This is to certify that the

dissertation entitled

Transform Encryption Coding

presented by

Chung Jung Kuo

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Electrical Engineering

John R. Deller Jr.
Major professor
John R. Deller, Jr., Ph.D.

Date May 16, 1990

TRANSFORM ENCRYPTION CODING

By

Chung Jung Kuo

A DISSERTATION

submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1990

ABSTRACT

TRANSFORM ENCRYPTION CODING

By

Chung Jung Kuo

This dissertation consists of two parts. The first part presents an independence technique for any m-dependent random variable and its application to data compression. The second part deals with new types of two-dimensional pseudo-noises, quasi m-arrays and the Gold code arrays. These pseudo-noises are used during the application of independence technique to ensure signal encryption.

In the first part of this dissertation, a technique to generate independent random variables from the m-dependent ones is proposed and its validity is proved. The probability density function of any resulting independent random variable is also shown to be Gaussian. This technique can be applied to any m-dependent signal, and the resulting independent signal is unrecognizable. Therefore, signal encryption is also achieved. Since these encrypted signals are independent Gaussian random variables, they can be quantized individually. A simple scalar quantizer can be used for these independent Gaussian random variables and the sum of mean-square quantization errors can be minimized. In addition, vector quantization is no longer necessary because these encrypted signals are already independent.

Secondly, two-dimensional quasi m-arrays and Gold code arrays are developed. These arrays are easily generated by the modulo-2 addition of m-sequences. The cyclic correlation properties of these arrays are studied. The cyclic auto-correlation of

any array is similar to a delta-function. In addition, the cyclic cross-correlation between any two arrays is small compared with their cyclic auto-correlation peaks. Therefore, these arrays have the quasi-orthogonal property. In addition, these arrays can be generated easily which is helpful for many applications.

Application of the independence technique to cosine and Hadamard transform image coding is also studied. The quasi m-arrays are used before transform coding to ensure the signal encryption. Therefore, this technique is named as *transform encryption coding*. Computer simulation shows a saving of 0.5 bit/pixel can be obtained with satisfactorily reconstructed images by transform encryption coding. However, 1 bit/pixel is required to have comparable results by transform coding. In addition, the *blocking effect* is also removed by the proposed transform encryption coding. While only an application of the proposed technique to the images is given, it can also be applied to other signals such as speech.

To My Family

ACKNOWLEDGEMENTS

I would like to acknowledge the late Dr. H. Rigas and Dr. J.R. Deller, my thesis advisors, for their invaluable inspiration and guidance throughout the course of this dissertation. Although Dr. Rigas can not see this work completed, her inspiration to this work is greatly appreciated. Special thanks go to Dr. Deller for his kindness in resuming the duty of advisor after Dr. Rigas passed away. I also like to thank Dr. A.K. Jain for his valuable comments and constructive suggestions which helped me a lot in this dissertation. Special thanks also go to Drs. P. Parker and M. Siegel for lending a hand and moral support. The consideration, concern, and support of Drs. Deller, Jain, Siegel, and Parker after Dr. Rigas passed away are also greatly appreciated. Without the help from my advisors and committee members, this work would never have been possible. The kindness of Dave Marks in shooting the pictures and the financial support offered by the Department of Electrical Engineering in the form of teaching assistantship are also appreciated.

Personally, I am in debt to my parents and like to thank them for their continuous encouragement and support throughout all these years. Without their sacrifice, I would not be who I am today.

Finally, I like to thank my wife, Chih J. Hsu, who takes good care of me and our baby, Andrew, such that I do not have to worry about anything but study. Her understanding, patience, and love during the period of my Ph.D. study is truly appreciated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

I.1 The Problems

For digital information transmission or storage, a continuous signal must be converted into a digital one. Therefore, how to efficiently quantize and code the continuous signals is important. The correlation between adjacent samples of signals is often high. Hence the redundancies in these samples are also large. These redundancies will decrease the quantization and coding efficiency if measures are not taken to reduce them. A technique to reduce these redundancies is transform coding [1]-[10]. The idea of transform coding is to transform the dependent signals into the transform coefficients where the redundancies are greatly reduced [7]-[10]. Instead of quantizing the original signal, these transform coefficients are quantized. The Karhunen-Loēve (K-L) transform is often considered as the "best" in the sense that it produces uncorrelated transform coefficients [1]-[12]. Since the K-L transform does not have a fast computation algorithm, it is usually used only as benchmark comparison for other unitary mathematical transformations. However, the K-L transform coefficients are still dependent, and redundancies remain . Therefore, vector quantization should be used to quantize these dependent K-L transform coefficients [1]-[10], [13]-[15]. Problems with the techniques of vector quantization are the difficulty in estimating the joint probability density function and the complexity in searching for the decision and reconstruction vectors. However, if scalar quantization is used for these dependent K-L transform coefficients, the efficiency is low. Therefore, the first part of this dissertation is to present a technique to produce the independent K-L transform coefficients from dependent ones. Hence the efficiency of scalar quantization on these K-L transform coefficients increases and vector quantization is no longer necessary.

When the bit rate is sufficiently low, the blocking effect [7], which results from nonoverlapping coding of each subimage, becomes highly visible. Reconstructed images exhibiting blocking effects can be very unpleasant visually, and often become the dominant degradation. This problem will also be solved in this dissertation.

Maximal length sequences (m-sequences) with good correlation properties are useful for radar and navigation applications [15]-[19]. Two dimensional maximum area arrays (m-arrays, pseudo-noises) have been studied by Normura et al. [20] and MacWilliams and Sloane [21]. Correlation properties of these m-arrays are similar to those of m-sequences. Applications of these m-arrays are found in coded aperture imaging [22], "add-on" data transmission [23], pattern synchronization [24], and code division image multiplexing [25]. Problems with these m-arrays are their symmetrical appearance, the restriction in the selection size, the necessity to generate long m-sequences, and the complexity in construction. Therefore, the second part of this dissertation is to provide new types of two-dimensional pseudo-noises to solve the problems above.

Sometimes, the security is another concern for the information transmission or storage. Most encryption techniques are either public- or private-key type based on algebraic coding theory [26]-[27]. However, these techniques either require large computational efforts or are easy to attack by unauthorized persons. In other words, there is a tradeoff in the calculation burden and security. This problem will also be considered and solved in this dissertation.

I.2 Purposes and Significances

A technique to generate independent random variables from generally dependent ones is first presented. The probability density function of these resulting independent random variables is also shown to be Gaussian. Therefore, a simple scalar quantizer is enough to quantize these independent Gaussian random variables and minimize the sum of mean-square quantization errors. In addition, vector quantization is no longer necessary. This independence technique can be applied to any signal, and the resulting signal is always unrecognizable. Therefore, the signal encryption is achieved, and the security of information during the transmission or storage is also maintained. Besides, the blocking effects can also be removed by this technique because the decryption process will spread the quantization error over the whole image.

Secondly, two new types of two-dimensional pseudo-noises, the quasi m-arrays and the Gold code arrays, are presented in this dissertation. These arrays are easy to generate and appear random. The cyclic auto-correlation of any quasi m-array is similar to a delta-function. In addition, the cyclic cross-correlation between any two quasi m-arrays is small compared with their cyclic auto-correlation peaks. The quasi m-arrays are so-named because their cyclic correlation properties are similar to those of the m-arrays. Two dimensional Gold code arrays generated by the same construction method are also studied. The correlation properties of these Gold code arrays are similar to those of the quasi m-arrays. In addition, the Gold code array provides families of arrays in which any two arrays have the same bound on their cyclic cross-correlation. Therefore, both the quasi m-arrays and the Gold code arrays have the quasi-orthogonal property.

Finally, the pseudo-noises are used in the application of independence technique during transform coding to achieve the signal encryption. To break this type of cryptography is difficult and time-consuming because there are too many pseudo-noises available. In addition, this encryption process requires only two FFTs which can be easily and quickly obtained.

In summary, the significances of this dissertation are the following.

1. Any m-dependent signal can be decomposed into independent Gaussian random variables by the proposed independence technique. Therefore, a simple scalar quantizer is enough to minimize the sum of mean-square quantization errors, and vector quantization is no longer necessary.
2. The blocking effect, which results from nonoverlapping coding of each subblock of signal, is removed. Therefore, the reconstructed signal from the proposed technique is much more acceptable by observors.
3. An easily obtained cryptography is proposed. Therefore, the security of information during the transmission or storage is guaranteed.
4. Two new types of two-dimensional pseudo-noises are developed. These pseudo-noises are easy to generate and appear random.
5. These pseudo-noises have the quasi-orthogonal properties which are useful for many applications.

I.3 Organization

The organization of this dissertation is as follows. In Chapter II, some background material is presented. Section II.1 shows the optimal scalar quantizer for a single random variable. This includes the Max and companding quantization. How to apply the Max quantization to a sequence of independent random variables is also reviewed. Section II.2 presents the fundamental considerations for transform image coding. Finally, the properties of m-sequences and Gold code sequences are discussed in Section II.3. The definition of correlation for these sequences are also given.

The independence theorem and its application are presented in Chapter III. The transform encryption coding is developed in Section III.1. A theorem to support the claim of the independence technique is proved. Some issues related to the independence theorem are also studied. In Section III.2, a computational algorithm for signal encryption is developed. This algorithm is proved to satisfy the independence theorem. Comparisons between this encryption technique and the other related techniques are discussed. The number of operations required by the encryption process are also calculated.

Chapter IV is devoted to the study of two-dimensional pseudo-noises. In Section IV.1, the construction method for the quasi m-arrays is proposed and the correlation properties of these arrays are investigated. Advantages and limitations of these arrays in applications are also studied. Some characteristics of the quasi m-arrays are also summarized in this section. Section IV.2 is similar to Section IV.2 except the attention is switched to the Gold code arrays.

The simulation results of transform coding are given in Chapter V. Section V.1 discusses the considerations for simulation. Simulation verifications of the independence theorem are also provided. The results of simulations are shown in Section V.2 These results are compared according to signal-to-noise ratio and paired-comparison method. Advantages and limitations of transform encrypted image coding are also discussed.

The conclusion is given in Chapter VI. Section VI.1 summarizes the limitations and advantages of the proposed transform encryption coding and two-dimensional pseudo-noises. Some recommendations for further work are also presented in Section VI.2.

Finally, a brief discussion about optical encryption are presented in the Appendix. Some computer simulations are shown there. A method to implement cyclic convolution by linear convolution is presented. Directions for further work are also discussed.

CHAPTER II

BACKGROUND

II.1 Scalar Quantization

An analog quantity that is to be processed by a digital computer or system is usually represented by an integer number proportional to its amplitude. During the conversion process, the analog quantity must be represented by a digital one with finite precision. This process is called quantization. The following contains an analytic treatment of the quantization process which can be applied to any signal [1]-[10].

Let x be a real scalar random variable with a known probability density function p(x). The quantization problem entails specification of a set of decision levels d_j and reconstruction level r_j such that if $d_j \le x < d_{j+1}$ then x is quantized to a reconstruction value r_j [10]. Decision and reconstruction levels are chosen to minimize some desired quantization error measure between x and r_j. The quantization error measure is usually defined as the mean-square error because this measure is tractable and correlates well with subjective criteria.

For 2^b quantization levels, the mean-square quantization error is

$$E = \sum_{j=0}^{2^b-1} \int_{d_j}^{d_{j+1}} (x-r_j)^2 p(x)dx. \qquad \text{(II.1)}$$

Setting the partial derivatives of the above error expression with respect to the decision and reconstruction levels equal to zero yields

$$r_j = 2d_j - r_{j-1} \tag{II.2a}$$

and

$$r_j = \frac{\int_{d_j}^{d_{j+1}} xp(x)dx}{\int_{d_j}^{d_{j+1}} p(x)d(x)}. \tag{II.2b}$$

Recursive solution of these equations for a given probability density function p(x) provides optimal values for the decision and reconstruction levels. The solutions for some given probability density functions were studied by Max [28]. Therefore, it is also called *Max quantization*. The mean-square error of Max quantization is

$$E_{min} = E[x^2] - \sum_{j=0}^{2^b-1} r_j^2 P[d_j \leq x < d_{j+1}]. \tag{II.3}$$

If the probability density function is uniform, then the decision and reconstruction levels are uniformly spaced in the range of x. Therefore, it is desirable to perform a nonlinear transformation on x such that the transformed variable is uniformly distributed. Hence a simple placement of decision and reconstruction levels can then be used. After quantization is made, an inverse nonlinear transformation must be taken. This method is called *companding quantization* [10]. For companding quantization, the transformed variable is $y = T[x]$, where T[x] is chosen such that the probability density function of y is uniform, that is, $p(y) = 1$ for $-0.5 \leq y \leq 0.5$. If x is a zero mean random variable, then the proper transformation function T[x] is the cumulative probability distribution of x. For example, let x be a zero mean Gaussian random variable, and

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}}. \qquad \text{(II.4)}$$

Then the forward transformation for companding quantization is

$$y = \frac{1}{2}\mathrm{erf}[\frac{x}{\sqrt{2}\sigma}], \qquad \text{(II.5)}$$

and the inverse transformation for companding quantization is

$$x = \sqrt{2}\sigma\mathrm{erf}^{-1}[2y]. \qquad \text{(II.6)}$$

Max quantization can also be applied to a sequence of independent random variables [29]-[33]. If b_i bits are allocated to the i^{th} random variable, then the bit allocation must sum to a fixed number B,

$$B = \sum_{i=1}^{N} b_i, \text{ where } b_i \geq 0 \text{ for } i = 1, 2, ..., N. \qquad \text{(II.7)}$$

How to choose the bit allocation b_i for a fixed B to minimize the sum of mean-square Max quantization errors has been studied by Segall [29]. Mathematically, the problem is

$$\min\sum_{i=1}^{N}\left\{E[x_i^2]-\sum_{j=0}^{2^{b_i}-1} r_{i,j}^2 P[d_{i,j} \leq x_i < d_{i,j+1}]\right\} \qquad \text{(II.8)}$$

subject to the constraint of Equation (II.7). This problem is solved with the help of Lagrange multipliers and requires the solution of a set of nonlinear equations. Many approximations to this solution have been proposed. The algorithm suggested by Wintz and Kurtenbach [31] is as follows.

1. Compute the bit allocation from

$$b_i = \frac{B}{N}+2\log_{10}\sigma_i^2-\frac{2}{N}\sum_{j=1}^{N}\log_{10}\sigma_i^2, \qquad \text{(II.9)}$$

   where σ_i^2 is the variance of the i^{th} sample.

2. Round off each bit b_i to its nearest integer value.

3. Modify the resultant bit allocation until Equation (II.7) is satisfied.

The derivation leading to the above algorithm is based on an exponential approximation for the error expression of Equation (II.3) between the mean-square quantization error of the i^{th} sample and its bit allocation b_i. Another similar approximation suggested by Huang and Schultheiss [32] is

$$b_i = \frac{B}{N}+\frac{1}{2}\log_2\sigma_i^2-\frac{1}{2N}\sum_{j=1}^{N}\log_2\sigma_i^2. \qquad (II.10)$$

This approximation is obtained without the constraint of Equation (II.7).

Direct application of quantization techniques to signals is undesirable because the correlation among samples of signals are large. A method to reduce these redundancies is transform coding which will be discussed in the next section.

II.2 Transform Image Coding

Transform image coding represents a "radical departure" from the classical form of image coding such as PCM, predictive, and interpolative coding in which the image signal is directly coded [1]-[10]. In transform image coding, a unitary mathematical transform is performed on the image data to produce a set of transform coefficients, which are then quantized and coded for transmission or storage [7]-[10]. Transform coding has proven to be an effective and practical coding method for monochrome, color, and multispectral images for both still images and real-time television.

Transform coding is used to achieve maximum reduction in the quantity of data to be transmitted, subject to the constraint that a reasonable amount of fidelity be preserved. In transform image coding, an N×N image is first subdivided into $(N/n)^2$ subimages with size n×n. A unitary mathematical transform is then performed on these subimages. The probability density function for each transform coefficient are estimated from the transform coefficients of these $(N/n)^2$ subimages. The resulting transform coefficients are then quantized and coded for transmission or storage. The performance of transform coding depends primarily on the (1) transformation, (2) quantization, (3) subimage size, and (4) subimage shape.

(1) Transformation

The "best" transformation from both objective and subjective viewpoint is the K-L transform [1]-[10]. This transform can produce uncorrelated transform coefficients which is suitable for quantization because all but the "nonlinear" redundancies among transform coefficients are removed. However, there is no efficient algorithm to compute K-L transform. Many other unitary mathematical transforms have been proposed and used for transform coding. These transforms are cosine, sine, Fourier, Hadamard, Harr, Slant, and Hartly. The coding performance of these transforms approaches that of K-L as the subimage size becomes large. Among them, cosine transform coding provides the best approximation to K-L transform coding when the image model is a first order Markov process [7]-[10].

(2) Quantization strategy

Both mean-square quantization error and subjective quality are sensitive to the number of bits used in the quantization of transform coefficients. The simplest strategy is to form normalized coefficients

$$y_{i,j} = \frac{y_{i,j}}{\sigma_{i,j}}, \tag{II.11}$$

and use the same quantizer for each coefficient. The probability density function for each transform coefficient is usually assumed to be Gaussian. Since there are $(N/n)^2$ subimages, the mean and variance of Gaussian density for each transform coefficient can be estimated. If η coefficients are retained and m bits are used to code each coefficient, then a total of $m\eta/n^2$ bits/pixel are required. For good quality reconstructions, about half the coefficients should be retained, in which case 7 bits must be used for each retained coefficient [8]. Therefore, $m\eta/n^2 \approx 3.5$ bits/pixel are required. This method is called *zonal sampling quantization* [10].

The best quantization strategy is the block quantization. In *block quantization*, a different quantizer is used for each transform coefficient. Each quantizer has different number of quantization bins and different spacing between bins. A total of B bits is used to code the coefficients. The bit allocation for each coefficient is usually set according to Equation (II.9) or (II.10), or a bit allocation map. Run-length coding [7]-[10] is required to determine the bit allocation when it is set according to some equations. This will increase the amount of overhead information required during transmission and storage. This problem can be solved by using a bit allocation map. However, this map can not be adapted to a changing image. Figure II.1 show some commonly used bit allocation maps [5], [7], [10]. The quality of reconstructed images by block quantization is about the same as those by the other quantizations, but usually only 1 bit/pixel is required [8].

(3) Subimage size

Mean-square quantization error decreases with increasing subimage size. However, most images contain significant correlations between pixels for only about 20 adjacent pixels, although this number is strongly dependent on the amount of detail in the image. Since the purpose of transformation is to produce the uncorrelated transform coefficients, 16×16 is a reasonable choice for subimage size [8]. However, the 8×8 subimage size does not significantly increase the quantization error. This argument can not be applied when subjective quality is the criterion of goodness. The subjective quality appears to be independent of subimage size when the subimage size is greater than 4×4 [8]. The subimage size is usually confined to the power of two to increase the computational efficiency of transformation.

There are fewer samples available in estimating the probability density function for each transform coefficient when the subimage size is large or the number of subimages is small. This will increase the estimation error in the mean and variance. For

```
8 8 8 7 7 7 5 5 4 4 4 4 4 4 4 4
8 8 7 6 5 5 3 3 3 3 3 2 2 2 2 2
8 7 6 4 4 4 3 3 2 2 2 2 2 2 2 2
7 6 4 3 2 2 2 2 1 1 1 1 0 0 0 0
7 5 4 2 2 2 2 1 1 1 0 0 0 0 0 0
7 5 4 2 2 2 1 1 0 0 0 0 0 0 0 0
5 3 3 2 2 1 1 0 0 0 0 0 0 0 0 0
5 3 3 2 1 1 0 0 0 0 0 0 0 0 0 0
4 3 2 1 1 0 0 0 0 0 0 0 0 0 0 0
4 3 2 1 1 0 0 0 0 0 0 0 0 0 0 0
4 3 2 1 0 0 0 0 0 0 0 0 0 0 0 0
4 3 2 1 0 0 0 0 0 0 0 0 0 0 0 0
4 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0
4 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0
4 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0
4 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0
```

(a)

```
7 6 5 4 3 3 2 2 2 1 1 1 1 1 0 0
6 5 4 4 3 3 2 2 1 1 1 1 1 0 0 0
5 4 4 3 3 2 2 2 1 1 1 1 1 0 0 0
4 4 3 3 3 2 2 2 1 1 1 1 1 0 0 0
3 3 3 3 2 2 2 1 1 1 1 1 0 0 0 0
3 3 2 2 2 2 2 1 1 1 1 1 0 0 0 0
2 2 2 2 2 2 1 1 1 1 1 0 0 0 0 0
2 2 2 2 1 1 1 1 1 1 1 0 0 0 0 0
2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

(b)

```
7 5 4 3 3 3 2 2 2 2 1 0 0 0 0 0
5 4 3 3 2 2 1 1 1 1 0 0 0 0 0 0
4 3 2 2 2 1 1 1 1 0 0 0 0 0 0 0
3 3 2 2 2 1 1 1 0 0 0 0 0 0 0 0
3 2 2 2 1 1 1 0 0 0 0 0 0 0 0 0
3 2 1 1 1 1 0 0 0 0 0 0 0 0 0 0
2 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
2 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

(c)

Figure II.1 Bit allocation maps.
(a) 1.5 bits/pixel.
(a) 1.0 bits/pixel.
(a) 0.5 bits/pixel.

example, there are 256 samples for each transform coefficient when the image size is 256×256 and the subimage size is 16×16. The number of samples reduces to only 64 when the subimage size increases to 32×32. Therefore, the subimage size can not be too large. Usually, the subimage size is limited to 16×16. Of course, this limitation is also dependent on the size of original image.

