

**EFFICIENT SIMULTANEOUS ENCRYPTION AND  
COMPRESSION OF DIGITAL VIDEOS IN  
COMPUTATIONALLY CONSTRAINED APPLICATIONS**

**By  
Nazar Al-hayani**

**Applied Computing Department, the University of Buckingham  
United Kingdom**

**A THESIS Submitted for the degree of Doctor of Philosophy in Secure  
Video Transmission To the school of Science and Medicine  
In the University of Buckingham**

**January 2015**

# ABSTRACT

This thesis is concerned with the secure video transmission over open and wireless network channels. This would facilitate adequate interaction in computationally constrained applications among trusted entities such as in disaster/conflict zones, secure airborne transmission of videos for intelligence/security or surveillance purposes, and secure video communication for law enforcing agencies in crime fighting or in proactive forensics. Video content is generally too large and vulnerable to eavesdropping when transmitted over open network channels so that compression and encryption become very essential for storage and/or transmission. In terms of security, wireless channels, are more vulnerable than other kinds of mediums to a variety of attacks and eavesdropping. Since wireless communication is the main mode in the above applications, protecting video transmissions from unauthorized access through such network channels is a must. The main and multi-faceted challenges that one faces in implementing such a task are related to competing, and to some extent conflicting, requirements of a number of standard control factors relating to the constrained bandwidth, reasonably high image quality at the receiving end, the execution time, and robustness against security attacks. Applying both compression and encryption techniques simultaneously is a very tough challenge due to the fact that we need to optimize the compression ratio, time complexity, security and the quality simultaneously.

There are different available image/video compression schemes that provide reasonable compression while attempting to maintain image quality, such as JPEG, MPEG and JPEG2000. The main approach to video compression is based on detecting and removing spatial correlation within the video frames as well as temporal correlations across the video frames. Temporal correlations are expected to be more evident across sequences of frames captured within a short period of time (often a fraction of a second). Correlation can be measured in terms of similarity between blocks of pixels. Frequency domain transforms such as the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT) have both been used restructure the frequency content (coefficients) to become amenable for efficient detection. JPEG and MPEG use DCT while JPEG2000 uses DWT. Removing spatial/temporal correlation encodes only one block from each class of equivalent (i.e. similar) blocks and remembering the position of all other block within the equivalence class. JPEG2000 compressed images

achieve higher image quality than JPEG for the same compression ratios, while DCT based coding suffer from noticeable distortion at high compression ratio but when applied to any block it is easy to isolate the significant coefficients from the non-significant ones.

Efficient video encryption in computationally constrained applications is another challenge on its own. It has long been recognised that selective encryption is the only viable approach to deal with the overwhelming file size. Selection can be made in the spatial or frequency domain. Efficiency of simultaneous compression and encryption is a good reason for us to apply selective encryption in the frequency domain.

In this thesis we develop a hybrid of DWT and DCT for improved image/video compression in terms of image quality, compression ratio, bandwidth, and efficiency. We shall also investigate other techniques that have similar properties to the DCT in terms of representation of significant wavelet coefficients. The statistical properties of wavelet transform high frequency sub-bands provide one such approach, and we also propose phase sensing as another alternative but very efficient scheme.

Simultaneous compression and encryption, in our investigations, were aimed at finding the best way of applying these two tasks in parallel by selecting some wavelet sub-bands for encryptions and applying compression on the other sub-bands. Since most spatial/temporal correlation appear in the high frequency wavelet sub-bands and the LL sub-bands of wavelet transformed images approximate the original images then we select the LL-sub-band data for encryption and the non-LL high frequency sub-band coefficients for compression. We also follow the common practice of using stream ciphers to meet efficiency requirements of real-time transmission. For key stream generation we investigated a number of schemes and the ultimate choice will depend on robustness to attacks.

The still image (i.e. RF's) are compressed with a modified EZW wavelet scheme by applying the DCT on the blocks of the wavelet sub-bands, selecting appropriate thresholds for determining significance of coefficients, and encrypting the EZW thresholds only with a simple 10-bit LFSR cipher This scheme is reasonably efficient in terms of processing time, compression ratio, image quality, as well was security robustness against statistical and frequency attack. However, many areas for improvements were identified as necessary to achieve the objectives of the thesis.

Through a process of refinement we developed and tested 3 different secure efficient video compression schemes, whereby at each step we improve the performance of the scheme in the previous step. Extensive experiments are conducted to test performance of the new scheme, at each refined stage, in terms of efficiency, compression ratio, image quality, and security robustness.

Depending on the aspects of compression that needs improvement at each refinement step, we replaced the previous block coding scheme with a more appropriate one from among the 3 above mentioned schemes (i.e. DCT, Edge sensing and phase sensing) for the reference frames or the non-reference ones. In subsequent refinement steps we apply encryption to a slightly expanded LL-sub-band using successively more secure stream ciphers, but with different approaches to key stream generation. In the first refinement step, encryption utilized two LFSRs seeded with three secret keys to scramble the significant wavelet LL-coefficients multiple times. In the second approach, the encryption algorithm utilises LFSR to scramble the wavelet coefficients of the edges extracted from the low frequency sub-band. These edges are mapped from the high frequency sub-bands using different threshold. Finally, use a version of the A5 cipher combined with chaotic logistic map to encrypt the significant parameters of the LL sub-band.

Our empirical results show that the refinement process achieves the ultimate objectives of the thesis, i.e. efficient secure video compression scheme that is scalable in terms of the frame size at about 100 fps and satisfying the following features; high compression, reasonable quality, and resistance to the statistical, frequency and the brute force attack with low computational processing. Although image quality fluctuates depending on video complexity, in the conclusion we recommend an adaptive implementation of our scheme.

Although this thesis does not deal with transmission tasks but the efficiency achieved in terms of video encryption and compression time as well as in compression ratios will be sufficient for real-time secure transmission of video using commercially available mobile computing devices.

*Dedicated to*

*My parent's soul and my family*

## **ACKNOWLEDGMENTS**

First and foremost, all thanks to Allah the Almighty for blessing me with the strength to achieve this goal.

I would like to thank my wife for her patience and endless love. Together we have been sharing each and every detail in the life. Her support and encouragement was in the end what made this thesis possible and for her I dedicate this work. Thank you my wife for colouring my life with great memories, and for being my second half, my soul mate, and my best friend.

I would like to dedicate this thesis and all the good things in my life to my sons, Mustafa, Yasser and Weasaam, who have been my continuous source of hope and determination to go on despite the difficult times I have been away from them. Their patience has often pushed me to put in my best efforts to be a better person. Also, many thank go to the rest of my family, brothers and sisters for their support, encouragement and praying during my study.

I cannot find words to express my sincere gratitude to my first supervisor Prof. Sabah Jassim for his guidance, understanding, patience, and editorial advice was essential to the completion of this thesis and has taught me innumerable lessons and insights on the workings of academic research in general, and most importantly, his friendship during these years.

I also gratefully acknowledge my second supervisor Dr. Naseer.Al-Jawad. Without his encouragement and patience I would not have been writing this thesis. His mentorship was essential in providing a well-rounded experience, excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research.

I would also like to thank my personal tutor (Dr. Harin Sellaheewa) for being a very good listener and for his continuous support and encouragement. I highly thank Mr. Ali Abu-Raghaif, Mahar Al-Aboodi and Suleyman Al-Showarah for unique insights. And also extend my thanks to Dr.Abdul Hassan N AL-dijalie of encouragement and predicative get to complete my studies.

I would like to express my thanks to the Ministry of Higher Education and Scientific Research (MHESR) in Iraq for sponsoring my PhD program of study.

Last but not least, many thanks are due to all my friends, fellow students, and all members of staff at the applied computing department, University of Buckingham for the fruitful collaboration work and for all their support and important discussions.

# ABBREVIATIONS

A5	Stream cipher
AES	Advanced Encryption Standard
ASC	Average Speed Computer
B-Frame	Bidirectional frames
bior2.2	BiorSplines2.2
bpp	Bit per pixel
CABAC	Context Adaptive Binary Arithmetic Coding
CAC	Chaotic Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CC	Correlation Coefficients
CF	Current Frame
CH5M	Method of Chapter 5
CH6M	Method of Chapter 6
CLM	Chaotic Logistic Map
CLT	Central Limit Theorem
coif2	Coiflets2
CR	Compression Ratio
CS	Compressed sensing
CVES	Chaotic Video Encryption Scheme
CVQ	Compressive Vector quantization
db1	Daubechies1
db4	Daubechies4
db9	Daubechies9
DCT	Discrete cosine Transform
DCTM	Transform matrix of DCT
DES	Data Encryption Standard
<i>Dist</i>	Distance
DWT	Discrete wavelet transform
EZW	Embedded Zero trees of Wavelet
FLV	Flash player

GSM	Global System for Mobile communications
H3	High frequency sub-band of level 3
HDTV	High-Definition Television
HE	Homomorphic Encryption
HEVC	High Efficiency Video Coding
HH	High-high frequency sub-band
HI	Histogram Intersection
HL	High -low frequency sub-band
HME	Hierarchical Motion Estimation
I-Frame	Intra-frame
ISO/IEC	International Standardization Organization/ International Electro-technical Commission
ITU-T	International Telecommunications Union
JDWCT-CVQ	Joint DWT, DCT and CVQ
JDWCT-CVQ-Edge	Joint DWT, DCT , CVQ and edge sensing
JDWCT-CVQ-Edges-phase sensing	Joint DWT, DCT, CVQ, Edges and phase sensing
JnCE	Joint Compression and Encryption
JPEG	Joint Picture Expert Group
JPSEC	Security for JPEG-2000
L1	Level 1
L2	Level 2
L3	Level 3
LFSR	Linear Feedback Shift Register
LH	low - High frequency sub-band
LL	low frequency sub-band
LL3	Low frequency sub-band of level 3
LM	Logistic Map
MDMF	Maximum Deviation Measuring Factor
MPEG	Motion Picture Expert Group
MEZW	Modified EZW
MEZWE	MEZW Encryption
MJ2	Motion JPEG2000

MQ-coder	binary arithmetic coder
MSE	Mean Square Error
n-RF	non-Reference Frame
P-Frame	Predictive frames
PMMLG	Modulo Multiplicative Linear Congruence Generators
PSNR	Peak Signal to Noise Ratio
PWLCM	Piece-Wise Linear Chaotic Maps
QMS	Quadrant Monotonic Search
RAS	Ronald Rivest, Adi Shamir and Leonard Adleman
rbio2.2	ReverseBior2.2
RD	Rate Distortion
RF	Reference Frame
RLE	Run-Length Encoding
RHSC	Relatively High Speed Computer
SC	Significance Coefficients
SC-LL3	Significance Coefficients of low frequency sub-band of level 3
SM	Sensing Matrix
SMa	Sine Map
SOC	Start of code-stream
SPIHT	Set Partitioning in Hierarchical Trees
SQ	Scalar Quantization
SqCE	Sequential Compression and Encryption
STD	Standard Deviation
sym2	Symlets2
THR	Threshold
VQ	Vector Quantization
WDR	Wavelet Difference Reduction
WT	Wavelet Transform
XOR	Exclusive Or

