Chapter 2
Various Problems in Visual Cryptography

2.1 Alignment Problems

Pixel expansion is an important parameter for Visual Cryptography Schemes (VCS) [16, 40, 47, 52]. However, most research in literature is dedicated to reduce pixel expansion at pixel level [58], i.e., to reduce number of subpixels that represent a pixel in original secret image. It is quite insufficient since final size of the transparencies of the VCS is affected not only by number of the subpixels, but also by size of the subpixels in the transparencies. However, reducing the size of the subpixels in transparencies is due to difficulties of the transparencies alignment [35, 58].

We notice that, final goal of reducing the pixel expansion is to shorten size of the transparencies that are distributed to the participants [58], because smaller transparencies are easier to be transported. However, the subpixels that are printed on the transparencies affect the final size of the transparencies, in fact, size of the transparencies is the product of size of the subpixels and number of the subpixels in each transparency. Unfortunately, there is a dilemma when one tries to determine the size of the subpixels: when the subpixel size is large, it is easy to align the shares (most publications in the literature require alignment of the shares precisely in the decrypting phase), but large subpixel size will lead to large transparencies. On the other hand, when the subpixel size is small, it is relatively hard to align the shares. From the viewpoint of VCS participants, the goal is to align the shares easily and have small transparencies as well. Table 2.1 shows the relationship between size of the subpixels of the transparencies and the ease to align them from experiential viewpoint.

In this chapter, we take the alignment problem of VCS into consideration [35], and prove that in order to visually recover the original secret image, it is not necessary to align the transparencies precisely. This study is restricted to the case when only one transparency is shifted.

F. Liu and W.Q. Yan, *Visual Cryptography for Image Processing and Security*,
DOI 10.1007/978-3-319-09644-5_2

Table 2.1 The advantages and disadvantages of different sizes of the sub-pixels printed on the transparencies

Size of subpixels	Advantages	Disadvantages
Larger	Easier to align	Larger transparencies size
Smaller	Smaller transparencies size	Hard to align

2.1.1 Precise Alignment of VCS

The shares of visual cryptography are printed on transparencies which need to be superimposed [16, 27, 29, 39, 45, 47, 52]. However, it is not very easy to do precise due to the fine resolution as well as printing noise [49]. Furthermore, many visual cryptography applications need to print shares on paper in which case scanning of the share is necessary [53]. The print and scan process can introduce noise as well which can make the alignment difficult [35, 57]. In this section, we consider the problem of printed and scanned visual cryptography shares. Due to the vulnerabilities in the spatial domain [18], we have developed a frequency domain alignment scheme. We employ the Walsh transform [1] to embed marks in both of the shares so as to find the alignment position of these shares.

Visual cryptography possesses these characteristics:

- Perfect security.
- Decryption (secret restoration) without the aid of a computing device.
- Robustness against lossy compression and distortion due to its binary attribute.

However, the shortcomings of visual cryptography are as salient as its merits. There are three main drawbacks in visual cryptography:

- It results in a loss of resolution [49]. The restored secret image has a resolution lower than that of the original secret image.
- Its original formulation is restricted to binary images [4, 6–8, 10, 12, 31, 37, 44, 46, 59, 61]. For color images, some additional processing such as halftoning and color-separation are required [9, 19–21, 23, 24, 62].
- The alignment of two shares is not easy to perform unless some special alignment marks are provided. The manual alignment procedure can be tedious especially for high resolution images [49].

We will focus on the third problem in this section. The shares of VC printed on transparencies are very difficult to be overlapped with proper alignment even if we ignore the printing errors. A wide variety of applications of visual cryptography would require the printing of the shares on paper like that of documents, checks, tickets, or cards. In such cases, scanning of the printed shares is inevitable for restoring the secret. The scanned shares (with printing, handling, and scanning errors) have to be superimposed in order to reconstruct the secret image which could be a photo, code or other such important information.

In this section, we concentrate on the applications of visual cryptography. i.e., to obtain the precise position of scanned shares which requires rotation and alignment correction. Putting alignment marks in the spatial domain is extremely vulnerable to cropping and editing. Therefore, we use the Walsh transform [1] domain to embed perceptually invisible alignment marks. We show that the Walsh transform helps in recovering the marks in spite of noise and we can precisely align the scanned shares to recover the secret.

In order to carry out the alignment, initially a spatial tag is marked beside the shares. In Fig. 2.1, we put a cross beside each share. For restoring the secret, the two crosses need to be precisely overlapped. If this is done, the secret image will be revealed. Another solution to this problem is by utilizing the scheme [3, 28]. This scheme shares a secret by using two protection images B and C. The procedure of visual cryptography is performed as: $A = B' \bigoplus C'$ where the secret A is divided into two shares B' and C' using VCS scheme. On these shares B' and C', images B and C are also visible. During restoration, images B and C are aligned to make them disappear (by canceling) revealing the secret in the process. An example of this technique is shown in Fig. 2.2, the cross beside the shares are the marks in Fig. 2.1.

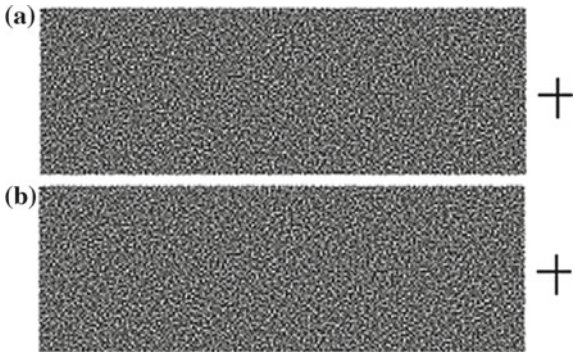

Fig. 2.1 Cross alignment for visual cryptography

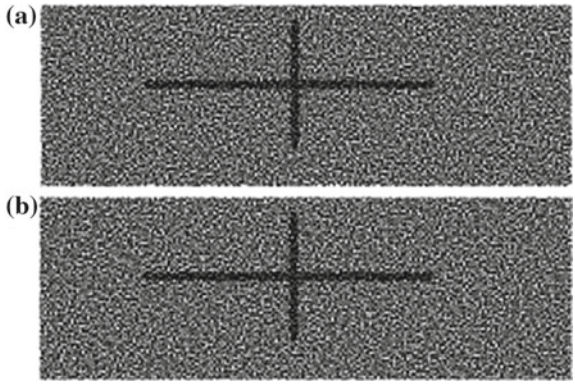

Fig. 2.2 Cross alignment by using extended visual cryptography

Actually, Figs. 2.1 and 2.2 belong to the same class of techniques since they both work in the spatial domain. The problem with this class is that the alignment marks are visible to an attacker and thus can be easily removed by cropping or localized image alteration. We therefore explore the alternative idea of using marks in the frequency domain. In particular, we consider the use of the Discrete Walsh Transform [1], which is useful for pulse signals and is distinct from the Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT) [1]. Walsh functions are a complete set of orthogonal functions with the value being only -1 and 1. We use the 2D Discrete Walsh Transform:

$$\omega_{xy}(u, v) = \frac{1}{N_x}\frac{1}{N_y}\sum_{x=0}^{N_x-1}\sum_{y=0}^{N_y-1} f(x, y) \cdot (-1)^{\alpha} \tag{2.1}$$

$$f(x, y) = \sum_{u=0}^{N_x-1}\sum_{v=0}^{N_y-1} \omega_{xy}(u, v) \cdot (-1)^{\alpha} \tag{2.2}$$

$$\alpha = \sum_{r=0}^{P_s-1} x_r \cdot u_r + \sum_{s=0}^{P_t-1} y_s \cdot v_s \tag{2.3}$$

where $f(x, y)$ is a pixel value of the image, (x, y) is its position, $\omega_{xy}(u, v)$ represents the transform coefficients, $N_x = 2^{P_x}$, $N_y = 2^{P_y}$, (P_x and P_y are positive integers), x_r, u_r, y_s and v_s are either 0 or 1. (i.e., one bit of x, u, y and v, respectively).

Unlike the Walsh transform [1], transforms [1] like DFT, DCT, and DWT are mainly used for continuous tone color images [20, 21, 23]. The results of applying these three transformations to a VC share is shown Fig. 2.3. In Fig. 2.3, the left image is a VC share. The subsequent images show the result of applying the Walsh, DCT, and the DFT transforms. The differences are quite apparent. Note that the bottom-left rectangle of the image for the Walsh transform is totally dark. This information can be exploited in removing noises by filtering the coefficients in this quadrant.

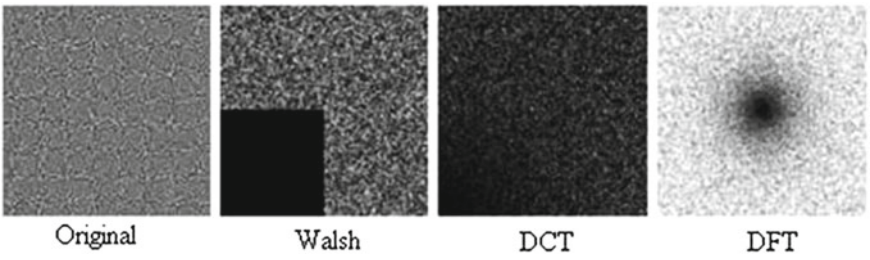

Fig. 2.3 The original shares and their transformations

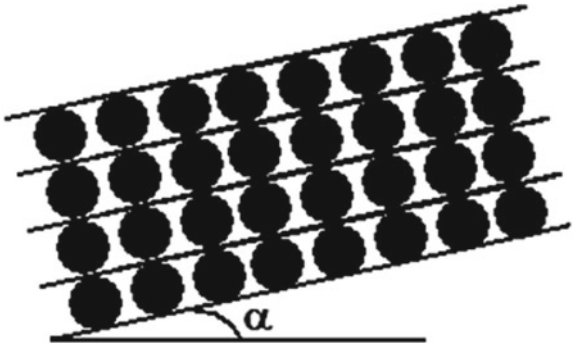

Fig. 2.4 Adjustment of visual cryptography shares

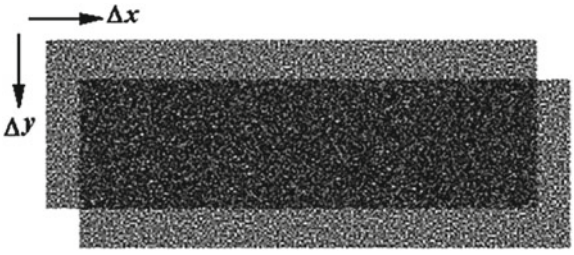

Fig. 2.5 The shift operation to the overlapping shares

In this section, we will describe our contributions. During encryption, we apply the Walsh transform on the shares. Then we embed marks in the high frequency coefficients of the transform. Then the inverse transform is applied to obtain the new shares with hidden marks that are printed on paper to be transmitted via public channels.

During the process of decryption, we scan the paper image and extract the marks by performing the Walsh transform to obtain the approximate alignment for shares superimposition. We then fine-tune the alignment by performing rotation and translation. The rotation is done by using:

The rotation adjustment in increments of angle α is done as shown in Fig. 2.4. The translation adjustment by x and y is done as shown in Fig. 2.5. The criteria for finding the best alignment position are that the superimposed image should have the least number of black pixels if we perform the XOR operation between them. This is because the XOR operation allows for perfect restoration of the secret image.

Figure 2.6 shows a share and the mark in the Walsh transform domain. The mark is in the form of a cross. Figure 2.7 is an example of a scanned marked share. Figure 2.8 shows the minimization of black pixels when the correct alignment is obtained.

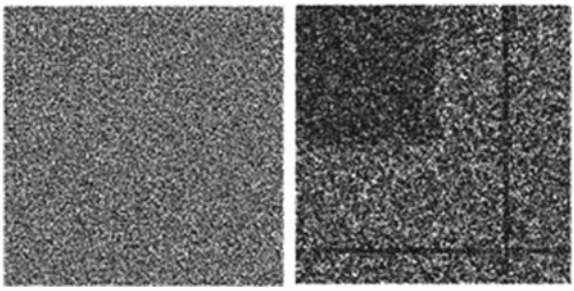

Fig. 2.6 Marked VC share in Walsh transform domain

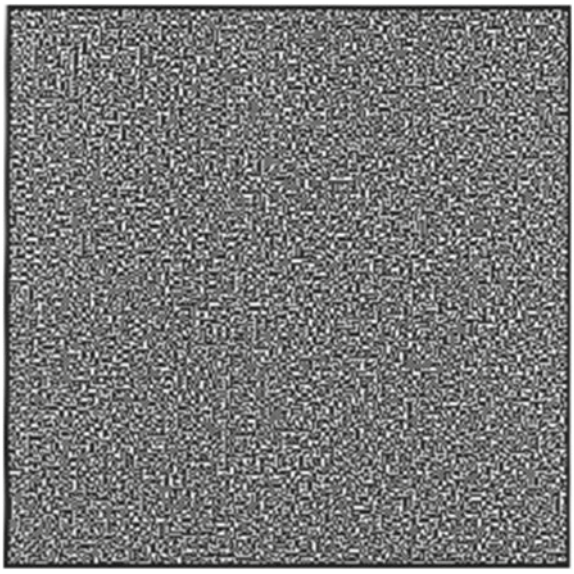

Fig. 2.7 The scanned watermarked VC shares

2.1.2 Visual Alignment of VCS

We found that, the of small subpixels is not critical [35]. The secret image can still be recovered visually even if the participants do not align the transparencies precisely. This phenomenon helps to determine the size of the printed on the transparencies.