(4) Subimage shape

Transforming two dimensional n×n subimage yields better performance than one dimensional $n^2 \times 1$ image [8]. For example, if the two dimensional subimage is 4×4, then an one dimensional subimage with size 16×1 is required to have comparable results. However, the gain in using the two dimensional subimage is only about 0.1-0.2 bit/pixel [8].

II.3 One-dimensional Pseudo-noises

The purpose of this background section is to discuss the codes used in communications and ranging systems–those that act as noise-like (but deterministic) carriers for information transmission. These code sequences are of much greater length than those commonly used for information transfer. This is because they are intended for bandwidth spreading and not for the direct transfer of information. Two of these code sequences will be discussed in here. They are the maximal length sequences (m-sequences) and Gold code sequences.

An *m-sequence*, by definition, is the longest sequence that can be generated by a given shift register or a delay element of a given length [15]-[19]. For a binary shift register sequence generator, the corresponding m-sequence has 2^m-1 bits, where m is the number of stages in the shift register sequence generator. Such a sequence generator consists of a shift register with an appropriate logic circuit. This logic circuit will feedback a logical combination of the state of its stages (two or more) to its input. The output of a sequence generator is a function of the outputs of the stages at the proceeding sample time. Table II.1 shows some possible feedback taps for different m-sequences, and Figure II.2 shows two typical 63-bit m-sequence generators.

Briefly stated, properties of the m-sequences are:

Property II.1: For any m-sequence, the number of ones is 2^{m-1} and the number of zeros is $2^{m-1}-1$, where m is the number of stages in the sequence generator.

Property II.2: The statistical distribution of ones and zeros is well defined. Relative positions of their runs change from sequence to sequence but the number of each run length does not.

Property II.3: The cyclic auto-correlation of any m-sequence is such that the correlation for all the cyclic phase shifts is -1 except the zero phase shift which is 2^m-1.

Property II.4: A modulo-2 addition of any m-sequence with a cyclic phase shifted replica of itself results in another cyclic phase shifted replica of itself which differs from either of the originals.

Property II.5: Every possible state, or m-tuple, of a given m-stage generator exists for one and only one clock pulse during the generation of a complete code cycle. The exception is that the all-zeros state does not occur and can not be allowed to occur.

Remark: Let $[a] \equiv [a_0, a_1, ..., a_{n_a-1}]$ be an m-sequence with length n_a, then $a_j = a_{j \bmod n_a}$ because of the cyclic property of m-sequences.

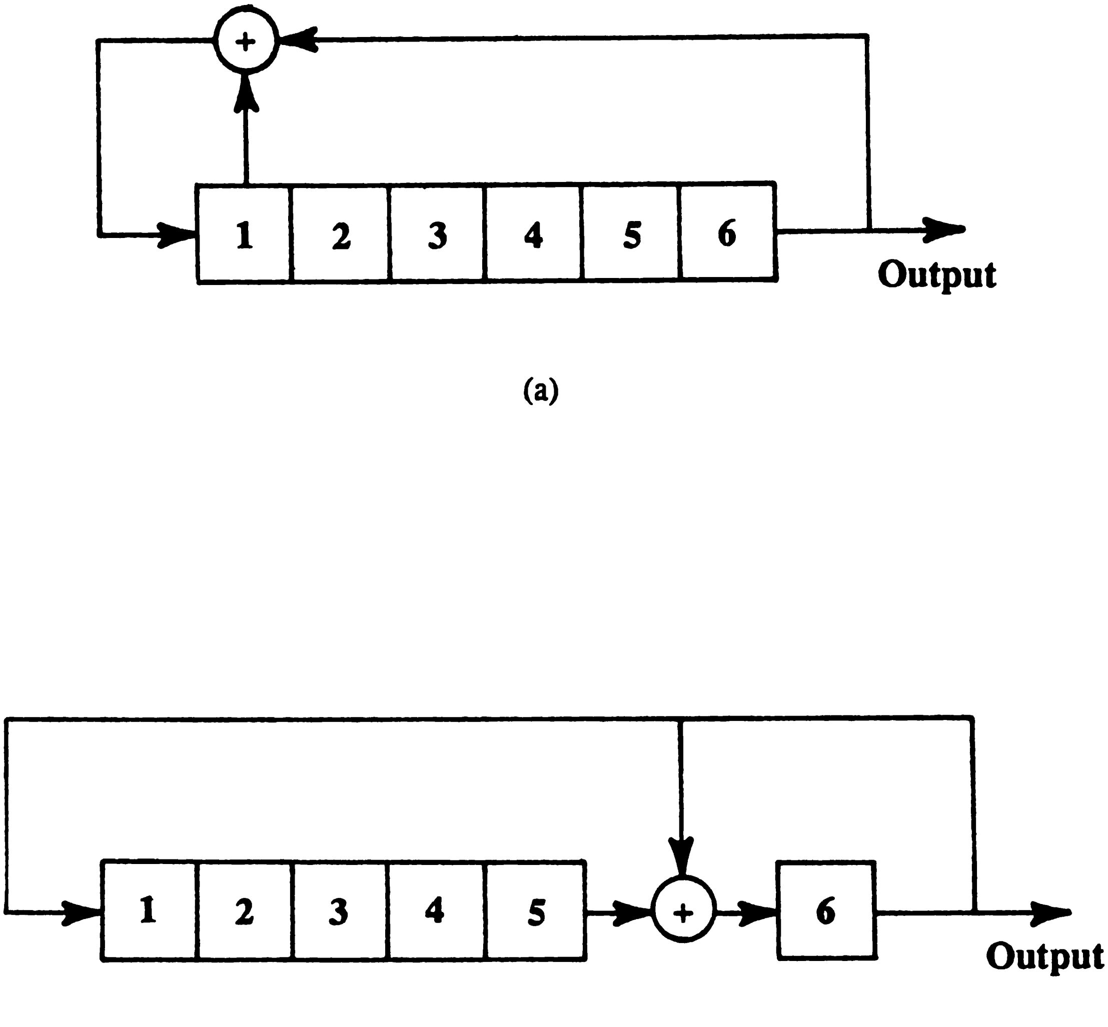
Let [a] and [a'] be any two m-sequences with the same length n_a. The following is an alternative definition for the cyclic correlation between these two sequences [16], [34].

Definition: Let "$\oplus$" be bit by bit modulo-2 addition and $[a]_i \equiv [a_i, a_{i+1}, ..., a_{n_a+i-1}]$. Then the cyclic correlation $\theta_{aa'}(i)$ between two m-sequences [a] and $[a']_i$ is the difference between the number of zeros and ones in $[a]\oplus[a']_i$.

Remark: The number of zeros in $[a]\oplus[a']_i$ is $[n_a+\theta_{aa'}(i)]/2$, while the number of ones is $[n_a-\theta_{aa'}(i)]/2$.

Table II.1 Feedback taps for m-sequence generator.

# of Stages	Code Size	Maximal Taps
2	3	[2,1]
3	7	[3,1]
4	15	[4,1]
5	31	[5,2] [5,4,3,2] [5,4,2,1]
6	63	[6,1] [6,5,3,2] [6,5,2,1]
7	127	[7,3] [7,1] [7,6,5,2] [7,6,4,2] [7,6,3,1] [7,4,3,2] [7,3,2,1] [7,6,5,4,2,1] [7,5,4,3,2,1]
8	255	[8,7,6,1] [8,6,5,3] [8,6,5,2] [8,6,5,1] [8,5,3,1] [8,4,3,2] [8,7,6,5,2,1] [8,6,4,3,2,1]
9	511	[9,4] [9,8,7,2] [9,8,6,5] [9,8,5,4] [9,8,4,1] [9,6,4,3] [9,5,3,2] [9,8,7,6,5,3] [9,7,6,4,3,1] [9,6,5,4,2,1]
10	1023	[10,3] [10,9,4,2] [10,9,4,1] [10,8,5,4] [10,8,5,1] [10,8,4,3] [10,8,3,2] [10,5,3,2] [10,5,2,1] [10,4,3,1]
11	2047	[11,1] [11,10,3,2] [11,9,8,3] [11,9,4,1] [11,8,6,2] [11,8,5,2] [11,7,3,2] [11,6,5,1] [11,5,3,1]
12	4095	[14,13,11,9] [14,13,4,2] [14,12,11,1] [14,12,2,1] [14,10,6,1] [14,6,4,2] [14,13,12,8,4,1] [14,13,12,7,6,3] [14,13,11,10,8,3] [14,13,6,5,3,1] [14,11,9,6,5,2] [14,10,6,5,4,1] [14,8,7,6,4,2]
13	8191	[13,4,3,1] [13,12,11,9,5,3] [13,12,11,5,2,1] [13,12,9,8,4,2] [13,12,8,7,6,5] [13,12,6,5,4,3] [13,11,8,7,4,1] [13,10,9,7,5,4] [13,9,8,7,5,1] [13,8,7,4,3,2]



(a)

(b)

Figure II.2 Two typical 63-bit m-sequence generators.
(a) Simple register configuration $[6,1]_s$.
(b) Equivalent modular configuration $[6,5]_m$.

Example II.1: Let [a] = [1110100]. Then n_a = 7, $[a]_1$ = [0111010], $[a]+[a]_1$ = [1001110], and $\theta_{aa}(1) = -1$. The number of zeros in $[a]+[a]_1$ is three ([7+(-1)]/2), while the number of ones is four ([7-(-1)]/2). Figure II.3 shows the cyclic correlations of two 63-bit m-sequences. These two m-sequences are generated by the feedback taps [6,1] and [6,5,2,1], respectively. The cyclic auto-correlation peak is 63, while the cyclic auto-correlation floor is always -1. The *cyclic auto-correlation floor* is the cyclic auto-correlation except the maximal peak. The cyclic cross-correlation values are 15, -17, and -1.

A different pair of m-sequences will have different bound (β) on the cyclic cross-correlation. As the size of m-sequences approaches infinity, the cyclic auto-correlation approaches a delta-function. In addition, the cyclic cross-correlation can be neglected when it is compared with the cyclic auto-correlation peak. Mathematically, let a(x) and a′(x) be two (2^m-1)-bit m-sequences, then

$$\begin{cases} \theta_{aa}(0) = \int_{b_L}^{b_U} a(x)a(x)dx = 2^m-1 \\ \theta_{aa'}(0) = \int_{b_L}^{b_U} a(x)a'(x)dx < \beta, \end{cases} \qquad \text{(II.12)}$$

where $\beta \ll 2^m-1$ as $m \rightarrow \infty$ and b_U-b_L equals the length of 2^m-1 bits. Two functions a(x) and a′(x) are defined to be *orthogonal* if

$$[a(x),a'(x)] = \int_{b_L}^{b_U} a(x)a'(x)dx = \begin{cases} \int_{b_L}^{b_U} a^2(x)dx & \text{for } a(x) = a'(x) \\ 0 & \text{for } a(x) \neq a'(x). \end{cases}$$

Therefore, these m-sequences have a "quasi-orthogonal property" according to the Equations (II.12) and (II.13).

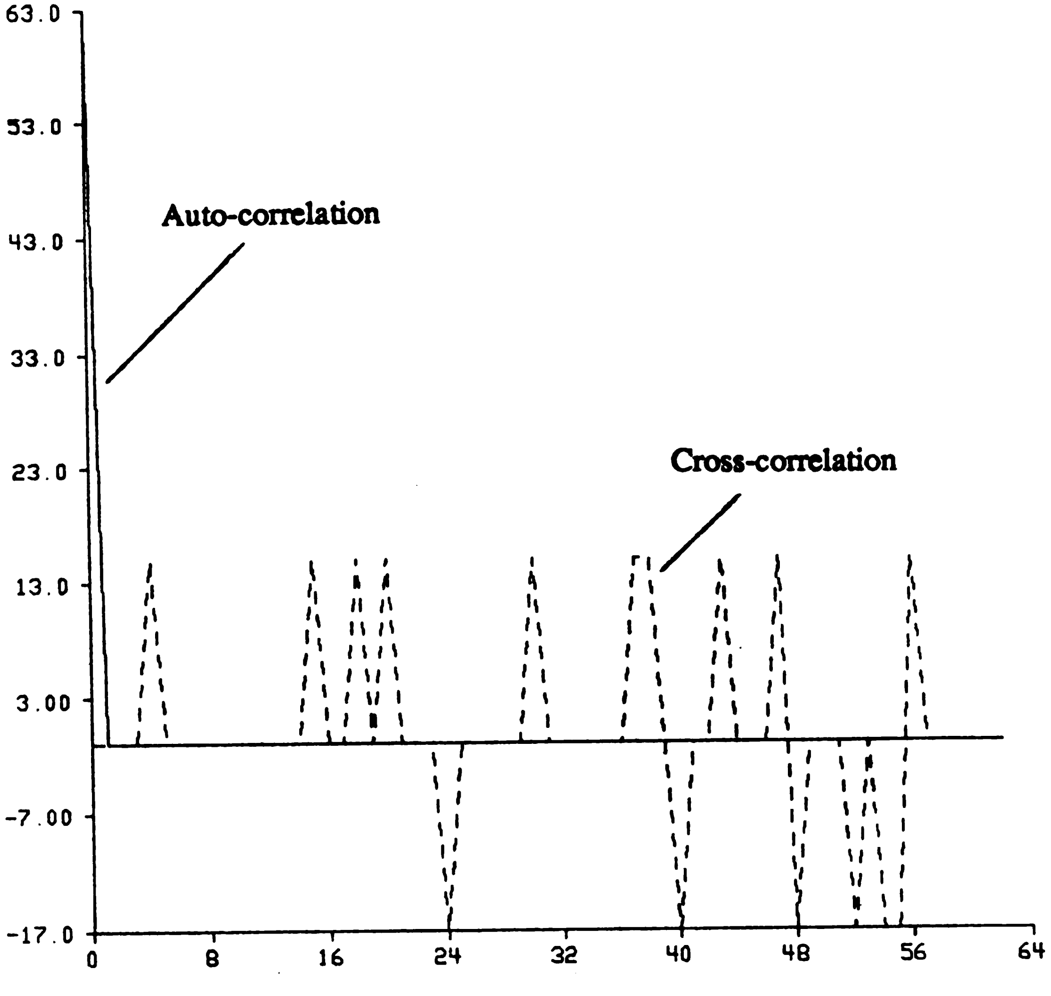


Figure II.3 Cyclic correlations of two 63-bit m-sequences.

A *Gold code sequence* is generated by modulo-2 addition of a pair (base pair) of m-sequences [15]-[19]. This pair of base m-sequences are added bit by bit by synchronous clocking. The two base m-sequences must have the same length and the resulting Gold code sequence has the same length as its base m-sequences. Figure II.4 shows a Gold code sequence generator. Every change in phase position between the two base m-sequences generates a new sequence. Therefore, a pair of m-sequences with length n can generate a family of n Gold code sequences. Every sequence in this family has the same cyclic auto-correlation peak as its base m-sequences. The bound on the cyclic cross-correlation between any two sequences in this family is also the same as that between the two base m-sequences. The drawback is the cyclic auto-correlation floor of any sequence in this set is not constant. However, the bound on the cyclic auto-correlation floor is also the same as the bound on the cyclic cross-correlation between the two base m-sequences.

The advantage of Gold code sequences is that they can provide a family of sequences in which any two sequences have the same bound on the cyclic cross-correlation and the cyclic auto-correlation floors. In other words, Gold code sequences offer families of sequences with the quasi-orthogonal property. Figure II.5 shows the cyclic correlations of two 63-bit Gold code sequences. These Gold sequences are generated by the m-sequences discussed in Example II.1.

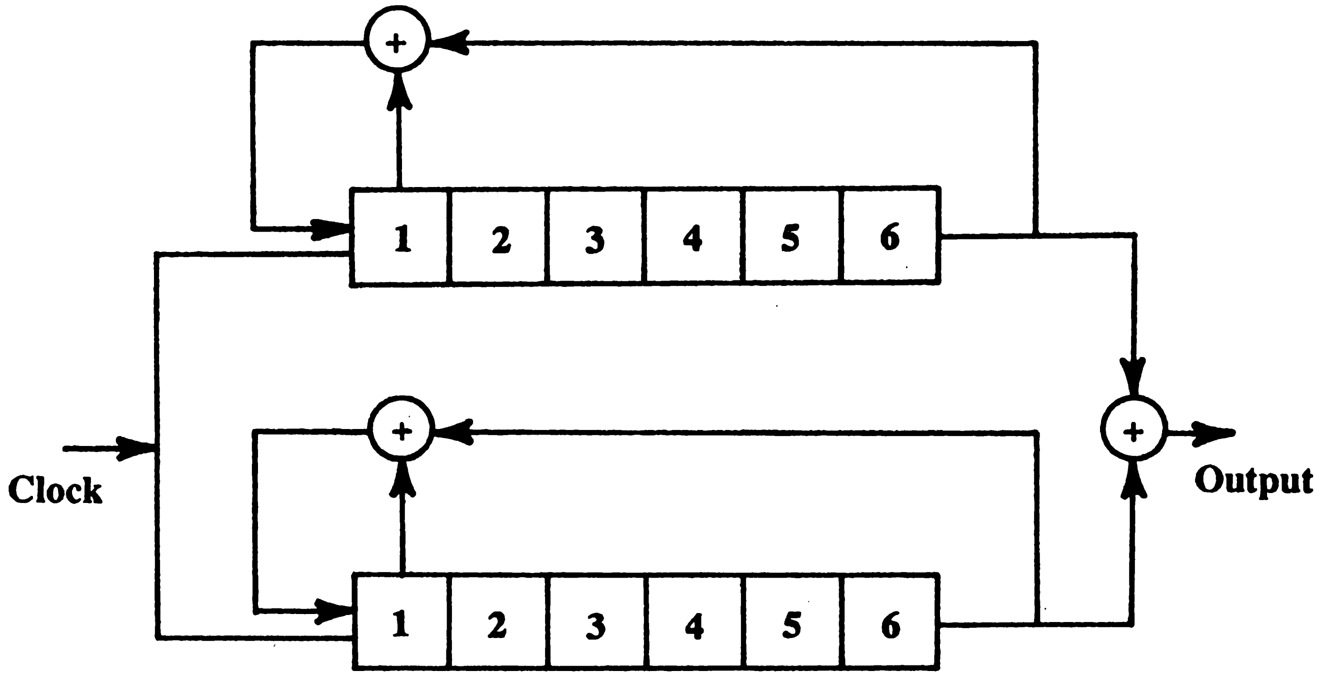


Figure II.4 A typical 63-bit Gold code sequence generator.

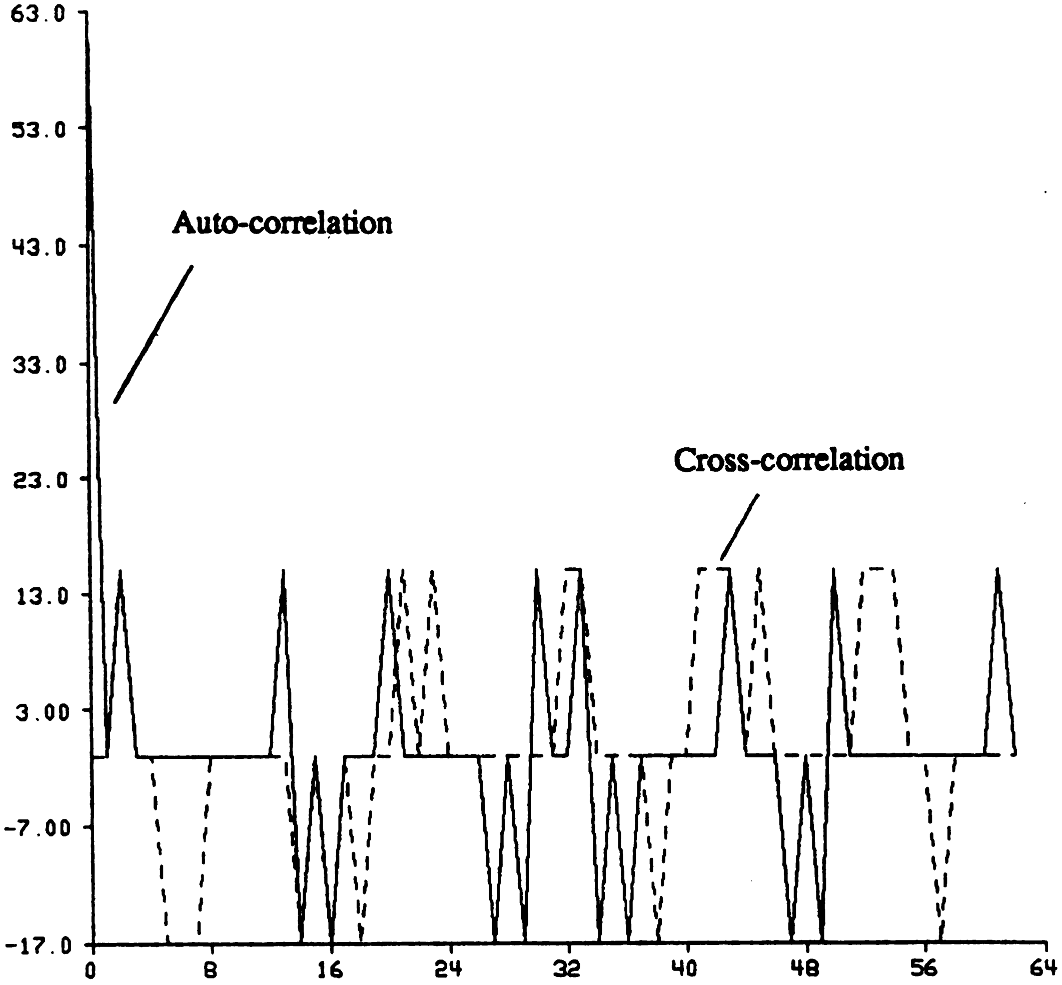


Figure II.5 Cyclic correlations of two 63-bit Gold code sequences.