# TABLE OF CONTENTS

ABSTRACT .....	ii
ACKNOWLEDGMENTS .....	vi
ABBREVIATIONS .....	viii
TABLE OF CONTENTS.....	xi
List of Figures .....	xiv
List of Tables .....	xvii
Declaration.....	xviii
Chapter 1 Introduction .....	1
1.1 Introduction to Image and Video Compression .....	2
1.1.1 Still image compression techniques and tools .....	3
1.1.2 Video compression tools and standards .....	10
1.2 Image and Video Encryption .....	13
1.3 Thesis Objectives, Main Approaches, and Contributions .....	15
1.4 Thesis organisation .....	18
1.5 List of Publications .....	19
Chapter 2 Literature Review .....	20
2.1 Image and video compression techniques.....	20
2.1.1 Image compression .....	20
2.1.2 Video compression.....	27
2.2 Image and Video Encryption .....	34
2.2.1 Stream ciphers.....	35
2.2.2 Sequential Compression and Encryption (SqCE) .....	39
2.2.3 Joint Compression and Encryption (JnCE).....	39
2.2.4 Homomorphic Encryption.....	43
2.2.5 Security for JPEG-2000 (JPSEC) .....	43
2.3 An over view of thesis investigations and approach.....	46
Chapter 3 Analysis of Still image Compression and Encryption.....	47
3.1 Image compression using wavelet transform.....	47
3.2 Image compression using DCT coding .....	53
3.2.1 The encoding process.....	53

3.2.2	DCT transform .....	53
3.2.3	Quantization .....	56
3.3	Optimized EZW technique for image compression and encryption .....	58
3.3.1	A Combined DCT and EZW image compression scheme .....	58
3.3.2	The Encryption Component of the MEZW .....	64
3.3.3	Conclusion .....	69
Chapter 4	Optimizing still Image compression & Encryption .....	70
4.1	The JDWCT-CVQ simultaneous compression and Encryption .....	70
4.1.1	Compressive Vector quantization (CVQ) .....	71
4.1.2	The proposed JDWCT-CVQ scheme .....	73
4.1.3	Experimental and analysis results .....	76
4.2	JDWCT-CVQ-Edge compression .....	85
4.2.1	JDWCT-CVQ-Edge scheme .....	85
4.2.2	Experimental work and analysis of results .....	86
4.3	Conclusion .....	93
Chapter 5	CVQ for Secured Video compression .....	95
5.1	Existing Video coding techniques .....	95
5.2	JDWCT- CVQ scheme for video compression and encryption .....	98
5.2.1	The compression algorithm .....	99
5.2.2	The encryption algorithm .....	101
5.3	Experimental work and analysis .....	103
5.3.1	The periodicity of reference frames .....	104
5.3.2	Compression analysis .....	107
5.3.3	Encryption analysis .....	111
5.4	Conclusion .....	116
Chapter 6	Edges Sensing for Simultaneous Video Compression and Encryption .....	117
6.1	Chaotic logistic map for pseudo-random number generation .....	118
6.2	Combined sine map and chaotic logistic map .....	121
6.3	The JDWCT-CVQ-Edge secure for video compression scheme .....	123
6.3.1	A revised Video Compression scheme .....	123
6.3.2	The Encryption scheme .....	125
6.4	Experimental work and results .....	127
6.4.1	Compression analysis .....	127
6.4.2	Testing the effect of using wavelet different filters .....	130
6.4.3	Encryption analysis .....	131
6.5	Conclusion .....	136
Chapter 7	Edge and Phase Sensing for Secure Video Compression .....	137

7.1	Phase Sensing.....	138
7.2	The A5 cipher description.....	143
7.3	JDWCT-CVQ-Edges-phase sensing scheme .....	145
7.3.1	The video compression scheme .....	146
7.3.2	The encryption proposed scheme.....	147
7.4	Experimental results.....	150
7.4.1	Analysis of compression result .....	151
7.4.2	Security analysis .....	164
7.5	Selective encryption based on AES .....	169
7.6	Conclusion .....	170
Chapter 8 Conclusion and Future work .....		172
References.....		178

# List of Figures

Figure 1.1 Desert image and its histogram of differences between neighbouring pixels .....	5
Figure 1.2 DCT encoding to Lena eye .....	7
Figure 1.3 Image (Lena) decomposition by DWT, (A) the first level decomposition and (B) second level decomposition .....	8
Figure 1.4 Histogram for original image (Lena) and for image after decomposed by DWT .....	9
Figure 1.5 Block diagram of image compression and decompression.....	10
Figure 1.6 coding order in MPEG.....	12
Figure 1.7 PB and D frame prediction .....	13
Figure 1.8 simultaneous video compression and encryption Approach.....	15
Figure 2. 1The block diagram of transform image coding	21
Figure 2.2 PB Mode	29
Figure 2.3 JPEG 2000 fundamental building blocks	30
Figure 2.4 the search area	32
Figure 2.5 The JPSEC creator and consumer	44
Figure 2.6 The structure of JPSEC stream	45
Figure 3. 2 (a) the sub-bands of WT i for 8x8image (b) the quad-tree for each coefficient HL3 and X2/HL2 sub-bands.....	49
Figure 3.3 Morton scan .....	49
Figure 3.4 Quantization transforms .....	51
Figure 3.5 (a) a block 8x8 pixels from Lena image that is DWT (b) Three-level scan order.....	51
Figure 3.6 (a) Iteration 1, threshold = 1024. (b) Iteration 6 threshold = 64 .....	52
Figure 3.7 Block diagram of image encoding based DCT coding .....	53
Figure 3.8 the DCT basis function of 8x8 arrays .....	55
Figure 3.9 Quantization tables (A) Low compression high quality (B) high compression Low quality .....	57
Figure 3.10 Zigzag scanning order of quantized DCT coefficients sequence .....	58
Figure 3.11 The Block diagram of proposal compression system .....	60
Figure 3.12 shows images test .....	62
Figure 3.13 the comparison results between MEZW and EZW .....	64
Figure 3.14 The scheme diagram of MEZWE system.....	65
Figure 3.15 Histogram of original and encrypted image .....	66
Figure 4.1 Parallel Image compression and encryption	71
Figure 4.2 Vector quantization coding and decoding	72
Figure 4.3 example of CVQ encoding	73
Figure 4.4 Block diagram of image compression based on hybrid DWT, DCT and CVQ.	75
Figure 4.5 Block diagram of the proposed encryption	76
Figure 4.6 Compression results after WT to level 2	77
Figure 4.7 The mean and STD of Compression results after WT to level-3 and level-2	78
Figure 4.8 Encrypted Images	81
Figure 4.9 Histogram of original and encrypted image	83
Figure 4.10 PSNR of encrypted images	85
Figure 4.11 Illustrated steps for image compression scheme	86
Figure 4.12 Illustration of the Empirical Rule	87
Figure 4.13 Experiment scheme to image compression through edges extraction for level 1 wavelet decomposition	88
Figure 4.14 CR and PSNR from level 1 sub-band (V1, H1, and D1).	89

Figure 4.15 Number of significant coefficients of the sub-bands (V1, H1 D1) for THR=1, 1.5, 2.	90
Figure 4.16 CR and PSNR of JDWCT-CVQ-Edge encoded images for different thresholds.	91
Figure 4.17 represent the comparison results between compression based on hybrid method and edges extraction	92
Figure 5.1 example of video frames encoding (A) display order (B) encoding order .....	97
Figure 5.2 displays corresponding blocks from an inter- frame and RF frame and the search area in the RF frame. ....	98
Figure 5.3 parallel video compression and encryption .....	99
Figure 5. 4 Block diagram of the video compression .....	101
Figure 5.5 Illustrates the Encryption scheme .....	102
Figure 5.6 shows sample frames of videos test.....	104
Figure 5.7 shows a periodic Reference Frame (RF) in video compression proposed scheme..	105
Figure 5.8 the effect of period RF on CR .....	106
Figure 5.9 the mean and STD of PSNR of tested videos at different RF period .....	107
Figure 5.10 displays CR for tested video.....	108
Figure 5.11 illustrates the execution time for video compression-encryption and decompression-decryption .....	109
Figure 5.12 (A1, C1) represents encrypted frame, (B1, D1) represents histogram of encrypted frame, (A2, C2) represents decrypted frame, (B2, D2) represents histogram of decrypted frame .....	112
Figure 5.13 (A) represents encrypted frame, (B) represents decrypted frame (C) represents histogram of encrypted frame, (D) represents histogram of decrypted frame .....	113
Figure 5.14 illustrates the correlation of two adjacent pixels in the encrypted and decrypted frames, (A1, C1, E1) represent the correlation test in horizontal direction of the encrypted frame in video 1,3 and 6 respectively, (A2, C2, E2)represents correlation test of encrypted frame in vertical direction,(B1, D1, F1) and (B2,D2, F2) represents the correlation test for decrypted frame in the horizontal and vertical direction respectively. ....	114
Figure 6.1 represents the curve of the orbit map.....	119
Figure 6.2 A cobweb diagram of the logistic map, showing chaotic behaviour for various values of initial condition ( $\mathbf{x0}$ ) and control parameters ( $\mathbf{r}$ ), $\mathbf{n}$ is the number of iterations.....	120
Figure 6.3 bifurcation diagram (A) for logistic map, (B) for sin map (C) for combine logistic and sin map .....	122
Figure 6.4 The simultaneous compression and encryption scheme .....	123
Figure 6.5 diagram illustrates the video compression scheme.....	125
Figure 6.6 represent Encryption scheme.....	127
Figure 6.7 CR for tested video.....	128
Figure 6.8 The execution time .....	129
Figure 6.9 illustrates PSNR of video frames at constant CR of 97%.....	129
Figure 6.10 CR, execution time and PSNR for different wavelet filters .....	131
Figure 6.11 shows the encrypted and decrypted frames of tested videos (A) encrypted-decompressed frame (B) decrypted-decompressed frame .....	133
Figure 6.12 Histogram of encrypted and decrypted frame selected from tested video (A) encrypted frame (B) decrypted frame .....	135
Figure 7. 1 The sinusoidal signal .....	138
Figure 7. 2 illustrates the mismatched block compression based on phase sensing .....	140
Figure 7.3 simultaneous video compression and encryption .....	143
Figure 7.4 illustrates the A5 structure.....	144
Figure 7.5 represents the process block diagram of the video compression .....	147