The usual way of tackling the alignment problem of the VCS is by adding frames to the shares [35]. To align the shares, one just needs to align the frames. Another study employs the Walsh transform to embed marks in both of the shares so as to find the alignment position of these shares. However, both the two methods need to align the transparencies precisely. Besides, Kobara and Imai calculated the visible space

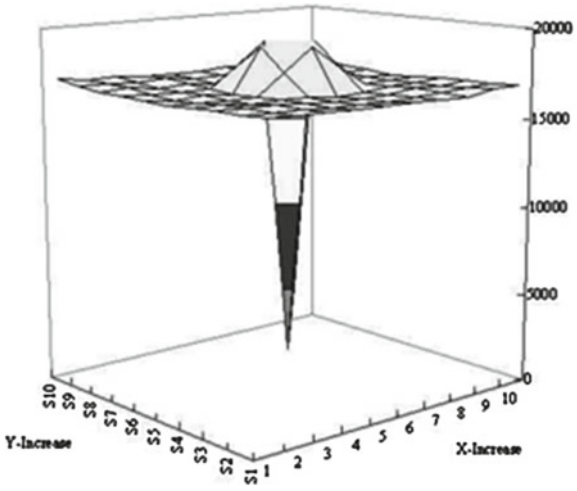

Fig. 2.8 Number of black pixels at various alignments

when viewing the transparencies. The results are somehow related to the alignment problem, but not exactly.

According to the traditional view, the subpixels of the transparencies should be aligned precisely, however, in this study, we point out that, to recover the secret image visually, it is not necessary to align the precisely. We will show that, by shifting one of the shares by some number (at most $m - 1$) of to the right (resp. left), one can still recover the secret image visually, for the reason that the average contrast $\widetilde{\alpha} \neq 0$ [5, 36]. This result can naturally be extended to the case when more than one share is shifted. However we leave the numerical analysis of this case as an open problem. So, in this chapter, we will only consider the case with only one share (transparency) being shifted by a number of subpixels. And we call the scheme with a share being shifted the shifted scheme, the basis matrices and share matrices of the shifted scheme are called the shifted basis matrices and shifted share matrices.

Generally, we aim at proving the conclusion that, the shifted scheme can visually recover the original secret image based on the (k, n)-VCS. However, it is noticed that this proof can be reduced to the proof based on the (2,2)-VCS in the case that only one share is shifted. The reason is as follows:

First, a (k, n)-DVCS consists of $\binom{n}{n}(k, k)$-VCS. For a set of k shares, if no share is shifted, then the k shares can recover the secret image obviously. And because we only consider the case when only one of the n shares is shifted, we only need to consider the k shares that contain the shifted share. i.e., we only need to prove our conclusion based on a (k, k)-VCS.

Second, denote the k shares of a $(k, k)-$VCS as $s_1, s_2, \ldots, s_k$, without loss of generality, let s_k be the share that is shifted, and let s_k be the resulting image of stack-

ing the remaining $k-1$ shares $s_1, s_2, \ldots, s_{k-1}$ together. Then the scheme becomes a (2, 2)-VCS, where the two shares are s'_k and s_k. Note that the stacking result of this (2,2)-VCS is the same as that of the previous (k, k)-DVCS. The previous (k, k)-VCS can visually recover the secret image if and only if s'_k and s_k can do so. Hence it is sufficient to prove the conclusion based on a (2,2)-VCS.

We analyze the structure of the basis matrix of the (2,2)-VCS. Denote M_0 and M_1 as the basis matrices of the (2, 2) -VCS, then the M_0 and M_1, without loss of generality, are in the following form:

$$M_0 = \begin{pmatrix} 1\cdots1 & 0\cdots0 & 1\cdots1 & 0\cdots0 \\ \underbrace{1\cdots1}_{a} & \underbrace{0\cdots0}_{b} & \underbrace{0\cdots0}_{c} & \underbrace{1\cdots1}_{d} \end{pmatrix} \tag{2.4}$$

and

$$M_1 = \begin{pmatrix} 1\cdots1 & 0\cdots0 & 1\cdots1 & 0\cdots0 \\ \underbrace{1\cdots1}_{a'} & \underbrace{0\cdots0}_{b'} & \underbrace{0\cdots0}_{c'} & \underbrace{1\cdots1}_{d'} \end{pmatrix} \tag{2.5}$$

where a, b, c, d, a', b', c' and d' are nonnegative integers satisfying $a + c + d = l$ and $a' + c' + d' = h$. According to the contrast and security property of Definition 1 [5], we have,

$$\begin{cases} a + b + c + d = a' + b' + c' + d' \\ a + c = a' + c' \\ a + d = a' + d' \\ b > b' \end{cases} \tag{2.6}$$

solving the above system, we get $a - a' = b - b' = c - c' = d - d'$. Let $e = b - b'$, hence by deleting identical columns of M_0 and M_1, we get,

$$M'_0 = \begin{pmatrix} 1\cdots1 & 0\cdots0 \\ \underbrace{1\cdots1}_{e} & \underbrace{0\cdots0}_{e} \end{pmatrix} \tag{2.7}$$

$$M'_1 = \begin{pmatrix} 1\cdots1 & 0\cdots0 \\ \underbrace{0\cdots0}_{e} & \underbrace{1\cdots1}_{e} \end{pmatrix} \tag{2.8}$$

where the number of columns in M_0 and M_1 is $2e$.

Now we know that the basis matrices of an arbitrary (2, 2)-VCS M_0 and M_1 contain the same number of identical columns $\binom{1}{1}$, $\binom{1}{0}$, $\binom{0}{1}$, $\binom{0}{0}$ apart from the sub-matrices M'_0 and M'_1, Hence, without loss of generality, they can be represented as the following form:

$$M_0 = \begin{pmatrix} 1\cdots1 & 0\cdots0 & 1\cdots1 & 0\cdots0 & 1\cdots1 & 0\cdots0 \\ \underbrace{1\cdots1}_{a'} & \underbrace{0\cdots0}_{b'} & \underbrace{0\cdots0}_{c} & \underbrace{1\cdots1}_{d} & \underbrace{1\cdots1}_{e} & \underbrace{0\cdots0}_{e} \end{pmatrix} \tag{2.9}$$

and

$$M_1 = \begin{pmatrix} 1\cdots1 & 0\cdots0 & 1\cdots1 & 0\cdots0 & 1\cdots1 & 0\cdots0 \\ \underbrace{1\cdots1}_{a'} & \underbrace{0\cdots0}_{b'} & \underbrace{0\cdots0}_{c} & \underbrace{1\cdots1}_{d} & \underbrace{0\cdots0}_{e} & \underbrace{1\cdots1}_{e} \end{pmatrix} \tag{2.10}$$

Let m be the pixel expansion, then it is obvious that $m = a' + b' + c + d + 2e$. The collections C_0 and C_1 contain all the permutations of the basis matrices M_0 and M_1, and hence each has $m!$ share matrices.

The shifted scheme is generated as follows:

Shift the second row of the $m!$ share matrices in C_0 (resp. C_1) to the left (resp. right) by r subpixels, and let $c_1, c_2, \ldots, c_r$ be the r-bit string that is shifted in, where each $c_i \in \{0, 1\}$ represents a subpixel. By the above discussion, we get $m!$ shifted share matrices for C_0 (resp. C_1). Take the share matrix $M_0 \in C_0$ as an example, then the shifted share matrix, denoted by $M_0^{(r)}$, is as follows:

$$M_0^{(r)} = \begin{pmatrix} *\cdots* & 1\cdots1 & 0\cdots0 & 1\cdots1 & 0\cdots0 & 1\cdots1 & 0\cdots0 \\ \underbrace{1\cdots1}_{a'} & \underbrace{0\cdots0}_{b'} & \underbrace{0\cdots0}_{c} & \underbrace{1\cdots1}_{d} & \underbrace{1\cdots1}_{e} & \underbrace{0\cdots0}_{e} & \underbrace{c_1\cdots c_r}_{r} \end{pmatrix} \tag{2.11}$$

where $c_1, c_2, \ldots, c_r$ of share 2 are the adjacent of the right pixel that are shifted in. By going through all $m!$ share matrices of C_0 and C_1 and all the possible string of $c_1, c_2, \ldots, c_r \in \{0, 1\}^r$, where $\{0, 1\}^r$ is the set of all the binary strings of length r, the shifted scheme is generated. Hence we have:

Theorem 2.1 *The shifted scheme of a VCS is a PVCS, where the average contrast of the shifted scheme is* $\bar{\alpha} = \frac{-(m-r)e}{m^2(m-1)}$, $1 \le r \le m-1$ *is the number of by which the share 2 (the second share) is shifted.*

Note that after a shift, the value of the average contrast is negative $\bar{\alpha} < 0$, which means that the recovered secret image is the complementary image of the original one, and the absolute value of $\bar{\alpha}$ reflects how clear the image can be viewed visually.

The above theorem shows that in order to align the transparencies when decrypting the VCS, one does not need to align the transparencies precisely. So, when the participants of a VCS want to align the transparencies, for example, the transparencies in the Example 2.1, they can first align the transparencies precisely in the vertical direction, and then move the second transparencies to the right then to the left in the horizontal direction. Then they will get the recovered secret image for three times. Furthermore, this phenomenon also helps to determine the size of the printed image on the transparencies.

In order to reduce the size of transparencies, one needs to reduce not only the pixel expansion, but also the size of each subpixel in the transparencies [11]. However,

smaller size of results in more difficulties when aligning the transparencies together. We study the alignment problem of the VCS [35], and proved that, the original secret image can be recovered visually when one of the transparencies is shifted by at most $m-1$ subpixels, and the average contrast becomes a $\bar{\alpha} = \frac{-(m-r)e}{m^2(m-1)}$. Our study is based on a DVCS, and the shifted scheme is a PVCS with less contrast but still visible. This result helps to determine size of the subpixel printed on the transparencies.

Our result can be extended to the case when l transparencies are shifted all together. In this case we only need to consider the resulting transparency of stacking all these shifted transparencies together, which is also equivalent to a $(2, 2)$-VCS. Further generalization when the l transparencies are shifted differently is possible, but numerical analysis becomes more complicated. We leave this as an open problem.

2.2 VCS Cheating Prevention

Most schemes (CIVCS) are based on a scheme (VCS) and are designed to avoid cheating when the secret image of the original VCS is to be recovered. However, all the known CIVCS have some drawbacks. Most usual drawbacks include the following: the scheme needs an online trusted authority, or it requires additional shares for the purpose of verification, or it has to sacrifice the properties by means of pixel expansion and contrast reduction of the original VCS [58] or it can only be based on such VCS with specific access structures. In this chapter, we introduce a new CIVCS that can be based on any VCS, including those with a general access structure [2], and show that them CIVCS can avoid all the above drawbacks. Moreover, their CIVCS does not care about whether the underlying operation is OR or XOR.

The cheating problem in VCS is quite interesting. The possibility of cheating activity in VCS has been studied. For cheating, the cheaters present some fake shares so that the stacking of fake and genuine shares together reveals a fake image, and the victims who cannot detect the cheating activities will be fooled to believe that the recovered fake image is the genuine secret image. This is terrible because the secret image is usually important to the victims.

Many studies focused on the cheating problems in VCS, and consequently many cheating immune visual cryptography schemes (CIVCS) have been proposed. We classify the techniques in these CIVCSs as follows:

- Make use of an online trusted authority who can verify the validity of the stacked shares.
- Generate extra verification shares to verify the validity of the stacked shares.
- Expand the pixel expansion of the scheme to embed extra authentication information [63].
- Generate more than n shares to reduce the possibility that the cheaters can correctly guess the distribution of the victims' shares.
- Make use of the genetic algorithm to encrypt homogeneous secret images.

By examining the above techniques, we found that the first technique is not practical in real applications, because the beauty of VCS is its simplicity, which is meant to be useful even when no computer networks is available. The second technique requires the extra verification shares, which inevitably increases the burden of the participants. The third and forth techniques increase the pixel expansion and reduce the contrast of the original VCS [5]. The fifth technique requires strong computational overhead and degrades the quality of the recovered secret image [60], where the secret image can only be a password. It is also noted that most CIVCS can only be based on a VCS with specific access structure, for example, the $(2, n)$ threshold access structure.

2.2.1 Definitions

We give some definitions about cheating:

Definition 2.1 A CVS is called a cheater if, during the reconstruction phase, he presents a fake share, which results in the recovered image to be different from the original secret image. A participant is called a victim if he cannot tell whether a recovered image is the original image and hence has to believe that the recovered image is the original one.

