Chapter 4

The Applications of the SOC Error Correction Techniques

4.1 Introduction

In this chapter, we combine the methods proposed in Chapters 2 and 3 to improve the recovery ability of the proposed sharing methods. That is, the error correction method using SOC-image will be applied to VQ sharing and progressive sharing. The SOC-image not only can be used in traditional image sharing method, but also the VQ sharing. The security of the method of SOC-image will be introduced. It is safe because hackers cannot get information from the SOC-image. The distribution of the values in the SOC-image is attractive so the appeared frequencies will be discussed. The SOC-image also can improve the quality of the recovered secret image in the progressive sharing method proposed in Chapter 2. The organization of this chapter is as follows. Section 4.2 presents some applications of the SOC-image, including the application to VQ, and the image modification using SOC-image. The security of the SOC-image will also be discussed here. In Section 4.3, two examples about the method mentioned in Section 4.2 will be shown. Finally, the summary is made in Section 4.4.

4.2 Some Applications of the SOC-image

SOC of VQ Indices:

By using the VQ sharing method proposed in Section 2.2.1, the code indices of the secret image is calculated according to the codebooks which are generated from the k most significant bits of the host images (the number of k is dependent on the sizes of the secret image and the host images). Then, the code indices will be shared and hidden among these host images which contain the information of codebooks or the mixed information of the codebooks. Before sharing the code indices, the image error correction method by SOC can be applied to the code indices. The SOC-image generation method is the same as that (the basic version) mentioned in Chapter 3, except that the data being processed here are the code indices instead of the gray values. The data of code indices can be treated as an image whose gray values are the code indices. By finding the same code index as the current one, the SOC-image can be generated after processing all the code indices. The size and the bits of each pixel in the SOC-image are decided by the sizes of the secret image, the codebook, and the code word. For example, if a 1024 x1024 secret image is quantized by a codebook which contains 256 codewords and each codeword is 4 x 4 = 16 dimensional, then the size of code indices is 256 x 256 and the SOC-image is 256 x 256, too. Each pixel of the SOC-image is 8 bits since there are 256 codewords in the codebook. If the codebook contains 512 codewords, then the size of each pixel in the SOC-image will become 9 bits; and if each code word is 2 x 2 = 4 dimensional, then the size of the SOC-image will come to 512 x 512. After the generation of the SOC-image, the code indices are shared among the host images using the method proposed in Chapter 2.

In the recovery phase, because of the using of the (r, n) threshold system, the secret image can always be recovered as long as no more than n-r shadow images are damaged. For example, if r = 3 and n = 5 in this system, and if 2 (= n-r) shadows are broken, then the secret image can still be recovered using the other 3 shadows.

However, if the number of the damaged shadows is over *n*-*r*, then the secret image cannot be perfectly recovered by any 3 shadows (because there are only 2 shadows being correct). In this bad condition, the SOC-image can still be applied to the repairing of the code indices. If 2 correct shadow images and one damaged shadow image are received, the code indices can be resolved first (although the code indices may be wrong). Then the error correction method by the SOC-image is applied to the code indices. After the repairing, the information of the code indices may become more precise. It should be noted that there also exist the information of the codebooks in the shadow images. If the shadow image is damaged, not only the code indices, but also the recovered codebook will be incorrect. Thus, the method of error correction will generate some strange blocks during the repairing. Even so, the SOC-image is still useful if the damaged area of the shadow image is local (or the damage is widely distributed but the damaged level is slight.

The recovering method of the SOC-image can be applied to not only the gray values of the secret image and the code indices, but also the codebooks. For example, if the codebook consists of 256 codewords and each codeword is 4 x 4 = 16 dimensional, the codebook can be assumed as a 16 x 256 image, and its SOC-image can be calculated. Thus, SOC technique can still be used when all the shadow images are damaged.

Security Analysis:

The users do not need to worry about that the detailed information of the secret image be retrieved by using the SOC-image alone. It is secure enough because the SOC-image does not record directly the information of the pixel gray values. In fact, the SOC-image just records the searching count of a pixel in the process of finding the previous point whose gray value is the same as that of the current pixel. Thus the

secret image or the secret code indices cannot be recovered by the SOC-image alone. Though there are some contours can be seen on the SOC-image, they can be scrambled by a random seed to prevent this condition. If the stealer wants to guess the information of the original secret image according to the data recorded in the SOC-image, there are 256! different results will be generated and needed to be tried what is the most possible one to produce the secret image(since there are 256 groups in the SOC-image and each group represents one value). It is a very huge amount of work so that retrieving the secret image from the SOC-image alone becomes extremely hard. Only when the shadow images are collected, and after the secret image or code indices (may be damaged) being recovered, the SOC-image can be used.