Figure 7.6 shows the encryption scheme .....	149
Figure 7.7 represent sample frames from test videos.....	151
Figure 7. 8 shows the achieved CR for tested video .....	152
Figure 7. 9 the mean and STD of PSNR for decompression and decryption videos .....	152
Figure 7.10 shows the bit rate comparison .....	153
Figure 7.11 CR comparison .....	154
Figure 7.12 shows execution time.....	154
Figure 7.13 PSNR for different video frames at constant CR of 97% .....	155
Figure 7.14 sample of recovered frame by three different methods (A) JDWCT-CVQ-Edges- phase sensing (B) DCT based(C) SPIHT.....	156
Figure 7.15 Comparison between phase sensing and RLE followed by Huffman encoding ....	157
Figure 7.16 represents sample frames size 512x512 pixels from test videos .....	157
Figure 7.17 illustrates the results of test videos with frame size 512x512 pixels .....	159
Figure7. 18 represents sample frames size 512x512x3 pixels from test videos .....	160
Figure7.19 illustrates the results of test videos with frame size 512x512x3 pixels .....	160
Figure 7. 20 represents sample frames size 240x320x3 pixels from test videos .....	161
Figure 7. 21 illustrates the results of test videos with frame size 240x320x3 pixels .....	161
Figure 7.22 the histogram of encrypted and decrypted frame (A) encrypted frame (B) decrypted frame(C) the histogram of encrypted frame (D) the histogram of decrypted frame .....	166
Figure 7.23 the correlation coefficient of adjacent pixels in encrypted and original frames ....	168
Figure 7.24 the mean and STD of PSNR for encrypted and decrypted videos.....	168
Figure 7. 25 illustrates the consumed time of selective encryption baser on AES .....	169

# List of Tables

Table2.1 video coding standard .....	28
Table 3.1 results of EZW and proposal encoder	60
Table 3.2 Compression results after encoded by EZW and EZW combined with DCT	63
Table3.3 The correlation coefficient between two adjacent pixels in original and encrypted image.	68
Table 4.1: Number of blocks post compression, at decomposition level 1, 2 and 3.	79
Table 4.2 the comparison results between MEZW and JDWCT-CVQ	80
Table 4.3 Correlation coefficients analysis	84
Table 4. 4 the comparison results between MEZW, JDWCT-CVQ and JDWCT-CVQ-Edge	92
Table 5.1 The Mean and STD of PSNR for decompression video at different period RF.....	106
Table 5.2 the Mean and Standard Deviation (STD) of PSNR for decompression and decryption video.....	108
Table 5. 3PSNR of reconstructed frame of endoscopic video .....	110
Table 5.4 the mean and STD of PSNR of tested video at constant CR = 97%.....	110
Table 5.5 the Correlation coefficient of adjacent pixels .....	113
Table 5. 6 the Mean and Standard Deviation (STD) of PSNR for encrypted video .....	115
Table6. 1 The Mean and Standard Deviation (STD) of PSNR for decompression and decryption video	128
Table 6.2 The Mean and Standard Deviation (STD) of PSNR for CH6M and CH5M method	130
Table 6.3 the Mean and Standard Deviation (STD) of PSNR for encrypted video	132
Table7. 1 the edges and mapping them in a zeroes matrix .....	142
Table7. 2 the clocking rule of A5 .....	145
Table7. 3 shows the mean and STD of PSNR and CR .....	155
Table7.4 the comparison results between coding frame size 512x512 and 256x320 pixels.....	158
Table 7. 5 shows the execution time by two different computers for coding 100 frames from 30 different videos with size (256x320) pixels in grayscale.....	163
Table 7. 6 shows the execution time by two different computers for coding 100 frames of 10 different coloured videos .....	164

# Declaration

*I hereby declare that all the work in my thesis entitled “**Efficient Simultaneous Encryption and Compression of Digital Videos in Computationally Constrained Applications**” is my own work except where due reference is made within the text of the thesis.*

*I also declare that, this thesis is not submitted for any degree at the University of Buckingham or any other university.*

**Nazar Al-hayani**

# Chapter 1

## Introduction

In recent years multimedia transmission through wireless channels has grown fast as a result of advances in wireless technology and availability of sensors and devices with increasing capacity for high resolution imaging. This has contributed to exponential increase in bandwidth and storage requirements far beyond the capabilities of existing open network channels and storage device. Therefore, data compression becomes essential for reducing the resulting heavy and ever increasing burden these requirements place on storage or transmission. The choice of the appropriate compression technique depends on the domain of application requirements in terms of competing factors such as image/video quality, bandwidth availability and efficiency. Moreover, many applications such as military images, video conferences, video telephony, digital TV and Internet streaming are transmitted through “unsecured” network channels such as the Internet. In such scenarios, video content is vulnerable to access or interception by unauthorized parties. Therefore, beside compression there is a need to deploy some security mechanism to protect the image/video content during transmission or in storage.

This thesis is concerned with efficient secure video transmission over open networks. The amount of uncompressed video data is too large for limited transmission channel. Therefore, video compression is an essential to achieve faster communication as possible through a limited bit-rate transmission channel. Over the last few decades, various methods and techniques for data and image compression have been developed. In general data compression algorithms can be categorized in two groups: lossy and lossless data compressions.

The general approach adopted in image./videos compression consist of 3 main steps: applying a frequency domain transform; a quantisation scheme which is meant to produce a sparse representation of the frequency data; and a coding scheme to map the output symbols onto a compact table of digital code words representing the compressed image/video. In this thesis we investigate and develop a video compression technique that is based on combining two main transforms, DWT and DCT, followed by Vector quantization and edge-phase sensing.

There are many well established cipher systems that have been developed over the years, Moreover; selective encryption technique is also performed to provide security when compressing data during transmission or storage. This technique will be shown to be fast enough to meet an efficient video streaming in constrained computational power.

In the first section of this chapter we shall describe general approaches to image and video compression. Section 1.2 describes available video and image encryption. Section 1.3 presents thesis objectives, main approaches, and contributions. Section 1.4 gives thesis organisation. Finally, list of publication is presented in section 1.5.

## **1.1 Introduction to Image and Video Compression**

Multimedia technologies have been advanced in recent years. Many applications such as military images, video conferences, video telephony, digital TV and internet streaming are transmitted through network channels. A digital image is composed of a finite number of elements known as pixels. In grey scale image, pixels are represented by unsigned 8-bits and having values ranging from 0 to 255. Pixels in colour images are represented by three colour channels; in Red, Green and Blue (RGB) colour space. Each component of RGB required 8-bits in binary representation and thus 24-bits per pixel. And there are other colour spaces such as  $Y C_B C_R$ . The Y component is luminance and  $C_B C_R$  represents chrominance components which define the actual colour.

Generally, the content of Multimedia objects (e.g. image/video) is too large in size with high bit rates when produced from uncompressed digital video streams. Accordingly, such objects require large storage space, and their transmission over network channels requires very high bandwidth. When the data rate is generated by video streams become greater than the available network bandwidth, the data transmission process will suffer from long delays and possible failure.

For example, if a digital monochrome video of spatial resolution of  $320 \times 240$  pixels is to be displayed at the standard rate of 25 frames per second (fps), then the corresponding bit rate is  $320 \times 240 \times 25 \times 8 = 15360000$  bits per second (bps). Now assuming the availability of a 56000 bps telephone network link, then we need to compress the video data by at least 275 times in order to accomplish the data transmission in real-time. Real-time is meant that the algorithm will run the video at the rate of the source supplying video. It is definitely impossible to transmit such a large volume of data (uncompressed video) over bandwidth limited networks used in low constraint devices.

The case of coloured images/videos is 3 times more demanding in terms of storage and bandwidth.

Therefore, data compression becomes essential before any video storage or transmission because it produces a low bit rates and reduces the time required for sequence video frames to be sent over different network channels (Shi and Sun 2008) (Annadurai 2007) (David, Motta and Bryant 2007).

Generally, the range frequencies required to transmit the audio are far narrower than that required for image/video transmission. Therefore, the audio also can be compressed at high quality using relatively low bit rate (high compression ratio).

### 1.1.1 Still image compression techniques and tools

Generally, compression of digital image/signals aims to faithfully represent digital data with fewer bits than necessary without degrading the amount of information conveyed by the data. This is possible because data and pixel values in images are not randomly dispersed within their spatial domain and include many redundancies. Compression is closely dependent on the amount of data redundancy in the generated image/video data. The amount of redundancy in an image has a measurable impact on the amount of information the image is conveying. Data compression techniques work by removing redundancies as much as possible and producing a compact representation of the data with little or no effect on the amount of information conveyed by the data. According to the theory of information, the amount of information a data/signal conveys is determined by the measure of Entropy which is computed in terms of the probability distribution of the symbols occurring in the data. According to the Information Theory, any symbol “S” in a signal/data holds an amount of Information (measured in bits) that reflects the frequency of its presence within the signal/data. The more a symbol “S” occurs in a signal the less its presence adds to the amount of information conveyed by the data. Shannon Theory of information models the amount of information a symbol “S” holds within a signal by the formula:

$$I(S) = \log_2 \left( \frac{1}{p} \right) = - \log_2 p \text{ bit} \quad 1.1$$

Where ‘ $p$ ’ is occurrence probability of “S” in the signal.

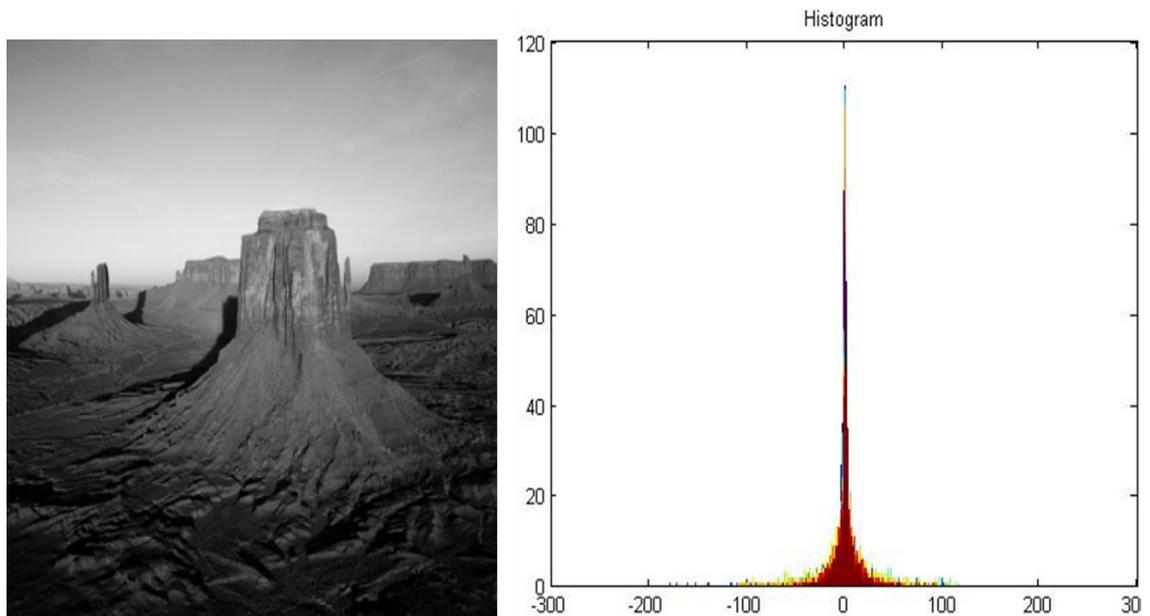
From the above equation 1.1, one can infer that a small value of ‘ $p$ ’, shows that more information the symbol holds and vice-versa.