Definition 2.2 A successful cheating on e victims is that a fake image is recovered in the reconstruction phase owing to the cheaters presenting fake shares, and e victims cannot tell whether the recovered image is the original one, that is, the victims cannot tell whether the cheaters presented fake shares or genuine ones.

Definition 2.3 A pe-secure CIVCS is a VCS such that the probability of cheating e victims successfully is no more than p_e.

Definition 2.4 A successful cheating method (SCM) is a cheating against a VCS that can succeed with probability 1.

In the practical sense of VCS, when we assume that a powerful cheater knows the basis matrices, it is reasonable to assume that all the other participants know such information as well. It is noted that the basis matrices require little memory to hold. It is also reasonable to assume that every participant knows the qualified sets where he belongs to. More precisely, we give the following assumption.

Assumption 2.1 For any participant $i \in V$ of a VCS, (s)he should know the following information:

- All the qualified sets in which i is a member.
- The basis matrices M_0 and M_1.

Knowing the assumed information about the VCS helps the cheaters to cheat; it also helps other genuine participants make use of the extra information to detect the existence of cheaters. Hence, in the rest of this paper, an SCM under the Assumption 2.1 means that the fake share can pass the victim's verification.

2.2.2 Attacks

In this section, we show that a forbidden set of participants can also recover the original secret image. We also show that all the cheating attacks can be detected. However, successful cheating does exist in the CIVCS, and this can be done by modifying an SCM for VCS, and we show that the SCM can be applied to [2] under Assumption 2.1 for the case of cheaters colluding.

2.2.2.1 Attack on Horng's CIVCS

Two CIVCSs are proposed by Horng et al., where the schemes only tackle the $(2, n)$-VCS. We found that the first CIVCS is not secure, that is, the confidentiality of the secret image cannot be guaranteed and any single participant (which forms a forbidden subset) can almost recover the secret image. This is not acceptable, even the recovered image has low visual quality than the original, as in many cases the content of the hidden image is more important [60].

First, we recall the CIVCS of Horng et al. as follows. Horng's CIVCS: assume the set of participants is $V = \{1, \ldots, n\}$. In the distribution phase, each participant is assigned a share S_i and a verification share V_i. The verification share V_i is divided into $n - 1$ regions $R_{i,j}$ where $1 \leq i, j \leq n$, $j \neq i$. Then the verification shares are generated by a (2, 2)-VCS with the secret image being the logos L_i which are chosen by the participants and sent to the dealer securely. So, the logo L_i will appear in $R_{i,j}$ when stacking the V_i and S_i.

In the reconstruction phase, the participant i first stacks the shares V_i and S_i to check whether S_j is a fake share or not. If the authentication is passed, the participants stack their shares to decrypt the secret image.

Our attack is given as follows.

Attack 2.2 According to the above construction, we get to know that a participant i owns the share S_i, the verification share V_i and the logo L_i. Since the region $R_{i,j}$ and the S_j constitute a (2, 2)-VCS with the secret image being L_i, and the participant i owns $R_{i,j}$ and L_i, it is clear that (s)he can restore part of the share S_j (the part corresponding to the region $R_{i,j}$), denoted by S'_j. By stacking S_i and S'_j in the region $R_{i,j}$, the participant i recovers the secret image in the region $R_{i,j}$. Repeat the above process for the remaining $n \geq 2$ regions in V_i, the participant i can eventually recover the whole secret image by himself.

It is noted that S'_j is not necessarily the same as S_j in the region $R_{i,j}$; this is because a (2,2)-VCS is not unique. However, the recovered images are very close and hence the secret image can almost be recovered with a good visual effect.

We give the experimental results for the scenario with three participants Alice, Bob and Carol, where the secret image is the word 'Secret', and the logo of Alice is 'L_1', the image S'_{23} is the concatenation of the images S'_2 (left) and S'_3 (right).

From the experimental results, we can observe that the secret image can be recovered by stacking the images S_1 and S'_{23}, although the result is not so clear. This is

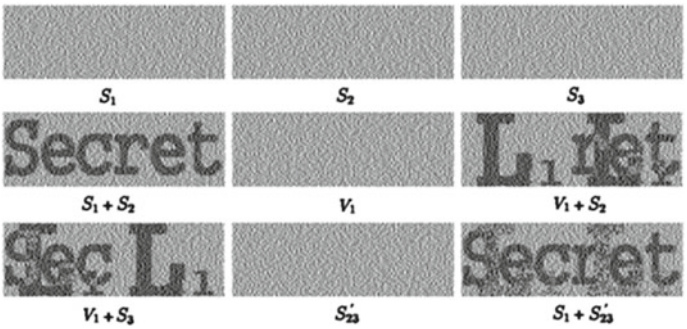

Fig. 2.9 Experimental results of the attack on Horng's CIVCS

not acceptable when the content of the secret image is what is meant to hide [25]. Here S'_{23} can be generated by V_1 and L_1 using a (2, 2)-VCS. This shows that the first CIVCS is not secure.

Cheating detection method:

- Check if any participant takes more than one share in the reconstruction phase; only cheaters could take more than one share (a cheater is assumed to take $r + 1$, 2 and $r + 3$ shares in the first, second and third attacks respectively).
- Check if the shares used to recover the original image form a qualified subset in G_m. It is apparent that there must exist cheaters if a forbidden set of participants attempts to recover the original image.
- Check if every share is necessary to recover a visual image. This can be done by seeing if the rest of shares (excluding the participant's genuine share) give a meaningful visual image (the fake image can be recovered only by the cheater's shares in the three attacks) [38].
- Check whether there exists a column permutation of the basis matrices M_0 and M_1 that correspond to the distribution of the stacked shares for each pixel in the recovered secret image (the distribution of the stacking shares for each black pixel may not agree with any permutation of the basis matrices M_0 and M_1 for the three attacks).

The first three cheating detection methods are to verify the access structure according to the first item of Assumption 2.1 of the last section, and the fourth cheating detection method is to verify the basis matrices according to the second item of Assumption 2.1.

2.2.2.2 Detectable Attacks

Hu and Tzeng have the three attacks on the traditional VCS. However, all the three attacks can be detected under the Assumption 2.1. The main idea of the three attacks are that a cheater generates r fake shares ($r = 2$ for the second attack), and the

stacking of these fake shares together with the cheater's own genuine share recovers a fake image chosen by the cheater (if the cheater does not have a genuine share, then just stack the fake shares). Furthermore, by stacking the victims' shares and the cheater's genuine share, the fake image appears.

However, the drawbacks of those attacks are clear. First, one cheater needs to take more than one share in the reconstruction phase, which may not be allowed in many VCS. Second, the fake shares can recover the fake image without the victims' shares, which is also a way to detect the cheating. Third, the distributions of the stacked shares may not agree with any permutation of the basis matrices, and this is also a way to detect cheating. Fourth, the size of the shares may not agree with the actual size of the subpixels and that of the victims' shares (the second attack). These drawbacks will incur the suspicion of the victims. Hence, the victims can detect these cheatings during the reconstruction phase.

2.2.2.3 Collaborating Cheaters

Given Assumption 2.1, it seems hard to find a SCM. Unfortunately, based on the detection methods, such an SCM does exist; in fact, the cheating example is indeed an SCM. However, Horng et al. only consider the cheating on $(2, n)$-VCS. Here we extended it to the general access structure [2].

Theorem 2.2 *Denote* (C_0, C_1) *as a VCS on access structure* Γ_m *and participant set* $V = \{1, 2, \ldots, n\}$. *Denote* $C = C_0 \bigcup C_1$. *If a sub-matrix of* t *participants (cheaters),* $p_1, \ldots, p_t$ *can uniquely determine a share matrix in the collection* C, *then there must exist a SCM under the Assumption 2.1 cheating the rest* $n \geq t$ *participants (victims),* $p'_1, \ldots, p'_{n-t}$. *More precisely, for any forbidden subset of* v *participants* $\{p'_{r_1}, p'_{r_2}, \ldots, p'_{r_v}\} \subseteq \{p'_{r_1}, p'_{r_2}, \ldots, p'_{r_{n-t}}\}$ *satisfying* $\{p'_{r_1}, p'_{r_2}, \ldots, p'_{r_v}\} \bigcup \{p_{r_1}, p_{r_2}, \ldots, p_{r_c}\} \subset \Gamma_m$ *where* $\{p_{r_1}, p_{r_2}, \ldots, p_{r_c}\} \subseteq \{p_1, p_2, \ldots, p_t\}$, *then* $\{p'_{r_1}, p'_{r_2}, \ldots, p'_{r_v}\}$ *can be successfully cheated by the collusion of cheaters* $p_1, p_2, \ldots, p_t$.

To make things clearer, we give the following experimental results.

Example 2.1 For the access structure $\Gamma_m = \{\{1, 2, 3\}, \{1, 3, 4\}, \{1, 2, 4\}, \{1, 5\}\}$, assume that the basis matrices of the VCS are:

The collection C_i is obtained by all the permutations of the basis matrix M_i, for $i = 0, 1$, and the collection $C = C_0 \bigcup C_1$. The dealer distributes the shares S_1, S_2, S_3, S_4 and S_5 to the participants 1, 2, 3, 4 and 5. It is easy to verify that the first three rows ca uniquely determine a share matrix in the collection C. The cheaters can generate the fake shares S_{f_1}, S_{f_2} and S_{f_3} according to Theorem 2.2.

$$M_0 = \begin{pmatrix} 111000 \\ 100100 \\ 010100 \\ 001100 \\ 110100 \end{pmatrix} \tag{2.12}$$

and

$$M_0 = \begin{pmatrix} 111000 \\ 100100 \\ 100010 \\ 100001 \\ 100110 \end{pmatrix} \tag{2.13}$$

2.2.2.4 Cheater Colluding

Based on the SCM of Theorem 2.2, we found that the CIVCS is still vulnerable against collusion by cheaters. In this section, we construct an SCM based on Hu and Tzeng's CIVCS under Assumption 2.1. First, recall the CIVCS of Hu and Tzeng as follows.

Hu and Tzeng's CIVCS: Given the original VCS for an access structure Γ_m, and let the basis matrices be M_0 and M_1, denote the pixel expansion of the original VCS as m. In the distribution phase, the dealer generates T_0 and T_1 as follows:

$$M_0 = \begin{pmatrix} 10 \mid \\ \cdots \mid M_0 \\ 10 \mid \end{pmatrix} \tag{2.14}$$

and

$$M_1 = \begin{pmatrix} 10 \mid \\ \cdots \mid M_1 \\ 10 \mid \end{pmatrix} \tag{2.15}$$

The dealer will use T_0 and T_1 as the basis matrices for CIVCS to generate shares $S_1, \ldots, S_n$ with pixel expansion $m+2$. The leading bits '10' of each row of T_0 and T_1 are treated as authentication for the CIVCS. Then for each participant $i (1 \leq i \leq n)$, choose a verification image and generate a verification share V_i as follows:

- For each white pixel in the verification image, put the pixel of $(m+2)$-dimensional [100...0] to V_i (after corresponding permutation as for the share S_i).
- For each black pixel in the verification image, put the pixel of $(m+2)$-dimensional [010...0] to V_i (after corresponding permutation as for the share S_i).

In the reconstruction phase, the participant i first stacks the shares V_i with all the other S_j to verify whether S_j is a fake share. If all the verifications pass, then the participants can stack their shares to decrypt the secret image.

The colluding attack on the above scheme is given as follows.

Attack 2.3. Since T_0 and T_1 are generated by simply concatenating the authentication and the basis matrices of the original VCS, and the authentication are the same for

all the shares, if the cheaters can locate the positions of the authentication subpixels, they can duplicate the authentication to the fake shares and make use of the SCM to generate the fake shares for the rest subpixels. In this way, the fake shares can pass the verification of the victim under Assumption 2.1, and hence the colluding forms a new SCM for the above CIVCS.

The positions of the authentication can be located as follows:
For a qualified subset of participants $\{p'_{r_1}, \ldots, p'_{r_v}\} \bigcup \{p_{r_1}, \ldots, p_{r_c}\} \subseteq \Gamma_m$, where $p_{r_1}, \ldots, p_{r_c}$ are part of the cheaters $p_1, \ldots, p_t$. Recall that $p_1, \ldots, p_t$ are the cheaters that can uniquely determine a share matrix in the collection $C = C_0 \bigcup C_1$. Denote the shares of $p_1, p_2, \ldots, p_t$ as $S_1, S_2, \ldots, S_t$, the verification shares as $V_1, V_2, \ldots, V_t$ and the verification images as $L_1, L_2, \ldots, L_t$.

Because the positions of the two authentications are at the position of the 1 (black subpixel) of a white pixel in the verification share and the position of the 1 (black subpixel) of a black pixel in the verification share, and there is only one 1 in each pixel of the verification shares, the cheaters can obtain the positions of the authentication by choosing their verification images accordingly. More precisely, for the corresponding positions in the verification images $L_1, L_2, \ldots, L_t$ denote the pixels in these positions as $P_{e_1}, P_{e_2}, \ldots, P_{e_t}$; if there exist both black and white pixels in $P_{e_1}, P_{e_2}, \ldots, P_{e_t}$, then the cheaters can locate the positions of the authentication precisely by finding the 1's in their verification shares $V_1, V_2, \ldots, V_t$. Hence, the cheaters only need to choose verification images that have both black and white pixels in the same positions. A simple way to achieve this is by choosing complementary verification images for two out of the t cheaters (Fig. 2.10). In fact there exist better methods to construct verification images $L_1, L_2, \ldots, L_t$ satisfying the above condition for arbitrary patterns.