Compression of the SOC-image

The SOC-image only records the searching count. Because there are usually many smooth areas in an image, this will cause the searching count usually small. That means that the gray value in the SOC-image is small, too. For example, the gray values from 1 to 9 appear more than 100000 times in the SOC-image of “Pepper”. To the contrary, the gray values which are greater than 200 seldom appear (each one appear in less than 100 times). The SOC-image can be efficiently compressed by the lossless compression method (e.g.: Huffman coding). The occurrence frequencies of some values of the SOC-image for “Pepper” and the size of the image after compressed using Huffman coding are shown in Tables 4.1 and 4.2, respectively.

Gray Value	1	2	3	50	51	52
Occurrence Frequencies	19125	20551	19124	524	536	495

100	101	102	207	208	209	210	211	212~255
203	193	200	10	6	5	0	2	0

Table 4.1 The occurrence frequencies of some values of the SOC-image for “Pepper”.

	The original image “Pepper”	The SOC-image for “Pepper”
The image size before the compression(bytes)	262,145	262,145
The image size after the compression(bytes)	246,112	181,241

Table 4.2 The compression results of the image “Pepper” and its SOC-image.

Improving the Recovered Effect of Progressive Sharing

The user can also apply the SOC-image technique on the progressive sharing. According to the progressive sharing method, the recovered image may be lossy if the number of the collected shadows does not reach the threshold to recover the lossless secret image. During the collecting stage, the user can use the SOC-image generated by the original secret image to adjust the recovered lossy image. Although the recovered secret image may become lossless after getting all the shadows, the error correction method of SOC-image is still useful if it needs a lot of time to receive all the shadows and user wants to get a better quality of the secret image immediately. Besides, there are some lossy progressive image sharing methods, too. The

SOC-image provides another method to improve the quality of the recovered image. We use the method mentioned in Section 2.2.2 to show the example. Table 4.2 presents the PSNRs of the recovered images obtained by different numbers of the shadows. It also shows the obtained PSNRs with and without the help of the SOC-image.

Number of received shadows	2	3	4	5
The PSNR(dB) of the recovered image without using SOC-image	32.92	37.28	40.32	lossless
The PSNR(dB) of the recovered image with the SOC-image	47.33	48.56	49.32	X

Table 4.3 The PSNRs of the recovered secret images by different numbers of the shadows with or without SOC-image.

4.3 Experimental Results

In the experiment, the same setting (r=3 and n=5) used in Section 2.3 is applied again. Fig. 4.1 is the secret image "Lena" which is the same as Fig. 2.6, and Fig. 4.2 are the 5 host images which have been shown in Fig. 2.7. After doing the VQ, the code indices of the secret image can be gotten. Thus an SOC-image of the code indices can be constructed by applying the SOC method to the code indices. Since each codeword is 4 x 4 = 16 dimensional, the size of the code indices and the generated SOC-image are both one-sixteenth of the secret image. The SOC-image of the code indices is shown in Fig. 4.3. After the sharing of the code indices and the hiding of the mixed information of the codebook, 5 shadows (shown in Fig. 2.8) can

be generated. Then, the SOC-image is stored distributely. During the recovery phase, it needs at least 3 shadows to reconstruct the codebooks and the code indices, which can generate a lossy version of the secret image (the image shown in Fig. 2.9(b)). However, if the 3 shadows are all damaged or modified, the recovered secret image will be damaged, too. Fig. 4.4 presents three damaged shadow images. There is no way to perfectly recover the secret image by any 3 shadows, for at least one of the three shadows is broken. However, the SOC-image can be helpful in this condition. Notably, the damaged secret image reconstructed from the three shadows shown in Fig. 4.4 is shown in Fig. 4.5, and Fig. 4.6 presents the secret image on which some repairing by the SOC-image has been done. The PSNRs of these two images are 12.97dB and 27.93dB, respectively (the PSNR of the recovered secret image using three non-damaged shadows is 34.01dB). We can see that the image before repairing is seriously damaged and quite noisy. It is hard to see the information from this image. However, after the repairing by the SOC-image, although the method does not completely correct the damaged pixel, the recovered secret image becomes much better. If the shadows are damaged seriously, such as shown in Fig. 4.7, the recovered image (Fig. 4.8) almost reveals no information about the secret. However, after the repairing using the SOC-image, the recovered image (Fig. 4.9) becomes much better than the damaged one. Even if the damaged shadows (Fig. 4.10) includes the 5th shadow (where the mixed information of the codebooks is hidden), the SOC-image is useful for repairing (Figures 4.11 and 4.12).