CHAPTER III

TRANSFORM ENCRYPTION CODING

III.1 Independence Technique

It is necessary to convert continuous signals into digital ones for digital information transmission or storage. This conversion is done by the quantization of continuous signals. The optimal quantization for signals is usually defined as the one that minimizes the expected mean-square quantization error using a fixed number of quantization levels or regions. Although the mean-square quantization error can be minimized over a data set empirically, it is time-consuming in doing so. The quantized data must be converted into digital symbols for digital information transmission and storage. However, direct digitization of these quantized data is inefficient. To have a high efficiency, a large amount of information is required for the mapping between the quantized data and the corresponding digital symbols. An efficient way is to approximate the data set by a probability density function and minimize the mean-square quantization error according to this approximated function. In this case, overhead information is greatly reduced because only the mean and variance are required. The optimal quantization for a single random variable is the Max quantization [28] discussed in Section II.2. For a sequence of dependent random variables, vector quantization is optimal. In vector quantization, the mean-square quantization error is minimized by jointly quantizing and reconstructing the whole sequence of dependent random variables according to their joint probability density function [1]-[10], [13]-[15]. The following theorem shows the condition under which the Max quantizer minimizes the mean-square quantization error for a sequence of random variables.

Theorem III.1: The Max quantizer minimizes the sum of mean-square quantization error for a sequence for random variables z_1, z_2, ..., z_N if and only if z_1, z_2, ..., z_N are independent random variables.

Proof:

(Necessity) Let us assume z_1, z_2, ..., z_N are dependent random variables. Therefore, the joint probability density function $f_{z_1, z_2, ..., z_N}(\zeta_1, \zeta_2, ..., \zeta_N)$ is nonseparable. In other words, the probability density function of any z_i is a function of z_1, z_2, ..., z_N. Hence the quantization levels of any z_i must be a function of z_1, z_2, ..., z_N to minimize the mean-square quantization error. However, the Max (scalar) quantizer chooses the quantization levels of any z_i according to z_i only. Hence the Max quantizer does not minimize the mean-square quantization error for any z_i. In addition, the Max quantizer does not minimize the sum of mean-square quantization error for z_1, z_2, ..., z_N. Therefore, z_1, z_2, ..., z_N are independent random variables when the Max quantizer minimizes the sum of mean-square quantization errors for z_1, z_2, ..., z_N.

(Sufficiency) Since z_1, z_2, ..., z_N are independent random variables, the probability density function of any z_i is a function of z_i only. In addition, the mean-square quantization error of any z_i is minimized by the Max quantization. Therefore, the sum of mean-square quantization errors for z_1, z_2, ..., z_N is also minimized by the Max quantizer.

Q.E.D.

Both scalar and vector quantizations are used to effectively quantize the signal according to its probability density function. However, it is much more difficult to implement vector quantization than scalar quantization. Problems with vector quantization are the following. First, the joint probability density function of random variables is not easily estimated. Secondly, the quantization regions are difficult to determine. Thirdly, searching for the nearest matching vector template to any input vector is time-consuming. Usually, only a partial search is carried out which makes vector quantization sub-optimal. Finally, large codebook storage and overhead information are necessary. Therefore, Max (scalar) quantization is still a popular technique for quantizing dependent random variables although it is not optimum. The proposed independence technique generates independent random variables from the dependent ones. In this case, Max quantization can be applied to the resulting independent random variables and the sum of mean-square quantization errors minimized in the process. Before the independence theorem is stated, the following definitions and the extended central limit theorem are needed.

Definition: x_1, x_2, ..., x_N are m-dependent random variables if $(x_1, x_2, ..., x_r)$ is independent of $(x_s, x_{s+1}, ..., x_N)$ for all integers $1 \leq r \leq s \leq N$ and $s-r \equiv m \geq 0$ [35]-[39].

Remarks: The samples of signals are often independent random variables as long as the samples are widely separated. Therefore, these samples can be considered to be m-dependent random variables. In addition, 0-dependent means independent. This definition can be extended to the n-dimensional case in which x_1, x_2, ..., x_N are n-dimensional vectors.

Theorem III.2 (Extended Central Limit Theorem): Let $x_1, x_2, ..., x_N$ be a sequence of m-dependent random variables with

$$E[x_i] = 0, \tag{III.1a}$$

$$E[x_i^2] = 1, \tag{III.1b}$$

$$E[x_i^8] < \infty, \tag{III.1c}$$

and

$$0 < C = \lim_{N\to\infty} \frac{1}{N} \mathrm{Var}[\sum_{i=1}^{N} x_i] < \infty. \qquad \text{(III.1d)}$$

Then there exist a constant A such that for all integer N and all real a

$$\left| P\left[\sum_{i=1}^{N} x_i/\sqrt{NC} \le a\right] - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{a} e^{\frac{-x^2}{2}} dx \right| \le \frac{A}{\sqrt{N}}, \qquad \text{(III.2)}$$

where A depends on the distribution of $x_1, x_2, ..., x_N$, but not on N.

Proof: See [35]-[39].

Remarks: The error in the Gaussian approximation is $O(1/\sqrt{N})$ by the theorem above [35]. In addition, central limit theorem is also true for two-dimensional m-dependent random vectors [36].

Definition: The *encrypted* random variables $y_1, y_2, ..., y_N$ are defined as the weighted sums of the random variables $x_1, x_2, ..., x_N$,

$$y_i = \sum_{j=1}^{N} a_{ij} x_{j,} \qquad \text{(III.3)}$$

where $a_{ij} \in \mathbf{R}$ for i, j = 1, 2, ..., N.

Definition: $y_1, y_2, ..., y_N$, are joint Gaussian random variables if and only if the sum $b_1y_1 + b_2y_2 + ... + b_Ny_N$ is a Gaussian random variable for any real $b_1, b_2, ..., b_N$ [40], [41].

Theorem III.3 (Independence Theorem): Let $x_1, x_2, ..., x_N$ be a sequence of m-dependent random variables, $y_1, y_2, ..., y_N$ be the corresponding encrypted random variables, and $z_1, z_2, ..., z_M$ be the K-L transform coefficients of any M encrypted random variables chosen from $y_1, y_2, ..., y_N$, where $M \le N$. Then $z_1, z_2, ..., z_M$ are independent Gaussian random variables as $N \to \infty$.

Proof:

By the definition of encryption, we have $y_i = \sum_{j=1}^{N} a_{ij}x_j$, where $a_{ij} \varepsilon \mathbf{R}$ for i, j = 1, 2, ..., N. Let $\overline{y}_1, \overline{y}_2, ..., \overline{y}_M$ be any M encrypted random variables chosen from $y_1, y_2, ..., y_N$, where $M \leq N$. Then we also have $\overline{y}_i = \sum_{j=1}^{N} \overline{a}_{ij}x_j$, where $\overline{a}_{ij} \varepsilon \mathbf{R}$ for i = 1, 2, ..., M and j = 1, 2, ..., N. For any real $b_1, b_2, ..., b_M$,

$$\begin{aligned}\overline{y} &\equiv b_1\overline{y}_1 + b_2\overline{y}_2 + \cdots + b_M\overline{y}_M \\ &= b_1 \sum_{j=1}^{N} \overline{a}_{1j}x_j + b_2 \sum_{j=1}^{N} \overline{a}_{2j}x_j + \cdots + b_M \sum_{j=1}^{N} \overline{a}_{Mj}x_j \\ &= \sum_{j=1}^{N} (b_1\overline{a}_{1j} + b_2\overline{a}_{2j} + \cdots + b_M\overline{a}_{Mj})x_j.\end{aligned}$$

Therefore, $\overline{y}$ is a Gaussian random variable as $N \rightarrow \infty$ by the extended central limit theorem. Consequently, $\overline{y}_1, \overline{y}_2, ..., \overline{y}_M$ are joint Gaussian random variables as $N \rightarrow \infty$ by the definition above. According to the definition of K-L transform, we have $z_i = \sum_{j=1}^{M} c_{ij}\overline{y}_j$, where $c_{ij} \varepsilon \mathbf{R}$ for i, j = 1, 2, ..., M. Therefore, $z_1, z_2, ..., z_M$ are also joint Gaussian random variables as $N \rightarrow \infty$. In addition, $z_1, z_2, ..., z_M$ are uncorrelated because of K-L transform. Therefore, $z_1, z_2, ..., z_M$ are independent (Gaussian) random variables as $N \rightarrow \infty$. **Q.E.D.**

Remarks: The K-L transform coefficients of any encrypted signal are asymptotically independent Gaussian random variables as the size of the m-dependent signal approach infinity. In addition, the probability density function of random variable $x_i, x_2, ..., x_N$ is not important in obtaining the independence results because of the extended central limit theorem. Therefore, simple Max quantizers can minimize the sum of mean-square quantization errors. Without this theorem, vector quantization must be used to minimize the quantization error. Because Max quantization is much simpler than vector quantization and most techniques of vector quantization are sub-optimal, the importance of independence theorem is clear.

Actually, the K-L transform coefficients of any m-dependent signal are also asymptotically independent Gaussian random variables as the signal size approach infinity according to the proof above. Therefore, it seems unnecessary to go through the encryption to obtain the independent Gaussian transform coefficients. The reason for doing so is as follows. In transform coding, the probability density function of each transform coefficient must be estimated first before the quantization. Therefore, the signals are usually partioned into blocks with the same size and then the transformation is performed on these blocks. If the block size is too large, then the estimation errors in the mean and variance will be large because fewer samples are available in the estimation. In addition, the number of calculations required by the transformation is also increased. For example, a 16-point FFT requires 64 multiplications, while only 32 multiplications are needed by four 4-point FFTs [7]-[10]. Therefore, the smaller the block size, the faster the transformation. However, the K-L transform coefficients are independent only when the block size is infinity. Therefore, there is a tradeoff in choosing the size of the blocks. With this independence theorem, this tradeoff is avoided. Asymptotically, the *encrypted* signals are joint Gaussian. As long as the joint Gaussian property is induced by encryption , then the K-L transform coefficients will be independent. Therefore, the block size can be small, and the K-L transform coefficients of these blocks are still independent random variables. In addition the estimation error in the mean and variance of transform coefficients can also be minimized because the block size can be small.

It has been observed by many researchers that the sum of mean-square Max quantization errors for any unitary mathematical transform decreases as the subimage size approach infinity [1]-[10]. However, no explanation has ever been given. With the proof of the independence theorem, an explanation is as follows. The K-L transform coefficients are asymptotically independent random variables as the subimage size increases. This is because the K-L transform coefficients are the weighted sums of the original signal samples. Therefore, they are asymptotically joint Gaussian random variables as the subimage size approaches infinity. In addition, they are also uncorrelated random variables because of the property of K-L transform. Hence they are asymptotically independent random variables. In addition, the performance of any unitary mathematical transform also approaches that of K-L as the block size approach infinity. Therefore, the unitary mathematical transform coefficients are asymptotically independent random variables as the subimage size increases. In addition, the sum of mean-square quantization errors will decrease when the transform coefficients are independent.

With this independence theorem and the Wintz and Kurtenbach algorithm, the best way to quantize the m-dependent signals, according to the material presented thus far, is the following.

1. Obtain the encrypted signal which is weighted sums of the original m-dependent signal.
2. Partion the encrypted signal into blocks, and then apply the K-L transform to each block.
3. Estimate the mean and variance for each transform coefficient. These transform coefficients are Gaussian distributed.
4. Apply the Wintz and Kurtenbach algorithm to obtain the bit allocation for each transform coefficient.
5. Apply Max quantization to each transform coefficient with the number of quantization levels obtained in step 4.

This algorithm is optimal for the following reasons. First, the K-L transform coefficients of encrypted m-dependent signals are asymptotically independent Gaussian random variables. Secondly, the Wintz and Kurtenbach algorithm obtains the best bit allocation for independent Gaussian random variables under an exponential approximation for the mean-square quantization error. Finally, the Max quantizer minimizes the sum of mean-square quantization errors for independent random variables. The bit allocation maps discussed in Figure II.2 can also be used. However, this will produce sub-optimal results.

For the application of independence theorem, the encrypted signal must be obtained first. The techniques of transform coding can then be applied to these encrypted signal for quantization. Therefore, this technique can be referred as *transform encryption coding*. A computational algorithm for signal encryption is presented in the next section.

III.2 Computational Algorithm for Signal Encryption

By the definition of encryption, Equation (III.3), we have

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_N \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} \tag{III.4}$$

or Y = AX. Apparently, the matrix A must have full rank in order for it to be possible to decrypt the encrypted signal Y (recover X).

It is well known that an $\overline{N}\times\overline{N}$ image can be arranged into an $\overline{N}^2\times1$ vector. Similarly, any n-dimensional signal can also be arranged into a vector. In addition, the samples of most signals have the m-dependent property. Therefore, the independence theorem can be directly applied to any n-dimensional signal for any n. Strictly speaking, the result of the independence theorem holds only when the signal size approaches infinity. When the image size is 128×128, the error of Gaussian approximation defined by Equation (III.2) is about 0.008, while this error reduces to about 0.004 when the image size is 256×256. Therefore, the larger the signal size, the better the Gaussian approximation and the independence results. However, it is difficult to calculate the signal encryption when signal size is too large. For example, if the image size is 256×256, then X is a 65536×1 vector. The direct computation of $Y_{65536\times1} = A_{65536\times65536}X_{65536\times1}$ is time-consuming because it requires 4,294,967,296 multiplications. In addition, to search a 65536×65536 matrix with full rank is also difficult. Therefore, the following algorithm is proposed as an easy way to achieve the signal encryption.

Algorithm: The encrypted signal Y can be obtained by cyclic scrambling the phase spectrum of an m-dependent signal X according to the *phase* spectrum of a *reference function* $\overline{A}$.

Proof:

Let $\overline{A} = [\overline{a}_1, \overline{a}_2, ..., \overline{a}_N]$, and $F[\overline{A}] \equiv |F[\overline{A}]| e^{j\phi_{\overline{A}}}$, where F is the Fourier transform. Then $F[Y] = e^{j\phi_{\overline{A}}}F[X]$ by the above algorithm. If $\overline{A}' \equiv F^{-1}[e^{j\phi_{\overline{A}}}] \equiv [\overline{a}'_1, \overline{a}'_2, ..., \overline{a}'_N]$, then we have $Y = \overline{A}' {*} X$, where "*" stands for the cyclic convolution. In other words,

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_N \end{bmatrix} = \begin{bmatrix} \overline{a}'_1 & \overline{a}'_2 & \cdots & \overline{a}'_N \\ \overline{a}'_N & \overline{a}'_1 & \cdots & \overline{a}'_{N-1} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \overline{a}'_2 & \overline{a}'_3 & \cdots & \overline{a}'_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} \qquad (III.5)$$

which has the same form as Equation (III.4). Therefore, the results of this algorithm satisfy the definition of signal encryption. **Q.E.D.**

Remarks: It is possible that the phase spectrum of a reference function is so regular that the encrypted signal can be recognized without any effort. Therefore, a specific class of reference functions with irregular phase spectra must be used to ensure the encryption. This type of function will be discussed in the next chapter. Although the phase of the complex number (0,0) is undefined, it can be assigned any value between 0 and 2π. Here, the phase of (0,0) is defined as zero. With this assignment, the phase spectrum of any reference function is well defined. Therefore, the decryption can be uniquely obtained by $X = F^{-1}[F[Y]e^{-j\phi_{\bar{A}}}]$. With this algorithm and the above assignment, the full rank requirement on A is relaxed. In addition, the encryption and decryption processes become simple because of the FFT and its cyclic property.

In other words, this encryption process amounts to the passing of the m-dependent periodic signal through an all-pass periodic phase filter. If the phase spectrum of the m-dependent signal is appropriately chosen, then the filtered output (encrypted signal) is uncognizable [42]. Actually, any cyclic filter can produce the encrypted signal although it maybe recognizable. For example, if a bandpass filter is used, the output is still in the form of encryption. A problem with this type of filter is as follows. Since the transform coding will operate on these encrypted signals, some quantization errors must occur. These quantization errors can be assumed to be unform all over the frequences. In the decryption, an inverse filter must be used. The quantization error will then be amplified or attenuated at most frequencies. This will create some noticeable error at specific frequencies. Therefore, an all-pass phase filter must be chosen to ensure this type of error does not appear in the reconstructed signal. In some ways, this signal encryption is also similar to phase modulation because the phase spectrum of the m-dependent signal is modulated by that of the reference function [43].

The transform encrypted image coding is also similar to the transform image coding using a spatial mask [44]-[46]. In the techniques of transform coding using a spatial mask, the image is passed through a spatial mask before transform coding. The amplitude and phase spectrum of this spatial mask is specially designed to match the human visual model. Usually, the size of this spatial mask is chosen from 3×3 to 9×9. In the transform encrypted image coding, the reference image (spatial mask) must be cyclic and as large as possible. In addition, the amplitude spectrum of reference image (spatial mask) must be constant to avoid the grid error pattern. These two techniques, of course, can be combined together to produce better results.

If the reference function is complex, then the encrypted signal is also complex. This will create the redundancies. Therefore, the reference function should be real, in which case the phase spectrum of the reference function will be an odd function. Instead of generating a real reference function and computing its phase spectrum, an odd function can be used to replace the phase spectrum of the reference function. This replacement can reduce the computational burden of the encryption and decryption processes because it is not necessary to calculate the phase spectrum of the reference function in this case. As can be seen in the proof of encryption algorithm, the key point is to provide a uniquely defined phase spectrum for the encryption process. Therefore, the binary phase spectrum of a reference function can also be used. The binary phase spectrum is usually defined as the following [47].

$$F[\overline{A}'] \equiv \begin{cases} 1 & \text{for } \pi/2 \leq \phi_{\overline{A}} \leq 3\pi/2 \\ -1 & \text{elsewhere.} \end{cases} \qquad \text{(III.6)}$$

In summary, there are two ways to achieve the signal encryption. In either case, the m-dependent signal is treated as a periodic one. The first way is to use a continuous or binary phase spectrum of a reference function to modulate the phase spectrum of the m-dependent one. The other way is to use an odd function (continuous or binary) to modulate the phase spectrum of an m-dependent signal. Both methods can be easily obtained through the use of the FFT. Although at least two additional N×N FFTs are required by the encryption process for two-dimensional images, they can be calculated easily. The number of complex additions or multiplications required by the encryption process is $4N^2\log_2 N$ when N is a power of 2. Since a 1024×1024 FFT can be calculated in real-time (1/30 second) by today's CMOS technology [48], the encryption process is definitely realizable in real-time when the signal size is 512×512.

Sometimes, security is another important issue during the information transmission and storage. The proposed transform encryption coding also ensures the security because the the information is scrambled and uncognizable when the phase spectrum of the reference function is sufficiently irregular. However, if the reference function used for encryption is known to an unauthorized person (*attack*), then the security is lost. Therefore, we need to have a specific type of reference function with the following properties to avoid these cases.

1. They are easy and fast to generate.
2. They must have the pseudo-random appearance.
3. There must be many this type of function available.

One of this type of function is the pseudo-noises which will be discussed in the next chapter. Since there are so many pseudo-noises available, an attack by unauthorized persons is time-consuming or almost impossible. In addition, the phase spectra of these pseudo-noises are irregular because of their pseudo-random appearance.

CHAPTER IV

TWO-DIMENSIONAL PSEUDO-NOISES

IV.1 Two-dimensional Quasi Maximal Area Arrays

Two-dimensional pseudo-noises, maximal area arrays (m-arrays), are useful for many applications such as coded aperture imaging [22], "add-on" data transmission [23], pattern synchronization [24], and code division image multiplexing [25] because of their pseudo-random and quasi-orthogonal properties. An m-array can be constructed by folding a corresponding m-sequence [20], [21]. For any m-sequence with length $n = 2^m-1 = 2^{k_1 k_2}-1$, the size of the corresponding m-array is $n_1 = 2^{k_1}-1$ and $n_2 = n/n_1$, where n_1 and n_2 are relatively prime. For example,

if $n = 15$, then $n_1 = 3$ and $n_2 = 5$ for $m = 4$, $k_1 = 2$, and $k_2 = 2$;

if $n = 63$, then $n_1 = 7$ and $n_2 = 9$ for $m = 6$, $k_1 = 3$, and $k_2 = 2$;

if $n = 511$, then $n_1 = 7$ and $n_2 = 73$ for $m = 9$, $k_1 = 3$, and $k_2 = 3$.

The *m-array* is then obtained by writing the m-sequence down the main diagonal of an $n_1 \times n_2$ matrix and continuing from the opposite side whenever an edge is reached. For example, an m-sequence $[a_0, a_1, ..., a_{14}]$ produces the corresponding m-array

$$\begin{bmatrix} a_0 & a_6 & a_{12} & a_3 & a_9 \\ a_{10} & a_1 & a_7 & a_{13} & a_4 \\ a_5 & a_{11} & a_2 & a_8 & a_{14} \end{bmatrix}.$$

If the m-sequence is [110101100100011], then the corresponding m-array is

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The cyclic correlation properties of m-arrays are similar to those of m-sequences. For example, the cyclic auto-correlation of any 3×5 m-array is

$$\begin{bmatrix} 15 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}.$$

Although m-sequences appear random to the human eye, the corresponding m-arrays have a nonrandom appearance. They are symmetric about a column of zeros. The size of an m-array is also restricted because it depends on the prime factor decomposition of the length of the corresponding m-sequence. Generating a long m-sequence to construct the corresponding m-array is also time-consuming. In addition, the construction method of the m-arrays is complicated. To solve these problems, the quasi m-array is developed.