Once the cheaters know the positions of the authentication subpixels, they can make use of the SCM to generate the fake shares for the remaining (other than the part used for authentication), while remaining the authentication intact. It is easy to verify that the above approach makes a new SCM for the above CIVCS.

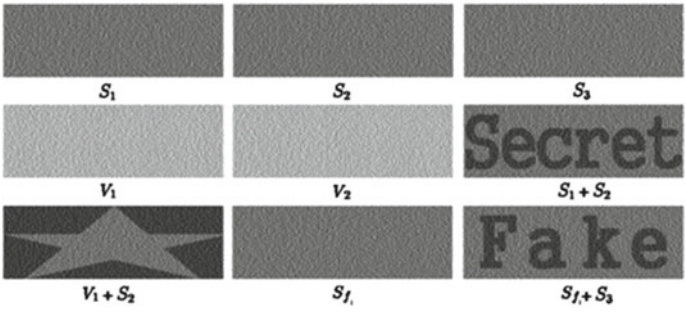

Fig. 2.10 Experimental results of the SCM on Hu and Tzeng's CIVCS under Assumption 2.1

To demonstrate how the above colluding attack works, we give some experimental results for the above SCM on the (2, 3)-CIVCS, where the participants p_1 and p_2 are the cheaters and p_3 is the victim.

Example 2.2 The secret image is the word 'Secret', the fake image is the word 'Fake', the logo of p_1 is a pentacle and the logo of p_2 is the complementary image of the pentacle. The size of these images is 120×200.

The basis matrices in this Example 2.2 are as follows:

From the experimental results, we can observe that the CIVCS is still cheatable, that is, a fake image appears when stacking the fake share S_{f_1} and the victim's share S_3, while using the verification process proposed by Hu and Tzeng, the victim cannot identify that S_{f_1} is a fake share.

$$M_0 = \begin{pmatrix} 10 \mid 100 \\ 10 \mid 100 \\ 10 \mid 100 \end{pmatrix} \tag{2.16}$$

and

$$M_1 = \begin{pmatrix} 10 \mid 100 \\ 10 \mid 010 \\ 10 \mid 001 \end{pmatrix} \tag{2.17}$$

2.2.2.5 New CIVCS

The results imply that the traditional VCS is not secure against cheater colluding under Assumption 2.1. We designed a new CIVCS to thwart against these SCMs for cheater colluding. A good CIVCS should satisfy the following properties to avoid some drawbacks of the known CIVCS:

- The CIVCS should not rely on the help of an online trusted authority.
- The CIVCS should not increase the pixel expansion of the original VCS.
- The CIVCS should not reduce the contrast of the original VCS.
- The CIVCS should be applicable to any VCS for general access structure Γ_m.
- The amount of the authentication information should be as small as possible, and the verification process does not have to rely on computing devices.
- The CIVCS should be able to detect the existence of cheaters, and it would be ideal if it is able to detect the actual cheaters.

In this section, we will discuss two methods to construct our CIVCS satisfying all the above required properties. However, because of the similarity of the two methods, we combine the main steps into the Construction 2.1, and we differentiate the steps of the two methods by using superscript *1 (for Method 1) and *2 (for Method 2).

Construction 2.1 Given a VCS (C_0, C_1) for the access structure G_m, denote its basis matrices as M_0 and M_1, then our construction of the CIVCS is as follows.

Distribution phase:

Step 1 Construct the n shares by using the original VCS and record all the share matrices that chosen for each pixel from the original secret image.

Step 2 Randomly choose t pixels from the original secret image as the authentication pixels (APs) for each participant i, and record the t share matrices, $M_1^i, M_2^i, \ldots, M_t^i$ of the APs, where $i \in V$ (note that the t APs are chosen separately for different participants).

Step 3*1 Distribute the ith share to the participant i and mark the t APs in the share of i securely, where a black AP is marked by a green box and a white AP is marked by a red box.

Step 3*2 Distribute the ith share to the participant i and tell the participant i the t share matrices $M_1^i, M_2^i, \ldots, M_t^i$ and mark the t APs in the share of i securely, where a black AP is marked by a green box and a white AP is marked by a red box.

Reconstruction phase:

Let $p_1, p_2, \ldots, p_r$ be members in a qualified subset, that is $\{p_1, p_2, \ldots, p_r\} \subseteq \Gamma_m$

Step 1*1 Each participant verifies whether the color of the recovered secret image at the positions of t APs agrees with the color of the APs he received from the dealer.

Step 1*2 Each participant verifies whether the distribution of the stacked shares agree with the share matrices of the t APs he received from the dealer.

Step 2*1 If the verification of Step 1*1 is passed, the participants stack their shares and recover the secret image, else reject the fake shares.

Step 2*2 If the verification of Step 1*2 is passed, the participants stack their shares and recover the secret image, else reject the fake shares and find out the cheaters whose shares do not agree with the distributions of the share matrices at the positions of the APs.

To demonstrate how Construction 2.1 works, we give the following Example 2.3.

Example 2.3 We give a simple example for a (2, 2)-CIVCS by using Method 1, where we let $t = 10$ for simplicity. The size of the secret image is 150×120; hence, we have $r = 1/1800$. We mark the shares by red and green boxes for white and black APs, respectively.

The following theorem shows the effectiveness of our CIVCS:

Theorem 2.3 *Denote the size of the secret image as $l \times h$ and denote the participants set as $V = \{1, 2, \ldots, n\}$. Let $p = \frac{s_s}{s_t}$, where s_s is the number of pixels in the fake image that have same color as the corresponding pixels (the pixels that at the same position) in the secret image, and s_t is the total number of pixels in the fake image (secret image). Then Construction 2.1 is a p_e-secure CIVCS with $p_e = max(1/\binom{l \cdot h}{t \cdot e}, p^{te})$, where e is the number of target victims and t is the number of APs for each share. The securer of each share is $r = \frac{t}{s_t}$.*

Table 2.2 Comparison of the amount of authentication information

Method 1	Method 2	1st CIVCS of Horng	2nd CIVCS of Horng	CIVCS of Hu & Tzeng
Number k	Number k and indices of t aPs the permutation of the basis matrices of t aPs	Verification share with s_t aPs	Enlarged share with $l \times s_t$ aPs	Enlarged share with $2 \times s_t$ aPs

To show the effectiveness of our CIVCS, we compare the amount of the authentication information each participant carries in our CIVCS and that of the CIVCS proposed by Horng et al. and Hu and Tzeng for the (k, n) access structure. Table 2.2 shows the comparison of the amount of information needed for authentication in our CIVCS.

In Table 2.2, s_t is the number of pixels in the secret image. Note that t should be far less than s_t; hence, it is obvious that the amount of the authentication information of the proposed CIVCS is far less than that of the CIVCS proposed by Horng et al. and Hu and Tzeng for the (k, n) access structure. Hence, our CIVCSs do not bring a heavy burden to the participants. The following Example 2.4 shows the effectiveness of our CIVCS.

Example 2.4 Suppose the fake image is shown as in Fig. 2.11. As the CIVCS of Example 2.3 depicts, assume that there is a cheater and a target victim in the CIVCS. Let $t = 10$, then the probability $\binom{l \cdot h}{t \cdot e} \simeq 1.02 \times 10^{-36}$, which can be neglected. The probability is even smaller when the value of t is larger. Since the probability of being the same color at the same position in the secret image (Fig. 4) and the fake image in Fig. 2.11 is 0.5944 (which can easily be verified), the probability $p^{10} \simeq 0.0055$, which is very small. Note that, for different fake images, the values of p can be different. All these values of p satisfy $p < 1$, because if $p = 1$, then the secret image is identical to the fake image, and there will be no cheating at all.

In this section, we first discussed the drawbacks of some known CIVCSs, and then proposed a new CIVCS which avoids all the previous drawbacks. Our CIVCS is constructed based on a known VCS and can be applied to all VCSs for general access structure [2]. It is also noted that our CIVCS works when the underlying operation

Fig. 2.11 'Fake' image with size 150 × 120

is XOR, although most discussions on CIVCS only consider the OR operation. Furthermore, our CIVCS can detect the cheaters or only detect the existence of cheaters depending on the amount of the authentication information provided. Our CIVCS achieves high security against the cheating attacks only with a small cost, only r of the secret pixels are revealed to each of the participant, and in real applications, one can set r to be a very small value, and hence the confidentiality of the secret image can be ensured.

2.3 Flipping Issues in VCS

Plane transformation visual cryptography takes a unique approach to some of the current shortcomings of visual cryptography techniques. Typically, the direction and placement of the encrypted shares are critical when attempting to recover the secret. Many schemes are highly dependant on this stacking order. Within this section, the scheme presented illustrates a technique whereby this restriction is loosened such that the number of acceptable alignment points is increased by performing a simple plane transform on one of the shares [35]. This results in the same secret being recovered when the shares correctly aligned. The technique has also been extended to encompass multiple secrets [22, 43, 48], each of which can be recovered depending on the type of transformation performed on the shares.

Many schemes within visual cryptography suffer from alignment issues and are dependant on how the shares are stacked together [35]. Loosening or removing this restriction would be a very desirable advance, as it enables an end user to recover the secret without having to work out how he must stack the shares. Figure 2.12 provides an example of this stacking and alignment problem. It can be observed that successful recovery is achieved when the shares are superimposed correctly. However, if the second share is transformed about its center point in the x-axis direction, then the secret cannot be recovered. Removing this limitation would improve the end users experience when it comes to recovering the hidden secret.

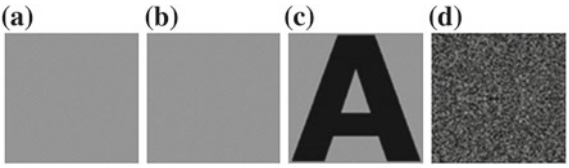

Fig. 2.12 Traditional visual cryptography decryption process. **a** Share one. **b** Share two. **c** Secret recovered by superimposing share two on share one. **d** Attempted secret recovery after flipping share two vertically and superimposing it on share one

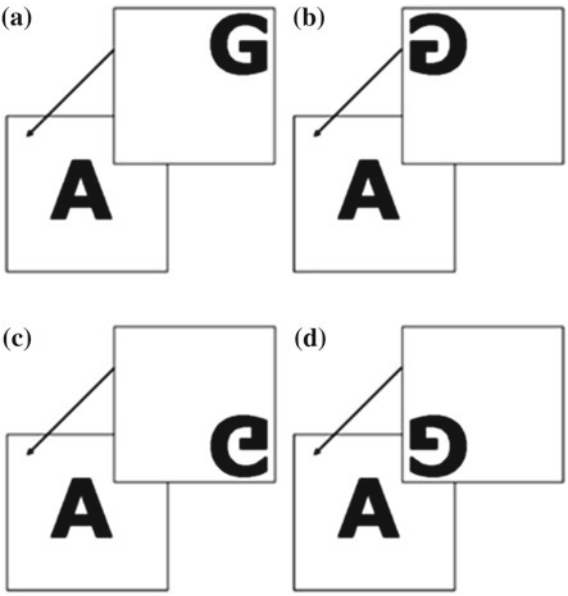

Fig. 2.13 Configurations under specific transformations. **a** Transformation one. No specific transformation. **b** Transformation two. *Vertical* transform. **c** Transformation three. *Horizontal* transform. **d** Transformation four. *Vertical* + *horizontal* transform

Creating shares in such a way that allows for secret recovery when the shares are superimposed after having been transformed was a valid line of research as it removes these specific types of restrictions which are demonstrated in Fig. 2.12.

The main idea is that one share is printed onto a normal white page, but the second is printed onto a transparent. This transparency is then transformed as previously mentioned. Figure 2.13 illustrates each of the transformations that each share undergoes in order to recover each of the secrets. Share one is marked with an 'A', share two is marked with a 'G'. The arrow denotes superimposing the shares in their specific configurations. After each of the transformation, the same or unique secrets can be recovered.

The term 'plane' used within this chapter refers to a flat two-dimensional surface. We used this term when describing the shares in order to illustrate the type of movement that they undergo using geometric expressions. Therefore the whole space is used when working in a two-dimensional Euclidean space.

When compared to the plethora of visual cryptography schemes [16] in use today, this scheme attempts to improve upon them by allowing the shares to be stacked in a variety of ways, after having been transformed about the horizontal, vertical, and a combination of both axes. This is a much more intuitive way to manipulate a quadrilateral in order to recover each of the secrets. Especially when dealing with two shares.