The amount of information for symbol  $S_i$  ( $i = 1, 2, \dots, n$ ) is called the Entropy ( $E$ ), where  $E$  can be described as below:

$$E = - \sum_{i=1}^n p_i \log_2 p_i \quad \text{bit} \quad 1.2$$

Shannon declares that a message of ' $n$ ' symbols can be coded to  $n \cdot E$  bit. A fixed length bit code representation for all symbols is not a realistic approach, but an optimal solution means that a minimum bit rate will be required for coding the more frequently occurring symbols while larger bit rate can be afforded for low occurring symbols. This approach is referred to in the literature as entropy coding. The fundamental concept in entropy coding is to assign the shortest code words to the symbols that occur more frequently, and the longest code words to those that appear infrequently. Accordingly, knowledge of the probability distribution of image spatial domain pixels (or the frequency coefficients), is essential for optimal compression. However, achieving optimality in this respect may have adverse effect on efficiency, and in many applications some loss of information may become necessary.

The main challenges to compression are the efficiency of detecting and locating the significant data, and coding the data in a compact manner that remain within the actual constraints of data storage and/or communication bandwidth. In digital images, data redundancy may exist in the form of correlation between neighbouring pixels. In fact, pixel values are very similar and high correlation in the non-edge smooth regions, where differences between neighbouring pixels are generally very small or zero. For example, Figure 1.1 shows a histogram of differences in grey value between each pixel and its adjacent pixel in the 320x320 Desert image. It can be seen from Figure 1.1 that the differences clustered around zero i.e. there is high correlation between adjacent pixels in raw image. But neighbouring pixels around significant features differ significantly in their values. Hence, image compression could benefit from knowledge/detection of the smooth regions as well as the significant regions so that it can remove the redundancies in the first while retaining as much information as possible in the latter.



*Figure 1.1 Desert image and its histogram of differences between neighbouring pixels*

Compression algorithms generally exploit statistical information about the frequency distribution of symbols/characters of the input data in order to reduce redundancies and to create new coding map of the data. Knowledge of the frequency distribution of symbols in the original data can be used to represent symbols with variable length bit strings and reduced redundancies. For example, in Figure 1.1, one can afford using long code words for symbols further away from the 0. There are various coding schemes that have been used in compression such as Huffman coding (Yang and Bourbakis 2005), Run Length Coding (RLE), Arithmetic coding LZW and etc. For more details see (Carreto-Castro, et al. 1993), (Pandur and Thiruvallur 2009) (David, Motta and Bryant 2007).

Image/video compression systems usually consist of three main steps: (1) de-correlation step which aims to reduce the amount of data redundancy; (2) a quantisation step; and (3) entropy encoding step. The de-correlation step is achieved by transformation of the uncompressed data from the spatial domain to the frequency domain where redundancies can be easily detected and removed. Often this step results in increased memory requirements because the output image data are real numbers rather than fixed length integer. However, the output data consist of large amount of small values (near 0) that are redundant. The second quantisation step, results in a small number of symbols that approximating the real-number data obtained in step 1. The entropy coding results in a coding table, to represent the quantised symbols in code words (bit strings) of variable length. The second step (quantisation) may result in loss of information. In

many applications certain level of such loss may be tolerated but has an effect on compression ratio and image quality. Accordingly, compression can be classified into two classes: lossless and lossy.

Generally compression exploits the statistical information in the image and is a reversible process by the decompression procedure. In the lossless case the reconstruction is perfect, and the image quality is preserved. However, lossless compression can only achieve low compression ratios. A very simple example of lossless compression skips the first two steps and use RLE, commonly used for any type of data, and basically replaces any run (sequence of adjacent) pixels of the same grey value by giving the pixel value followed by the length of the run. For our purposes we shall focus more on Lossy compression but we aim to minimize the amount of loss, and achieve acceptable processing time as well as image quality.

Lossy compression is irreversible, because some information contained in the original data will be lost primarily during the quantisation step and the original data cannot be recovered exactly. It is typically used for images and sound compression where a little bit of loss in resolution is frequently undetectable. Therefore, the decompressed image contains degradations relative to the original image. Generally, more compression can be achieved at the expense of more distortion. However, there is always a trade-off between compression ratio, retrieve data in computational time and quality. The higher compression ratio is, the smaller the compressed data size is and the lower the bandwidth requirement is but the output will suffer from lower quality.

There are a variety of transforms that can be used for the de-correlation step, which normally project the image spatial domain (considered as a vector in a high-dimensional vector space) into another domain where the most significant image features can be approximated by a vector in a smaller dimensional subspace. The frequency domain transforms of Discrete Cosine Transforms (DCT) and the Discrete Wavelet transforms (DWT) are widely accepted for their capacity to re-organise input data in such a way that redundant transformed data can be detected easily. The existing image and video compressions tools of JPEG and MPEG are based on DCT while JPEG2000 is based on DWT. The DCT transforms the original data from spatial domain to frequency domain, while the DWT decompose the original data into different frequency sub-bands (low frequency sub-band and high frequency sub-bands). On their own, these transformations do not realize any compression, but they de-correlate the original data and compact the

significant information in certain frequency coefficients that form a relatively small set compared other frequency coefficients.

For efficiency purposes, the original uncompressed image may be first divided into smaller non-overlapping blocks and the transform is applied on each block separately. In the case of the DCT, the blocks are usually of size 8x8, but unfortunately this may result in blocky effects depending on the compression ratio.

For an 8x8 image block  $B = (b_{ij})$  the DCT transform outputs an 8x8 frequency coefficient matrix  $T(B)$  so that for each  $i=1..8$  and  $j=1..8$ .

$$T(B) = \begin{cases} \sqrt{\frac{1}{n}} \cos \frac{(2j+1)i\pi}{2n}, & i = 0; j = 0, 1, \dots, n-1 \\ \sqrt{\frac{2}{n}} \cos \frac{(2j+1)i\pi}{2n}, & i = 1; j = 0, 1, \dots, n-1 \end{cases}$$

All the coefficients of  $T(B)$  appear in ascending order of frequency magnitude in zigzag manner starting from the top left corner (see chapter 3 for description of DCT based image coding). Usually, the lower order coefficients contain significant information about a particular block and this information are more important in human perception. Therefore, in this thesis we shall be calculating DCT for only the top left triangle part of the  $T(B)$  block where the energy is locate. This method will reduce the computational time of DCT to the half and become more suitable to low constraint applications, see Figure 1.2.

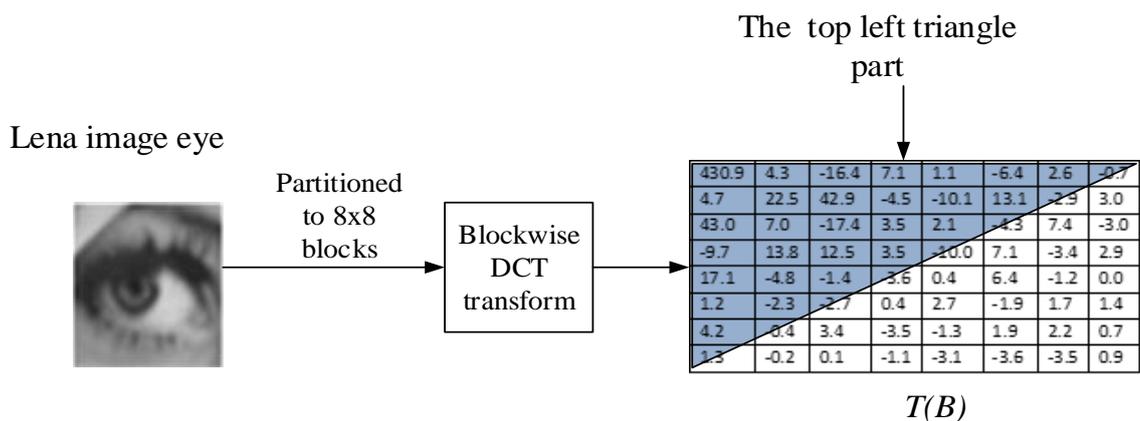


Figure 1.2 DCT encoding to Lena eye

JPEG2000 is DWT based coding method and produces higher quality compressed images compared to the earlier JPEG tool. On the other hand, JPEG2000 is still not so popular and the commercial options for JPEG2000 are not that great (Acharya and Tsai 2005) (Vetrivel, Suba and Athisha 2010) (David, Motta and Bryant 2007). The DWT decomposes an input image into four sub-bands labelled as LL1, HL1 (V1), LH1 (H1) and HH1 (D1) as shown in Figure 1.3 A. Subsequent decompositions are possible in many different ways. In the pyramid decomposition scheme, the LL1 (low frequency sub-band) is further decomposed into the next level of four sub-bands as shown in Figure 1.3 B. The low frequency sub-band (LL) represents an approximation of the original image. Therefore, the histogram distributions of LL sub-band is identical to the histogram of original image. The non-LL sub-bands (high frequency sub-bands) contains image features such as edges and corners. The non-LL sub-bands have Laplacian distribution and the significant coefficients (image features) are furthest away from the mean of each high frequency sub-band, this property remains valid at all level of decomposition as shown in Figure 1.4 (Al-Jawad, Ehlers and Jassim 2006). JPEG2000 and many recent compression schemes exploit these properties to gain knowledge about the non-smooth regions and increase compression ratio over JPEG (Ma 2002) (Ehlers 2008).