In the second example, the SOC-image is applied to progressive sharing. The left column of Fig. 4.13 shows three images which are the secret image recovered by the progressive sharing method. From top to bottom, it uses 2, 3, and 4 shadows to recover the image, respectively. The right column of Fig. 4.13 is the recovered secret images using the same numbers of the shadows as their left images, followed by

applying the error correction using the SOC-image. The PSNRs of these six images are listed in Table 4.3.

4.4 Summary

In this chapter, we showed the benefits of the SOC-image. The error correction of the SOC-image can be widely used in gray values, code indices, and others. The correcting ability of using SOC-image of code indices is even better than of gray values. The user does not need to worry about that the SOC-image will blunder out the detail of the secret image. The SOC-image alone is useless. Only with the recovered secret image, the user can use the SOC-image to correct the damaged file. The occurrence frequencies of the values in the SOC-image show that the SOC-image can be efficiently compressed by some lossless method. The application of SOC-image to the progressive sharing is introduced in this chapter, too. If the user does not want to wait another incoming shadow or if some shadows are lost, then he/she can use the SOC-image to improve the already-recovered low-quality secret image. It should be known that even with the SOC-image, the more the number of shadows, the better quality of the recovered image. Thus the property of the progressive sharing still exists. Using the SOC-image does not conflict with the progressive sharing.

Fig. 4.1 The original 1024x1024 secret image (Fig. 2.6).

Fig. 4.2 The five host images of size 512x512 each (Fig. 2.7).

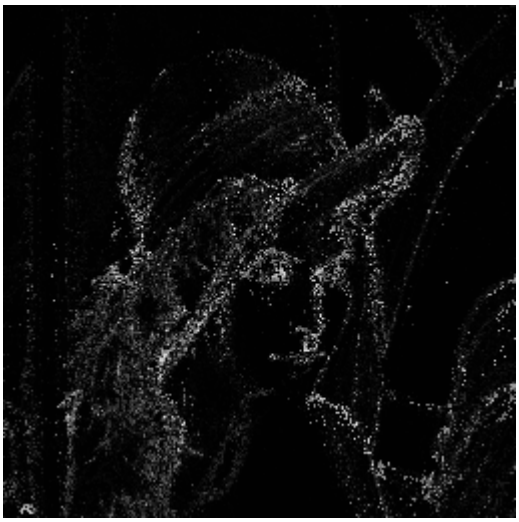

Fig. 4.3 The SOC-image of the code indices of Fig. 4.1.

Fig. 4.4 Three damaged shadow images.