Removing the specific stacking order required by the majority of the previous schemes is a great advantage, as it allows for easier secret recovery. Illustrated within this section is one main idea which accomplishes two goals, the same secret recovery

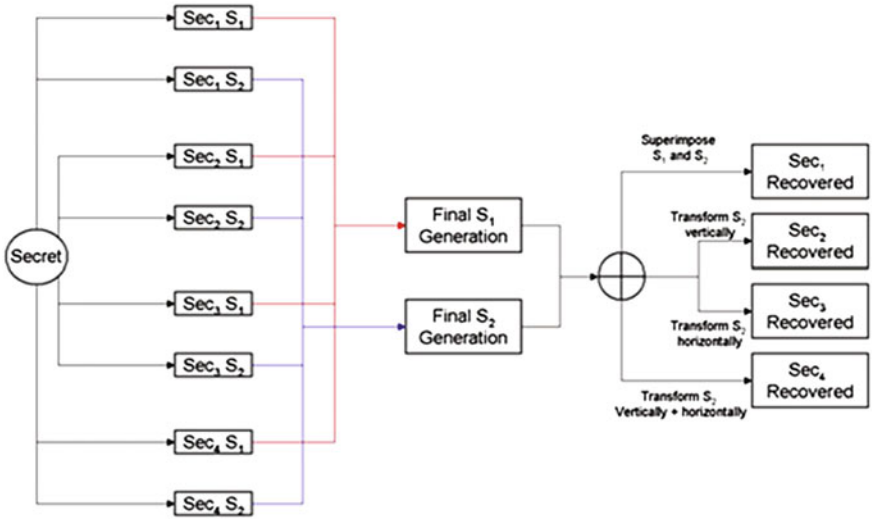

Fig. 2.14 Plane transform visual cryptography flowchart

based on different transforms and unique secret recovery based on the same set of transformations. Ideally, the same secret is used, this means that no matter how the shares are stacked, the same results are obtained. The unique secrets are illustrated to prove that it is possible for unique secrets to be shares as well.

The steps involved in order to create the resulting two shares can be examined in Fig. 2.14. Figure 2.14 provides a flowchart of the proposed system which details each of the corresponding actions required. Each of these steps is detailed below.

It can be observed from Fig. 2.14 that a secret is input and four sets of shares are generated accordingly, $\text{Sec}_1 S_1 \rightarrow \text{Sec}_4 S_1$ for the set of secrets belonging to share one and $\text{Sec}_1 S_2 \rightarrow \text{Sec}_4 S_2$ for the set of secrets belong to share two. Where $\text{Sec}_1 S_1$ refers to share one from secret and $\text{Sec}_1 S_2$ refers to share two from the corresponding set of secrets.

Whether one secret is input (recovering the same secret for each transform: $\text{Sec}_1 = \text{Sec}_2 = \text{Sec}_3 = \text{Sec}_4$) or four secrets (unique secret recovery for each transform), four sets of shares are generated. Based on these sets of shares, the final set of two shares is generated which allows for the recovery. When the final S_1 and the final S_2 are superimposed, Sec_1 is recovered. When final S_2 is transformed vertically about its center point on the x-axis, Sec_2 can be recovered. Sec_3 can be observed when final S_2 is transformed about its center point along the y-axis in a horizontal direction. Finally, Sec_4 is revealed after final S_2 is transformed about both center points of each axis.

The algorithm required is presented within Algorithm (2.1), which provides a pseudocode implementation of the process. Further details are presented in the fol-

lowing sections. They provide more insight into what happens during each of the steps. This algorithm simply provides a computational reference how the process is executed.

This transformation requires a lot of thought when creating a suitable scheme that can recover black and white pixels accordingly. Some pixel configurations may be representing white pixels, while, after a vertical transformation the pixel representation required is black.

Algorithm 2.1: Pseudo code for generating two shares of plan transform VCS

Input : One secret four times or four secrets Sec_i, $i = \overline{1,4}$.
Output: Final two shares S_1 and S_2.

for $i = \overline{1,4}$ **do**
  $(\text{Sec}_i S_1, \text{Sec}_i S_2) = GenVCShares(\text{Sec}_i)$;
end
for $i = \overline{1,4}$ **do**
  $ExpVCShares(\text{Sec}_i S_1, \text{Sec}_i S_2)$;
end
for $i = \overline{1,4}$ **do**
  $ProcVCShares(\text{Sec}_i S_1, \text{Sec}_i S_2)$;
end
$S_1 = \oplus_i^4 \text{Sec}_i S_1$;
$S_2 = \oplus_i^4 \text{Sec}_i S_2$;
Return S_1, S_2;

2.3.1 Share Generating

The shares are generated using a combination of processes. A size invariant scheme is used initially and then using these size invariant shares [26], it is expanded into a more traditional scheme where one pixel from the invariant shares is represented by a 2×2 block. This is the general process used to create the final share. Each of the invariant shares patterns is used to create a new suitable pattern capable of recovering each of the secrets.

The structure of this scheme is described by a Boolean n-vector $V = [v_0, v_1]^T$, where v_i represents the color of the pixel in the ith shared image. If $v_i = 1$ then the pixel is black, otherwise, if $v_i = 0$ then the pixel is white. To reconstruct the secret, traditional ORing is applied to the pixels in V. The recovered secret can be viewed as the difference of probabilities with which a black pixel in the reconstructed image is generated from a white and black pixel in the secret image. As with [3, 16], $n \times m$ sets of matrices need to be defined for the scheme (in this case 2×2):

C_0={ All the matrices obtained by permuting the columns of $\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$}

C_1={All the matrices obtained by permuting the columns of $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$}

Because this scheme uses no pixel expansion [50], m is always equal to one and n is based on the type of scheme being used, for example a (2, 2) scheme, $n = 2$. Using the defined sets of matrices C_0 and C_1, $n \times m$ Boolean matrices S_0 and S_1 are chosen at random from C_0 and C_1, respectively: $S_0=\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ and $S_1=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

To share a white pixel, one of the columns in S_0 is chosen and to share a black pixel, one of the columns in S_1 is chosen. This chosen column vector $V = [v_0, v_1]^T$ defines the color of each pixel in the corresponding shared image. Each v_i is interpreted as black if $v_i = 1$ and as white if $v_i = 0$. Sharing a black pixel for example [14], one column is chosen at random in S_1, resulting in the vector: $V = [0, 1]^T$. Therefore, the ith element determines the color of the pixels in the ith shared image, thus in this (2, 2) example, v_1 is white in the first shared image, v_2 is black in the second shared image.

2.3.2 Share Expansion

After the shares for each identical or unique secret have been generated, each set of shares for each secret is expanded into a 2×2 block and inserted into the final set of shares by the $processShare(\cdot)$ function from Algorithm (2.1). The following steps are involved when $processShare(\cdot)$ is executing. This function generates the final set of shares required in order to successfully recover the secrets.

Generating the final S_1 is a relatively simple procedure where each of the corresponding expanded shares is placed into the following coordinates on the share:

- $\text{Sec}_1 S_1$ no change, leave its current pixel locations intact.
- $\text{Sec}_2 S_1$ shifts its pixel locations one pixel to the right, in order to fill in the space to the right of $\text{Sec}_1 S_1$'s pixels.
- $\text{Sec}_3 S_1$ shifts its pixel locations down one pixel, this fills in the space beneath $\text{Sec}_1 S_1$'s pixels.
- $\text{Sec}_4 S_1$ shifts its pixel locations down one and right one, this fills in the final space remaining on the final share.

Generating the final S_2 is more challenging. The reason is that the transformations that this share undergoes need to be taken into consideration so that the correct black and white pixels can be represented. Accurate reconstruction is very difficult because four different situations arise due to the transforms.

Final S_2 can be generated according to the following scheme:

- $\text{Sec}_1 S_2$ has no change, leaves its current pixel locations intact.
- $\text{Sec}_2 S_2$ places its pixels in the same locations as those which belong to $\text{Sec}_2 S_1$, but its vertical inverse must be placed at those locations.
- $\text{Sec}_3 S_2$ places its pixels in the same locations as those which belong to $\text{Sec}_3 S_1$,but its horizontal inverse must be placed at those locations.

- $\text{Sec}_4 S_2$ places its corresponding vertical and horizontal inverse pixels at the same coordinates as those of $\text{Sec}_4 S_1$.

No change is made to the placement of the first set of secret shares, this corresponds to simply superimposing each of the shares in the traditional way. The inverse of the pixel locations is required in order to reconstruct each of the secrets after a specific transformation occurs. Determining the inverse pixel patterns required for each of the specific transformed patterns proved to be rather difficult in terms of alignment [35].

After a transform on a pixel block was performed, simply supplying the inverse at a pixels transformed location was not possible. This is down to the fact that other pixels may be required at that location in order to provide a white pixel representation at one instance, but a black pixel at another.

This resulted in a compromise between full secret recovery and a probabilistic secret recovery which would be closer to a "best effort" style of recovery. This best effort is mostly a trade-off between visual representation and resulting contrast [5]. The results from this process are good when the same secret is to be recovered after each transformation. The recovered quality would be similar in terms of contrast of the scheme which employ halftoning [3, 62]. The contrast ratio is typically around 1/4. The contrast suffers, when different secrets are added. The recovered secrets remain readable, but a much lower contrast is available. This is due to the nature of the scheme, completely new patterns have to be generated which must represent a unique letter each time. Using the same letter as the secret, the same patterns can be selected, therefore giving a higher contrast. This is not possible when using unique secrets.

Another important aspect of the scheme that must be mentioned and analyzed is the security. Traditional VC schemes exhibit good security due to the nature of the patterns that are chosen to represent pixels from the original. If a white pixel is to be represented then each pattern used to represent the white pixel is placed in each share. Similarly, corresponding patterns are placed in each share when a black pixel is to be represented. This results in a potential attacker (who has obtained one of the shares) having to make a 50/50 choice for each pixel pattern in order to guess the correct corresponding share. It can be observed that this is not feasible at all.

Based on each of the individual shares that are created for each of the secrets, a new pattern is created which is capable of revealing the secret while being transformed invariant. These new patterns work in the same way as the traditional patterns. An attacker would have to generate identical or corresponding patterns for each of the pixel representations. Correctly guessing those patterns to reveal one of the secrets is extremely unlikely, guessing the correct patterns that four secrets are revealed is even more unlikely again. The probabilities drop even further when four unique secrets are examined.

Randomness of the generated shares can also be examined in a security context. Visually, the shares do not leak any information about the secrets that are hidden within. On further inspection of the shares, the distribution of the pixels is uniform. This makes it much harder for an attacker to concentrate on a specific area on the share in order to force information to leak out regarding the secret.

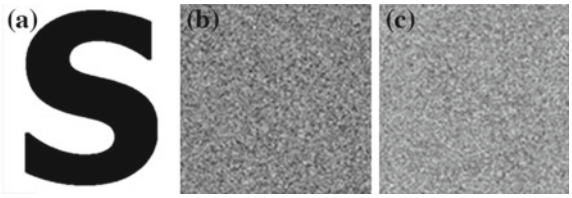

Fig. 2.15 The corresponding shares where all the secrets are identical. **a** Original secret. **b** Final S_1. **c** Final S_2

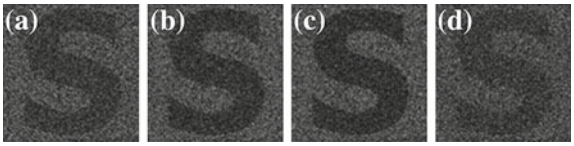

Fig. 2.16 The same secret recovered after different plane transformations. **a** Share two no transformation. **b** Share two transformed about the *horizontal axis*. **c** Share two transformed about the *vertical axis*. **d** Share two transformed about the *horizontal* and *vertical axis*

A number of results are presented within this chapter which show the capability of the scheme discussed. The two shares that are generated using this scheme are depicted in Fig. 2.15. These shares look like normal visual cryptography shares and do not give away any information about the secret or secrets concealed within.

When superimposed, these shares can recover the secret 'S'. Figure 2.16 provides the results of each of the transformations which the share can be made to go through in order to recover the same secret. Figure 2.16a is simply share one superimposed upon share two. Figure 2.16b shows the secret recovery after the share two has been transformed about the horizontal axis. Figure 2.16c highlights the secret recovery after the share two has been transformed about the vertical axis and Fig. 2.16d provides the final secret recovery after the share two has been transformed in both the horizontal and vertical axis.

In the following results, multiple and unique secrets have been embedded within a set of shares [22, 48]. Using the same technique as previously described, each of the secrets can be recovered. Figure 2.17 provides each of the secrets along with their corresponding shares. Each secret has its own set of decryption blocks embedded within the shares so that as each of the secrets is recovered, no information leaks out with regard to the other secrets. This is vital in any multi-secret sharing visual cryptography scheme [11, 41, 42, 51, 54].

The recovered results are presented within Fig. 2.18. Figure 2.18a shows the first 'T' from the secret 'TEST'. Figure 2.18b–d provide the remaining results after specific transformations have been performed on the second share as it is superimposed.