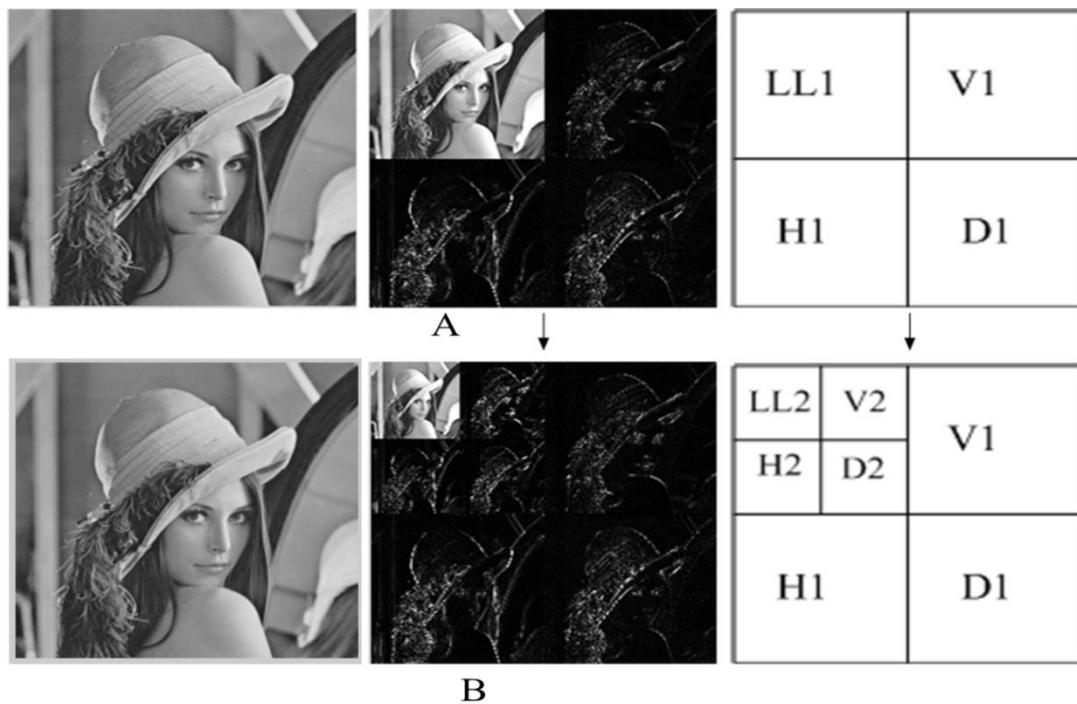
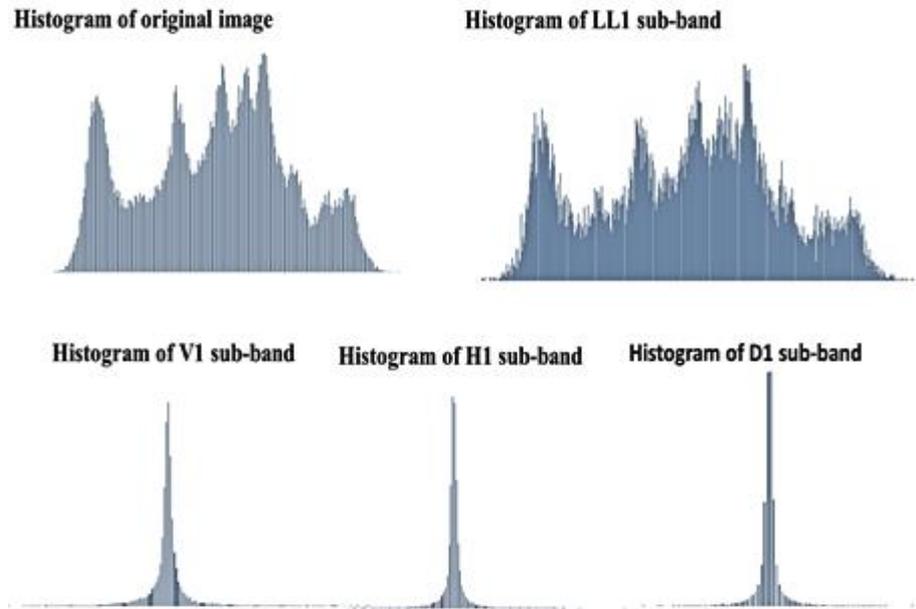


Figure 1.3 Image (Lena) decomposition by DWT, (A) the first level decomposition and (B) second level decomposition



*Figure 1.4 Histogram for original image (Lena) and for image after decomposed by DWT*

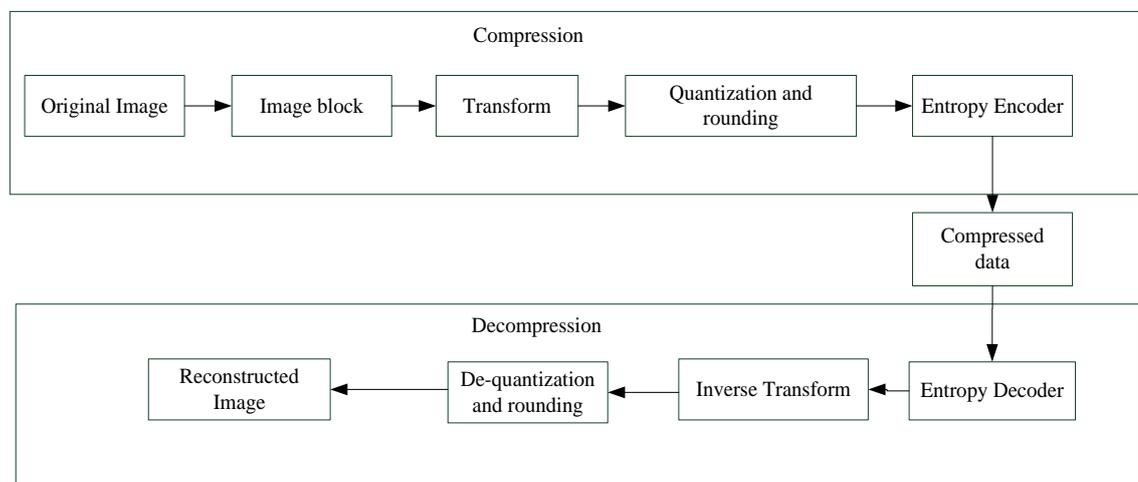
The Standard Deviation (STD) of the non-LL sub-bands can be used to reveal these significant coefficients which correspond to edges and corners.

Having transformed the input image, the second reduction in entropy step of compression in each DCT block (or DWT sub-band) is achieved by removing the non-significant information which is traditionally done by applying a scalar quantization of the transformed coefficients. The quantization is based on the human visual system which is more perceptive to low frequency information than high frequencies. The simplest quantization can be done by dividing the block contents by a quantization factor, chosen in terms of the desired compression ratio, and rounding frequency coefficients to the nearest integer value. This is an irreversible process because cannot recover the lost information by inverse process. In the final step, the quantized coefficients are losslessly encoded using entropy encoding such as Run-Length Encoding(RLE), Huffman coding and arithmetic coding. Consequently, the entropy of quantized data can be represented by a variable bit rate compared with raw image pixel values that have fixed size bit representation. The basic idea in entropy coding is to assign the shortest bit-string code (known as word) to the symbols that occur more frequently, and the longest code words to those that appear infrequently.

A more advanced quantization technique for image compression is called Vector Quantization (VQ). In VQ the image is subdivided into non-overlapping blocks, each block is considering a vector called a code vector. The encoder will construct a list consisting of all code vectors is called a codebook. Then for each code vector, the closet

matched vector is determined and its index in codebook is encoded and transmitted to the decoder. At the decoder, the index is decoded and replaces it with the associated code vector with the same codebook as at the encoder. This technique will be demonstrated in more detail in chapter 4.

The image decompression procedure is just an inverse of compression process which starts by using the code book to recover the quantized values and finally applying the inverse of the DCT/DWT frequency transform. The following is a block diagram that covers both compression and de-compression.



*Figure 1.5 Block diagram of image compression and decompression*

When compressing RGB colour images, they are first converted to the  $Y C_B C_R$ . The human visual system is more sensitive to the Y component than other two components. So that, the Y component can be encoded at low compression ratio and the  $C_B C_R$  components may be encoded at high compression ratio.

In this thesis we focused our investigations primarily on monochrome grey scale images/videos. However, we shall test the performance of a multi-channel version of our final monochrome coding scheme on colour videos to demonstrate the viability and suitability of our proposed scheme for efficient secure coding of colour videos (see chapter 7). (Shi and Sun 2008) (Blelloch 2001) (Sayood 2012)

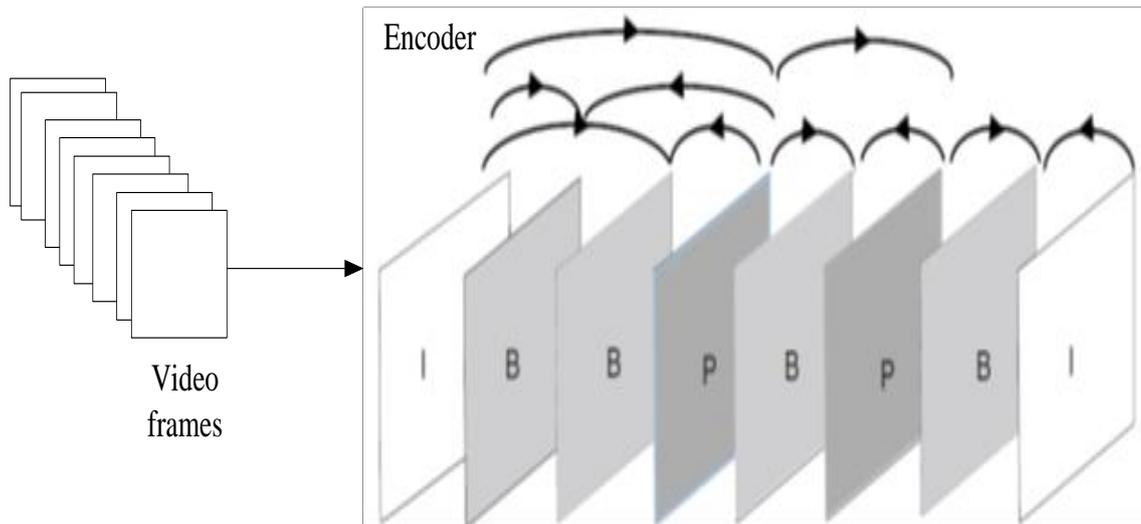
### 1.1.2 Video compression tools and standards

Digital video is a sequence of images, each called a frame. In video sequences there are two types of redundancies that could be noticed. The first type is the spatial redundancy

(also called the intra-frame redundancy) that exists in each frame i.e. correlation between pixels within each frame. The second type is the temporal redundancy which is concerned with correlation between successive frames of video that are usually very similar. Video compression techniques are therefore based on reducing redundancies in both spatial and temporal directions. A common technique for video compression starts by encoding the first frame (Intra-frame) using a still image compression procedure. Then encoding each successive frame (Inter-frame) by determining the differences between the Intra and inter frame and encoding these differences and transmitting the differences (motion vector data) rather the whole frame over again. However, this approach results in accumulating errors and hence requires frequent corrections.

There are several methods to determine the motion vector; one such method is block based motion compensation which starts by sub-dividing the current frame into uniform non overlapping blocks. Then each block is compared with a set of candidate blocks in the preceding frame to select the one that best match the current block. However, motion estimation is time consuming in video compression process. Several standards like MPEG-x, H.26x, (x=1, 2, 3, 4) have been developed over the years and are commonly used for video compression. MPEG-x has been designed for storage (DVD) and TV broadcast, while H.261, 263 and H.264 are usually optimized for real-time video communication. Both standards are motion compensation based in Intra-frame coding.