Definition: Let [a] and [b] be any two m-sequences, where $[z] \equiv [z_0, z_1, ..., z_{n_z-1}]$ and $z = a$ or b. Then the *two-dimensional quasi m-array* [A] is

$$[A] \equiv \begin{bmatrix} a_0 \oplus b_0 & a_0 \oplus b_1 & \cdots & a_0 \oplus b_{n_b-1} \\ a_1 \oplus b_0 & a_1 \oplus b_1 & \cdots & a_1 \oplus b_{n_b-1} \\ \vdots & \vdots & & \vdots \\ a_{n_a-1} \oplus b_0 & a_{n_a-1} \oplus b_1 & \cdots & a_{n_a-1} \oplus b_{n_b-1} \end{bmatrix}. \qquad \text{(IV.1)}$$

By definition, every row or column of a quasi m-array is either an m-sequence or its complement. Therefore, the quasi m-arrays preserve all the properties of the m-sequences in both rows and columns. It is also easier and faster to generate the quasi m-arrays than the m-arrays. In addition, more freedom is available in selecting the size of the quasi m-arrays. For example, the size of a quasi m-array can be 7×7, 7×15, 7×31, and so on. Finally, the quasi m-array is not symmetric when two different m-sequences are used for construction. However, the quasi m-array is symmetric about the diagonal when only one m-sequence is used. For example, if [a] = [b] = [1110100], then the corresponding quasi m-array becomes

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

These m-arrays can be used as reference functions for transform encryption coding for the following reasons. First, they are easy and fast to generate. Secondly, they appear random which ensures security. Figure IV.1 shows a quasi m-array and its binary phase spectrum. However, the number of the quasi m-arrays available is limited. For example, there are only 256 255×255 quasi m-arrays available. Therefore, in the next section a new type of array is derived which can provide many arrays with the pseudo-random property. Before this type of array is stated, let us study the cyclic correlation properties of quasi m-arrays. If $[A]_{i,j}$ is defined as the i^{th} and j^{th} cyclic shift of [A] in column and row respectively, that is,

$$[A]_{i,j} \equiv \begin{bmatrix} a_i \oplus b_j & a_i \oplus b_{j+1} & \cdots & a_i \oplus b_{n_b+j-1} \\ a_{i+1} \oplus b_j & a_{i+1} \oplus b_{j+1} & \cdots & a_{i+1} \oplus b_{n_b+j-1} \\ \vdots & \vdots & & \vdots \\ a_{n_a+i-1} \oplus b_j & a_{n_a+i-1} \oplus b_{j+1} & \cdots & a_{n_a+i-1} \oplus b_{n_b+j-1} \end{bmatrix}. \qquad (IV.2)$$

Then the cyclic correlation properties of the quasi m-arrays are given by the following.

Theorem IV.1: The cyclic auto-correlation $\Theta_{AA}(i,j)$ of a two-dimensional quasi m-array [A] is

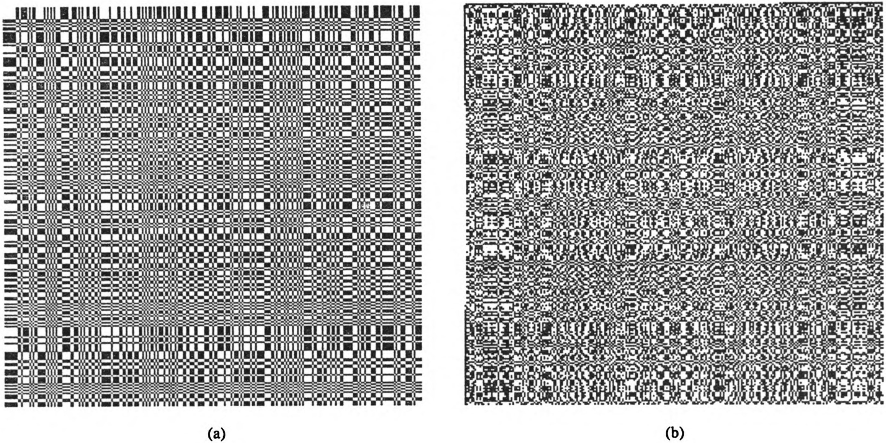


Figure IV.1 A 255×255 quasi m-array and its binary phase spectrum.

(a) Quasi m-array.

(b) Binary phase spectrum.

$$\Theta_{AA}(i,j) = \begin{cases} n_a n_b & \text{for } i = j = 0 \\ -n_b & \text{for } i \neq j = 0 \\ -n_a & \text{for } j \neq i = 0 \\ 1 & \text{for } i \neq 0 \text{ and } j \neq 0. \end{cases} \qquad \text{(IV.3)}$$

Proof:

Case 1: $\Theta_{AA}(0,0)$

By the definition of correlation, $\Theta_{AA}(0,0)$ is the difference between the number of zeros and ones in $[A]\oplus[A]_{0,0}$. But $[A]\oplus[A]_{0,0}$

$$\begin{bmatrix} a_0\oplus b_0\oplus a_0\oplus b_0 & a_0\oplus b_1\oplus a_0\oplus b_1 & \cdots & a_0\oplus b_{n_b-1}\oplus a_0\oplus b_{n_b-1} \\ a_1\oplus b_0\oplus a_1\oplus b_0 & a_1\oplus b_1\oplus a_1\oplus b_1 & \cdots & a_1\oplus b_{n_b-1}\oplus a_1\oplus b_{n_b-1} \\ \vdots & \vdots & & \vdots \\ a_{n_a-1}\oplus b_0\oplus a_{n_a-1}\oplus b_0 & a_{n_a-1}\oplus b_1\oplus a_{n_a-1}\oplus b_1 & \cdots & a_{n_a-1}\oplus b_{n_b-1}\oplus a_{n_a-1}\oplus b_{n_b-1} \end{bmatrix}$$

is a matrix whose elements are all zero. Therefore, $\Theta_{AA}(0,0) = n_a n_b$. (The number of zeros in $[A]\oplus[A]_{0,0}$ is $n_a n_b$.)

Case 2: $\Theta_{AA}(i,0)$, $i \neq 0$

Similarly to *Case 1*, the matrix $[A]\oplus[A]_{i,0}$ can be written as

$$\begin{bmatrix} a_0\oplus b_0\oplus a_i\oplus b_0 & a_0\oplus b_1\oplus a_i\oplus b_1 & \cdots & a_0\oplus b_{n_b-1}\oplus a_i\oplus b_{n_b-1} \\ a_1\oplus b_0\oplus a_{i+1}\oplus b_0 & a_1\oplus b_1\oplus a_{i+1}\oplus b_1 & \cdots & a_1\oplus b_{n_b-1}\oplus a_{i+1}\oplus b_{n_b-1} \\ \vdots & \vdots & & \vdots \\ a_{n_a-1}\oplus b_0\oplus a_{n_a+i-1}\oplus b_0 & a_{n_a-1}\oplus b_1\oplus a_{n_a+i-1}\oplus b_1 & \cdots & a_{n_a-1}\oplus b_{n_b-1}\oplus a_{n_a+i-1}\oplus b_{n_b-1} \end{bmatrix}$$

which becomes

$$\begin{bmatrix} a_0 \oplus a_i & a_0 \oplus a_i & \cdots & a_0 \oplus a_i \\ a_1 \oplus a_{i+1} & a_1 \oplus a_{i+1} & \cdots & a_1 \oplus a_{i+1} \\ \vdots & \vdots & & \vdots \\ a_{n_a-1} \oplus a_{n_a+i-1} & a_{n_a-1} \oplus a_{n_a+i-1} & \cdots & a_{n_a-1} \oplus a_{n_a+i-1} \end{bmatrix}.$$

By Property II.4, we have

$$[A] \oplus [A]_{i,0} = \begin{bmatrix} a_I & a_I & \cdots & a_I \\ a_{I+1} & a_{I+1} & \cdots & a_{I+1} \\ \vdots & \vdots & & \vdots \\ a_{n_a+I-1} & a_{n_a+I-1} & \cdots & a_{n_a+I-1} \end{bmatrix},$$

where $I \neq 0$ and $I \neq i$. Therefore, $\Theta_{AA}(i,0) = -n_b$ by the definition of correlation. (There are n_b columns in $[A] \oplus [A]_{i,0}$, and each column is an m-sequence.)

Case 3: $\Theta_{AA}(0,j)$, $j \neq 0$

Similarly to *Case 2*, $\Theta_{AA}(0,j) = -n_a$.

Case 4: $\Theta_{AA}(i,j)$, $i \neq 0$ and $j \neq 0$

Similarly to *Case 2*, the matrix $[A] \oplus [A]_{i,j}$ can be written as

$$\begin{bmatrix} a_0 \oplus b_0 \oplus a_i \oplus b_j & a_0 \oplus b_1 \oplus a_i \oplus b_{j+1} & \cdots & a_0 \oplus b_{n_b-1} \oplus a_i \oplus b_{n_b+j-1} \\ a_1 \oplus b_0 \oplus a_{i+1} \oplus b_j & a_1 \oplus b_1 \oplus a_{i+1} \oplus b_{j+1} & \cdots & a_1 \oplus b_{n_b-1} \oplus a_{i+1} \oplus b_{n_b+j-1} \\ \vdots & \vdots & & \vdots \\ a_{n_a-1} \oplus b_0 \oplus a_{n_a+i-1} \oplus b_j & a_{n_a-1} \oplus b_1 \oplus a_{n_a+i-1} \oplus b_{j+1} & \cdots & a_{n_a-1} \oplus b_{n_b-1} \oplus a_{n_a+i-1} \oplus b_{n_b+j-1} \end{bmatrix}.$$

By Property II.4, we have

$$[A] \oplus [A]_{i,j} = \begin{bmatrix} a_I \oplus b_J & a_I \oplus b_{J+1} & \cdots & a_I \oplus b_{n_b+J-1} \\ a_{I+1} \oplus b_J & a_{I+1} \oplus b_{J+1} & \cdots & a_{I+1} \oplus b_{n_b+J-1} \\ \vdots & \vdots & & \vdots \\ a_{n_a+I-1} \oplus b_J & a_{n_a+I-1} \oplus b_{J+1} & \cdots & a_{n_a+I-1} \oplus b_{n_b+J-1} \end{bmatrix},$$

where $I \neq 0$, $I \neq i$, $J \neq 0$, and $J \neq j$. Therefore, we have

$$\Theta_{AA}(i,j) = \frac{n_a-1}{2}\frac{n_b-1}{2}+\frac{n_a+1}{2}\frac{n_b+1}{2}-\frac{n_a-1}{2}\frac{n_b+1}{2}-\frac{n_a+1}{2}\frac{n_b-1}{2} = 1$$

by the definition of correlation. **Q.E.D.**

Theorem IV.2: The cross-correlation $\Theta_{AA'}(i,j)$ between two different two-dimensional quasi m-arrays [A] and [A′] is

$$\Theta_{AA'}(i,j) = \theta_{aa'}(i)\theta_{bb'}(j), \tag{IV.4}$$

where [A′] is the quasi m-array generated by [a′] and [b′].

Proof:

Similarly to the proof of Theorem IV.1, the matrix $[A]\oplus[A']_{i,j}$ can be written as

$$\begin{bmatrix} a_0\oplus b_0\oplus a'_i\oplus b'_j & a_0\oplus b_1\oplus a'_i\oplus b'_{j+1} & \cdots & a_0\oplus b_{n_b-1}\oplus a'_i\oplus b'_{n_b+j-1} \\ a_1\oplus b_0\oplus a'_{i+1}\oplus b'_j & a_1\oplus b_1\oplus a'_{i+1}\oplus b'_{j+1} & \cdots & a_1\oplus b_{n_b-1}\oplus a'_{i+1}\oplus b'_{n_b+j-1} \\ \vdots & \vdots & & \vdots \\ a_{n_a-1}\oplus b_0\oplus a'_{n_a+i-1}\oplus b'_j & a_{n_a-1}\oplus b_1\oplus a'_{n_a+i-1}\oplus b'_{j+1} & \cdots & a_{n_a-1}\oplus b_{n_b-1}\oplus a'_{n_a+i-1}\oplus b'_{n_b+j-1} \end{bmatrix}.$$

Now, let us define

$$[z_0, z_1, \ldots, z_{n_z-1}]\oplus[z'_k, z'_{k+1}, \ldots, z'_{n_z+k-1}] \equiv [\vec{z}^k_0, \vec{z}^k_1, \ldots, \vec{z}^k_{n_z-1}]$$

and the cross-correlation between $[z_0, \ldots, z_{n_z-1}]$ and $[z'_k, \ldots, z'_{n_z+k-1}]$ is $\theta_{zz'}(k)$, where z = a or b, and k = i or j. Then

$$[A]\oplus[A']_{i,j} = \begin{bmatrix} \vec{a}^i_0\oplus\vec{b}^j_0 & \vec{a}^i_0\oplus\vec{b}^j_1 & \cdots & \vec{a}^i_0\oplus\vec{b}^j_{n_b-1} \\ \vec{a}^i_1\oplus\vec{b}^j_0 & \vec{a}^i_1\oplus\vec{b}^j_1 & \cdots & \vec{a}^i_1\oplus\vec{b}^j_{n_b-1} \\ \vdots & \vdots & & \vdots \\ \vec{a}^i_{n_a-1}\oplus\vec{b}^j_0 & \vec{a}^i_{n_a-1}\oplus\vec{b}^j_1 & \cdots & \vec{a}^i_{n_a-1}\oplus\vec{b}^j_{n_b-1} \end{bmatrix}.$$

By the definition of correlation, we have

$$\Theta_{AA'} = \frac{[n_a+\theta_{aa'}(i)]}{2}\frac{[n_b+\theta_{bb'}(j)]}{2} + \frac{[n_a-\theta_{aa'}(i)]}{2}\frac{[n_b-\theta_{bb'}(j)]}{2}$$

$$-\frac{[n_a+\theta_{aa'}(i)]}{2}\frac{[n_b-\theta_{bb'}(j)]}{2} - \frac{[n_a-\theta_{aa'}(i)]}{2}\frac{[n_b+\theta_{bb'}(j)]}{2}$$

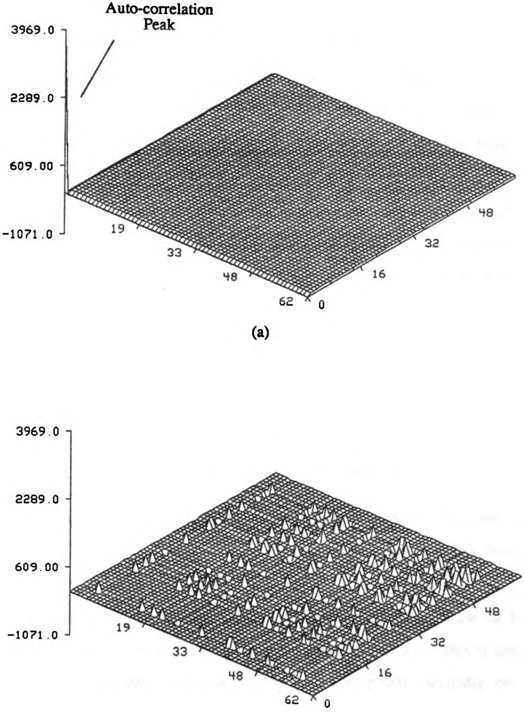
$$= \theta_{aa'}(i)\theta_{bb'}(j). \qquad \textbf{Q.E.D.}$$

Remarks: The cyclic auto-correlation of any quasi m-array can also be written as

$$n_a n_b \begin{bmatrix} 1 & -1/n_b & \dots & -1/n_b \\ -1/n_a & 1/n_a n_b & \dots & 1/n_a n_b \\ \vdots & \vdots & & \vdots \\ -1/n_a & 1/n_a n_b & \cdots & 1/n_a n_b \end{bmatrix}.$$

The difference between the m-arrays and the quasi m-arrays is the shape of cyclic auto-correlation. The cyclic auto-correlation floor of any m-array is constant, while that of a quasi m-array is not. The cyclic cross-correlation between two different quasi m-arrays is the product of the cyclic cross-correlations between the generating m-sequences of the quasi m-arrays. For a different pair of the m-sequences, the bound on their cyclic cross-correlation is different. Therefore, a different pair of the quasi m-arrays has a different bound on their cyclic cross-correlation. As the size of the quasi m-array approaches infinity, the cyclic auto-correlation approaches a delta-function. In addition, the cyclic cross-correlation is negligible when compared with the cyclic auto-correlation peak. Therefore, these quasi m-arrays are quasi-orthogonal. These quasi m-arrays are so-named because their cyclic correlation properties are similar to those of m-arrays.

Example IV.1: Figure IV.2 shows the cyclic correlations of two 63×63 quasi m-arrays. Each quasi m-array is generated by an m-sequence. The two m-sequences discussed in Example II.1 were used to generate these quasi m-arrays. The cyclic cross-correlation values are $(-1)^2$, $(-17)^2$, 15^2, $(-1)(15)$, $(-1)(-17)$, and $(-17)(15)$. In addition,



(a)

(b)

Figure IV.2 Cyclic correlations of two 63×63 quasi m-arrays.
(a) Cyclic auto-correlation.
(b) Cyclic cross-correlation.

the cyclic auto-correlation values are 63^2, (-1)(63), and 1. These values are obtained from the theorems above and verified by computer simulation.

In summary, properties of the quasi m-arrays are the following.

Property IV.1: For any $n_a \times n_b$ quasi m-array, the number of zeros is $(n_a n_b+1)/2$ and the number of ones is $(n_a n_b-1)/2$, where n_a and n_b are the length of the m-sequences in the rows and columns respectively.

Property IV.2: The statistical distribution of ones and zeros in each row or column is well defined. Relative positions of their runs change from sequence to sequence but the number of each run length does not. Every row or column is an m-sequence or its complement.

Property IV.3: The cyclic auto-correlation of a quasi m-array is as follows:

$$\Theta_{AA}(i,j) = \begin{cases} n_a n_b & \text{for } i = j = 0 \\ -n_b & \text{for } i \neq j = 0 \\ -n_a & \text{for } j \neq i = 0 \\ 1 & \text{for } i \neq 0 \text{ and } j \neq 0 \end{cases} \qquad \text{(IV.3)}$$

Property IV.4: A modulo-2 addition of a quasi m-array with a cyclic phase shifted replica of itself results in another cyclic phase shifted replica which differs from either of the originals.

The advantages of the quasi m-arrays over those of the m-arrays are the following. First, they are easier and faster to generate than the m-arrays. This is useful for transform encryption coding and code division multiplexing [25]. Secondly, the selection size of the quasi m-arrays is more flexible than that of the m-arrays. This is also useful for applications because many quasi m-arrays with different sizes and the quasi-orthogonal property can be easily generated [23]-[25]. Finally, the random appearance of the quasi m-arrays is also useful for transform encryption coding and the testing of texture segmentation algorithms [49], [50].

IV.2 Two-dimensional Gold Code Arrays

Gold code sequences are generated by the modulo-2 addition of a pair of m-sequences. Therefore, the two-dimensional Gold code array is defined as the following:

Definition: The *two-dimensional Gold code array* is

$$[G] \equiv \begin{bmatrix} \bar{a}_0 \oplus \bar{b}_0 & \bar{a}_0 \oplus \bar{b}_1 & \cdots & \bar{a}_0 \oplus \bar{b}_{n_b-1} \\ \bar{a}_1 \oplus \bar{b}_0 & \bar{a}_1 \oplus \bar{b}_1 & \cdots & \bar{a}_1 \oplus \bar{b}_{n_b-1} \\ \vdots & \vdots & & \vdots \\ \bar{a}_{n_a-1} \oplus \bar{b}_0 & \bar{a}_{n_a-1} \oplus \bar{b}_1 & \cdots & \bar{a}_{n_a-1} \oplus \bar{b}_{n_b-1} \end{bmatrix}. \qquad \text{(V.5)}$$

where $[\bar{z}] \equiv [z] \oplus [z']$ and $z = a$ or b. In other words, $[\bar{z}]$ is a Gold code sequence. $[z]$ and $[z']$ are the m-sequences with length n_z.

These Gold code arrays can also be used as the reference function for transform encryption coding. It can be inferred from the definition above that there are many Gold code arrays available. For example, the number of 255×255 Gold code arrays available is 936,360,000. Therefore, it is almost impossible to attack transform encryption coding when these arrays are used as reference functions. Figure IV.3 shows a Gold code array and its binary phase spectrum. The Gold code arrays and the quasi m-arrays are generated by the same construction method. Similarly to the Gold code sequences, we have

$$[G]^{p,q} \equiv \begin{bmatrix} a_0 \oplus a'_p \oplus b_0 \oplus b'_q & \cdots & a_0 \oplus a'_p \oplus b_{n_b-1} \oplus b'_{n_b-1+q} \\ a_1 \oplus a'_{p+1} \oplus b_0 \oplus b'_q & \cdots & a_1 \oplus a'_{p+1} \oplus b_{n_b-1} \oplus b'_{n_b+q-1} \\ \vdots & & \vdots \\ a_{n_a-1} \oplus a'_{n_a-1+p} \oplus b_0 \oplus b'_q & \cdots & a_{n_a-1} \oplus a'_{n_a+p-1} \oplus b_{n_b-1} \oplus b'_{n_b+q-1} \end{bmatrix}, \qquad \text{(IV.6)}$$

and $[G] \neq [G]^{p,q}$. In other words, $[G]$ and $[G]^{p,q}$ are two different Gold code arrays generated by the same m-sequences with different relative phase shifts.