Using a simple transform, accurate and effective secret recovery can be achieved. No rotation is required, what is needed is a simple geometric transformation. This

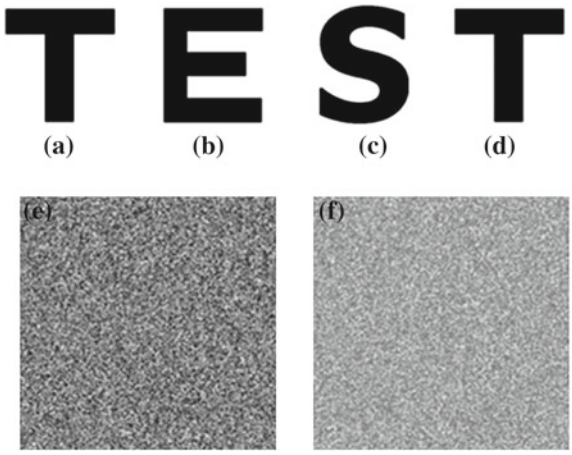

Fig. 2.17 The corresponding shares when all the secrets are unique. **a** Original secret one. **b** Original secret two. **c** Original secret three. **d** Original secret four. **e** Final S_1. **f** Final S_2

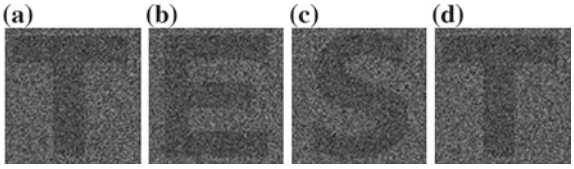

Fig. 2.18 The same secret recovered at different plane transformations. **a** Share two no transformation. **b** Share two transformed about the *horizontal axis*. **c** Share two transformed about the *vertical axis*. **d** Share two transformed about the *horizontal* and *vertical axis*

helps users recover secrets almost immediately without having to determine the correct angle and stacking order of the shares.

Testing these shares can be done very easily and quickly using the very simple Microsoft Paint program. The final S_1 can be loaded into the application, the final S_2 can be pasted on top and set to have a transparent background. Using the flip / rotate option [34], final S_2 can be manipulated vertically, horizontally and both in order to test the validity of the results.

From these results it is clear that contrast improvements can be made in particular when transforming share two twice, in both axial directions. The secret is still readable but the contrast does suffer. From the results and discussion presented, it is easy to see the advantages of a scheme likes this have over existing schemes. Reducing the alignment problem to a simple transform while being able to recover four identical or unique secrets is a great advantage to the end user [35]. This scheme removes the onus on the user when aligning and recovering the secrets.

This type of invariant placement of shares should be considered in the future when new cutting-edge VC schemes are being proposed. Making secret recovery easy for the end user is highly valuable and may help to push VC into the mainstream.

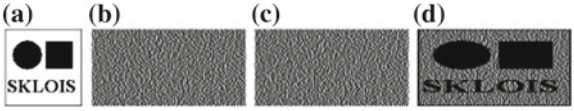

Fig. 2.19 An example of traditional VCS with pixel expansion 2, **a** is the original secret image with image size 100 × 100, **b** and **c** are the share images with image size 200 × 100, **d** is the recovered secret image with image size 200 × 100

2.4 Distortion Problems

For visual cryptography scheme (VCS) [3], normally, the size of the recovered secret image will be expanded by $m (\geq 1)$ times of the original secret image. In most cases, m is not a square number, hence the recovered secret image will be distorted. Sometimes, m is too large that will bring much inconvenience to the participants to carry the share images. In this section, we introduce a visual cryptography scheme which simulated the principle of fountains. The proposed scheme has two advantages: non-distortion and flexible (with respect to the pixel expansion). Furthermore, the presented scheme can be applied to any VCS that is under the pixel by pixel encryption model [15], such as VCS for general access structure [2], color VCS and extended VCS [3, 28], the VCS does not restrict to any specific underlying operation. Compared with other non-distortion schemes, the scheme discussed in this chapter is more general and simpler, real flexible, it has competitive visual quality for the recovered secret image.

In general, the recovered secret image of VCS will be expanded by (≥ 1) times over the size of the original secret image i.e., the size is m. However, in most cases, m is not a square number, hence the recovered secret image will be distorted. An example of distorted VCS can be found in Fig. 2.19.

In Fig. 2.19, the circle and square are compromised to an oval and a rectangle respectively and hence lead to the loss of information. This will not be allowed, especially when the aspect ratio is viewed as important information of the secret image [56]. To avoid distortion, many methods have been proposed. Naor and Shamir recommended adding extra to retain the value of m as a square number. In such a case, the pixel expansion of the scheme will increase significantly for some m and meanwhile may degrade the visual quality of the scheme [60]. Yang et al. proposed some aspect ratio invariant VCS's which relied on adding dummy to the shares, such methods also increase the overall pixel expansion [56]. Beside, their method is complicated, how to design a mapping pattern that reduces the number of dummy to the minimum is [58], as they said, a huge challenge, especially for some pixel expansions and secret image sizes [11, 32, 33].

Sometimes, m is so large that will bring much inconvenience to the participants to carry them. Some other studies, hence, consider size invariant VCS [26], i.e., VCS with no pixel expansion [17, 50]. For such schemes, the recovered secret image will have no distortion. The size invariant VCS's are usually called scheme (PVCS) [13, 55] for the reason that a secret pixel can only be recovered with a certain probability. In contrast to PVCS, the traditional VCS's are called deterministic visual cryp-

tography schemes (DVCS), which means that a secret pixel can be recovered deterministically. Because of PVCS's probabilistic nature, the recovered secret images of PVCS often have bad visual quality. Usually, better visual quality of the recovered secret image requires larger pixel expansion.

Definition 2.5 (Probabilistic VCS) Let k, n and m' be nonnegative integers, $\bar{l}$ and $\bar{h}$ be positive numbers, satisfying $2 \leq k \leq n$ and $0 \leq \bar{l} < \bar{h} \leq m$. The two collections of $n \times m'$ binary matrices (C_0, C_1) constitute a scheme, (k, n)-PVCS, if the following properties are satisfied:

Contrast For the collection C_0 and a share matrix $s \in C_0$, by v a vector resulting from the OR of any k out of the n rows of s. If $\overline{w(v)}$ denotes the average of the Hamming weights of v, over all the share matrices in C_0, then $\overline{w(v)} \leq \bar{l}$

Contrast For the collection C_1, the value of $\overline{w(v)}$ satisfies $\overline{w(v)} \geq \bar{h}$.

Security For any $i_1 < i_2 < \ldots < i_t$ in $1, 2, \ldots, n$ with $t < k$, the two collections of $t \times m'$ matrices $D_j, j = 0, 1$, obtained by restricting each $n \times m'$ matrix in $C_j, j = 0, 1$, to rows $i_1, i_2, \ldots, i_t$, are indistinguishable in the sense that they contain the same matrices with the same frequencies.

The definition of PVCS only considers the case with $n \times 1$ share matrices, we extend this definition to the $n \times m'$ case. And the definition of PVCS used the factor β to reflect the contrast, we use the values $\bar{l}$ and $\bar{h}$ to reflect the contrast. The common point of the three definitions of PVCS is that, for a particular pixel in the original secret image, the qualified participants can only correctly represent it in the recovered secret image with a certain probability. Because human eyes always average the high frequency black and white dots into gray areas, so the average value of the Hamming weight of the black dots in the area reflects the grayness of the area. The PVCS does not require the satisfaction of the difference in grayness for each pixel in the recovered secret image as the DVCS does. It only reflects the difference in grayness in the overall view.

The contrast [5] of the DVCS is fulfilled for each pixel (consisting of m subpixels) in the recovered secret image, however, this is quite different in the PVCS. The application of the average contrast, denoted by $\bar{\alpha}$. This term is often used in the PVCS, where the traditional contrast of the PVCS does not exist. Here we define the average contrast to be the average value of the overall contrast of the recovered secret image, i.e., the mean value of the contrast of all the pixels in the recovered secret image. According to our definition of the contrast $\alpha = (h - l)/m$, the average contrast can be calculated by the formula $\bar{\alpha} = (\bar{h} - \bar{l})/m'$ where $\bar{l}$ and $\bar{h}$ are the mean values of $w(v)$ for the black and white pixels in the overall recovered secret image respectively [30], and m is the pixel expansion of the PVCS. Because the number of pixels is large in the recovered secret image, the values $\bar{l}$ and $\bar{h}$ are equivalent to the mean values of the $w(v)$ in the collections C_1 and C_0 respectively. Note that, the DVCS also has the average contrast, and many proposed DVCSs in the literature have $\bar{\alpha} = \alpha$.

When comparing DVCS that has $\bar{\alpha} = \alpha$, in the overall view, the visual quality of the recovered secret image of the PVCS is the same as the visual quality of the recovered secret image of a DVCS. However, because of the probabilistic nature, a PVCS is disadvantaged in displaying the details of the original secret image, especially for the white background areas in the recovered secret image. A simple construction of PVCS based on a given DVCS (we will call it the original DVCS hereafter) can be as follows:

Construction 2.2 (PVCS) Denote (C_0, C_1) as the share matrix collections of a (k, n)-DVCS with pixel expansion m. The $n \times m$ share matrix collections of a (k, n)-PVCS, denoted by (C_0, C_1), can be generated by restricting each share matrix in C_0 and C_1 to its first m columns respectively.

According to the Construction 2.2 of PVCS, we have the following lemma:

Lemma 2.1 *The Construction (PVCS) generates a* (k, n)*-PVCS based on an original* (k, n)*-DVCS, where the average contrast of* (k, n)*-PVCS equals to the contrast of* (k, n)*-DVCS, i.e.,* $\bar{\alpha} = \alpha$.

2.4.1 The Fountain Algorithm

The main idea of our scheme is reflected by Fig. 2.20. Imagine a pool with several water nozzles as depicted in Fig. 2.20. The water with the same speed. In such a case the water will fill up the pool. Think of a blank image as a pool which has no distortion to the shape of original secret image (only differs in the size), think of the secret pixels of the original secret image as water injection nozzles that are evenly distributed in the pool, think of each secret pixel as water drops. As a result, the pool will be filled up by secret pixels, and hence becomes a share image. Note that, each water nozzle sprays water with the same speed, hence, each nozzle will spray almost the same number of pixels into the pool. We do the same process to all the share images, we get a VCS with no distortion. Certainly, the stacking of the share images will recover the secret image visually.

For the case of Fig. 2.20, the size of the secret image is 6×6, where each secret pixel is a water nozzle. The size of the share image can be flexible and its size equals to the size of the pool. The water nozzles (secret pixels) spray water (subpixels) and fill up the pool (secret image). Clearly, the generated share images will have no distortion with the secret image.

Formally, we give the following Algorithm (2.2).

In the above Algorithm (2.2), the new position (p', q') of a pixel at position (p, q) in the original secret image can be calculated as follows: $p' = p\sqrt{m_N} + X$ and $q' = q\sqrt{m_N} + Y$, X and Y are shown in Fig. 2.20.

Denote the length (resp. width) of the secret image as e (resp. f), then the length (resp. width) of the pool will be $e\sqrt{m_N}$ (resp. $f\sqrt{m_N}$), if $e\sqrt{m_N}$ (resp. $f\sqrt{m_N}$) is not an integer, then we will use $e\sqrt{m_N}$ (resp. $f\sqrt{m_N}$) instead.

Algorithm 2.2: The fountain algorithm

Input : The original secret image S_I, overall pixel expansion (pool expansion) m_N, an original DVCS with pixel expansion m_o.

Output: The non-distortion share images $S_1, S_2, \ldots, S_n$.

Step 1. Generate a blank image (pool), M, that is m_N times of the size of the original secret image and has no distortion, i.e., the length (resp. width) of M is $\sqrt{m_N}$ times of that of S_I. Generate n blank share images $S_1, S_2, \ldots, S_n$, which have the same size of M.

Step 2. For a secret pixel at position (p, q) in the original secret image, initialize an empty list $L_{p,q}$ which is used to store the positions of pixels in M (or $S_1, S_2, \ldots, S_n$).

Step 3. Distribute the secret pixels (water injection nozzles) of the original secret image evenly into the blank image M. Note that the corresponding coordinates of a pixel (p, q) of the original secret image is (p', q') in M now.

Step 4. For each subpixel in the blank image M, and the nearest secret pixel (water injection nozzle), suppose the position of the secret pixel is (p', q'). Add the position of the subpixel to list $L_{p,q}$.

Step 5. Sort each list $L_{p,q}$ with ascending order with respect to the distance to the secret pixel (water injection nozzle) (p', q').

Step 6. Denote $|L_{p,q}|$ as the number of positions in $L_{p,q}$. Encrypt the secret pixel (p, q) by applying the original DVCS in order, by $\lceil \frac{|L_{p,q}|}{m_0} \rceil$ times and distribute the pixels of the shares in order, to the positions of $L_{p,q}$ in $S_1, S_2, \ldots, S_n$ respectively, while discarding the redundant subpixels.

By saying "applying the original DVCS in order", we mean applying the DVCS by several times and concatenating the output shares (subpixels) in order, for each participants respectively.