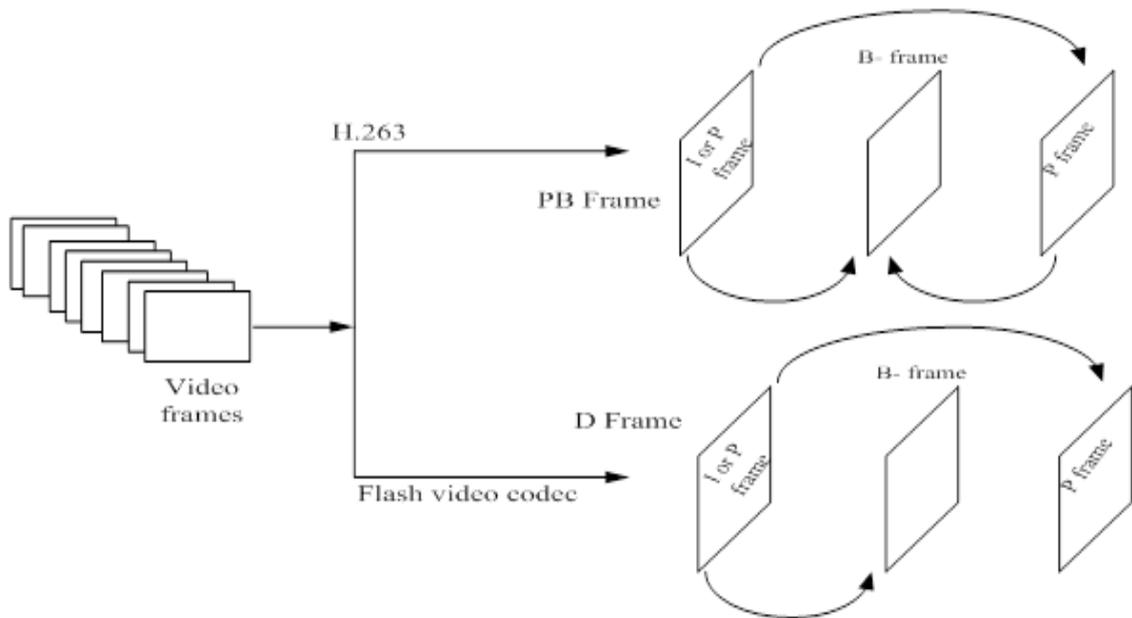
To expedite motion vector prediction, MPEG separates the video into three types of frame: intra-frame (I), Predictive frames (P) and Bidirectional frames (B). The P frames are predicted directly from previous I frame. B frames are predicted from both previous (I or P) and next frame (I or P) as shown in Figure 1.6.



*Figure 1.6 coding order in MPEG*

The H.263 video compression tool use PB-frame mode for temporal encoding. PB-mode consists of two frames: P frame predicted from the latest decode frame and B frame predicted from both last frame (P or I valid) and the next frame P. The flash video codec (flash player) replaces the B frames with D frames. D frames only use the past P or I frame for motion vector prediction. The codec D frames achieve less compression ratio compared with H.263 but it improves the real-time processing, see Figure 1.7. We will describe this topic in more detail in chapter 2.

JPEG, MPEG-x and H.26x are DCT based image/video coding. However, the image based DCT coding suffers from noticeable distortion at high compression ratio (low bit rate). In this thesis we shall investigate the use of a hybrid of DWT and DCT for video compression to achieve a better performance than DCT based coding in term of quality.



*Figure 1.7 PB and D frame prediction*

Motion JPEG2000, also known as MJ2, employs only intra-frame. MJ2 compresses each frame individually by using JPEG2000 algorithms for still image compression.

## **1.2 Image and Video Encryption**

For many organisations that rely in their work on the transmission of digital media objects over open network channels, protecting their contents from hackers and eavesdroppers has become an essential step in protecting their knowledge asset. Being the main and most commonly used security mechanism, encryption techniques are the natural scheme to be used to protect video data against security violations in storage and in transit. Encryption has been used for centuries for the protection of secrets and a wealth of ciphers have been developed within the wider area of information theory.

Suitability of image/video encryption algorithms is dependent on various factors such as security level, computational time and their impact on compression efficiency. Modern data encryption ciphers such as AES, DES, and RSA generally work well on data stream input such as audio, video and text. Most of these ciphers work by acting on blocks of certain sizes. When dealing with offline secure video streaming, the above encryption algorithms can provide strong security, but are not fast enough for video streaming due the large size of the data input unless considerable amount of computational power are deployed. Stream ciphers are the alternative type of ciphers

that may be provide more efficiency, because they work on small size units of data. Examples of such ciphers include the Linear Feedback Shift Register (LFSR). In recent years, chaotic maps have been used to create stronger stream ciphers.

When dealing with both tasks of compression and encryption, the choice of a specific cipher is not the only influencing factor. Much consideration has been given to whether security necessitates the encryption of the entire compressed data, and whether these tasks can be implemented in parallel. Such decisions also have an influence on the choice of the appropriate cipher. Note that distorting an image in a way that renders recognition of the regions/objects of interest very difficult is akin to securing the image content. This idea have led to the concept of selective encryption which works by encrypting a small amount of the image data with a difficult to break cipher while using simple shuffling, if necessary, on the remainder. For example, encrypting certain wavelet sub-band(s) such as the LL-sub-band with a strong cipher may be sufficient to secure the image content. Selective encryption is useful for efficiency and will be investigated in this thesis. Moreover, the fact that image/video data have certain spatial redundancies that have been exploited for compression presents a choice between two approaches: Sequentially Compress and Encrypt (SqCE) or Joint Compression and Encryption (JnCE) by interleaving the encryption task with the steps of compression (i.e. the de-correlation step, quantisation, and entropy coding). In SqCE compression and encryption are done independently, inefficient and in the case of videos it complicates the effort of exploiting temporal redundancies across adjacent frames. For the JnCE approach, the encryption algorithm can implicitly have access to data at various compression processing steps, which is known SqCE compared with the JnCE; therefore the JnCE is better than SqCE in low constrained applications.

In this thesis, we incorporate selective encryption into the JnCE framework and refer to this modified approach as Simultaneous Compression and Encryption (SCE) which allows the data to be securely exchanged. Figure 1.8 depicts our SCE framework, taking into account our use of wavelet transforms as the de-correlation tool.

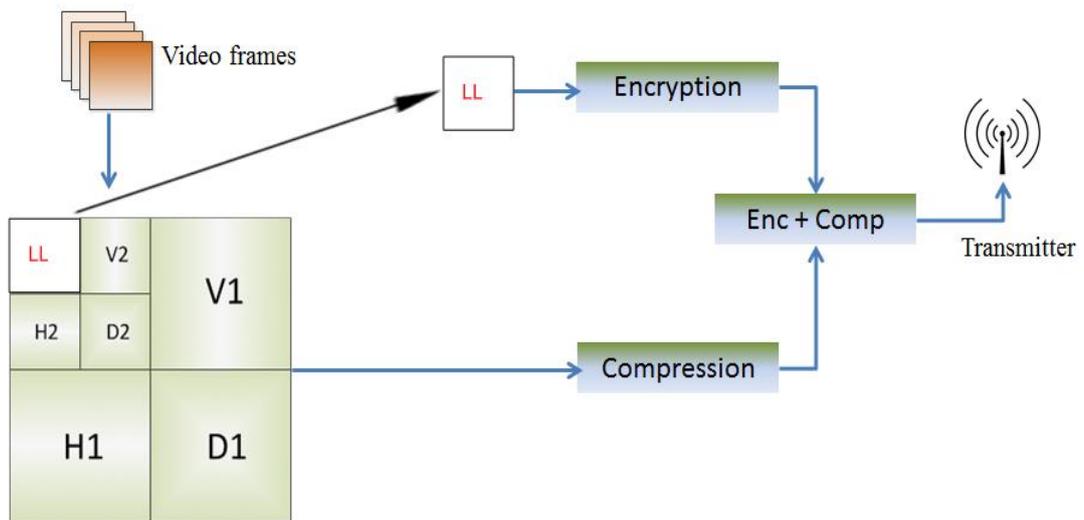


Figure 1.8 simultaneous video compression and encryption Approach

### 1.3 Thesis Objectives, Main Approaches, and Contributions

This project is aimed at meeting a growing demand for secure video transmission as a mean of gathering timely reliable awareness of events from unconventional locations while availability and affordability of sophisticated computing and communication facilities and infrastructure are restricted to the very basic. This is a typical scenario in many third world countries and in particular countries that are facing widespread international terrorism, organised crime, and internal civil conflicts while their law enforcing agencies and armed forces are deprived of material and skilled human resources. My research project is meant to utilise commercially available and affordable mobile devices that could be incorporated and integrated into its modest computing and communication infrastructure, this to facilitate remote intelligence gathering from around the country through secure communications with their personal who may be equipped with modest handheld devices that are endowed with reasonable cameras. The computing power requirements is estimated to be equivalent to speed computer equipped with Intel (R) i5 processor 2.3 GHz and RAM 4 GB.

In this project the concept of JnCE scheme can be based on the ability to process colour videos at a minimum rate of 30 frames per second. The standard frame size on the mobile devices we envisage in this thesis is 256x320 pixels. We will consider larger size standard definition video frames in this thesis of 512x512 pixels. Moreover, to

further reflect the limited computing and storage capabilities of some participant, we shall assume that the handheld devices do not save of videos for offline transmission.

The developed scheme described in chapter 7 can be used by legitimate armed forces of under-developed nations by providing very low cost maintenance of secure communication as a cheap alternative of using costly systems such as satellite communications and/or early warning aircrafts. In these scenarios there are no fundamental reasons why the gathered intelligence videos should be of High Definition (HD), but due to our assumption on the constrained devices capabilities we will not experiment with HD videos. By no mean this is a shortcoming of the developed algorithms but rather a reflection of scares and constrained computational power affordable in the above circumstances. Moreover, we plan to widen the scope of our research in the future to include such investigations.

For my sponsor, this project fits into their effort to build capacity in developing advanced and secure digital technology solutions to fit their needs in a volatile region. The implementation of a secure video transmission system that can be used with high mobility low-cost constrained devices is only one step in their aim, this for achieving self-reliance in the development of cost effective enabling digital technology systems to maintain and adaptable for use in different applications including civil applications. Below is a list of use scenarios that the intended system can be utilised with basic modifications if necessary:

1. In dangerous zones of conflict, the intended system should work inside and on the periphery of the theatre of operations for the secure timely online relay of video content without saving for later use. However, the encrypted and compressed videos would be available for storage at the Command, Control, Computers, Communication and Intelligence (4CI) units.
2. The implementation of the same technology on unmanned aerial vehicle systems or fixed CCTVs as a general surveillance activity for information/intelligence gathering on possible terrorist and criminal. Such activities provide vital information for 4CI units in preparation of necessary tactical plans, or to protect vital infrastructure and industrial installations.
3. The use of the developed solution is by no mean limited to military and security purposes. In fact it can provide vital enabling information for civil applications,

such as the monitoring of critical installations like oil pipes, borders, green fields for agriculture and environmental purposes, etc.

