Quality Controllable Progressive Secret Image Sharing – Discrete Cosine Transform Approach

Wen-Pinn Fang

***Abstract*—**This paper proposes a novel secret image sharing method which can control the quality of shares. Based on the theories of secret image sharing and frequency domain transform, a multi-threshold secret image sharing is achieved. The method contains more properties compared with conventional secret image sharing which just has only one threshold. The properties of the proposed method include fault tolerance, small size of shares, secure, multi-threshold and progressive transmission. The method not only has the advantages of conventional secret image sharing, but also is more flexible. Users can set several thresholds in the method. When the number of shares is less than the smallest threshold, no secret will be revealed. While the numbers of shares collected are more than the smallest threshold, a low quality secret image will be revealed. The quality of recovered secret image will be better after collecting more shares. When the amount of shares is over the maximum threshold, the recovered secret image is the same as the original secret image.

***Keywords*—**Secret sharing, Progressive, Share, Fault-tolerance, Threshold, Transmission.

I. INTRODUCTION

In an (r, n) image sharing system [1-3], n shares $\{L_1, L_2, ..., L_n\}$ are created for a given image, e.g., Lena. The image can be revealed when r shares are received, while less than r shares reveal nothing about the image. With only sharing, nobody (even the company organizer) can view the image without attending a public meeting. Therefore, sharing is a valuable safety process especially in a company where no employee/investor alone should be trusted. Significantly, the original image can be discarded after sharing; moreover, each of the r shares is $1/r$ of the size of the given image. Therefore, the sharing process does not waste storage space. Dissimilar to the traditional secret image sharing, the method proposed in this paper can control the amount of information released by means of the amount of shares collected. With this feature, a majority rule will be established. For example, if a dealer has some shares but his share is a noisy image, we can say that he has no right to get the information. If there are two dealers with individual share images, the one with the more informative share has more power.

The rest of this paper is organized as follows: the background knowledge is show in Section II; the method is proposed in Section III; Quality Control is discussed in Section IV, Experimental results are shown in Section V. Finally, the discussion is represented in Section VI.

Manuscript received March 16, 2007; The work was supported by R.O.C. NSC project NSC 96-2218-E-264-001- Revised received May 30, 2007

W.P. Fang, was with Yuanpei University, HsinChu, Taiwan. He is now with the Department of Computer Science and Information Engineering (e-mail: wpfang@mail.ypu.edu.tw).

II. BACKGROUND KNOWLEDGE

Before describing the method, it is necessary to explain some basic knowledge. The basic knowledge includes discrete cosine transform, zig-zag scan and the kernel of secret image sharing. The detail is shown as below:

A. Discrete cosine transformation (DCT)

Discrete cosine transform is one of the most frequently used transformation for image compression, Equation(1) is a 2-D DCT equation for 8×8 non-overlapping block

$$F(u,v)=\frac{c(u)c(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\left(\frac{(2i+1)u\pi}{16}\right)\cos\left(\frac{(2j+1)v\pi}{16}\right)f(i,j) \quad (1)$$

$$c(e)=\begin{cases}\frac{1}{\sqrt{2}}, & if\ \ e=0,\\ 1, & if\ \ e\neq 0\end{cases}.$$

Here, $F(u,v)$ and $f(i,j)$ present a DCT coefficient at the coordinate (u,v) and a pixel value at the coordinate (i,j), respectively. $F(0,0)$ is called the direct current (DC) component, which corresponds to an average intensity value of each block in the spatial domain, $F(u,v)$ is called the alternating current(AC) component, in which $u\neq0$ and $v\neq0$.

B. Zig-Zag scan

We transform a 2-D data to a series of number. For the sake of sorting the DCT coefficient by the importance, we adopt zig-zag scan. The scan order is shown in Fig.1.

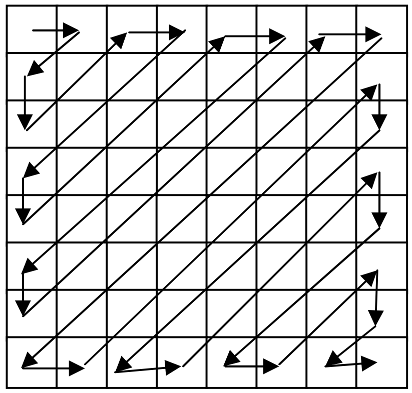


Fig. 1 zig-zag ordering for the proposed method

C. Secret Image Sharing

Secret sharing was first introduced by Shamir [4]. It is a reliable method for the protection of cryptographic key with many good properties. It is a perfect threshold scheme, with the size of each share not exceeding the size of the secret and the security does not rely on unproven mathematical assumptions. It is presented in Fig. 2 as mentioned in Ref [5]:

In 2002, Thien and Lin extend the scheme to image [1], named "Secret Image Sharing". They change the values of a_j into a corresponding pixel value of a secret image. According to their design, the shares are very small. They also proved that secured.