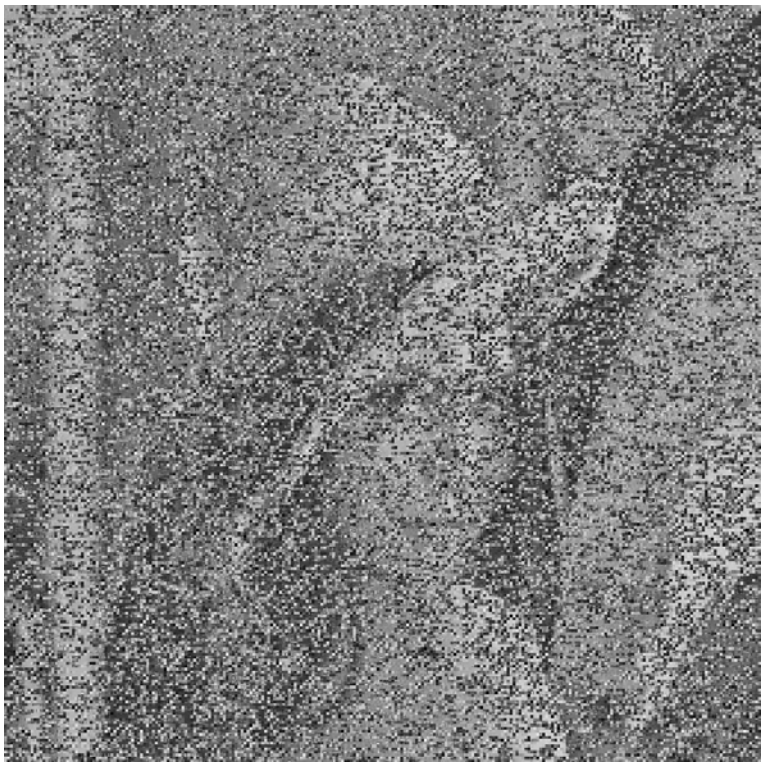

Fig. 4.5 The secret image recovered by the three damaged shadow images (Fig. 4.4). The PSNR is about 12.97 dB.

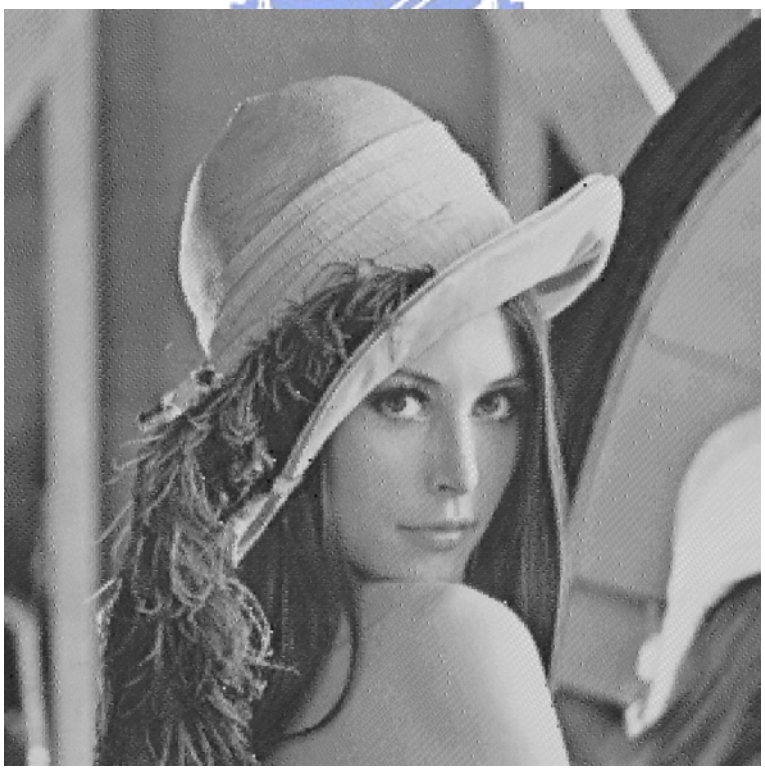

Fig. 4.6 The corrected secret image using the damaged shadow images (Fig. 4.4) and the SOC-image (Fig. 4.3). The PSNR is about 27.93 dB.

