end{cases} \quad \text{where EL} = 0 \tag{17}$$

$$d_k'(i,j) = \begin{cases} d_k''(i,j) + 1 & \text{if } d_k''(i,j) = -1 \\ d_k''(i,j) - 1 & \text{if } d_k''(i,j) = 1 \\ d_k''(i,j) & \text{if } d_k''(i,j) = 0 \end{cases} \quad \text{where EL} = 0 \tag{18}$$

$$\text{data} = \begin{cases} 1 & \text{if } d''_k(i,j) \in \{2 \times EL + 1, -2 \times EL - 1\} \\ 0 & \text{if } d''_k(i,j) \in \{2 \times EL, -2 \times EL\} \end{cases} \quad (19)$$
where EL>0

$$d'_k(i,j) = \begin{cases} d''_k(i,j) - EL & \text{if } d''_k(i,j) = 2EL \\ d''_k(i,j) - EL - 1 & \text{if } d''_k(i,j) = 2EL + 1 \\ d''_k(i,j) + EL & \text{if } d''_k(i,j) = -2EL \\ d''_k(i,j) + EL + 1 & \text{if } d''_k(i,j) = -2EL + 1 \end{cases} \quad (20)$$
where EL>0

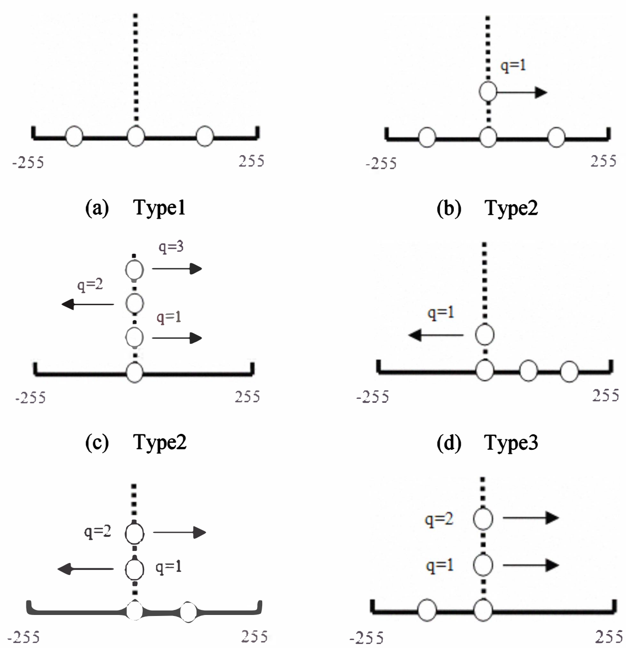
$$W'(i,j) = WI'(i,j) + WD'(i,j) \quad (21)$$



(a)Original (b)Shift by EL

(c)EL=2 embed (d)EL=1 embed

Fig.10 Embedding when EL>0



(a) Type1 (b) Type2

(c) Type2 (d) Type3

(e) Type3 (f) Type4

Fig.11 Embedding type in 2×2 blocks

VI. Experimental Results

Peak Signal to Noise Ratio (PSNR) is one of standard which measure the images that information they loss. As Eq. (22) shows, H and W are image size, $f(x,y)$ and $g(x,y)$ are images that original and be changed. The higher value of PSNR is, the better quality of image we get. Stego-image cannot be noticed directly if PSNR≥30.

When EL=0 to 5 in $u \times v = 2 \times 2$ blocks, the results of PSNR and capacity show as TABLE II, TABLE III, Fig.12 and Fig.13. We can note the more EL we set, the more capacity we can embed, but the less PSNR of image quality we get. Thus, the EL what we set is more important in embedding. In TABLE II and TABLE III, we also can note the capacity of image F16 is more than Tank, because the difference of histogram is affected by image which is smooth or texture.

For upgrading the quality of grayscale secret image, we embed not only binary halftone image, but also the loss information which in halftoning transform. The loss information is less than original one, and it is good in VQ compress. Relation of the size of codebook and code words with image quality and rate of compress are closely. For image quality and size of codebook, we employ the code words that vector is 8×8, the size of codebook and each vector that is compressed need as TABLE IV shows. The image quality can be well when codebook is larger, but the capacity we need is higher.

How to choose the size of codebook is the same important with EL. In TABLE V, WV is the information which loss in halftoning transform, WI is halftone secret image, $VQ + WI$ is their sum, and W' is the quality of image which inverse halftoning and feedback from loss information.

Fig.14 (a) is employing the method which Mese and Vaidyanathan [13] proposed, (b) is employing the method that Chung and Wu [12] proposed, and (c) is our method. Although the capacity gains more 10 thousand, it upgrades the quality in image recover.

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255 \times 255}{\frac{1}{H \times W} \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} [f(x,y) - g(x,y)]^2} \mathrm{dB} \quad (22)$$

TABLE II Capacity and PSNR when EL=0-2 in 2×2 blocks

EL / Test Image	EL=0		EL=1		EL=2	
	Hiding capacity	*PSNR*	*Hiding capacity*	*PSNR*	*Hiding capacity*	*PSNR*
Boat	15309	49.520	42852	43.910	66320	40.852
Elain	15486	49.552	41244	43.897	63741	40.811
F16	36253	49.795	85669	44.728	115581	42.211
Pepper	18350	49.580	52385	44.046	81809	41.148
Scence	16203	49.559	44385	43.936	67375	40.887
Tank	16424	49.561	30578	43.813	51575	40.594