Our main approach is based on the SCE version of JnCE framework to achieve computational efficiency. It will benefit from the wealth of knowledge achieved over the years in existing work on video encryption and compression techniques developed according to existing standards including MPEG-x and H.26x. Compression will be based on a hybrid of: DWT, DCT, VQ, wavelet based edge detection and phase sensing. We aim to achieve lossy compression on the high frequency sub-bands and simultaneously applying selective encryption to the low frequency sub-band. We shall investigate various ways of dealing with Intra-frame Reference Frame (RF) and the Inter-frame as non-Reference Frame (n-RF) to reduce temporal redundancies and we shall determine the most suitable RF updating policy to maintain efficiency, high quality and to reduce processing time. We shall also benefit from existing knowledge in the fields of digital signal processing and data transmission over wireless communication system. In particular we shall study efficient ways of mapping the sparse blocks output from the various de-correlation procedures.

For security objectives, we adopt the selective encryption approach due to its suitability for constrained application. Three selective encryption algorithms will be investigated in chapter 5, 6 and 7 respectively and applied to different components of wavelet coefficients independently of the compression procedures. We develop and test the performance of different stream ciphers, whereby the key stream is generated in different ways but analogous to the A5 stream cipher and clocked by chaotic logistic map.

The experimental results show that the proposed algorithms for video compression and encryption are having the following features: high compression ratio, acceptable quality, resistance to the brute force and statistical attack and low computational processing to meet our constrained requirements.

## 1.4 Thesis organisation

This thesis is organised as follows;

- Chapter 2 gives a literature review of different kinds of image/video compression and encryption and recent work on sequential and joint compression - encryption techniques for image/video data.
- Chapter 3 describes the analysis of still image compression and encryption based on combining DWT and DCT to improve the compression efficiency of EZW. We demonstrate that the DCT reduces the loops scan of EZW encoding. On other hand, the computational time of this combined scheme is relatively high and has relatively slow processing time.
- Chapter 4 proposes two approaches to optimize the image compression and encryption. The first approach is based on Joint DWT, DCT and Compressive Vector quantization (JDWCT-CVQ). The second approach (JDWCT-CVQ-Edge) uses the edges detection combined with the JDWCT-CVQ to improve the image compression-encryption.
- Chapter 5 discusses the approach of using JDWCT-CVQ for secure video compression.
- Chapter 6 propose edge sensing to improve the security of the video encryption and to optimize the computational performance for video compression and encryption that were used in chapter 5.
- Chapter 7 proposes phase sensing to optimize the video compression efficiency of the approach in the previous chapter. Moreover, the chapter proposes combined chaotic logistic map with A5 cipher to real-time secure video transmission.
- Chapter 8 presents general conclusions and future work.

## 1.5 List of Publications

- Al-Hayani, Nazar, Naseer Al-Jawad, and Sabah Jassim. "Simultaneous video compression and encryption for real-time secure transmission." In Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium on, pp. 240-245. IEEE, 2013.
- N. Al-Hayani, N. Al-Jawad and S. A. Jassim, "Simultaneous compression and encryption for secure real-time secure transmission of sensitive video transmission," in International Society for Optics and Photonics, 2014.
- N. Al-Hayani, N. Al-Jawad and S. Jassim, "Simultaneous edge sensing compression and encryption for real-time video transmission," in International Society for Optics and Photonics, 2014.

# Chapter 2

## Literature Review

Over the years, variety image/video compression techniques have been developed to meet a number of conditions on compression ratio, image/frame quality and bandwidth. Data Encryption, on the other hand, has a long history but video encryptions is more recent and require special considerations due to the large size of image/video data and the presence of spatial as well as temporal redundancies. In this chapter we review the literature on existing image/video compression and image/video encryption. Section 2.1 will be devoted for compression, while section 2.2 covers the literature on the combined online compression and encryption for image/video. We shall end the chapter by describing the various refinement steps we adopted in our investigations that culminated in final proposed secure compression of video (SCE) scheme.

### 2.1 Image and video compression techniques

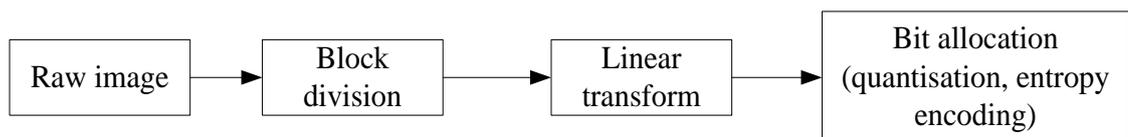
In this section a critical review of existing compression techniques will be conducted highlighting the advantages and shortcomings in relation to suitability for these thesis objectives. Although this thesis is concerned with secure videos under constrained conditions, but the starting point is on image compression techniques which we review in section 2.1.1. In section 2.1.2 the video compression standards, JPE2000 and Inter-frame compression techniques are presented.

#### 2.1.1 Image compression

Transform image coding has developed at the early stages as the main theoretical framework of image compression and refers to encoding the spatial image pixels into compact data representation. It is appropriately used in coding prediction error in motion compensated predictive coding of sequence of video frames.

Usually, the transform image coding works in three steps as shown in Figure 2.1. Firstly, the raw image is divided into blocks size 8x8 or 16x16 pixels. Then, each block is linearly transformed by orthogonal transformation (orthogonal i.e. the dot product of the row with itself is one and with different rows is zero and the same think for columns. The inverse of orthogonal is its transpose; the product of the data transformed

with the inverse of orthogonal transform reconstructs the original data). The orthogonal transform retain the energy in the original image data and faithfully represents the significant image information in terms of a basis of few elements of transformed vectors, also called atoms, which can be used to recover the significant data of original image. The few transformed elements are less correlated compared with raw pixels. Subsequently, quantisation converts the non-significant information (high frequency confidents) to zero. The quantisation processing is related with image quality consideration, if high quality required the quantisation step will be small and vice versa(the quantisation will be explain in more detail in chapter 3) . The three functions quantisation, truncation and entropy encoding are called bit allocation as shown in Figure 2.1The quantized coefficients(after truncation) are collected by zigzag scan. This produces a string (one dimension) of coefficients that starts with non-zero coefficients (low frequency coefficients) and ends with many coefficients being zero. A code known Run Length Encoding (RLE) is very effective to encoding these scanned coefficients. The RLE works by reducing the size of repeating strings, called run, is typically encoding into two bytes. The first represents the number of coefficients in the run and the second byte is the coefficient value in the run. For instance, a run of 8 zeros coefficients will be encoded with two byte (8 and zero) rather than encoding each zero separately. Finally, the entropy encoding will be assigned short bit code word to coefficients that occur more frequently and longer bit code word to coefficient that occur less often. (Rao and Yip 2010) (Shi and Sun 2008).



*Figure 2. 1The block diagram of transform image coding*

The role of the linear coding transform, both in lossless and lossy compression, is to decorrelate spatial domain of the image. A variety of such transforms are available for this task, which project the high-dimensional image spatial domain into another vector subspace generated by a small number of base vectors, called atoms, sufficient for approximating the most significant image features. Transforms like Singular Value Decompositions (SVD) and Principle Component Analysis (PCA) are applicable but not practical. The SVD is not efficient because it relies on solving an eigenvalue problem obtained from the image itself, while the PCA require solving the eigenvalue problem of the covariance matrix of a large training set of images of the same type of content,

(see (Kenneth 1996)). In general, the Karhunen-Loeve transforms (KLT), which is related to PCA has the best efficiency in term of energy compaction, but its coefficients (base coefficients) are not fixed and depend on the data to be compressed, as mention above, and calculating these coefficients is slow. Therefore, the discrete frequency domain transforms provide an efficient alternative and the most popular transform used for image compression is the Discrete Cosine Transform (DCT).

The DCT is used in two international image/video compression standards, Joint Photographic Experts Group (JPEG), and Motion Picture Experts Group (MPEG). It transforms the pixels from the spatial domain (correlated data) into frequency domain (uncorrelated data) (Cabeen and Gent 1998). In order to achieve better compression, the image is subdivided into blocks (of 8x8 pixels), and each block as sum of sinusoidal functions (of 2 variables) of varying magnitude and frequency. DCT compact most energy in the low frequency coefficients that appear at the top left corner of the output block (see page 7, in Chapter 1 for an illustrating example), representing the most significant information of the block. As one moves towards the bottom left corner of the DCT block the number of high frequency coefficients increases while the human eye ability to realize better image quality from their inclusion decreases. Therefore, quantising the DCT coefficients simply rounds down to 0 as many as necessary coefficients diagonally starting from the bottom right corner while a sensible rounding of the rest of coefficients create a relatively small number of symbols to be encoded in the final step. In JPEG and MPEG, after quantization, the RLE and Huffman encoding are applied to each block, after setting to zero the insignificant coefficients in the bottom right region of the DCT output block. Li and Kuo (Li, Li and Kuo 1996) has shown that the coding rate of JPEG will be enhanced by 30% when Huffman coder is used and by 5% in Arithmetic coder if the traditional scan is replaced by layer scan, these concepts of scan are used in JPEG2000. In chapter 3, we shall revisit this scheme with more details.

Several other lossy image compression algorithms have been developed, over the last few years, that use discrete wavelet transforms decomposition rather than the DCT for the de-correlation of the spatial domain image pixels. The Embedded Zero Tree Wavelet (EZW) (Valens 1999), is the first such scheme. In the EZW, the input image is wavelet decomposed, using a given wavelet filter, to multiple levels using the pyramid decomposition scheme. This will provide a multi-resolution analysis of the frequency domain where at each level of decomposition, the finer details are detected more than

the previous at level(s). This forms the bases on the so called “Morton” scan which maps the coefficients into a 1-dimensional vector, starting with each LL coefficient, followed by its 3 descendent coefficients at the same level (LH, HL, HH), followed by the four children at the next higher sub-band, and so on (see figure 3.1 in chapter 3). The EZW exploits this zigzag type of coefficient ordering to create a quad tree structure by two passes (dominant pass, and subordinate pass. The dominant pass determines the significant coefficients by a number of iterations that begin with an initial threshold ( $Th$ ) determined as a number greater than the maximum absolute value in the block and its descendants as long as one or more are  $\geq TH/2$ . The coefficients that are  $\geq TH/2$  are declared significant, and removed from the quad-tree. The threshold is halved in each iteration and the scanning resumes. The lower the threshold gets the better the image quality is maintained. Hence, the iteration stops when all coefficients have been removed from the quad-tree or the threshold reached a minimum value that depend on the required quality. As a result of dominant pass one of 4 symbols (R (zero tree root)), I (Isolated zero), P (positive), N (Negative)) are assigned to each coefficient dependent on original sign and its relation to its descendants and ancestors. The second subordinate pass assigns binary codes to the coefficients. The various steps of EZW will be described in more details in the next chapter.