Note that the overall pixel expansion m_N of our scheme is not necessarily equal to the pixel expansion of the original DVCS m_0, and it can be any value larger than 0.

In order to make things clear, we give the Example 2.5 for the (2, 2)-VCS, where the share matrix collections are as follows.

$$C_0 = \{\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}\} \text{ and } C_1 = \{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\}$$

Example 2.5 The recovered secret images of the presented scheme can be found in Fig. 2.21.

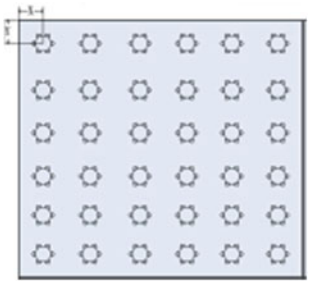

Fig. 2.20 A pool with 36 water injection nozzles

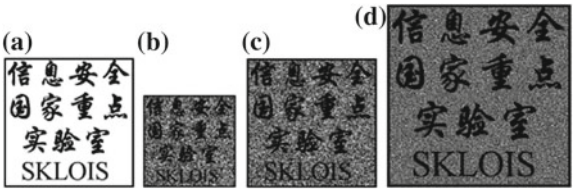

Fig. 2.21 **a** is the original secret image with size 300 × 300, **b** is the recovered secret image with overall pixel expansion $m_N = 0.5$ and image size 213 × 213, **c** is the recovered secret image with overall pixel expansion $m_N = 1$ and image size 300 × 300, **d** is the recovered secret image with overall pixel expansion $m_N = 2$ and image size 425 × 425

As depicted in Fig. 2.21, by comparing the three recovered secret images (b), (c) and (d), we can observe that, larger pixel expansion will result in better visual quality, and smaller pixel expansion will compromise poorer visual quality. Our scheme is flexible with respect to the compromise between the visual quality and overall pixel expansion of the recovered secret image. Formally, we give the following Theorem 2.4.

Theorem 2.4 *The fountain algorithm (2.2) generates a PVCS with no distortion and the size of its share images and recovered secret image can be flexible.*

2.4.2 Improving VC Quality

Suppose that the pixel expansion of the original DVCS is m_0 and the pool expansion is m_N. When the pool expansion m_N is not a multiple of the pixel expansion m_o, the pool expansion can be divided into two parts: the multiple part and the remaining part. Denote $d = \lceil \frac{m_N}{m_0} \rceil$, $m_N = t \cdot m_0 + 1$, $0 < t < m_0$, the multiple part contains $d \times m_0$ and the remaining part contains t (resp.$0 < t < m_0$) subpixels. The multiple part can be filled by repeating the original DVCS for d times. The remaining part can be filled by choosing t columns from the basis matrices (resp. the remaining part is filled by a PVCS with pixel expansion t). So when m_N is not a multiple of m_0, pool expansion will be filled by $d \times m_0$ from the original DVCS and t from a PVCS. The probabilistic function will add some visual-noise to the recovered image, which will blur the details in the recovered image. Thus the visual quality of the recovered image will be degraded. So we would like to remove the PVCS part. Our strategy is: the remaining part is assigned by m_0 with probability t/m_0 or assigned by no with probability $(m_0 - t)/m_0$. On average, the remaining part is assigned by t subpixels. From an overall view, a pixel of the original secret image (a water nozzle) is assigned by $\lceil \frac{m_N}{m_0} \rceil \cdot m_0$ with probability $(m_0 - t)/m_0$, and is assigned $\lceil \frac{m_N}{m_0} \rceil \cdot m_0$ with probability t/m_0. Suppose there is a Boolean matrix the same size as the original secret image, then there is a one-to-one mapping between a secret pixel and an entry in the Boolean matrix. If the secret pixel is assigned by $\lfloor \frac{m_N}{m_0} \rfloor \cdot m_0$ subpixels, we denote the corresponding entry as 0, else if the secret pixel is assigned

Algorithm 2.3: The fountain algorithm.

Input : The original secret image S_I, overall pixel expansion m_N, an original DVCS with pixel expansion m_0.

Output: The non-distortion shares $S_1, S_2, \ldots, S_n$.

Pre-process Let $s = \lfloor \frac{m_N}{m_0} \rfloor \cdot m_0$, $t = \lceil \frac{m_N}{m_0} \rceil \cdot m_0$ where s and t satisfy $s \times m_0 \leq m_N \leq t \times m_0$. Let a and b be two nonnegative real numbers satisfying $a + b = 1$ and $a \times (s \times m_0) + b \times (t \times m_0) = m_N$. Suppose the size of S_I is $m \times n$. Then we generate an $m \times n$ random Boolean matrix D, in which 0 appears with probability a and 1 appears with probability b. Then there is a one-to-one mapping between the pixels of the original secret image and the entries of D.

Step 1-3. Step 1-3 are as same as that of algorithm 2.2.

Step 4. Step 4 For each secret pixel (water injection nozzle) in the blank image M, if the entry of D is 0, and s/m_0 nearest and undistributed subpixels, else if the entry of D is 1, and $t \times m_0$ nearest and undistributed subpixels. Suppose the position of the secret pixel is (p', q'). Add the positions of the to list $L_{p,q}$.

Step 5. Encrypt the secret pixel (p, q) by applying the original DVCS in order, by s or t times and distribute the subpixels of the shares in order, to the positions of $L_{p,q}$ in $S_1, S_2, \ldots, S_n$ respectively. The undistributed in the pool are simply set to black. If the entry in D is 0, we distribute $s \times m_0$ for the corresponding pixel of the original secret image. If the entry in D is 1, we distribute $t \times m_0$ for the corresponding pixel of the original secret image.

by $\lceil \frac{m_N}{m_0} \rceil \cdot m_o$ subpixels, we denote the corresponding entry as 1. Then we will get a Boolean matrix for which $t \times m_0$ proportion of its entries are 1, and the entries of 1 are evenly distributed. Meanwhile the entries of 0 are evenly distributed in the Boolean matrix too. For example, for a (2,2)-DVCS with pixel expansion 2. Suppose the pool is three times as large as the original secret image. We distribute two for 50 % water nozzles and four for the remaining 50 % water nozzles, where there will be three for each water nozzle on average. And the two cases (two for a water nozzle, four for a water nozzle) are evenly distributed in the pool.

In the above construction, if the pool expansion m_N is a multiple of the pixel expansion m_0, hence every water nozzle will be assigned by m_N subpixels. If the pool expansion m_N is smaller than the pixel expansion of the original DVCS m_0, then each water nozzle will be assigned by m_0 with probability m_N/m_0 or assigned by no with probability $(m_0 - m_N)/m_0$, which implies that $(m_0 - m_N)/m_0$ of the secret pixels in the original secret image are lost in the recovered secret image on average.

In the following, we give a comparison for Fig. (2.21) and Algorithm (2.3) for (2, 2)-VCS, where the original DVCS is the same as that of Example 2.1

Example 2.6 Suppose that the pool is 1.37311 (this value can be arbitrarily chosen) times as large as that of the original secret image. Thus the length (resp. width) of the pool is 1.1718 times the length (resp. width) of the original secret image. The parameters in the stage of pre-process of Algorithm (2.3) are $m_N = 1.37311$, $m_0 = 2$, $s = 0$ and $t = 1$. In Algorithm (2.3), we assign one or two subpixels for

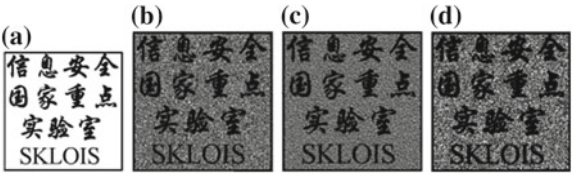

Fig. 2.22 **a** is the original secret image characters with image size 300 × 300. **b** and **c** are the recovered secret images of Algorithm 2.2 and Algorithm 2.3 with image size 352 × 352 respectively. **d** is the recovered secret image of Yang's VCS with image size 352 × 352

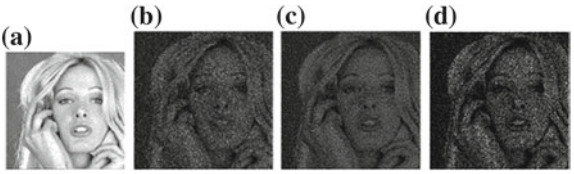

Fig. 2.23 **a** is the original secret image Human face with image size 512 × 512. **b** and **c** are the recovered secret images of Fig. 2.21 and Algorithm 2.3 with image size 600 × 600 respectively. **d** is the recovered secret image of Yang's VCS with image size 600 × 600

each secret pixel (water injection nozzle), for which about 37.311 % secret pixels are assigned with two (filled by a (2, 2)-DVCS) and about 62.689 % secret pixels are assigned with one subpixel (filled by a (2, 2)-PVCS with pixel expansion 1). In Algorithm (2.3), we assign two for 68.6555 % secret pixels (water injection nozzles) and assign no subpixel for 31.3445 % secret pixels (water injection nozzles).

We make use of two types of secret images: characters and human face. The original secret images are in the first column. The visual quality of Algorithm (2.3) can be found in the second column of Figs. 2.22 and 2.23. The visual quality of Construction 3 can be found in the third column of Figs. 2.22 and 2.23.

As depicted in Figs. 2.22 and 2.23, by comparing the recovered secret images (generated by Algorithm (2.3) and that of Algorithm (2.3), we can observe that, the recovered secret images for both constructions are clear and one can easily identify the contents of the original secret image. One also can observe that Construction 2.3 results in better visual quality than Construction 2 with respect to the evenness. Particularly, the recovered secret image is much more even at the white background areas.

2.5 Thin Line Problems (TLP)

Traditionally, the SIVCS is only suitable to encrypt coarse secret images that do not contain much detail information. The reason is that, SIVCS can only recover the secret image from an overall view point, each secret pixel can only be correctly

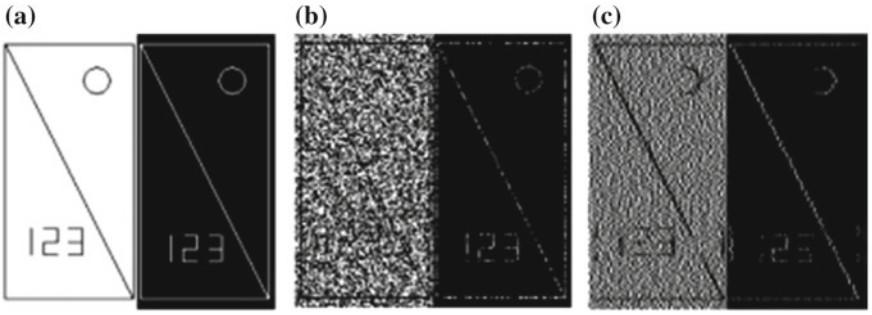

Fig. 2.24 The visual quality for secret images with thin lines, the image size is: 200 × 200. **a** Secret images **b** TLP-1 **c** TLP-2

represented with a certain probability in the recovered secret image. In such a case the thin lines, in the secret image, are usually unclear and misrepresented in the recovered secret image of SIVCS, where we call such phenomena the thin line problem (TLP). In this section, we classify the TLP into three types.

According to the recovered secret image **b** of Fig. 2.24, for P-SIVCS, the visual quality of the recovered secret image is seriously degraded. One can observe that, there are many chaotic pixels appear in the recovered secret image, especially for the white background areas. It is hard to identify the thin lines from the white background. We call this type of thin line problem as the first type thin line problem (TLP-1).

According to the recovered secret image **c** of Fig. 2.24, it is clear that the thin lines can be seen more clearly especially the horizon lines, diagonal lines and the right part of the circle, i.e., the TLP-1 is avoided in the Construction 2.2. The reason is that, it has smaller variance of the darkness level of each block of two secret pixels. However, according to Construction 2.2, because every m of $B_{m,b}$ blocks are encrypted by b of M_1 and $m - b$ of M_0 alternatively, it is possible that the patterns in the secret image can be falsely recovered, especially for images only consisting of thin lines, where the blocks on a thin line may be always encrypted by M_0 (resp. M_1), which means the thin line may be missing if it is a black (resp. white) thin line on the white (resp. black) background. This problem can be clearly observed in (c) of the Fig. 2.24, where the vertical lines and the left part of the circle are missing. We call this type of thin line problem as the second type thin line problem (TLP-2).