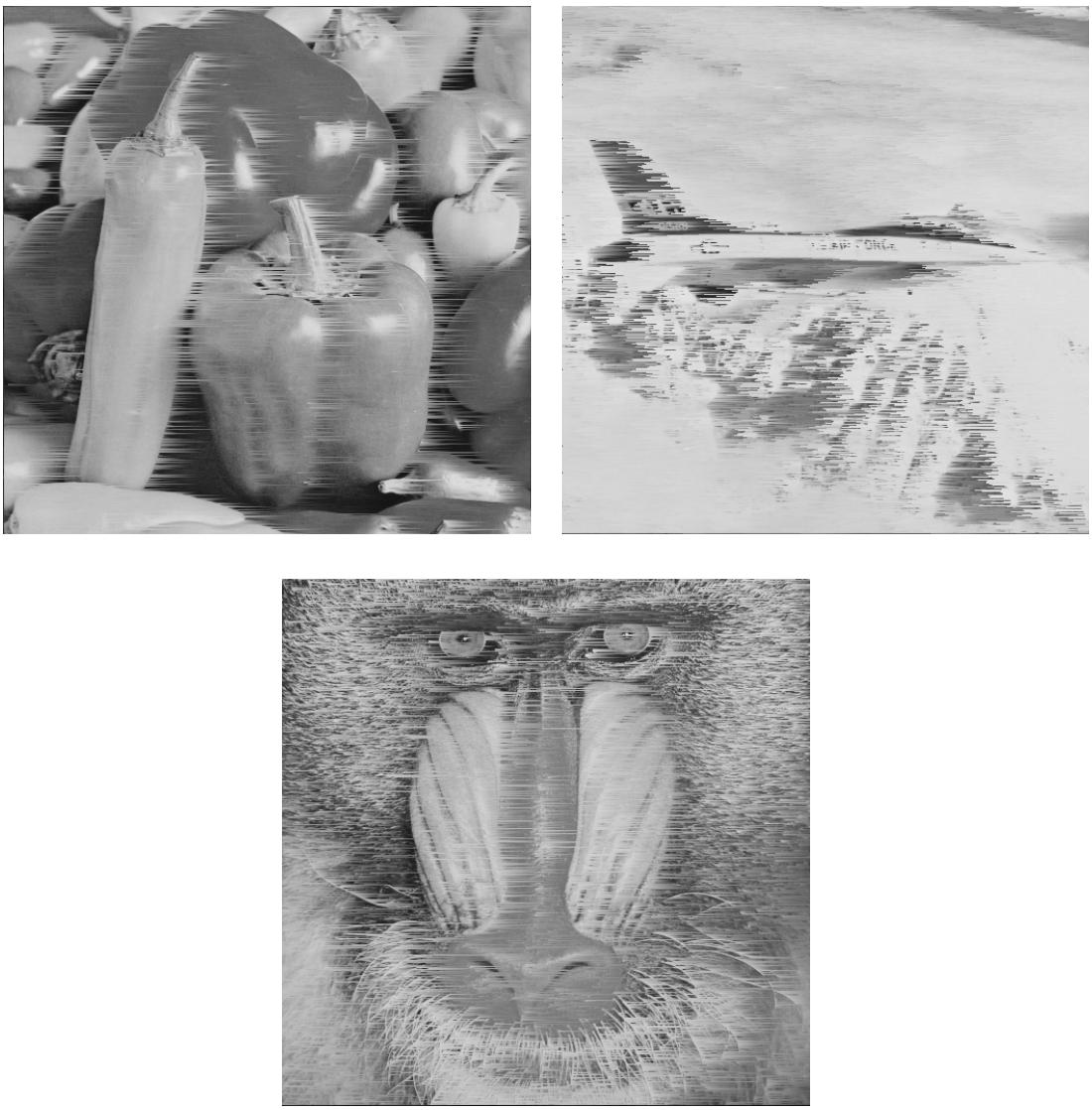

Fig. 4.7 Three seriously damaged shadow images.

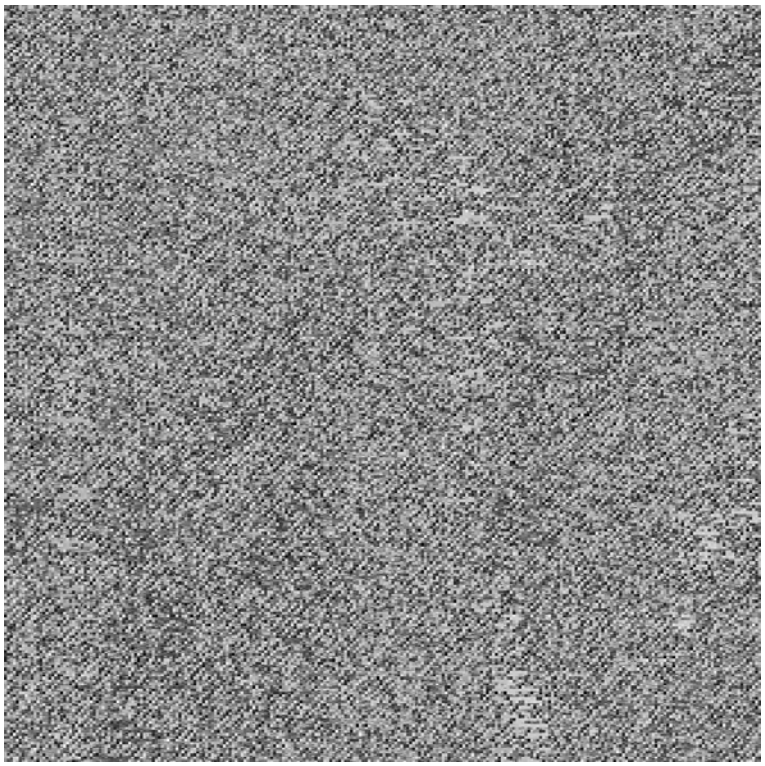

Fig. 4.8 The secret image recovered by the three damaged shadow images (Fig. 4.7).