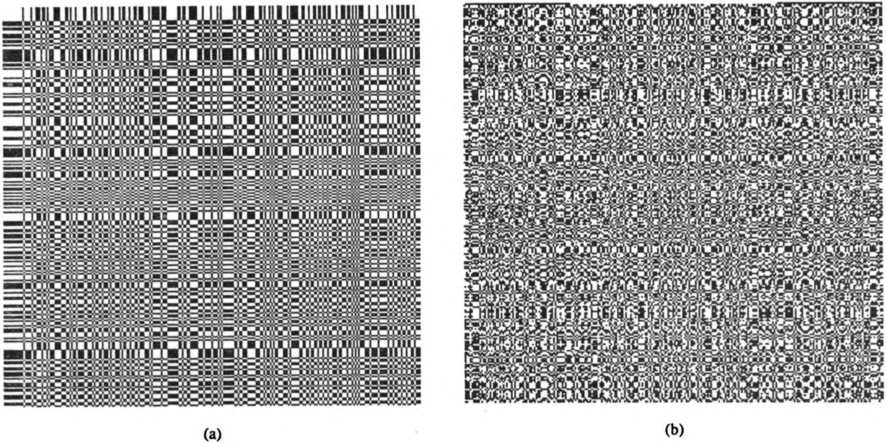


Figure IV.3 A 255×255 Gold code array and its binary phase spectrum.

(a) Gold code array.

(b) Binary phase spectrum.

Theorem IV.3: The cyclic auto-correlation $\Theta_{GG}(i,j)$ of a two-dimensional Gold code array [G] is

$$\Theta_{GG}(i,j) = \begin{cases} n_a n_b & \text{for } i = j = 0 \\ n_b \theta_{aa'}(I'-I) & \text{for } i \neq j = 0 \\ n_a \theta_{bb'}(J'-J) & \text{for } j \neq i = 0 \\ \theta_{aa'}(I'-I)\theta_{bb'}(J'-J) & \text{for } i \neq 0 \text{ and } j \neq 0, \end{cases} \qquad \text{(IV.7)}$$

where $I \neq 0$, $I \neq i$, $I' \neq 0$, $I' \neq i$, $J \neq 0$, $J \neq j$, $J' \neq 0$, and $J' \neq j$.

Proof:

Case 1: $\Theta_{GG}(0,0)$

Similarly to the proof of Theorems IV.1 and IV.2, the matrix $[G]\oplus[G]_{0,0}$ is

$$\begin{bmatrix} \overline{a}_0\oplus\overline{b}_0\oplus\overline{a}_0\oplus\overline{b}_0 & \overline{a}_0\oplus\overline{b}_1\oplus\overline{a}_0\oplus\overline{b}_1 & \cdots & \overline{a}_0\oplus\overline{b}_{n_b-1}\oplus\overline{a}_0\oplus\overline{b}_{n_b-1} \\ \overline{a}_1\oplus\overline{b}_0\oplus\overline{a}_1\oplus\overline{b}_0 & \overline{a}_1\oplus\overline{b}_1\oplus\overline{a}_1\oplus\overline{b}_1 & \cdots & \overline{a}_1\oplus\overline{b}_{n_b-1}\oplus\overline{a}_1\oplus\overline{b}_{n_b-1} \\ \vdots & \vdots & & \vdots \\ \overline{a}_{n_a-1}\oplus\overline{b}_0\oplus\overline{a}_{n_a-1}\oplus\overline{b}_0 & \overline{a}_{n_a-1}\oplus\overline{b}_1\oplus\overline{a}_{n_a-1}\oplus\overline{b}_1 & \cdots & \overline{a}_{n_a-1}\oplus\overline{b}_{n_b-1}\oplus\overline{a}_{n_a-1}\oplus\overline{b}_{n_b-1} \end{bmatrix}$$

whose elements are all zeros. Therefore, $\Theta_{GG}(0,0) = n_a n_b$.

Case 2: $\Theta_{GG}(i,0)$, $i \neq 0$

Similarly to *Case 1*, the matrix $[G]\oplus[G]_{i,0}$ is

$$\begin{bmatrix} \overline{a}_0\oplus\overline{b}_0\oplus\overline{a}_i\oplus\overline{b}_0 & \overline{a}_0\oplus\overline{b}_0\oplus\overline{a}_i\oplus\overline{b}_0 & \cdots & \overline{a}_0\oplus\overline{b}_0\oplus\overline{a}_i\oplus\overline{b}_0 \\ \overline{a}_1\oplus\overline{b}_0\oplus\overline{a}_{i+1}\oplus\overline{b}_0 & \overline{a}_1\oplus\overline{b}_0\oplus\overline{a}_{i+1}\oplus\overline{b}_0 & \cdots & \overline{a}_1\oplus\overline{b}_0\oplus\overline{a}_{i+1}\oplus\overline{b}_0 \\ \vdots & \vdots & & \vdots \\ \overline{a}_{n_a-1}\oplus\overline{b}_0\oplus\overline{a}_{n_a+i-1}\oplus\overline{b}_0 & \overline{a}_{n_a-1}\oplus\overline{b}_0\oplus\overline{a}_{n_a+i-1}\oplus\overline{b}_0 & \cdots & \overline{a}_{n_a-1}\oplus\overline{b}_0\oplus\overline{a}_{n_a+i-1}\oplus\overline{b}_0 \end{bmatrix}$$

which becomes

$$\begin{bmatrix} a_0 \oplus a'_0 \oplus a_i \oplus a'_i & \cdots & a_0 \oplus a'_0 \oplus a_i \oplus a'_i \\ a_1 \oplus a'_1 \oplus a_{i+1} \oplus a'_{i+1} & \cdots & a_1 \oplus a'_1 \oplus a_{i+1} \oplus a'_{i+1} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ a_{n_a-1} \oplus a'_{n_a-1} \oplus a_{n_a+i-1} \oplus a'_{n_a+i-1} & \cdots & a_{n_a-1} \oplus a'_{n_a-1} \oplus a_{n_a+i-1} \oplus a'_{n_a+i-1} \end{bmatrix}.$$

In other words,

$$[G] \oplus [G]_{i,0} = \begin{bmatrix} a_I \oplus a'_{I'} & \cdots & a_I \oplus a'_{I'} \\ a_{I+1} \oplus a'_{I'+1} & \cdots & a_{I+1} \oplus a'_{I'+1} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ a_{n_a+I-1} \oplus a'_{n_a+I'-1} & \cdots & a_{n_a+I-1} \oplus a'_{n_a+I'-1} \end{bmatrix}.$$

Therefore, we have $\Theta_{GG}(i,0) = n_b \theta_{aa'}(I'-I)$ from the proofs of Theorems IV.1 and IV.2. (I and I′ are different numbers because [a] and [a′] are different m-sequences.)

Case 3: $\Theta_{GG}(0,j)$, $j \neq 0$

Similarly to *Case 2*, $\Theta_{GG}(0,j) = n_a \theta_{bb'}(J'-J)$.

Case 4: $\Theta_{GG}(i,j)$, $i \neq 0$ and $j \neq 0$

From the proofs of Theorems IV.1 and IV.2, we have $\Theta_{GG}(i,j) = \theta_{aa'}(I'-I)\theta_{bb'}(J'-J)$.

Q.E.D.

Theorem IV.4: The cross-correlation $\Theta_{GG^{p,q}}(i,j)$ between two different Gold code arrays [G] and $[G]^{p,q}$ is

$$\Theta_{GG^{p,q}}(i,j) = \theta_{aa'}(Ip-I)\theta_{bb'}(Jq-J), \qquad \text{(IV.8)}$$

where $I \neq 0$, $I \neq i$, $Ip \neq 0$, $Ip \neq p+i$, $J \neq 0$, $J \neq j$, $Jq \neq 0$, and $Jq \neq q+j$.

Proof:

Similarly to the proof of Theorem IV.2, we have

$$[G]+[G]^{p,q}{}_{i,j} = \begin{bmatrix} a_0 \oplus a'_0 \oplus b_0 \oplus b'_0 \oplus a_i \oplus a'_{p+i} \oplus b_j \oplus b'_{q+j} & \cdots & a_0 \oplus a'_0 \oplus b_{n_b-1} \oplus b'_{n_b-1} \oplus a_i \oplus a'_{p+i} \oplus b_{n_b+j-1} \oplus b'_{n_b+q+j-1} \\ a_1 \oplus a'_1 \oplus b_0 \oplus b'_0 \oplus a_{i+1} \oplus a'_{p+i+1} \oplus b_j \oplus b'_{q+j} & \cdots & a_1 \oplus a'_1 \oplus b_{n_b-1} \oplus b'_{n_b-1} \oplus a_{i+1} \oplus a'_{p+i+1} \oplus b_{n_b+j-1} \oplus b_{n_b+q+j-1} \\ \vdots & & \vdots \\ a_{n_a-1} \oplus a'_{n_a-1} \oplus b_0 \oplus b'_0 \oplus a_{n_a+i-1} \oplus a'_{n_a+p+i-1} \oplus b_j \oplus b_{q+j} & \cdots & a_{n_a-1} \oplus a'_{n_a-1} \oplus b_{n_b-1} \oplus b'_{n_b-1} \oplus a_{n_a+i-1} \oplus a'_{n_a+p+i-1} \oplus b_{n_b+j-1} \oplus b'_{n_b+q+j-1} \end{bmatrix}.$$

By the proof of Theorem IV.2 and Property II.4, we have

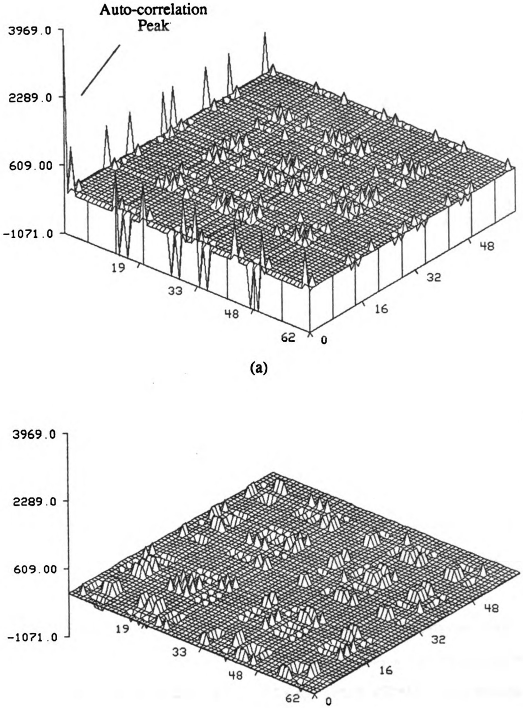
$$[G]\oplus[G]^{p,q_{i,j}} = \begin{bmatrix} a_I \oplus a'_{Ip} \oplus b_J \oplus b'_{Jq} & \cdots & a_I \oplus a'_{Ip} \oplus b_{n_b+J-1} \oplus b'_{n_b+Jq-1} \\ a_{I+1} \oplus a'_{Ip+1} \oplus b_J \oplus b'_{Jq} & \cdots & a_{I+1} \oplus a'_{Ip+1} \oplus b_{n_b+J-1} \oplus b'_{n_b+Jq-1} \\ \vdots & & \vdots \\ a_{n_a+I-1} \oplus a'_{n_a+Ip-1} \oplus b_J \oplus b'_{Jq} & \cdots & a_{n_a+I-1} \oplus a'_{n_a+Ip-1} \oplus b_{n_b+J-1} \oplus b'_{n_b+Jq-1} \end{bmatrix}.$$

Therefore, $\Theta_{GG^{p,q}} = \theta_{aa'}(Ip-I)\theta_{bb'}(Jq-J)$ from the proof of Theorem IV.3. **Q.E.D.**

Remarks: Similarly to the quasi m-arrays, every row or column of a Gold code array is a Gold code sequence or its complement. In addition, these Gold code arrays are also quasi-orthogonal.

For the quasi m-arrays, a different pair of arrays has a different bound on its cyclic cross-correlation. On the contrary, the Gold code array provides families of arrays in which any two arrays have the same bound on their cyclic cross-correlation and auto-correlation floors. In addition, the cyclic auto-correlation peak of the Gold code array and that of the quasi m-array are the same when both arrays have the same size. These correlation properties are important for the application of code division multiplexing [25].

Example IV.2: Figure IV.4 shows the cyclic correlations of two 63×63 Gold code arrays. These arrays are generated by the m-sequences discussed in Example II.1. The cyclic cross-correlation values of these Gold code arrays are the same as those of the quasi m-arrays discussed in Example IV.1. In addition, the cyclic auto-correlation values are 63^2, (-17)(63), (-1)(63), (15)(63), $(-1)^2$, $(-17)^2$, 15^2, (-1)(15), (-1)(-17), and (-17)(15). These values are also obtained from the theorems above and verified by computer simulation.



(a)

(b)

Figure IV.4 Cyclic correlations of two 63×63 Gold code arrays.

(a) Cyclic auto-correlation.

(b) Cyclic cross-correlation.

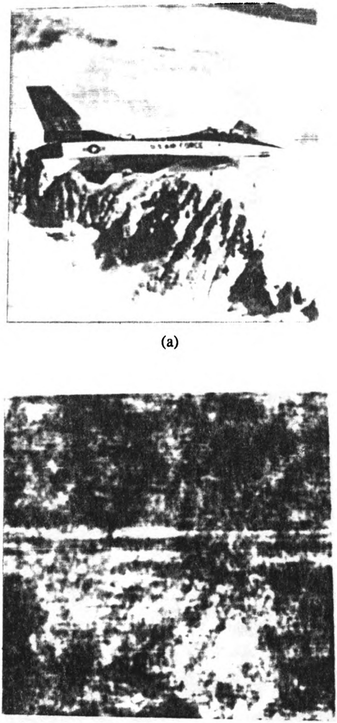
CHAPTER V

COMPUTER SIMULATION

V.1 Simulation Studies

In this section, simulation studies of the developed transform encrypted image coding technique are presented. The reference functions used for encryption are the pseudo-noises discussed in the previous chapter. These pseudo-noises have the following characteristics. First, they are easy and fast to generate. Secondly, they have a random appearance. Finally, it is difficult to compromise the security of these arrays because there are so many pseudo-noises available. The size of these pseudo-noises $(2^N-1)\times(2^N-1)$, where $N \varepsilon \mathrm{N}$. Because of the properties of these pseudo-noises, the encrypted images can be obtained easily and remain secure.

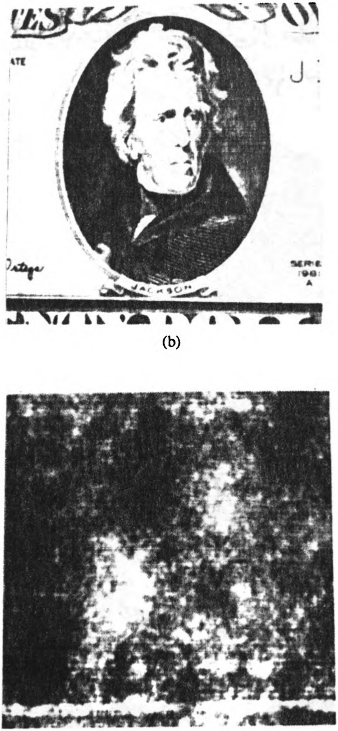
Figure V.1 shows the original images and their encrypted images used in the simulation. The size of these images is 256×256. The gray-scaled values for the original images are from 0 to 255. Five randomly chosen 255×255 quasi m-arrays are used for the image encryption respectively. The gray-scaled values of the encrypted image depend on the original images and reference functions. In some applications, they have to be negative to ensure a uniform amplitude spectrum. These encrypted images are scaled and shown by using gray-scaled values from 0 to 255. Figure V.1 shows these encrypted images are unrecognizable. In addition, all the high contrast



(a)

(a′)

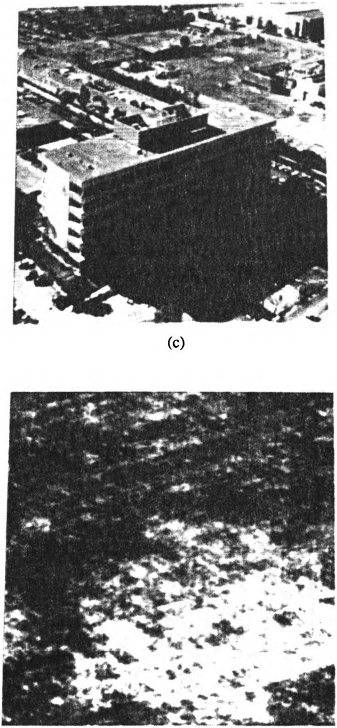
Figure V.1 The natural images and their encrypted images used in the simulation.
(a), (b), (c), (d), (e) Natural images.
(a′), (b′), (c′), (d′), (e′) Encrypted images.



(b)

(b′)

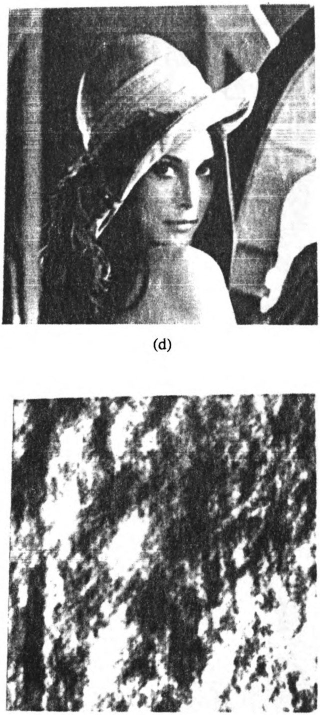
Figure V.1 (cont'd.)



(c)

(c′)

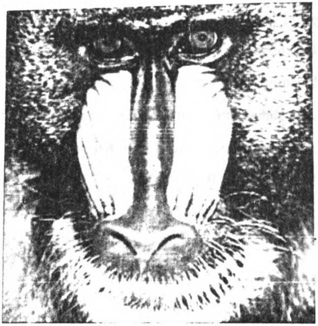
Figure V.1 (cont'd.)



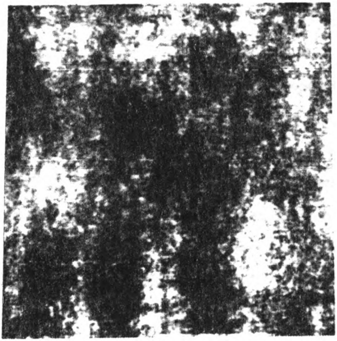
(d)

(d′)

Figure V.1 (cont'd.)



(e)

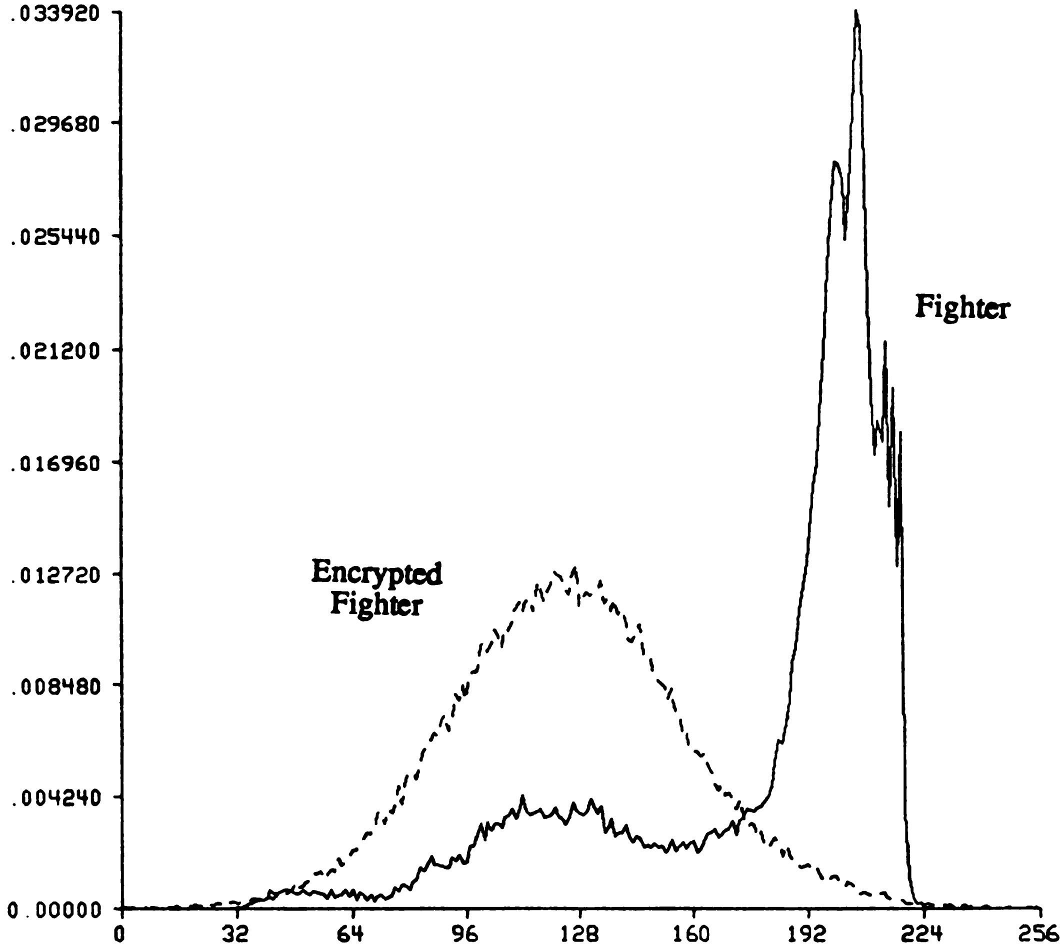


(e′)

Figure V.1 (cont'd.)