TABLE III Capacity and PSNR when EL=3-5 in 2×2 blocks

EL / Test Image	EL=3		EL=4		EL=5	
	Hiding capacity	*PSNR*	*Hiding capacity*	*PSNR*	*Hiding capacity*	*PSNR*
Boat	86697	38.845	103723	37.390	117774	36.271
Elain	83859	38.775	101266	37.307	116379	36.188
F16	134076	40.559	146056	39.330	154197	38.363
Pepper	105954	39.321	125341	38.029	140213	37.069
Scence	85972	38.864	101566	37.381	114243	36.223
Tank	67370	38.436	81512	36.833	95250	35.592

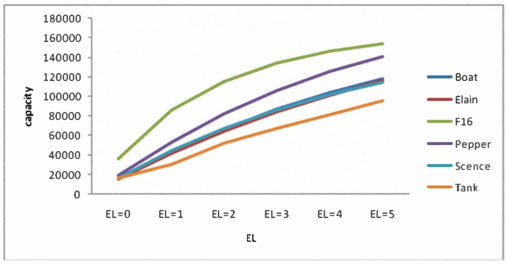


Fig.12 Relation of Capacity and EL

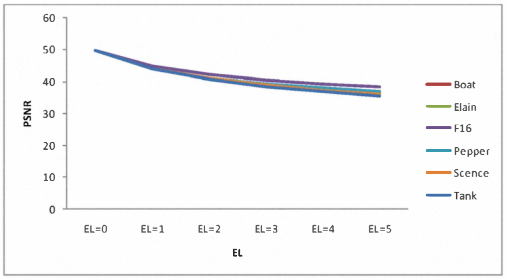


Fig.13 Relation of PSNR and EL

TABLE IV RELATION OF CODEBOOK AND CODE WORDS IN 8×8 VECTOR

Size of codebook	Size of code words(bit)
256	8
512	9
1024	10
2048	11
4096	12
8192	13
16384	14
32768	15
65536	16
131072	17

TABLE V RELATION OF CODEBOOK, WV, WI, VQ+WI, AND W'PSNR

codebook	WV(bit)	WI(bit)	VQ+WI(bit)	W'(PSNR)
256	8192	65536	73728	31.7610
512	9216	65536	74752	31.8434
1024	10240	65536	75776	32.0088
2048	11264	65536	76800	32.3129
4096	12288	65536	77824	33.0645
8192	13312	65536	78848	34.6216
16384	14336	65536	79872	38.9696

(a) LUT[12]
PSNR=29.063
Embed capacity
=65536(bit)

(b) ELIH[11]
PSNR=30.6165
Embed capacity
=65536(bit)

(C)our method
PSNR=38.9696
Embed capacity=79872(bit)

Fig.14 The quality of recovering secret image

VII. CONCLUSION

In this paper, we propose a reversible data hiding technique not only combine the halftoning but also employ VQ compress to improve the image quality. Experimental results show that our method improve the capacity, quality of stego-image and recovering grayscale secret image in extracting. At the same time, the cover image can recover completely. Believing it has great help in medical images, e-passports and so on.

Although our method improves them, it has a problem that makes limit in application. When image is processed by legitimate compression, the information we embed will loss. In future, we will study the methods to improve this problem.

REFERENCES

[1] J.Tian, Reversible data embedding using a difference expansion, IEEE Transactions on Circuits and Systems for Video Technology 13 (8) (2003) 890– 896.

[2] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Transactions on Image Processing 13 (8) (2004) 1147–1156.

[3] K. T. Lin, Digital image hidden in an image using n-graylevel encoding, The 1st international conference on information science and engineering, 2009.

[4] Honge Ren, Chunwu Chang and Jian Zhang, Reversible image hiding algorithm based on pixels difference, Department of information and computer engineering university of northeast forestry, 2009.

[5] TsaiP, HuYC, YehHL. Reversible image hiding scheme using predictive coding and histogram shifting, Signal Processing (2009) 1129–43.

[6] LinCC, Tai WL, Chang CC, Multilevel reversible data hiding based on histogram modification of difference images, Pattern recognition (2008) 3582–91.

[7] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Transactions on ircuits and Systems for Video Technology 16 (3) (2006) 354–362.

[8] Hao Luo, Fa-Xin Yu, Hua Chen, Zheng-Liang Huang, Hui Li, Ping-Hui Wang, Reversible data hiding based on block median preservation, journal of Information ciences 181 (2) (2010) 308-328.

[9] K.S. Kim, M.J. Lee, H.Y. Lee, H.K. Lee, Reversible data hiding exploiting spatial correlation between sub-sampled images, Pattern recognition 42 (11) 489 (2009) 3083–3096.

[10] H. R. Kang, Digital Color Halftoning, SPIE Optical Engineering Press, 1999.

[11] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," Proc. SID 17(2), 75—77 (1976).

[12] K.L. Chung, S.T. Wu, Inverse halftoning algorithm using edge-based lookup table approach, Image Processing, IEEE Transactions on image processing, VOL. 14, NO. 10, OCTOBER 2005.

[13] M. Mese and P.P. Vaidyanathan, Look Up Table (LUT) inverse halftoning, Proc. of IEEE ISCAS, Geneva, June 2000.

[14] N. Damera-Venkata, T. D. Kite, M. Ven kataraman, and B. L. Evans, Fast blind inverse halftoning, in Proc. Image Processing (1998) 64-68.

[15] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-8, no. 11, pp. 679–698, Nov.1986.

[16] Linde, Y., Buzo, A., and Gray, R.M.: An algorithm for vector quantization design, IEEE Trans. Commun. 28 (1) (1980) 84–95.

[17] Lin, Y.C., and Tai, S.C.: A fast Linde-Buzo-Gray algorithm in image vector quantization, IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process 45 (3) (1998) 432–435.

[18] USC.SIPI-The USC-SIPI. Image Database (2008). [online] http://sipi.usc.edu/services/database/Databse.html.