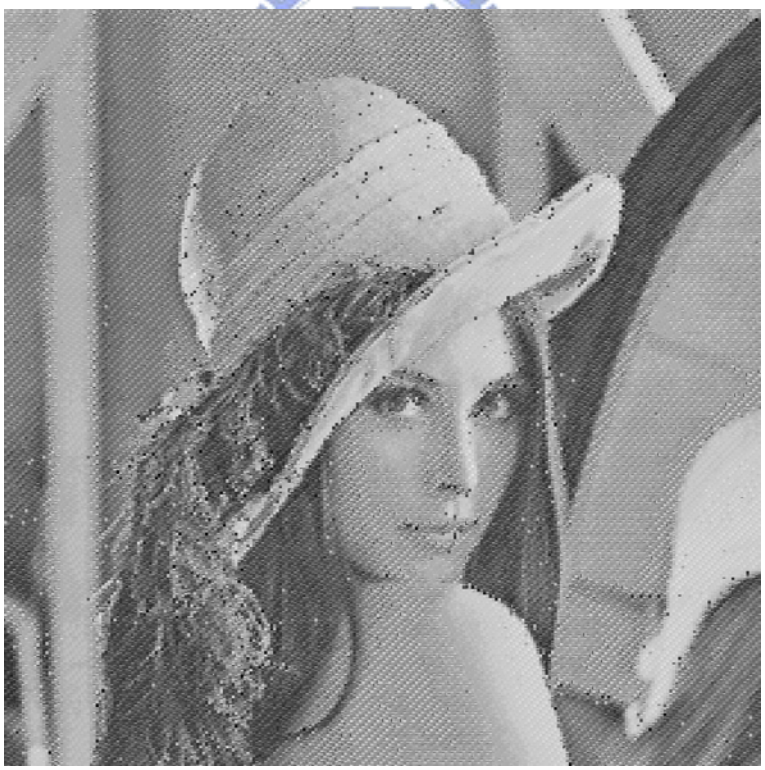

Fig. 4.9 The corrected secret image using the damaged shadow images (Fig. 4.7) and the SOC-image (Fig. 4.3).

Fig. 4.10 Three damaged shadow images included of the 5th shadow.

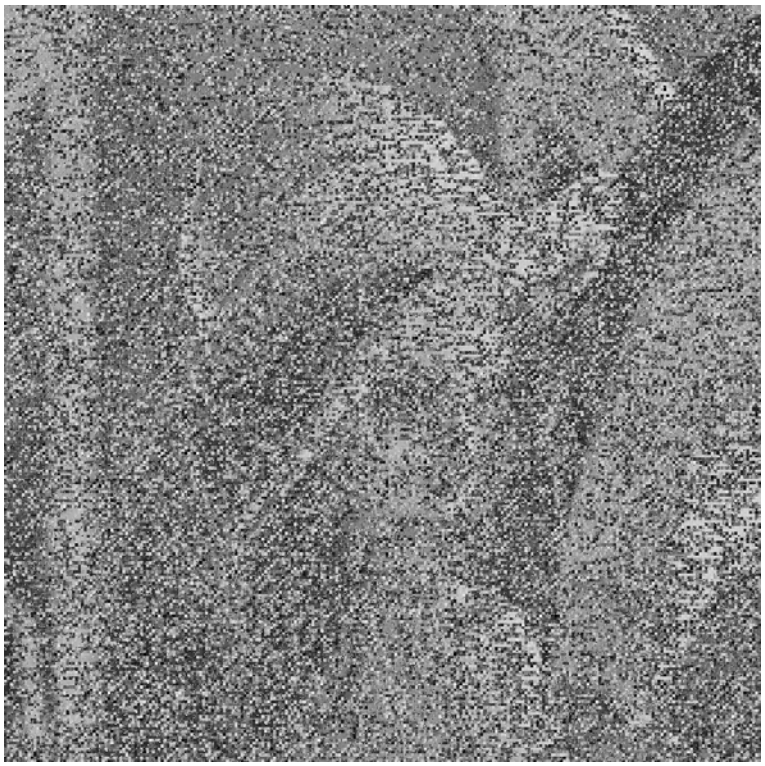

Fig. 4.11 The secret image recovered by the three damaged shadow images (Fig. 4.10).