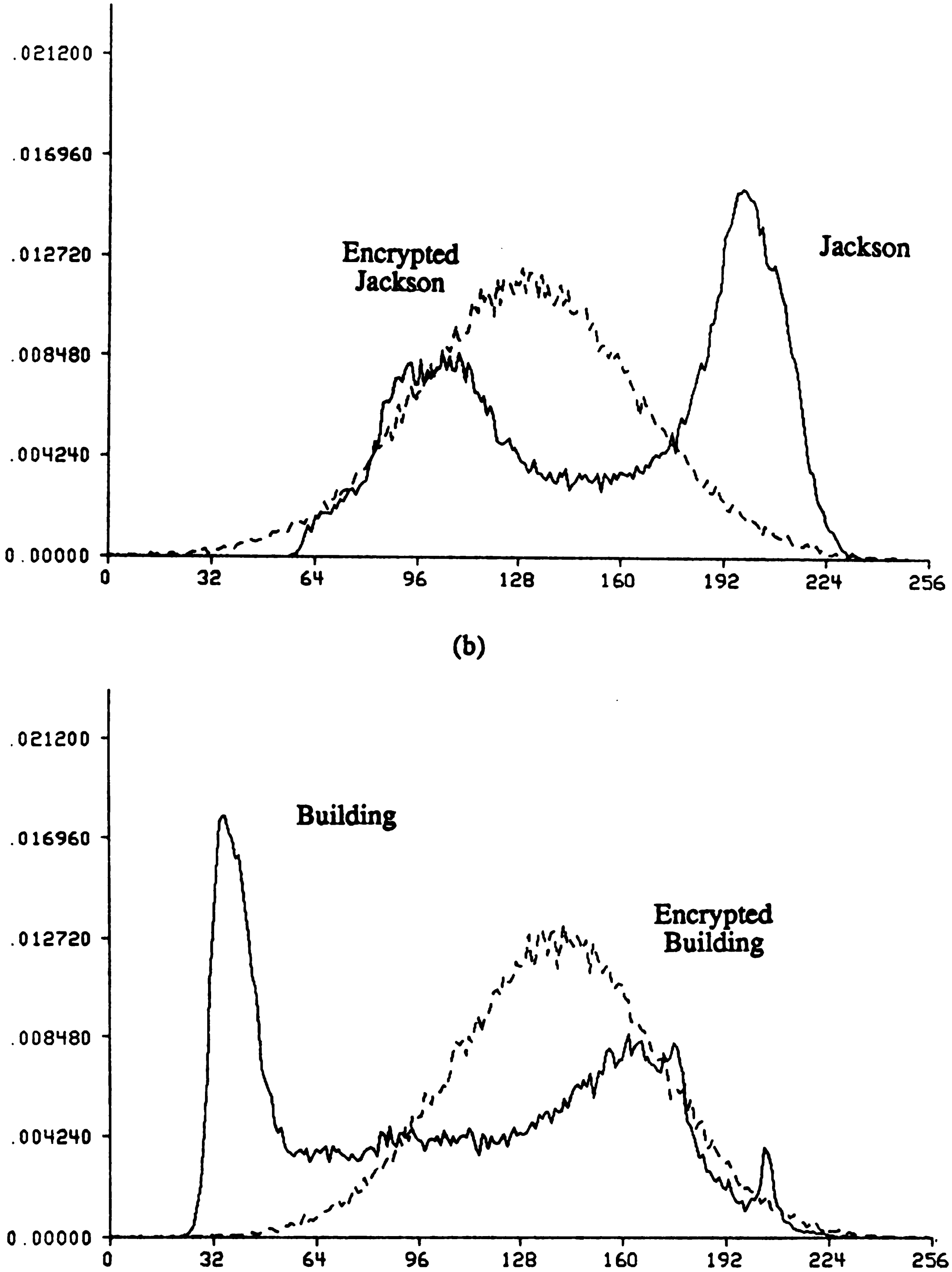
and detail of the original images are lost, and the encrypted images appear more uniform than the original images. Figure V.2 shows the estimated probability density functions for the original and encrypted images. The estimated probability density functions for the original images exhibit no regularity, while those for the encrypted images approximate Gaussian density functions. For the approximation errors defined in Equation (III.2), they are 0.00859, 0.00298, 0.00714, 0.01496, and 0.01095 for Fighter, Jackson, and Building, Lenna, and Mandrill respectively. These errors verify the extended central limit theorem because they are on the order of 0.00391 ($1/\sqrt{256^2}$). From these results, the signal encryption with size 256×256 is a good choice because the approximation error is about from 0.3% to 1.5% which is acceptable in most engineering applications.

In the independence theorem, the K-L transform is used to obtained independent Gaussian random variables. It is well known that the K-L transform is difficult to calculate and is without an efficient computation algorithm [1]-[10], [51]. Fortunately, many other unitary mathematical transforms have been proposed and their performances have been shown to approach that of the K-L transform [7]-[10]. In addition, these unitary mathematical transforms have fast computation algorithms [7]-[10]. These transformations include cosine, sine, Fourier, Hadamard, Harr, Slant, and Hartly. In the computer simulation, the K-L transform is replaced by the cosine and Hadamard transforms. The Hadamard and Fourier transforms share the same computational algorithm [1]-[10], [52]. The difference is that the transform coefficients of the Hadamard are real, while those of the Fourier are complex. Sometimes, there is a restriction on the type of signal that can be transformed by a specific transformation. For example, cosine (sine) transform can only transform even (odd) functions. However, there is no such restriction in using the Hadamard transform. Therefore, Hadamard transform is widely used in transform coding. Since cosine transform was defined as the international standard for transform coding by CCITT [53], some simulation results for cosine



(a)

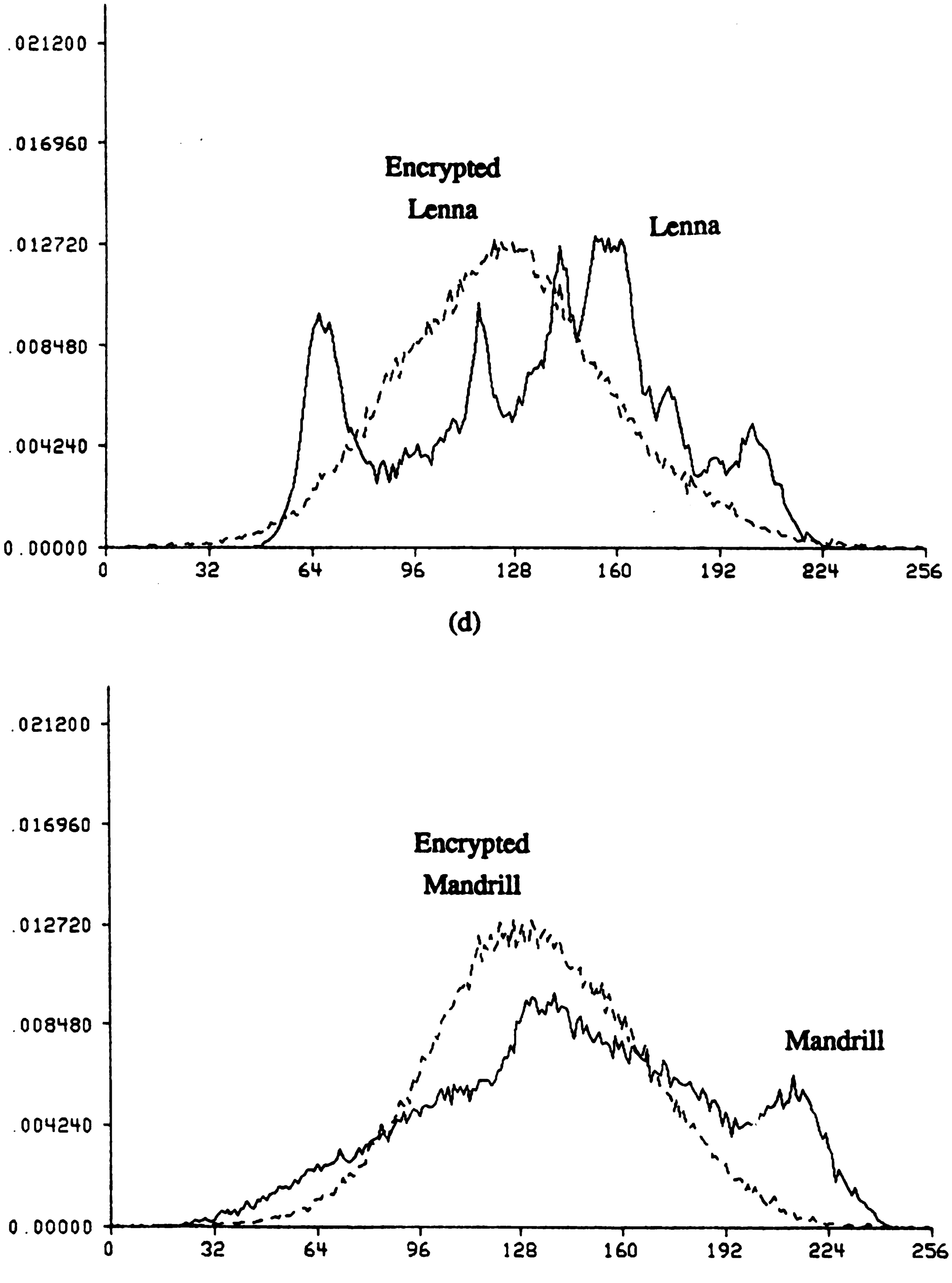
Figure V.2 The estimated probability density functions for the natural and encrypted images.



(b)

(c)

Figure V.2 (cont'd.)



(d)

(e)

Figure V.2 (cont'd.)

transform coding are also studied. Although only cosine and Hadamard transform coding are studied, similar results can be expected if other transformations are used.

To perform transform image coding, the image must be partitioned into many subimages to estimate the probability density function for each transform coefficient. A unitary mathematical transform is then performed on these subimages. The larger the subimage size, the better the coding performance because the K-L transform coefficients are asymptotically independent Gaussian random variables. In addition, the performance of the unitary mathematical transforms will also approach that of the K-L transform. However, the coding performance does not improve significantly after subimage size 16×16 in practice [1]-[10]. In addition, the computational efficiency of transformation decreases when the subimage size increases. The number of samples available for the estimation of probability density functions for different transform coefficients also decreases as the subimage size increases. Therefore, the subimage size is always chosen from 4×4 to 16×16 in practice. The subimage size used in these simulations is 16×16.

The quantization technique used in the simulation is the Wintz and Kurtenbach technique [31] discussed in Section II.2. The probability density function for each transform coefficient is assumed to be Gaussian because of the independence theorem. Since the cosine and Hadamard transforms are used in the simulation instead of the K-L transform, this assumption is true only when the subimage size is large. This is another reason why the 16×16 subimage size is chosen. Companding quantization is used instead of Max quantization because of its easy implementation. As long as the mean and variance of each transform coefficient is estimated, the quantization levels and reconstructed values are easily obtained.

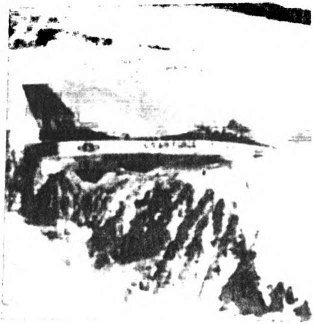
Usually, the bit allocation is set according to a map regardless of the particular image. Cosine transform coding for three different maps is also studied. The bit allocation maps are the ones shown in Figure II.1. The advantage of using these maps is that no run-length coding for bit allocation is required. However, the drawback is that the bit allocation cannot be changed according to the image statistics.

V.2 Results

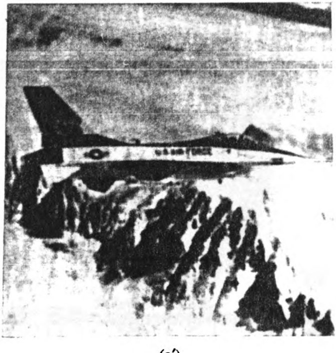
Different cases are tested in the computer simulation. First, the Hadamard transform image coding is studied for Fighter, Jackson, and Building. For the Hadamard transform encrypted image coding, the 255×255 quasi m-arrays are first used as the reference functions to produce the encrypted images. The size of original images is also reduced to 255×255. Since 255 is not a power of 2, a prime factor FFT must be used to compute the encrypted images [54], [55]. After the encrypted images are obtained, the Hadamard transform encrypted image coding is investigated. The drawback is that more computation is required to compute the encrypted images using a 255×255 FFT than a 256×256 FFT. However, code division image multiplexing is possible because of the quasi-orthogonal property of the quasi m-arrays [25]. To obviate the need for prime factor FFTs above, the 255×255 quasi m-arrays are put into 256×256 matrices as the reference functions with the boundaries filled with zeros. The experiment above is then repeated. The quasi orthogonal property of the quasi m-arrays is lost when they are zero padded. However, the speed of 256×256 FFT is about 3 times faster than that of 255×255 FFT [25]. Therefore, the speed of encryption is greatly increased. In this case, three 256×256 FFTs are required by the encryption process. To reduce the number of FFTs required in this process, the following case is tested. The quasi m-arrays are first put into 256×256 matrices with the boundaries filled with zeros. The binary phase spectra of these 256×256 matrices are calculated according to Equation (III.5). The binary phase spectrum of a quasi m-array is shown in Figure IV.2. These binary phase spectra are stored inside the computer. The encrypted images are then obtained by cyclic scrambling the phase spectra of the original images according to these binary phase spectra stored in the computer. The

Hadamard transform encrypted image coding is then studied. The encrypted images shown in Figure V.1 are the results of this case. The advantage of this case is that the number of FFTs required by the encryption process is reduced to two which increases the speed of encryption by 1.5 times. The disadvantage is that the binary phase spectrum must be computed and stored first. In all four cases, the encrypted images appear similar.

Figure V.3, V.4, and V.5 show the reconstructed images of Hadamard transform image and encrypted image coding at nominal 0.74, 1.09, and 1.52 bit/pixel, respectively. The encrypted images used for transform coding are shown in Figure IV.2. Apparently, the results of transform encrypted image coding are much better than those of transform image coding. The transform encrypted image coding preserves the edge, contrast, and detail better than the transform image coding. In the transform image coding, the quantization error only appears at the subimages where the error occurs. Therefore, the error is localized. The ***blocking effect***, which results from nonoverlapping coding of each subimage, is highly visible and very unpleasant visually. This effect can be observed in Figures V.3, V.4, and V.5. This blocking effect can be removed by the overlapping blocks [7]-[10]. However, the efficiency of data compression will be decrease because of the redundancies between blocks increase. On the contrary, the quantization error spreads over the whole image in the transform encrypted image coding because of the required decryption. Therefore, the error is less noticeable. In addition, blocking effect is removed because the error associated with nonoverlapping coding of each subimage spreads over the whole image during the process of decryption. Table V.1 shows the objective simulation results, signal-to-noise (SNR), at nominal 0.74, 1.09, and 1.52 bits/pixel, respectively. The SNR is defined as the total signal energy over total noise energy,

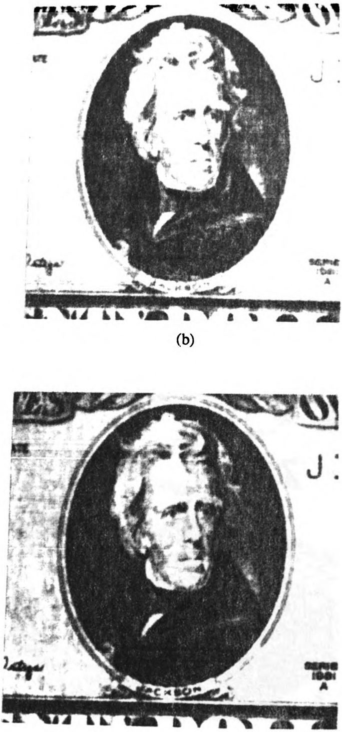


(a)



(a′)

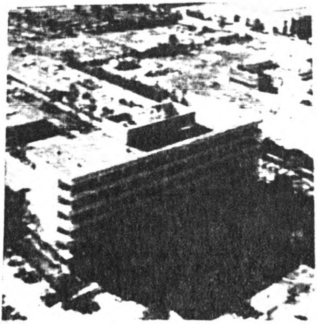
Figure V.3 The reconstructed images from Hadamard transform coding at 1.52 bits/pixel.
(a), (b), (c) From transform coding.
(a′), (b′), (c′) From transform encryption coding.



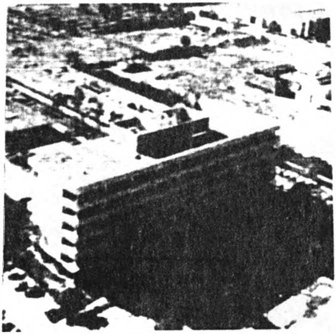
(b)

(b′)

Figure V.3 (cont'd.)

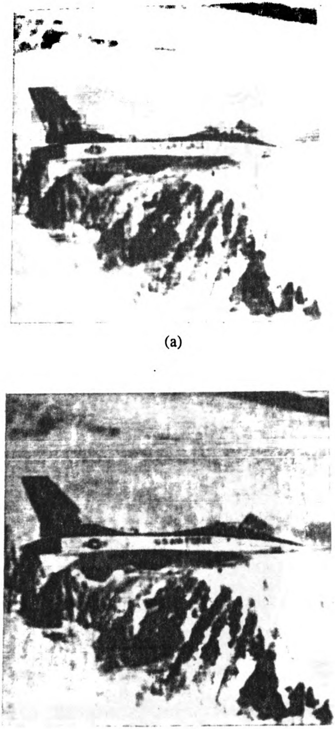


(c)



(c′)

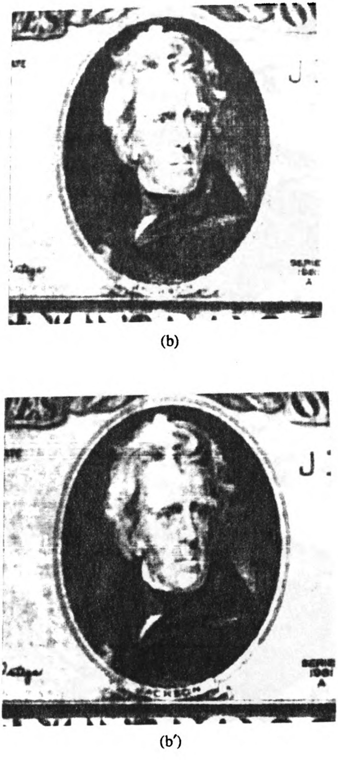
Figure V.3 (cont'd.)



(a)

(a′)

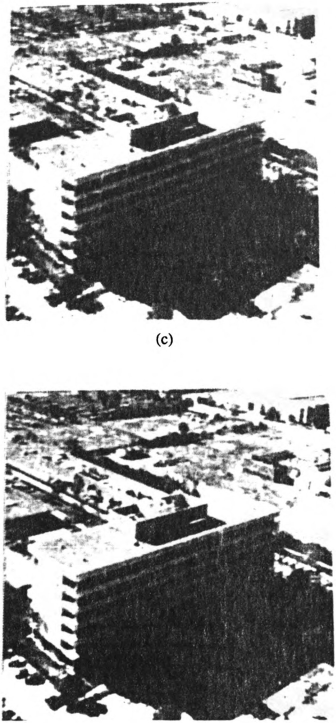
Figure V.4 The reconstructed images from Hadamard transform coding at 1.09 bits/pixel.
(a), (b), (c) From transform coding.
(a′), (b′), (c′) From transform encryption coding.



(b)

(b′)

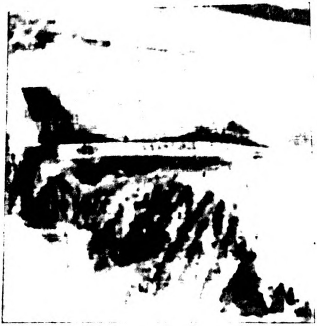
Figure V.4 (cont'd.)



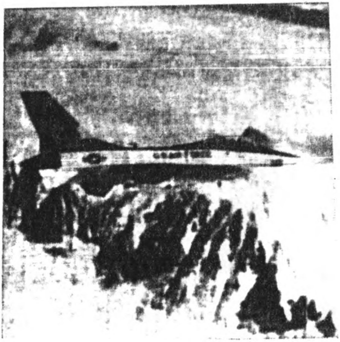
(c)

(c′)

Figure V.4 (cont'd.)

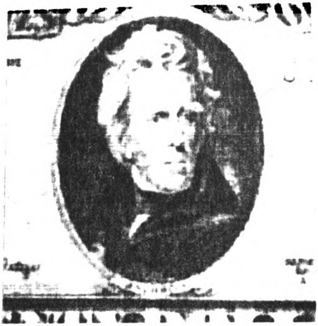


(a)

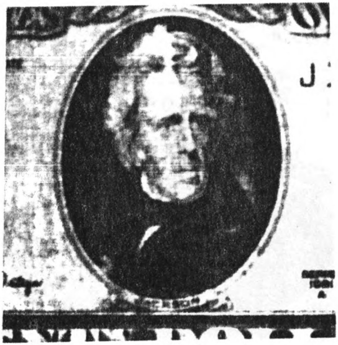


(a′)

Figure V.5 The reconstructed images from Hadamard transform coding at 0.74 bit/pixel.
(a), (b), (c) From transform coding.
(a′), (b′), (c′) From transform encryption coding.