Initialization Phase

1. D choose w distinct, non-zero elements of **Zp**, denoted x_i, $1 \le i \le w$ (this is where we require p≥w+1). For $1 \le i \le w$, D gives the value x_i to P_i. The values x_i are public.

Share Distribution

2. Suppose D wants to share a key $K \in Z_p$.D secretly chooses (indepently at random) t-1 elements of $\mathbf{Z_p}$, $a_1, \ldots a_{t-1}$.
3. For $1 \le i \le w$,D computes $y_i = a(x_i)$, where
$$a(x) = K + \sum_{j=1}^{t-1} a_j x^j \mod p.$$
4. For $1 \le i \le w$,D gives the share y_i to P_i

Fig. 2 The Shamir (t-w)-threshold scheme in $\mathbf{Z_p}$

For example, if the first 3 pixel values in a image are 22,22,18, the first pixel of 3 shares will be

$$22+22\times1+18\times12=62,$$

$$22+22\times2+18\times22=138$$

and

$$22+22\times3+22\times32=34.$$

That is the first pixels of 3 shares are 62,138,34.

Note that all operations are in finite-field (mod 251)

In decoding phase, after collecting the shares, we can get 62,138 and 34. Then we can generate equation as below

$$\begin{cases} a_0+a_1+a_2=62 \\ a_0+2a_1+4a_2=138 \\ a_0+3a_1+9a_2=34 \end{cases}$$

After solving the equations, we can get a_0=22, a_1=22,a_2=18. these solutions are the same as the pixel values of original image.

Peak signal-to-noise ratio (PSNR)

As mentioned in WIKIPEDIA, PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR is most commonly used as a measure of quality of reconstruction in image compression. While calculating PSNR, we determine the mean squared error (MSE) first, which is derived from two $m\times n$ monochrome images, I and K.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left\|I(i,j)-K(i,j)\right\|^2 \tag{2}$$

The PSNR is defined as:

$$PSNR = 20\log_{10}\left(\frac{255}{MSE}\right) \tag{3}$$

Typical values for the PSNR in image compression are between 30 and 40 dB.

III. Proposed Method

In this paper, we proposed a secret image sharing method with multi threshold to recovery secret image. There are two phases shall be explain: encoding and decoding phases. The encoding and decoding method are show as below:

Before describe the methods, we need to define some symbols.

O: original image
t^{th}: threshold
p^{th}: PSNR which after collection t^{th} shares

A. Encoding phase

In this phase, there are two main steps (1) pre-compute : decide the qualities and threshold (2) generate shares. They are state below.

Pre-compute:

Step 1. decide the thresholds and predefined-quality.

Step 2. divide original image O into non-overlapped 8×8 blocks

Step 3. transform every blocks to frequency domain by discrete cosine transform.

Step 4. for every blocks, transfer the coefficients to a series of number by zig-zag scan.

Step 5. let i=64

Step 6. set the value of the series obtain in Step 4. from 64^{th} to i^{th} be zero.

Step 7. transform every blocks to spatial domain by Inverse discrete cosine transform.

Step 8. compute the PSNR between original image the image made from step 6

Step 9. record the PSNR p^{th} that obtain in Step 8.

Step 10. let i=i-1

Step 11. repeat step 5 to Step 10 until i=1

Step 12. look-up the table to decide thresholds t

Generate shares:

Step 1. As show in Fig. 3, there are six steps to generate shares.

Step 2. divide original image into non-overlapped 8×8 blocks

Step 3. transform every blocks to frequency domain by discrete cosine transform and quantization

Step 4. rearrange the coefficients to a series of number by zig-zag scan.

Step 5. partition the series of number by look up the result of section III.

Step 6. share every partition

Step 7. merge all partition into shares

B. Decoding phase

We can get the recovery image by reversing the encoding phase. For example, if the thresholds are 4,5,6 and we collect 4 shares. We can recover 20 numbers. The meaning of this 20 number is the coefficient of DCT from DC to the 20th coefficient. We can get the recovery with about PSNR=35 db. In the same way, if we get 5 shares, we can recover 20+20=40 numbers. The PSNR will increase to near 39 dB. If we get 6 shares, we can get about 40 dB. The loss is because we work in mod 25. If we work in GF(2^8), then , while collecting 6 shares, the recovery will be lossless. But the computing time needed is more than that using mod 251.