Fig. 4.12 The corrected secret image using the damaged shadow images (Fig. 4.10) and the SOC-image (Fig. 4.3).

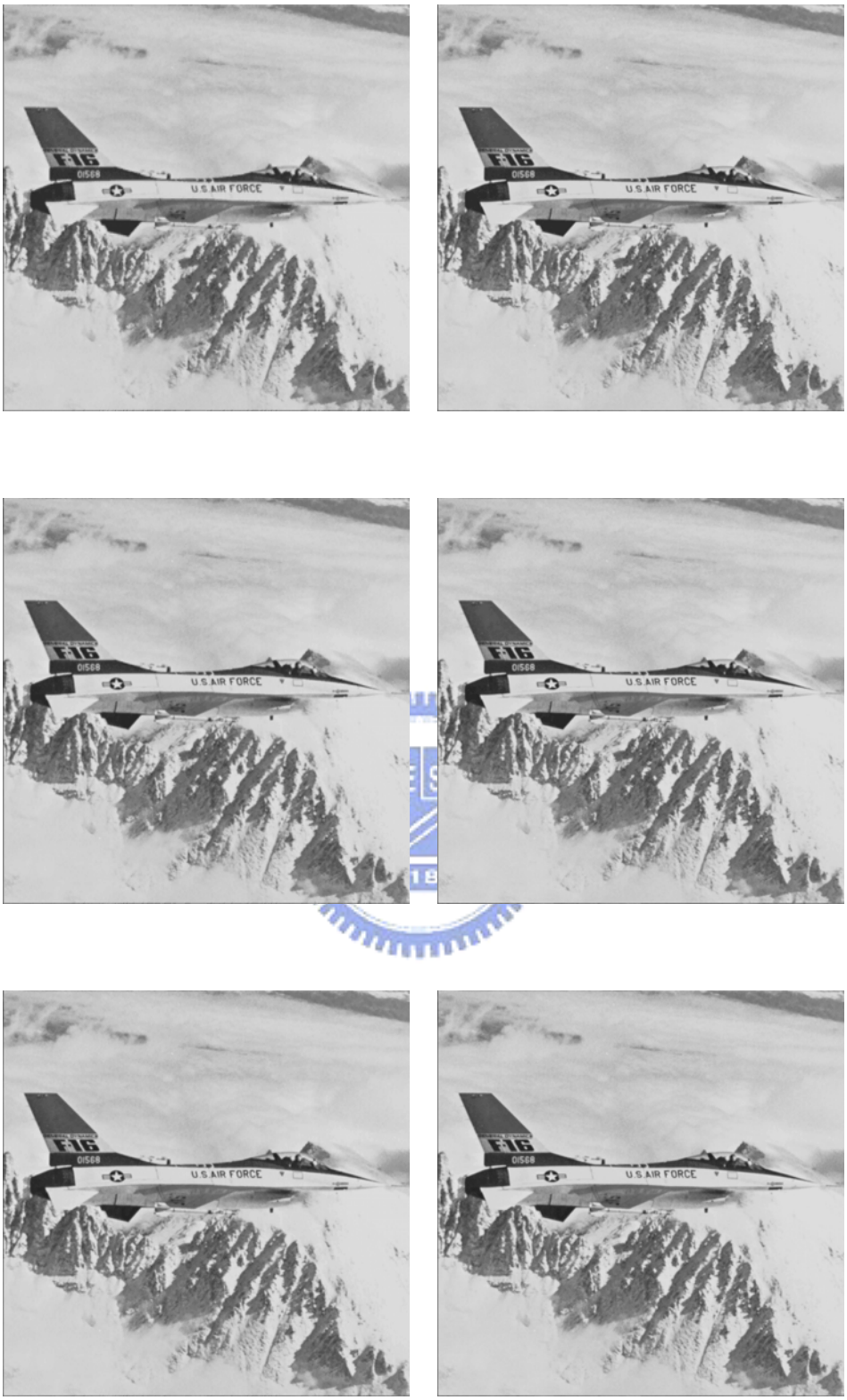

Fig. 4.13 The left column is the recovered secret images using 2, 3, and 4 shadows, respectively, and the right column is the corrected images obtained by using its left image and the SOC-image.