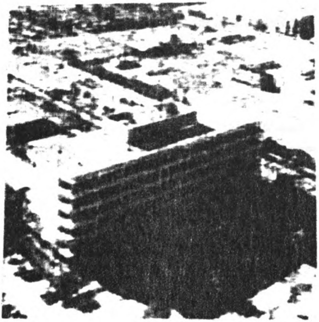


(b)

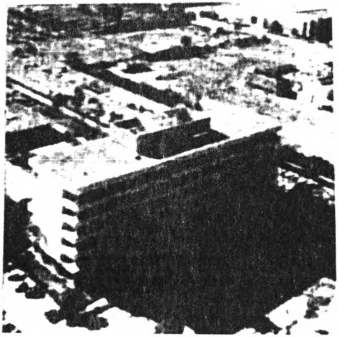


(b′)

Figure V.5 (cont'd.)



(c)



(c′)

Figure V.5 (cont'd.)

Table V.1 Simulation results of Hadamard transform coding.

Images		SNR @ nominal bits/pixel		
Fighter	A	296 @ 1.52	238 @ 1.09	185 @ 0.74
	B	777 @ 1.52	525 @ 1.09	356 @ 0.74
	C	794 @ 1.52	544 @ 1.09	361 @ 0.74
	D	788 @ 1.52	541 @ 1.09	360 @ 0.74
Jackson	A	282 @ 1.52	210 @ 1.09	161 @ 0.74
	B	531 @ 1.52	332 @ 1.09	223 @ 0.74
	C	495 @ 1.52	329 @ 1.09	219 @ 0.74
	D	502 @ 1.52	315 @ 1.09	221 @ 0.74
Building	A	120 @ 1.52	86.8 @ 1.09	62.4 @ 0.74
	B	189 @ 1.52	122 @ 1.09	83.0 @ 0.74
	C	192 @ 1.52	123 @ 1.09	83.9 @ 0.74
	D	191 @ 1.52	121 @ 1.09	84.0 @ 0.74

A. Reconstructed images obtained by transform coding.

B. Reconstructed images obtained by transform encryption coding. 255×255 quasi m-arrays are used as the reference images. The size of images is also reduced to 255×255.

C. Reconstructed images obtained by transform encryption coding. The 256×256 version of 255×255 quasi m-arrays with boundary filled with zero are used as the reference images.

D. Reconstructed images obtained by transform encryption coding. The 256×256 binary phase spectra of quasi m-arrays are used to encrypt the images.

$$\mathrm{SNR} \equiv \frac{\sum\limits_{i,j} f(i,j)^2}{\sum\limits_{i,j} [f(i,j)-\hat{f}(i,j)]^2}, \tag{V.1}$$

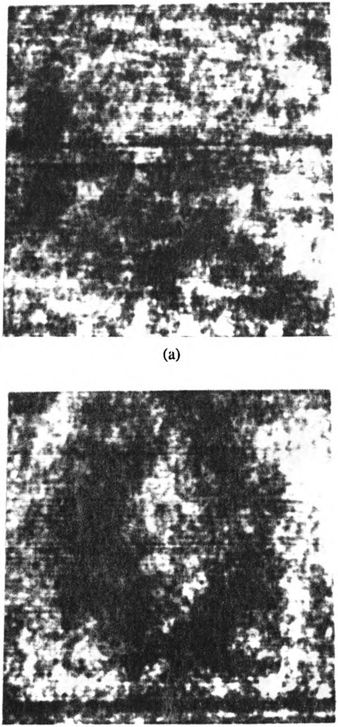
where f(i,j) and $\hat{f}$(i,j) are the original and reconstructed image respectively. The simulation results show a great improvement in the SNR at different bits/pixel. The SNR of Hadamard transform encrypted image coding is about 1.34 - 2.66 times higher than that of the Hadamard transform image coding. This improvement depends on the types of image and the bits/pixel used. Although 1.09 bits/pixel is required by the conventional Hadamard transform image coding to have an acceptably reconstructed image, the same quality image can be obtained by the Hadamard transform encrypted image coding at only 0.74 bit/pixel.

Objective measure is not always suitable for the evaluation of image quality. Therefore, a subjective paired-comparison method [8] is also conducted. The paired-comparison method involves the showing of two images to observers at a time and asking them to express a preference. The image pairs used in this measure are shown in Figure V.3, V.4, and V.5. Twenty observers were asked to do the comparison. All the observers preferred the results of transform encrypted image coding to those of transform image coding. Although the quality of reconstructed images does improve from transform image coding to transform encrypted image coding, there are some discrepancies. The improvement for the high contrast image such as Fighter is obvious, while, the improvement for the other type of image is not.

An explanation for the improvement of transform encrypted image coding for the high contrast image is as follows. For the high contrast image, the image data are less uniform. Therefore, the variance of transform coefficients is large. Sometimes, the distribution of these transform coefficients can not be approximated well by a Gaussian density function. Therefore, as expected, the mean-square quantization error is large. In addition, blocking effect at the region of high contrast is very obvious. On the contrary, the encrypted image seems more uniform than the original image. Therefore, blocking effect is not so serious as in transform coding. In addition, the decryption process will spread the quantization error over the whole image. Therefore, the reconstructed image from transform encryption coding are more acceptable.

A correct phase spectrum of reference function must be used to decrypt the encrypted image. If a wrong phase spectrum is used, the original image can not be recovered. Figure V.6 shows the results of this case. Therefore, a correct phase spectrum must be given to recover the original image, otherwise, a search for the correct phase spectrum is necessary. Since there are so many quasi m-arrays and Gold code arrays available, the search for the phase spectrum of a correct array is time-consuming. Hence the security of information is achieved because unauthorized persons do not have the knowledge of the phase spectrum of the reference function and it is too difficult to find this information.

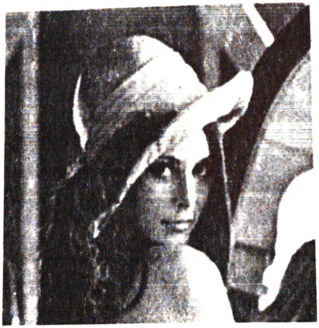
Lenna and Mandrill are used for the study of cosine transform coding because their approximation errors for Gaussian distribution are the largest among the test images. Figures V.7, V.8, and V.9 show the results of cosine transform coding when three different bit allocation maps shown in Figure II.1 are used. Table V.2 shows the SNR at 0.5, 1, and 1.5 bits/pixel, respectively. The SNR of cosine transform encrypted image coding is about 1.12-1.97 times better than that of cosine transform image coding. It is evident that the results of cosine transform encryption coding are better than those of cosine transform coding when the same number of bits/pixel is used. In addition, the blocking effect is removed and the results are more acceptable. The subjective paired-comparison method also shows the results of transform encrypted image coding is better than those of transform image coding. In general, the results of transform encryption coding at 0.5 bit/pixel have the same good quality as those of transform coding. Although adaptive transform coding can also achieve about 0.5 bit/pixel, it usually requires a significant amount of calculation and overhead



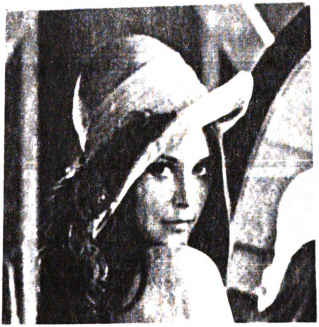
(a)

(b)

Figure V.6 The decrypted images according to the wrong reference functions.
(a) The decrypted Fighter.
(b) The decrypted Jackson.



(a)

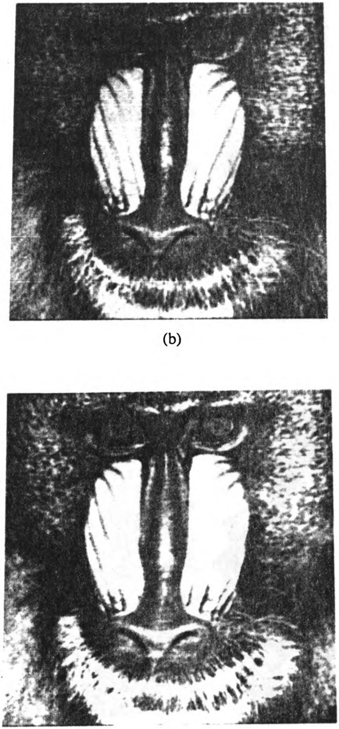


(a′)

Figure V.7 The reconstructed images from cosine transform coding at 1.5 bits/pixel.

(a), (b) From transform coding.

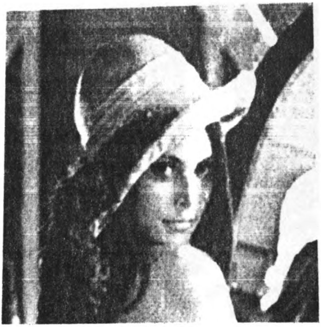
(a′), (b′) From transform encryption coding.



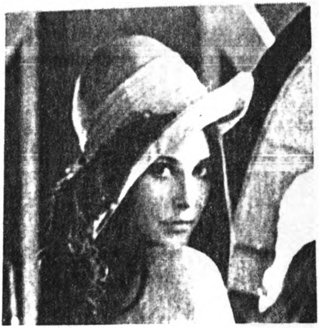
(b)

(b′)

Figure V.7 (cont'd.)

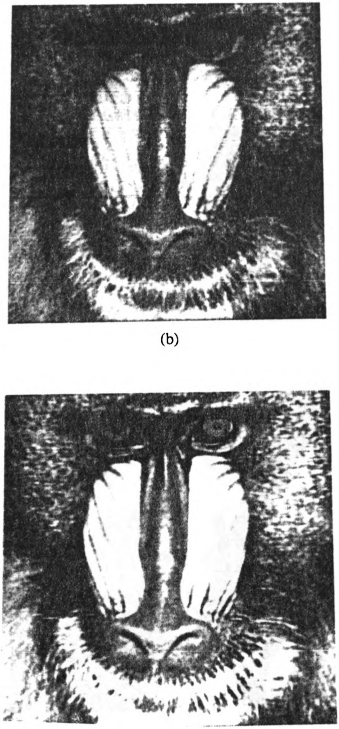


(a)



(a′)

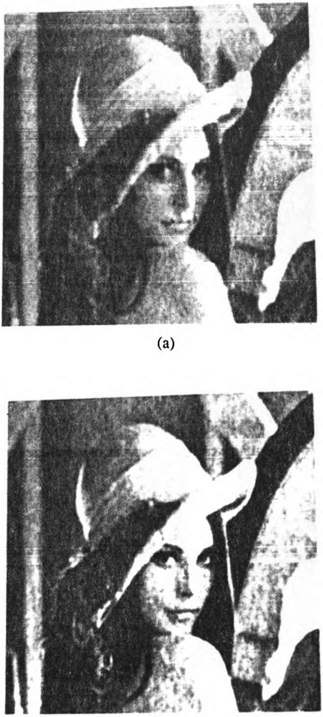
Figure V.8 The reconstructed images from cosine transform coding at 1.0 bit/pixel.
(a), (b) From transform coding.
(a′), (b′) From transform encryption coding.



(b)

(b′)

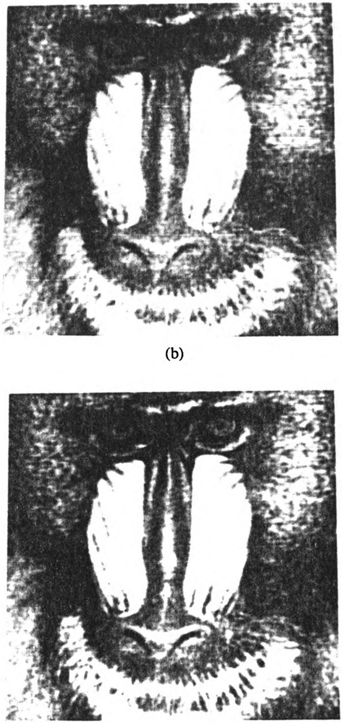
Figure V.8 (cont'd.)



(a)

(a′)

Figure V.9 The reconstructed images from cosine transform coding at 0.5 bit/pixel.
(a), (b) From transform coding.
(a′), (b′) From transform encryption coding.



(b)

(b′)

Figure V.9 (cont'd.)

Table V.2 Simulation results of cosine transform coding.

Images		SNR @ bits/pixel		
Lenna	A	397 @ 1.50	270 @ 1.00	182 @ 0.50
	B	730 @ 1.50	531 @ 1.00	280 @ 0.50
Mandrill	A	117 @ 1.50	95.7 @ 1.00	67.6 @ 0.50
	B	134 @ 1.50	118 @ 1.00	75.9 @ 0.50

A. Reconstructed images obtained by transform coding.

B. Reconstructed images obtained by transform encryption coding. The 256×256 binary phase spectra of quasi m-arrays are used to encrypt the images.

information. With the technique of transform encryption coding, 0.5 bit/pixel can be easily obtained without these drawbacks. In addition, this technique is compatible with all the other techniques of transform coding. Therefore, an image transmitted or storaged at less than 0.5 bit/pixel can be obtained when transform encryption coding is employed with the other techniques.

In summary, the transform encrypted image coding outperforms the transform image coding according to objective and subjective criteria. 0.5 bit/pixel can be expected by transform encryption coding according to the simulation results shown above, while 1 bit/pixel is necessary by transform coding. The only drawback is the encryption process which requires at least two N×N FFTs. Since FFT can be calculated easily and quickly by today's technology, this will not be a problem in practice. In addition, transform encrypted image coding also offers much higher security than transform image coding.

CHAPTER VI

CONCLUSION

VI.1 Limitations and Advantages

An independence technique for m-dependent random variables is first developed in this dissertation. The resulting independent random variables are also shown to follow the Gaussian distribution. The application of this technique to the transform image coding is also investigated. Since the transform coefficients are independent Gaussian random variables, a simple Max quantizer can achieve the optimal quantization. In addition, the redundancies in the transform coefficients are also removed. This technique can be directly applied to any image or speech signal because these signals have the m-dependent property. Although the cryptography is required by the proposed independence technique, the encrypted signal can be obtained easily and quickly by the algorithm developed in this dissertation. In addition, transform encryption coding has the higher security than transform coding. The simulation results show a great improvement in the coding efficiency. Satisfactorily reconstructed image from the cosine/Hadamard transform encrypted image coding at 0.5 bit/pixel were obtained. However, the similar results were obtained by cosine/Hadamard transform image coding at 1 bit/pixel. The results of transform encrypted image coding outperform those of transform image coding under the objective and subjective criteria. Higher coding efficiency can be expected if K-L transform is used. In addition, the technique of transform encryption coding is compatible with the all other techniques of transform coding. Therefore, they can be employed together to produce a better results at less than 0.5 bit/pixel. In addition, the blocking effect is also removed by the decryption process. Although only the application of the independence technique to image signals is studied, similar results can also be expected in speech signals.

Two-dimensional quasi m-arrays and Gold code arrays are also studied in this dissertation. These arrays have the quasi-orthogonal properties which are useful for many applications. The cyclic auto-correlation of any array is close to the delta-function. In addition, the cyclic cross-correlation between any two arrays is small compared with their cyclic auto-correlation peak. These arrays also have the pseudo-random property which is suitable for the reference function used in transform encryption coding. In addition, the number of these arrays available is so large that a successful attack by unauthorized persons is almost impossible. In summary, these arrays have the following advantages in applications.

1. They can be easily and quickly generated.
2. The size of these arrays is flexible.
3. These arrays are pseudo-random.
4. There are many of this type of array available.

The only limitation is that the cyclic auto-correlation floor of these arrays in not constant.

N-dimensional quasi m-hypercubes and Gold code hypercubes can also be generated by the same construction method proposed in this dissertation. A quasi-orthogonal property of these hypercubes is also expected. These n-dimensional hypercubes are much easier to generate than the n-dimensional Welti codes [56]. In addition, these hypercubes can also be used in the n-dimensional transform encryption coding.

VI.2 Recommendations for Further Work

Transform coding has been studied for more than two decades, and many techniques have been proposed [1]-[10]. All these techniques can be employed with the transform encryption coding. The following is a summary of some important directions for further work.

1. Image signals share many similarities with speech signals. Therefore, similar results from transform encrypted speech coding can also be expected. The problems is how good the transform encrypted speech coding can be under the objective and subjective criteria. This is an interesting problem for researchers in speech coding to study.

2. There are many transformations which can be used for transform coding. The cosine transform is the best approximation to K-L transform when the image model is a first order Markov process [7]-[10]. Since the encrypted signals are always joint Gaussian, a comparison of these transformations based on the joint Gaussian images is necessary.

3. There are many techniques in transform coding such as visual, thresholded, adaptive, and hybrid transform coding [1]-[10], [57]. These techniques can always be employed with the transform encrypted coding. How good the coding performance can be when these techniques are merged is also an interesting problem.

4. Noise is always presented during the information transmission and storage. Therefore, a study of transform encryption coding under the presence of noise is also necessary.

5. The optimal bit allocation for independent random variables proposed by Segall are under the assumption that the bit allocation can be any real number [29]. This is an incorrect assumption because the quantization levels are always integers

and then the bit allocation cannot be any real number. How to solve this optimal bit allocation problem under the assumption of integer quantization levels is also essential.

6. The independence theorem holds only when the K-L transform is used. However, an efficient computation algorithm for K-L transform does not yet exist [1]-[10], [51]. Although the performance of many unitary mathematical transformations is close to that of K-L, they are not exactly the same. Therefore, an efficient computation algorithm for K-L transform must be investigated.

The two-dimensional pseudo-noises developed in this dissertation are useful for many applications. Some possible topics for further work are the following.

1. The m-sequences or m-arrays can be represented by the primitive polynomial under the finite field [15], [16]. How can one represent the quasi m-arrays and the Gold code arrays by the primitive polynomial? What other properties do they have? These are some interesting issues in understanding the properties of two-dimensional finite fields and primitive polynomials.

2. Artificial even-odd textures (random patterns, pseudo-noises) are useful for the study of texture segmentation algorithms [49]. Some even-odd textures can be discriminated by these algorithms, but some cannot [57], [58]. What is the reason? Is it because of the correlation between the even and odd texture? The quasi m-arrays are indeed even textures. Is there any odd texture with quasi-orthogonal property? If so, can this type of even-odd texture be discriminated? This might be an intesting problem for researchers in computer vision.

APPENDIX

OPTICAL ENCRYPTION

Two dimensional convolution and correlation can be easily performed by optics because of the Fourier transform ability of converging lens. [59]-[62]. In addition, an optical system processes the information at the speed of light which is impossible with today's electronic technology. A typical optical information processing system is shown in Figure A.1. The first lens is to take the Fourier transform of the input object. The inverse Fourier transform of the filtered output at the filter plane is then taken by the second lens and displayed at the output plane. Although two-dimensional convolution and correlation can be performed by a match (VanderLught) filter [59]-[62], the filter synthesis is difficult and time-consuming. In addition, the light efficiency of matched filter is low. Real-time devices that can modulate the phase or amplitude of light are already available [63]-[65]. Two-dimensional convolution and correlation can be performed by these devices. In addition, the light efficiency of these devices is also better than that of the matched filter. Most devices of this type are controlled by a digital computer. Therefore, filters can be stored in the computer and filter switching is easy. An optical system with such real-time devices is called an electro-optics system. This system is important because of its programmability and high light efficiency.

An image formation system usually consists of lenses and recording devices. If the images obtained by such system are to be used in digital transmission or storage, then the quantization is necessary. To have a high quantization efficiency, the encryption must be performed on these images to produce the joint Gaussian encrypted

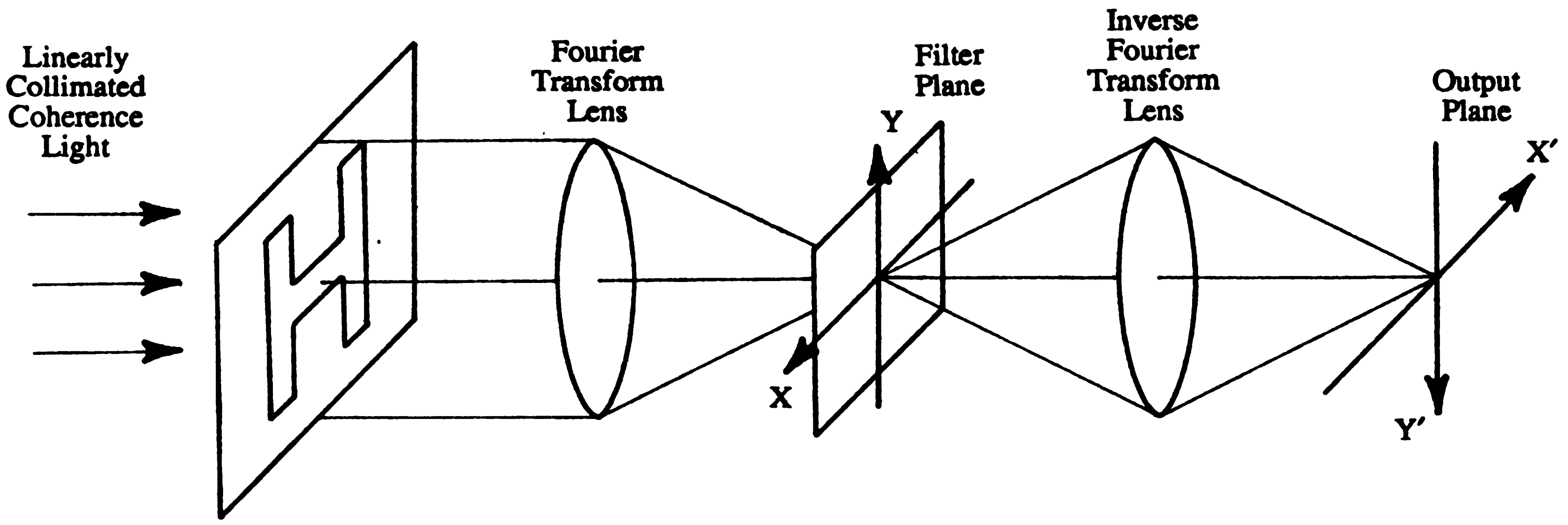


Figure A.1 An optical system for information processing.

image. Since the encryption is obtained by convolution, it can also be obtained by an optical or electro-optical system. Obtaining the encrypted image by optics is a better way than by electronics because all that is necessary is to modify the image formation system. In addition, the filter used for encryption can be changed easily and quickly by computer.