C. Example

In Step 4, based on the explanations in section II the number of coefficients shall be the same as the number of thresholds. So we partition the series of number depending on the predefined number of threshold. Here we focus on one of the block. Because every block has 8×8=64 pixels, after Step2 and Step 3, the series will has 64 numbers. For example, we presume the thresholds are 4, 5, 6. We partition the 64 number into 3 clusters; first cluster has 20 numbers, the second cluster has 20 numbers and the third cluster has 24 numbers. For every cluster, we share them by the corresponding threshold. If the first cluster has 20 numbers, and the corresponding threshold is 4, its share size will be 20/4=5. If the second cluster has 20 numbers and the corresponding threshold is 5, its share size will be 20/5=4. If the third cluster has 24 shares, and the corresponding threshold is 24, its share size will be 24/6=4. In the last step, the share size will be 5+4+4=13.

IV. Quality Control

We can control the quality by means of the numbers of shares we collected, and by designing the partition of sharing. As mentioned in Section II, DC is the average intensity value of each block in the spatial domain, and AC is the detail part. It is possible to control the quality of recovery image by setting the number of threshold. It is decided by which coefficient and how many coefficients are adopted as in Section III. In Fig.4, we present an experiment to show the relationship between the number of coefficient and the PSNRs. The quality of image recovered is proportional to the number of coefficient included in the cluster.

There are a question may be arise that if the predefined PSNR in low threshold is smaller than the value that just only remain DC. We can choice different pixel has different threshold.

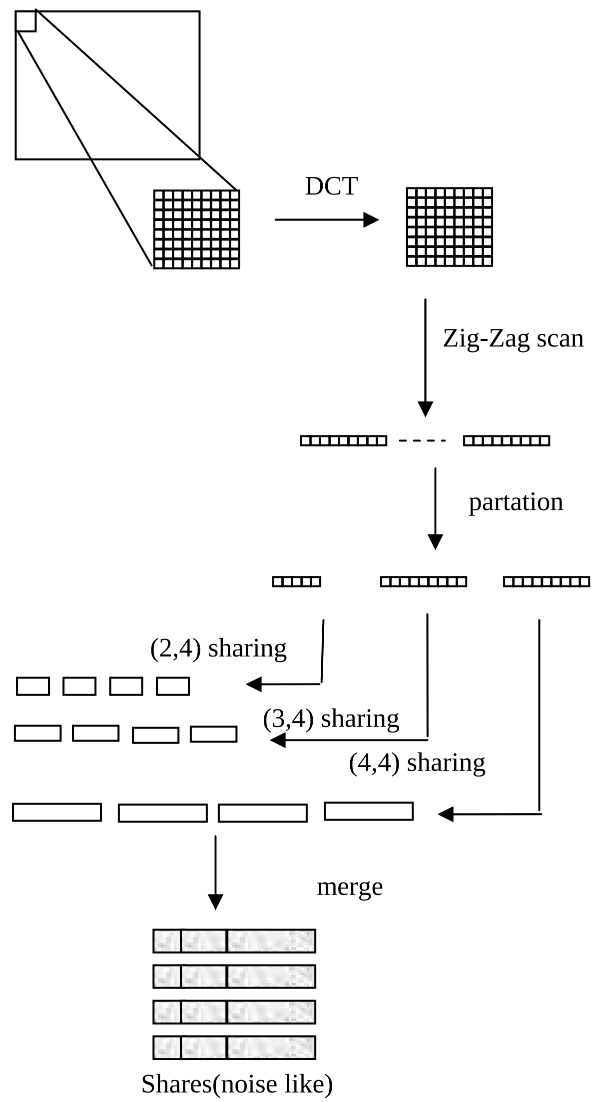


Fig.3 the encoding procedure

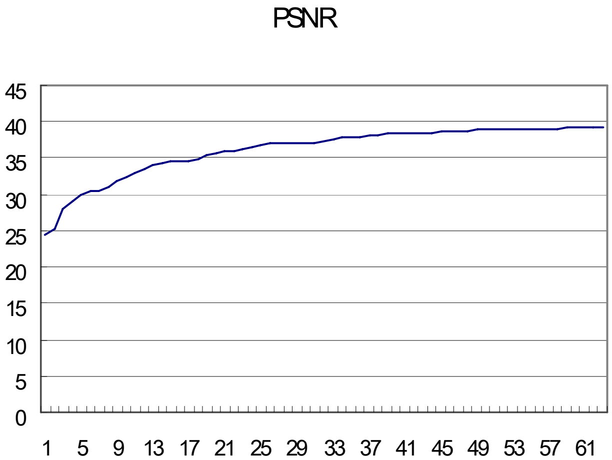


Fig. 4 the number of share vs. quality of recovery image, the vertical axis is PSNR in dB, the horizontal axis is the number of coefficients