One way to solve the TLP-1 and TLP-2 is to replace thin lines by thick lines in the secret images. They also calculated the reference thickness of the lines which can be found in Table 1. However, if the secret information in the secret image is characters, maps or geometry shapes etc., then after replace the thin lines by thick lines. One needs to enlarge the secret image and put down the given amount of secret information. This process will result in larger share images. Recall that the main advantage of SIVCS is the ability to generate smaller share images. Hence, for

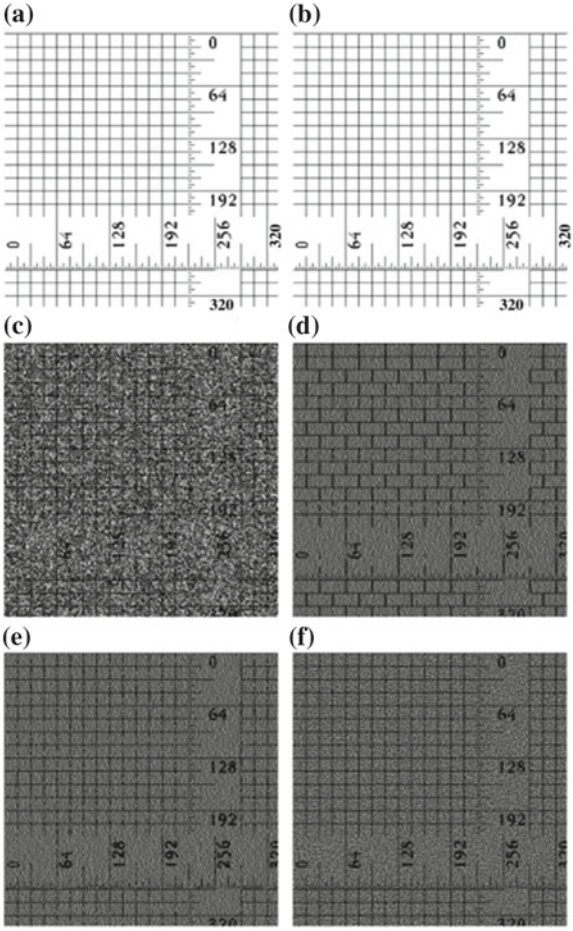

Fig. 2.25 Experimental results for image ruler, the image size is: 500 × 500. **a, b** Secret images **c** TLP-3 **d** TLP-2 **e, f** TLP has been avoided

Yang's solution for TLP-1 and TLP-2, the advantage of SIVCS on the pixel expansion is no more.

Another problem of the recovered secret image **c** of Fig. 2.24 is that, a thin line in the secret image may be represented by a thicker line. Particularly, the vertical and diagonal thin lines, that are with width 1, are represented by lines with width 2. The reason of this problem is that, a $B_{m,b}$ block may be encrypted by M_1 (resp. M_0), and in the recovered secret image, the $B_{m,b}$ block is represented by m pixels which contains h (resp. l) black pixels, and these black pixels spread evenly in the m positions of the $B_{m,b}$ block, hence, the human eyes will view the block as a uniform area, i.e., the thin lines become as thick as the size of the block. We call this problem the third thin line problem (TLP-3).

One also can observe the thin line problems TLP-1, TLP-2 and TLP-3 in the images **c** and **d** of Fig. 2.25, where we use the fine image Ruler as the original secret image.

In Fig. 2.25 shows, all the three thin line problems TLP-1, TLP-2 and TLP-3 are avoided. For the TLP-3, taking the encryption of b black pixels in a block $B_{r_s,b}$ as example, because the black pixels and white pixels are encrypted separately, the $b \cdot h/m$ black pixels in the recovered secret image only spread evenly in the original b positions of the b black pixels in $B_{r_s,b}$. Similarly, for the $r_s - b$ white pixels in $B_{r_s,b}$, the $(r_s - b) \cdot l/m$ black pixels in recovered secret image only spread evenly in the original $r_s \times b$ positions of the $r_s \times b$ white pixels in $B_{r_s,b}$. Hence, the average darkness level for the white and black pixels are different, and the human eyes can identify the difference, i.e., the TLP-3 problem is avoided in the recovered secret image of Construction 2.4.

The thin line problem is, more or less, a common problem for all kinds of SIVCS. There may be no perfect solution for the secret image which is a simple and regular line image.

References

1. Akansu AN, Haddad RA (1992) Multiresolution signal decomposition: Transforms, subbands, and wavelets. Academic Press, Boston, MA
2. Ateniese G, Blundo C, Santis AD, Stinson DR (1996) Visual cryptography for general access structures. Inf Comput 129:86–106
3. Ateniese G, Blundo C, Santis AD, Stinson DR (2001) Extended capabilities for visual cryptography. ACM Theory Comput Sci 250(1–2):143–161
4. Biham E, Itzkovitz A (1997) Visual cryptography with polarization. In CRYPTO '98
5. Blundo C, Santis A, Stinson DR (1999) On the contrast in visual cryptography schemes. J Cryptol 12(4):261–289
6. Blundo C, Bonis A, Santis A (2001) Improved schemes for visual cryptography. Des Codes Cryptogr 24:255–278
7. Blundo C, D'Arco P, Santis A, Stinson DR (2003) Contrast optimal threshold visual cryptography schemes. SIAM J Discrete Math 16(2):224–261
8. Bose M, Mukerjee R (2010) Optimal (k, n) visual cryptographic schemes for general k. Des Codes Cryptogr 55:19–35
9. Chang CY (2000) Visual cryptography for color images (Masters Thesis). National Central University (Taiwan)
10. Chen TH, Tsao KH, Wei KC (2008) Multiple-image encryption by rotating random grids. International Conference on Intelligent Systems Design and Applications 3:252–256
11. Chen YF, Chan YK, Huang CC, Tsai MH, Chu YP (2007) A multiple-level visual secret-sharing scheme without image size expansion. Inf Sci 177:4696–4710
12. Cimato S, Prisco R, Santis A (2005) Optimal colored threshold visual cryptography schemes. Des Codes Cryptogr 35:311–335
13. Cimato S, Prisco RD, Santis AD (2006) Probabilistic visual cryptography schemes. Comput J 49(1):97–107
14. Cimato S, Yang CN (2011) Visual cryptography and secret image sharing. CRC Press, Taylor & Francis, UK
15. Degara-Quintela N, Perez-Gonzalez F (2003) Visible encryption: using paper as a secure channel, security and watermarking of multimedia contents. In: Proceedings of SPIE'03, 5020

16. Droste S (1996) New results on visual cryptography. Springer, Berlin CRYPTO'96.
17. Fang WP (2009) Non-expansion visual secret sharing in reversible style. Int J Comput Sci Netw Secur 9(2):204–208
18. Floyd RW, Steinberg L (1976) An adaptive algorithm for spatial grey scale. In: Proceedings of the Society of Information Display. 17:75–77
19. Hou YC, Chang CY, Lin F (1999) Visual cryptography for colour images based on colour decomposition. In Proceedings of 5th conference on information management. 584–591
20. Hou YC, Chang CY, Tu SF (2001) Visual cryptography for color images based on halftone technology. In: Proceedings of international conference on information systems, analysis and synthesis (world multiconference on systemics, cybernetics and informatics)
21. Hou YC (2003) Visual cryptography for color images. Pattern Recognit 36:1619–1629
22. Hou YC, Xu CS (2003) A probability-based optimization model for sharing multiple secret images without pixel expansion. J Inf Technol Soc 2:19–38
23. Hou YC, Tu CF (2004) Visual cryptography techniques for colour images without pixel expansion. J Inf Technol Soc 1:95–110
24. Hou YC, Tu SF (2005) A visual cryptographic technique for chromatic images using multi-pixel encoding method. J Res Pract Infor Technol 37(2):179–191
25. Hsu HC, Chen TS, Lin YH (2004) The ringed shadow image technology of visual cryptography by applying diverse rotating angles to hide the secret sharing. Netw Sens Control 2:996–1001
26. Ito R, Kuwakado H, Tanaka H (1999) Image size invariant visual cryptography. IEICE transactions on fundamentals of electronics, communications and computer sciences. E82-A(10), 2172–2177
27. Jin D (2003) Progressive color visual cryptography (Masters Thesis). National University of Singapore, Singapore
28. Katoh T, Imai H (1996) An extended construction method for visual secret sharing schemes. IEICE Trans J79–A(8):1344–1351
29. Kobara K, Imai H (1996) Limiting the visible space visual secret sharing schemes and their application to human identification. Springer, Berlin, pp 185–195 ASIACRYPT '96. LNCS
30. Koga H (2002) A general formula of the (t, n)-threshold visual secret sharing scheme. Springer, Berlin, pp 328–345 LNCS 2501 (ASIACRYPT '02)
31. Krause M, Simon HU (2003) Determining the optimal contrast for secret sharing schemes in visual cryptography. Comb Probab Comput 12(3):285–299
32. Kuwakado H, Tanaka H (2004) Size-reduced visual secret sharing scheme. IEICE Trans Fundam E87–A(5):1193–1197
33. Lin CH (2002) Visual cryptography for color images with image size invariable shares (Masters Thesis). National Central University, Taiwan
34. Lin SJ, Chen SK, Lin JC (2010) Flip visual cryptography (FVC) with perfect security, conditionally-optimal contrast, and no expansion. J Vis Commun Image Represent 21(8):900–916
35. Liu F, Wu CK, Lin XJ (2009) The alignment problem of visual cryptography schemes. Des Codes Cryptogr 50(2):215–227
36. Liu F, Wu CK, Lin XJ (2010) A new definition of the contrast of visual cryptography scheme. Inf Process Lett 110:241–246
37. Liu F, Wu CK, Lin XJ (2010) Step construction of visual cryptography schemes. IEEE Trans Inf Forensics Secur 5(1):27–38
38. Liu F, Wu CK (2011) Embedded meaningful share visual cryptography schemes. IEEE Trans Inf Forensics Secur 6(2):307–322
39. Monoth T, Anto B (2010) Tamperproof transmission of fingerprints using visual cryptography schemes. Procedia Comput Sci 2:143–148
40. Naor M, Shamir A (1995) Visual cryptography. Springer, Berlin, pp 1–12 LNCS 950 (EUROCRYPT '94)
41. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
42. Shyu SJ (2006) Efficient visual secret sharing scheme for color images. Pattern Recognit 39(5):866–880

43. Shyu SJ, Huang SY, Lee YK, Wang RZ, Chen K (2007) Sharing multiple secrets in visual cryptography. Pattern Recognit 40(12):3633–3651
44. Shyu SJ (2007) Image encryption by random grids. Pattern Recognit 40:1014–1031
45. Surekha B, Swamy G, Rao KS (2010) A multiple watermarking technique for images based on visual cryptography. Comput Appl 1:77–81
46. Viet DQ, Kurosawa K (2004) Almost ideal contrast visual cryptography with reversing. In Topics in Cryptology - CT-RSA. Springer, Berlin, pp 353–365
47. Weir J, Yan W (2010) A comprehensive study of visual cryptography. transactions on data hiding and multimedia security. Springer, Berlin
48. Weir J, Yan W (2009) Sharing multiple secrets using visual cryptography. In IEEE international symposium on circuits and systems (ISCAS'09). 509–512
49. Weir J, Yan W (2010) Resolution variant visual cryptography for street view of Google maps. In IEEE international symposium on circuits and systems (ISCAS'10)
50. Wu XY, Wong DS, Li Q (2009) Threshold visual cryptography scheme for color images with no pixel expansion. In: Proceedings of the second symposium international computer science and computational technology (ISCSCT'09). 310–315
51. Wu HC, Chang CC (2005) Sharing visual multi-secrets using circle shares. Comput Stand Interfaces 28:123–135
52. Wu C, Chen L (1998) A study on visual cryptography (Masters Thesis). National Chiao Tung University, Taiwan
53. Yan W, Duo J, Kankanhalli MS (2004) Visual cryptography for print and scan applications. In: Proceedings of IEEE international symposium on circuits and systems (ISCAS'04), Canada. 572–575
54. Yang CN, Laih CS (2000) New coloured visual secret sharing schemes. Des. Codes Cryptogr. 20:325–335
55. Yang CN (2004) New visual secret sharing schemes using probabilistic method. Pattern Recognit Lett 25:481–494
56. Yang CN, Chen TS (2005) Aspect ratio invariant visual secret sharing schemes with minimum pixel expansion. Pattern Recognit Lett 26:193–206
57. Yang CN, Chen TS (2005) Size-adjustable visual secret sharing schemes. IEICE Trans Fundam E88–A(9):2471–2474
58. Yang CN, Chen TS (2006) New size-reduced visual secret sharing schemes with half reduction of shadow size. IEICE Trans Fundam E89–A(2):620–625
59. Yang CN, Chen TS (2006) Reduce shadow size in aspect ratio invariant visual secret sharing schemes using a square block-wise operation. Pattern Recognit 39:1300–1314
60. Yang CN, Chen TS (2007) Extended visual secret sharing schemes: improving the shadow image quality. Int J Pattern Recognit Artif Intell 21(5):879–898
61. Yang CN, Chen TS (2007) Visual secret sharing scheme: prioritizing the secret pixels with different pixel expansions to enhance the image contrast. Opt Eng 46(9):097005
62. Zhou Z, Arce GR, Crescenzo GD (2006) Halftone visual cryptography. IEEE Trans Image Process 15(8):2441–2453
63. Zhu BS, Wu JK, Kankanhalli MS (2003) Print signatures for document authentication. In: Proceedings of ACM conference on computer and communications security. 145–153

http://www.springer.com/978-3-319-09643-8

Visual Cryptography for Image Processing and Security
Theory, Methods, and Applications
Liu, F.; Yan, W.Q.
2014, XVI, 145 p. 49 illus., 6 illus. in color., Hardcover
ISBN: 978-3-319-09643-8