Figure A.2 shows a proposed system for an encrypted image formation system. This system is important because the encrypted images are directly obtainable by the recording device and ready for transform coding. The liquid crystal light valve is used to convert an incoherent optical image into a coherent one which is suitable as an input for any coherent optical information processing system. This conversion is necessary because the Fourier transform property of converging lens holds only when a coherent (laser) light source is used. The lens takes the Fourier transform of the input optical wave, and displays the results at the output plane. Therefore, it is the output optical wave (amplitude and phase) that is equivalent to the result of Fourier transform. The filter must be changed easily for security consideration. This can be done by using a programmable magneto-optics device. In addition, the requirement of an all-pass filter is also achieved because the magneto-optics device can be set to modulate the phase of light only. The output of this system can be recorded directly by the CCD. Although the amplitude of the optical wave for an image after the liquid crystal light valve is always positive, that for the encrypted image can be negative. In other words, there can be a 180° phase difference between the optical wave of the original image and the encrypted one. However, CCD can only detect the light intensity and cannot distinguish the phase difference in the optical wave. This problem can be solved by interferometry [59] shown in Figure A.2. CCD always records the light intensity of an encrypted image twice. The first time is with the shutter closed which will record the light intensity of the encrypted image only. Therefore, the amplitude of the optical wave for an encrypted image can be obtained by the square root of the

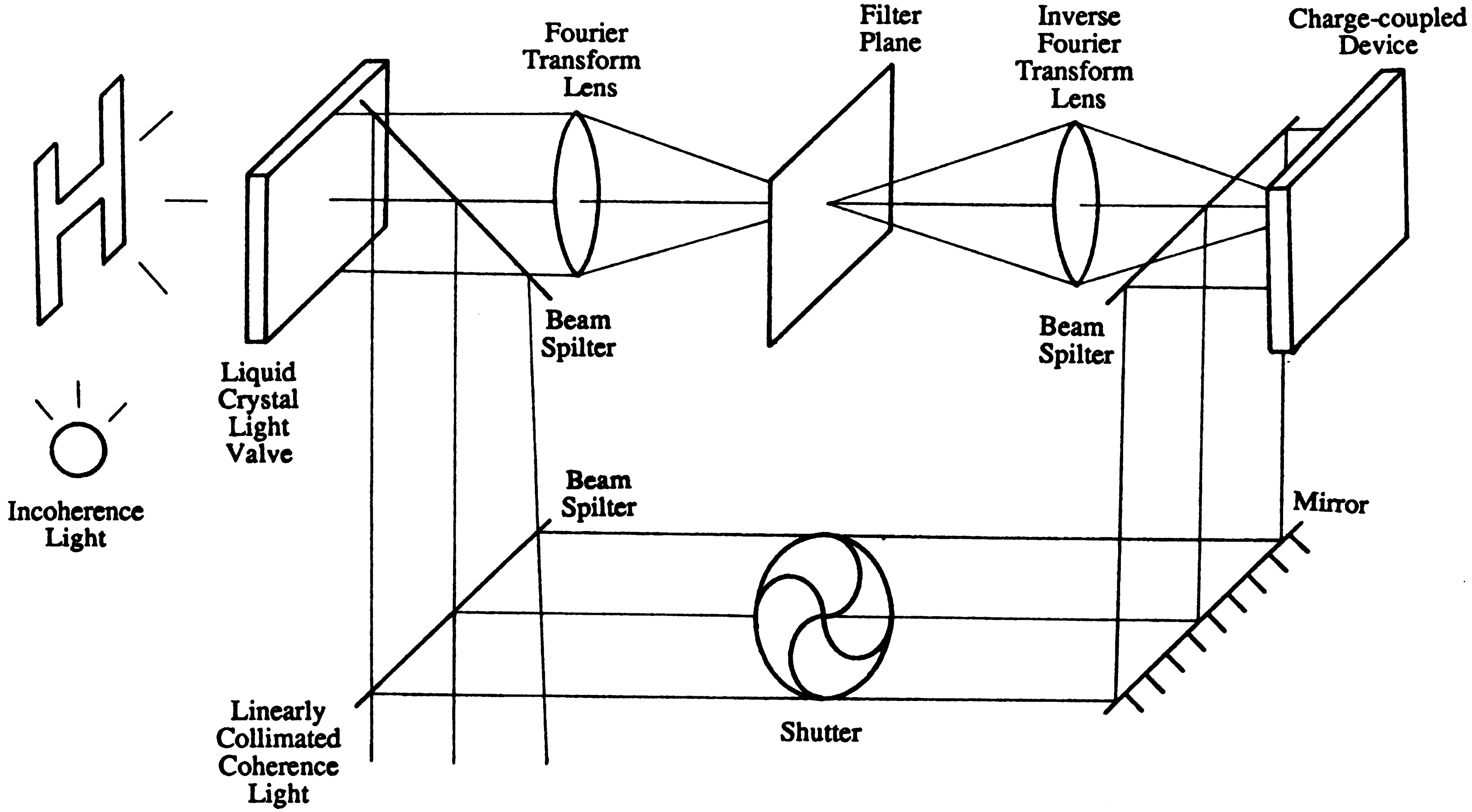
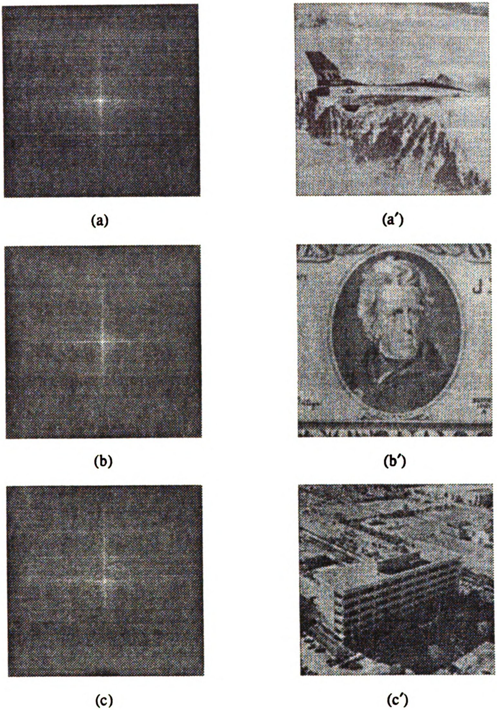


Figure A.2 An optical system for encrypted image formation.

CCD output. The second time is with the shutter open which will record the interference between the optical wave of the encrypted image and the reference optical wave. Since the difference between the two optical path lengths is set to be smaller than the coherence length of the laser, there will be a constructive or destructive interference at the output plane. The constructive interference corresponds to the no phase difference condition between the optical waves, while the destructive one corresponds to a 180° phase difference. Therefore, the 0° or 180° phase shift in the optical wave of the encrypted image can be determined by the output of CCD at this time.

Usually, only linear convolution is performed by an optical system. This will degrade the joint Gaussian property of the encrypted images. In addition, it also degrades the performance of transform encryption coding. All of these are because the definition of encryption is not fully satisfied. In addition, linear convolution also doubles the space-bandwidth product of the encrypted images. A simple simulation is used to see the effect above. The original images are Fighter, Jackson, and Building shown in Figure V.1. The original size of the encrypted images is 512×512 because the space-bandwidth product is doubled by the linear convolution. The size of these encrypted images are reduced to 256×256 to have compatible resolution with the original 256×256 images. These 256×256 encrypted images are then decrypted by an optical linear deconvolution. Figure A.3 shows the simulation results. The SNR of simulation results for Fighter, Jackson, and Building, are 350, 246, and 89, respectively. The noise occurs because the resolution of the encrypted image is reduced from 512×512 to 256×256.

To obtain better results, cyclic convolution must be implemented in optics. Therefore, the space-bandwidth product of the encrypted image will not be doubled. In addition, the joint Gaussian property of the encrypted signals can be maintained. Indeed, cyclic convolution can be implemented by a linear convolution. This is done as follows. Let us assume there are two signals of length N. Then the cyclic



(a)

(a′)

(b)

(b′)

(c)

(c′)

Figure A.3 Optical encrypted and decrypted images by linear convolution.
(a), (b), (c) Encrypted images.
(a′), (b′), (c′) Decrypted images.

convolution of these two signals can be obtained by the linear convolution of one signal with a periodic version of the other. The *periodic* signal consists of two identical copies of the original sequence. Therefore, the total length of the periodic signal is 2N in this case. Figure A.4 shows the method for obtaining cyclic convolution using linear convolution. The total length of the linear convolution output is 3N, while only the middle output with length N is the result of cyclic convolution.

Although cyclic convolution can be obtained by linear convolution, cyclic deconvolution by linear deconvolution is not yet available. This is an interesting problem in optical information processing and deserves further attention.

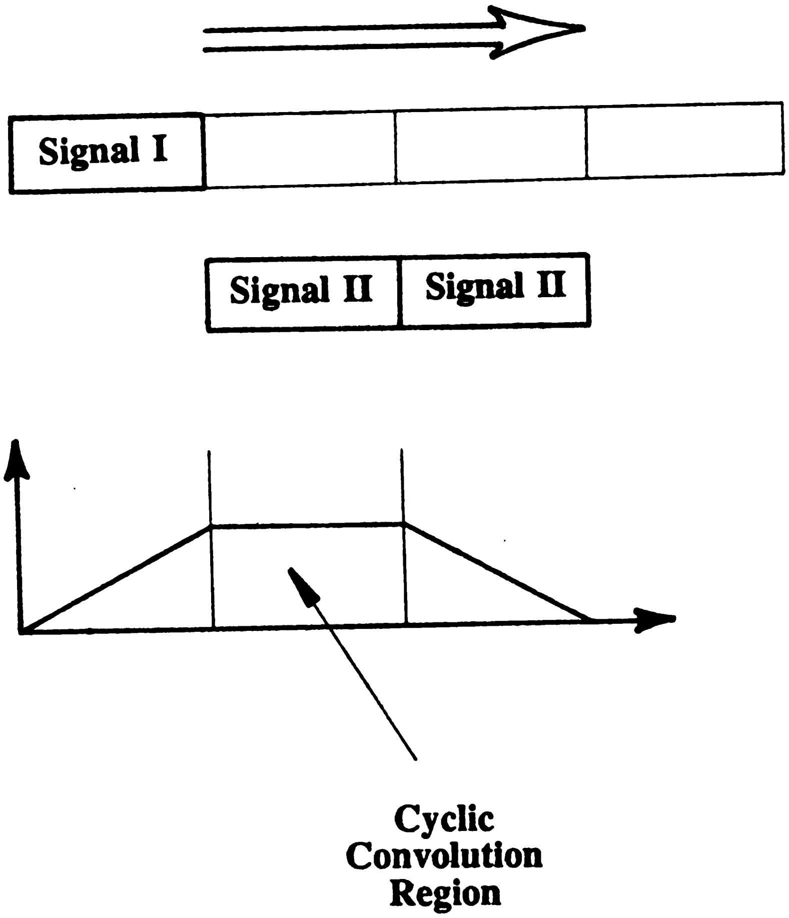


Figure A.4 The method of obtaining cyclic convolution using linear convolution.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] A.G. Tescher & J.A. Saghri, "Advances in transform image coding," *Optical Engineering*, vol. 26, pp. 563-569, July 1987.

[2] A.G. Tescher & J.A. Saghri, "Adaptive transform coding and image quality," *Optical Engineering*, vol. 25, pp. 979-983, August 1986.

[3] H.G. Musmann, P. Pirsch, & H.J. Grallert, "Advances in Picture coding," *Proceedings of the IEEE*, vol. 73, pp. 523-548, April 1985.

[4] M. Kunt, A. Ikonomopoulos, & M Kocher, "Second-generation image-coding techniques," *Proceedings of the IEEE*, vol. 73, pp. 549-574, April 1985.

[5] A.K. Jain, "Image data compression: a review," *Proceedings of the IEEE*, vol. 69, pp. 349-389, March 1981.

[6] A.N. Netravali & J.O. Limb, "Picture coding: a review," *Proceedings of the IEEE*, vol. 68, pp. 366-406, March 1980.

[7] J.S. Lim, *Two-dimensional Signal and Image Processing*, Englewood Cliffs: Prentice Hall, 1990.

[8] R.C. Gonzalez & P. Wintz, *Digital Image Processing*, Reading: Addison-Wesley, 1987.

[9] R.J. Clarke, *Transform Coding of Images*, London: Academic Press, 1985.

[10] W.K. Pratt, *Digital Image Processing*, New York: John Willey & Sons, 1978.

[11] V.R. Algam & D.J. Sakrison, "On the optimality of the Karhunen-Loeve expansion," *IEEE Transactions on Information Theory*, vol. IT-15, pp. 319-320, March 1969.

[12] G.L. Turin, "On the optimum energy distribution in sequential detection," *IEEE Transactions on Information Theory*, vol. IT-15, pp. 321, March 1969.

[13] N.M. Nasrabadi & R.A. King, "Image coding using vector quantization: a review," *IEEE Transactions on Communications*, vol. COM-36, pp. 957-971, August 1988.

[14] H.C. Tseng & T.R. Fischer, "Transform & hybrid transform/DPCM coding of images using pyramid vector quantization," *IEEE Transactions on Communications*, vol. COM-35, pp. 79-86, January 1987.

[15] J.G. Proakis, *Digital Communications*, New York: McGraw Hill, 1989.

[16] R.J. McEliece, *Finite Fields for Computer Scientists and Engineers*, Boston: Kluwer Academic, 1987.

[17] R. Skaug & J.F. Hjelmstad, *Spread Spectrum in Communication*, London: Peter Peregrinus, 1985.

[18] R.C. Dixon, *Spread Spectrum Systems*, New York: John Willey & Sons, 1984.

[19] D.V. Sarwate & M.B. Pursley, "Cross-correlation properties of pseudorandom and related sequences: a review," *Proceedings of the IEEE*, vol. 68, pp. 593-619, May 1980.

[20] T. Nomura, H. Miyakawa, H. Imai & A. Fukuda, "A theory of two-dimensional linear recurring arrays," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 775-785, November 1972.

[21] F.J. MacWilliams & N.J.A. Sloane, "Pseudo-random sequences and arrays," *Proceedings of the IEEE*, vol. 64, pp. 1715-1729, December 1976.

[22] N. Ohyama, T. Honda, & J. Tsujiuchi, "An advanced coded imaging without sidelobes," *Optics Communications*, vol. 27, pp. 339-344, December 1978.

[23] W. Szepanski, "Compatibility problems in add-on data transmission for TV channels," *Proceedings of the Second Symposium on Electromagnetic Compatibility*, pp. 263-268, June 1977.

[24] S.W. Golomb & H. Taylor, "Two-dimensional synchronization patterns for minimum ambiguity," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 600-604, July 1982.

[25] C.J. Kuo & H. Rigas, "Image multiplexing by code division technique," *SPIE Proceeding*, vol. 1153, August 1989.

[26] T.R.N. Rao & K.H. Nam, "Private-key algebraic-code encryptions," *IEEE Transactions on Information Theory*, vol. 35, pp. 829-833, July 1989.

[27] W. Diffie & M.E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644-654, November 1976.

[28] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. IT-6, pp. 7-12, March 1960.

[29] A. Segall, "Bit allocation and encoding for vector sources," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 162-169, March 1976.

[30] L.D. Davisson, "Rate-distortion theory and application," *Proceedings of the IEEE*, vol. 60, pp. 800-808, July 1972.

[31] P.A. Wintz & A.J. Kurtenbach, "Waveform error control in PCM telemetry," *IEEE Transactions on Information Theory*, vol. IT-14, pp. 650-661, September 1968.

[32] J.J.Y. Huang & P.M. Schultheiss, "Block quantization of correlated Gaussian random variables," *IEEE Transactions on Communications Systems*, vol. CS-11, pp. 289-296, September 1963.

[33] H.P. Kramer & M.V. Mathews, "A linear coding for transmitting a set of correlated signals," *IRE Transactions on Information Theory*, vol. IT-2, pp. 41-45, September 1956.

[34] S. Lin & D.J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs: Prentice-Hall, 1983.

[35] C. Stein, "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables," *Sixth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 2, pp. 583-602, 1972.

[36] L. Heinrich, *A Method for the Derivation of Limit Theorems for Some Classes of Dependent Random Variables*, Berlin: Akademie Der Wissenschaften Der DDR, 1983.

[37] K.N. Berk, "A central limit theorem for m-dependent random variables with unbounded m," *Annual Probability*, vol. 1, pp. 352-354, April 1973.

[38] M. Resenblatt, "A central limit theorem and a strong mixing condition," *Proceeding of National Academy of Sciences*, vol. 42, pp. 43-47, 1956.

[39] W. Hoeffding & H. Robbins, "The central limit theorem for dependent random variables," *Duke Mathematical Journal*, vol. 15, pp. 773-780, 1948.

[40] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York: McGraw Hill, 1984.

[41] J.M. Wozencraft & I.M. Jacobs, *Principles of Communication Engineering*, New York: John Wiley & Sons, 1965.

[42] A.V. Oppenheim & J.S. Lim, "The importance of phase in signals," *Proceedings of the IEEE*, vol. 69, pp. 529-541, May 1981.

[43] A.B. Carlson, *Communication Systems*, New York: McGraw Hill, 1986.

[44] A.N. Netravali & B. Prasada, "Adaptive quantization of picture signals using spatial masking," *Proceedings of the IEEE*, vol. 65, pp. 536-548, April 1977.

[45] J.L. Mannos & D.J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 525-536, July 1974.

[46] Z.L. Budrikis, "Visual fidelity criterion and modeling," *Proceedings of the IEEE*, vol. 60, pp. 771-779, July 1972.

[47] J.L. Horner & H.O. Barteit, "Two-bit correlation," *Applied Optics*, vol. 24, pp. 2889-2893, September 1985.

[48] S.Y. Kung, R.E. Owen, & J.G. Nash, *VLSI Signal Processing, II*, New York: IEEE Publication, 1986.

[49] A. Rosenfeld & A.C. Kak, *Digital Picture Processing vols. 1 & 2*, New York: Academic Press, 1982.

[50] B. Julesz & I.R. Bergen, "Textons, the fundamental elements in preattentive vision and perception of textures," *The Bell System Technical Journal*, vol. 62, pp. 1619-1645, July-August 1983.

[51] W.D. Ray & R.M. Driver, "Further decomposition of the Karhunen-Loeve series representation of a stationary random process," *IEEE Transactions on Information Theory*, vol. IT-16, pp. 663-668, November 1970.

[52] K.G. Beauchamp, *Application of Walsh and Related Functions*, London: Academic Press, 1984.

[53] R. Lucky, Distinguished lecture, Michigan State University, April 14, 1989.

[54] A.V. Oppenheim & R.W. Schafer, *Digital Signal Processing*, Englewood Cliffs: Prestice Hall, 1975.

[55] J.S. Lim & A.V. Oppenheim, *Advanced Topics in Signal Processing*, Englewood Cliffs: Prentice Hall, 1988.

[56] H.D. Luke, "Sets of one and higher dimensional Welti codes and complementary codes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-21, pp. 170-179, March 1985.

[57] B. Julesz, E.N. Gilbert, & J.D. Victor, "Visual discrimination of textures with identical third-order statistics," *Biological Cybernetics*, vol. 31, pp. 137-140, 1978.

[58] F. Farrokhnia, private communications, 1989.

[59] E. Hecht, *Optics*, Reading: Addison-Wesley, 1987.

[60] J.L. Horner, *Optical Signal Processing*, San Diego: Academic Press, 1987.

[61] F.T.S. Yu, *Optical Information Processing*, New York: John Willey & Sons, 1983.

[62] J.D. Gaskill, *Linear Systems, Fourier Transforms, and Optics*, New York: John Willey & Sons, 1978.

[63] A.D. Fisher, L.C. Ling, J.N. Lee, & R.C. Fukuda, "Photoemitter membrane light modulator," *Optical Engineering*, vol. 25, pp. 261-268, Feburary 1986.

[64] W.E. Rose, D. Psaltis, & R.H. Anderson, "Two-dimensional magneto-optic spatial light modulator for signal processing," *Optical Engineering*, vol. 22, pp. 485-490, July-August 1983.

[65] D.R. Pape & L.J. Hornbeck, "Characteristics of the deformable mirror device for optical information processing," *Optical Engineering*, vol. 22, pp. 675-681, November-December 1983.