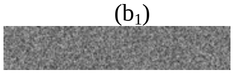
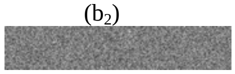
V. Experiment Result

In this section, the experiment result is demonstrated. The original image is 512×512, with the thresholds being 4, 5 and 6. As show in Fig.5, the shares are noise. No one can get any information from it. After collecting any 4 shares, we can recover the image as in Fig.5 (c). After collecting more shares, say , 5 or 6 shares, the quality will be improved as shown in Fig. 5(d) and (e)

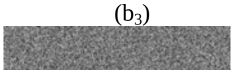
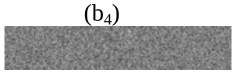
(a)

(b_1) (b_2)

(b_3) (b_4)

(b_5) (b_6)

(c) (d)

(e)

Fig. 5 the experiment (a) is the original image (b1)-(b2) are the shares, (c) is the recover image by any 4 shares (PSNR=35.1)(d) is the recover image by 5 shares (PSR=38.2) and (e) is the recover image by 6 Shares (identical to the original image)

VI. Conclusion and Discussion

In this paper we proposed a progressive secret image sharing method with multi-threshold. The method not only maintains the advantages of conventional secret image sharing, but also has other properties. For example, we can set some threshold to control the user's priority in application. We also can consider the method in Ref.1 as one case of our application. When setting as single threshold, the method we get is the same as Ref.1.

There are some reports discussing progressive secret image sharing. For example, as in Ref.6, it practices progressive transmission of visual sharing. It was applied in transparencies. Our method can be applied in the digital images. S.K. Chen, and J.C. Lin [7] proposed fault tolerance and progressive transmission method. The method adopted the technique of vector quantization (VQ). The recovered image was limited by the VQ approach. That is why their recovered image is not lossless. With our method operated in GF(2^p), we can recover the image loseless if we spend more computing time. In [8-9], they proposed good approaches, but they do not consider quality control of recover image. Finally, we proposed a more flexible method for secret image sharing. In [10], they proposed a scheme for progressive image sharing by bit-plane approach. Our method is different from their method. The proposed method share a image in frequency domain. Because there are many standard image file formats encode images in frequency domain, our method is more suitable than the method mentioned in [10].

In summary, in this paper, we proposed a general progressive secret image sharing method. The conversional secret image sharing is a special case of ours. If we set the amount of thresholds is one, we can get the same result of [1].

References

[1] C.C. Thien and J.C. Lin, Secret Image Sharing, Computers & Graphics, Vol. 26, No. 5, 2002, pp. 765-770.
[2] C.C. Thien and J.C. Lin, An Image-Sharing Method with User-Friendly Shadow Images, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 12, 2003, pp. 1161-1169.
[3] C.C. Thein, W.P. Fang and J.C. Lin, Sharing Secret Images by Using Base-transform and Small-size Host images, International Journal of Computer Science and Network Security, Vol.6, No.7, 2006, pp.219-225.

[4] A. Shamir, “How to share a secret,” CACM, 22(11), pp. 612-613, November 1979.

[5] D.R. Stinson, Cryptography Theory and Pratice, p.327,CRC,U.S.A. ,1995

[6] [6] W.P. Fang and J. C. Lin, Progressive Viewing and Sharing of Sensitive Images, Pattern Recognition and Image Analysis ,Vol.16, No.42006, 8, pp.638-642.

[7] S.K. Chen, and J.C. Lin, Fault-Tolerant and Progressive Transmission of Vector-Quantizated Images”, WSEAS Transactions on Signal Processing , 2006,5.

[8] C.W. Chan and C.C. Chang, A Scheme for Threshold Multi-secret Sharing, Applied Mathematics and Computation, Vol. 166, No.1, pp. 1-14, 2005.

[9] C.C. Chang and R.J. Hwang Sharing Secret Images Using Shadow Codebooks, Information Science, Vol. 111 pp. 335-345, 1998.

[10] S.K. Chen and J. C. Lin, Fault-tolerant and progressive transmission of images, Pattern recognition, Vol. 38, no.12, pp. 2466-2471, 2005.