The main disadvantage with EZW is the increased computational cost of the iterations in the dominant pass. The number of iterations has an impact on the image quality. There have been many attempts to improve the performance of EZW. For higher compression ratio, the output data of EZW is input to the Huffman encoder (Janaki, Tamilarasi and others 2011). The performance was tested with different quantisation threshold values, and it was shown that increasing the threshold results in wavelet coefficients and descendants become insignificant and thereby improved compression ratio and bit per pixel needed. However, the computational cost of Huffman coding is relatively high making this modified EZW is inefficient.

The threshold of quantization value was also noted, by others, to have obvious effect on image compression when using EZW scheme. Shingate and Sontakke (Shingate, Sontakke and Talbar 2010) investigated the effect of a chosen threshold, when using EZW, on various image parameters such as compression ratio, Peak Signal to Noise Ratio (PSNR), processing time, and reconstruction time. The results show that small thresholds improve the bit per pixel, as well as other image parameters. This is because

the dominant and subordinate pass will be used several times which in turn increases encoding time. Increasing the threshold has the opposite effect.

In 1996 Said and Pearlman (Said and Pearlman 1996) made a significant improvement on the compression ratio and PSNR of their EZW implementation. Their improved version is known as Set Partitioning in Hierarchical Trees (SPIHT). In SPIHT the quad tree is divided and partitioned into three lists; list of insignificant set (LIS), list of insignificant pixel (LIP) and list of significant pixel (LSP). Initially LSP is empty. The threshold of quantization is measured and compared with LIP to extract the significant coefficients (i.e. have magnitude larger than the threshold) to be moved to the LSP and their signs are also coded. For each pixel in the LIP, one bit is used to describe its significance. If it is not significant, the pixel remains in the LIP and no more bits are generated. Similarly, each set of LIS is tested, the insignificant sets remain in the LIS and the significant sets are moved to LIP. Finally, each pixel in the LSP is refined with one bit. The aforementioned procedure is then repeated for the subsequent resolution.

The EZW and SPIHT algorithms implement tree structures to detect the significant wavelet coefficients. Jun and Wells (Tian and Jr 1996) proposed a new algorithm known as Wavelet Difference Reduction (WDR) to encode direct location of wavelet coefficients. It can select a region on compression image to be processed for increased resolution WDR present lower bit rate of the significant coefficients indices by reducing binary expansion of significant coefficients. WDR based on two principles: the Differential Encoding (DE) and Binary Reduction (BR). In DE the difference of indices for the significant wavelet coefficients value is taken. Then, BR used to represent the binary form of data produced from DE with short binary length by dropping the most significant bit for each binary form.

Recognising the benefits of the EZW for encoding the wavelet coefficient triggered interest in using EZW quantization in the DCT domain. An alternative quantization strategy for DCT based image compression is described in (Monro and Dickson 1997), where the image is partitioned into non-overlapping blocks and DCT is applied to each block. The DCT coefficients are then rearranged to form a hierarchical sub-band structure, and the zero tree coding algorithms to generate compressed image bit stream. The reported results show that the proposed scheme outperforms the EZW and JPEG in compression ratio.

Furthermore, to benefit from the properties of both DCT and DWT transforms in data compression, a combination scheme based on DCT and DWT schemes has been presented in (Shrestha and Wahid 2010) and (Benchikh and Corinthios 2011). A hybrid DWT-DCT approach was proposed by Shrestha and Wahid in (Shrestha and Wahid 2010) for image and video compression. This approach starts by breaking the image into block size (16x16) and each block is decomposed by DWT. The high frequency sub-bands are discarded, the low frequency sub-bands are further decomposed by DWT and the high frequency sub-bands are neglected again. This process could continue but the authors stopped at level 2. As a result, from each block this methods computes low frequency LL2 sub-band to be processed further by the JPEG quantization after applying DCT. During decompression, zero values are padded in place of the detail high frequency sub-bands. This method is equivalent to applying a low pass filter before applying JPEG. Naturally, this would result in high compression ratio and high efficiency but the recovered decompressed image is of degraded quality and blurred edges as a result of discarding all high frequency sub-bands. It is worth noting that at Buckingham, Jinming Ma, developed an efficient WT-based Region Of Interest (ROI) image/video compression system (Codec) for telemedicine applications, whereby the high frequency coefficients outside the ROI were discarded (not computed) followed by a simple quantisation and entropy coding. The system was tested on the then constrained computing power PC's and demonstrated the maintenance of high quality in the ROI at the expense of degraded quality outside the ROI, (Ma 2002).

Benchikh and Corinthios in (Benchikh and Corinthios 2011) proposed another image compression technique based on a hybrid DWT and DCT in which the image is decomposed by DWT to level three. The DCT is applied on the low frequency sub-band at level three. The high frequency DWT sub-bands coefficients are compared with two thresholds. The chosen threshold is equal or less than the smallest coefficient in low frequency sub-band. The first threshold is chosen from low frequency sub-band of level two and compared with high frequency sub-bands of level one. Threshold two is chosen from low frequency sub-band of level three and compared with high frequency sub-bands of level 3. The reasons for threshold selection are not discussed/mentioned in that paper.

Note that it has been established that the high frequency sub-bands of wavelet decomposed images have a Laplacian distribution and the significant coefficients are the furthest away from their mean (Al-Jawad, Ehlers and Jassim 2006). In fact using,

various statistically determined thresholds for the high frequency sub-bands and inverting the wavelet transform results in marginal-to-modest loss of image quality. In this thesis we shall exploit the Laplace distribution property to turn the high frequency wavelet sub-bands into sparse prior to applying DCT as a way of detecting and organising the “significant” coefficients that needed for a set level of image quality.

Visual redundancy in video and images has been considered in video and image compression. As mentioned earlier, the human vision system does not respond with equal sensitivity to all visual information i.e. some information (redundancy) has less significance than other information in human vision system, such as edges or textual regions which correspond to high frequency coefficients (Annadurai 2007) (ch.5, 5.4 psycho-visual redundancies).

Consequently, several researchers have improved coding efficiency by removing some visual redundancies. Liu and et al in (Liu, et al. 2007) utilized edge detection and image inpainting for still image compression. Image inpainting is used to recover missing regions of image by smoothly propagating information from the surrounding areas. The original image is analysed in the encoder. Then some regions of image are intentionally removed and some (edge-related) information is extracted from these removed regions and an edge map is sent to decoder, the decoder will use the edge map as assistance information to guide image inpainting for image restoration. An image coding technique, known as edge-based perceptual image coding, is defined in (Niu, et al. 2012) whereby the encoder extracts the significant edges at very low bit rate from the background image to be transmitted, and then the edges region refined by a residual coding technique based on edges dilation and sequential scanning in the edge direction. The decoder will estimate the trajectories of significant edges based on low bit rate of background image. Improved quality of the compressed image comes at the expense of higher complexity due to the cost of good edge detection. The Laplacian distribution of high frequency sub-band coefficients provide an efficient, though not perfect, edge detection procedure that serve the same purpose above but at a lower cost.

Compressed sensing (CS) is a newly emerging signal sensing/processing technique which aims to efficiently acquiring and reconstructing a signal. Basically CS mitigates the stringent requirement of the Shannon-Nyquist sampling theory for the necessary number of samples needed to reconstruct sparse signals. In CS, the sparse signal of length ‘n’ can be reconstructed from far fewer than ‘n’ measurements via  $\ell_1$ -minimization or other recovery techniques. To some extent, the technique of CS

attempts to find a way to directly sense the data in compressed form data lower sampling rate rather than sampling at high rate and then compressing the sampled data. Presently, the CS is lossy compression and requires heavy computations. ( (Goyal, et al., 2008), (Gan, 2007), (Deng, et al., 2010)).

From the previous discussion, one can see that the core objective of removing redundancies and correlation in image data depends on using efficient procedures for detecting the significant data (non- zero elements) in transformed image and remembering their location in order to recover the image at the decoder. Our investigations will pursue the same objectives by investigating. Among other things, new approaches of combining DCT, wavelet transforms, and edge detection. In all these cases we need to make sure they can be incorporated into an efficient video compression and facilitate selective encryption.

### **2.1.2 Video compression**

Video compression can be seen as compression of a sequence of frames (i.e. image compression with a temporal component) but it may not be necessary that all frames go through the same procedure. Generally, video compression standards employ hybrid coding scheme which is based on removing temporal, spatial and entropy redundancies. The intra-frame (sometimes called Reference Frame (RF)) compression is used to reduce redundancy through pixels within the frame i.e. still image compression method. The resulting compressed frame is referred to as the I-frame. The temporal redundancy between sequences of frames is removed by identifying the difference between a frame and its predecessor frame and encoding these differences, this refers as inter or P (Predictive) frame. When a frame is encoded based on both previous and next frames it will be referred to as a B (Bidirectional) frame (Rajagopal and Shenbagavalli 2013).

#### **2.1.2.1 Video compression standards**

There are two formal organizations that describe video coding standards: the International Telecommunications Union (ITU-T) and the International Standardization Organization/ International Electro-technical Commission (ISO/IEC). The ITU-T video coding standards has recommended H.26x (like H.261, H.262, H.263 and H.264) to be used for real-time video communication such as videoconference and video telephony while the ISO/IEC standards, recommends MPEG-x (such as MPEG-1, MPEG-2 and

MPEG-4), to be mainly used for storage (DVD) and broadcast audio and video streams. Table2.1 summarizes the applications of each video coding standard (Zeng, et al. 2013) (Wiegand, et al. 2003) (Richardson 2011).

Standard Organization	Video coding standards	Typical Range Bit rate	Application
ITU-T	H.261	40 Kbps-2 Mbps	ISDN, Video Phones
ITU-T	H.262	1-25 Mbps	SD/HD broadcast, DVD ,HDV
ITU-T	H.263	20 Kbps-4 Mbps	Video conference, streaming internet, video over 3G wireless
ITU-T	H.264	64 Kbps- 25 Mbps	Video conference, digital TV broadcast, Mobile phone camera
ITU-T	H.26L	$\leq 64$ Kbps	Internet application
ITU-T	H.264/AVC, VC-1	20-200Kbps	Video conference, Video telephony
ISO/IEC	MPEG-1	1.2 Mbps	CD-ROM
ISO/IEC	MPEG-2	4-80 Mbps	DVD video, HDTV, Blu-ray Disk
ISO/IEC	MPEG-3	20-40 Mbps	HDTV
ISO/IEC	MPEG-4	24-1024 Kbps	Video over 3G wireless

*Table2.1 video coding standard*

We shall now briefly review some of the standards of type H.26x which are used for real-time video communication.

### **The H.26x**

This refers to a group of video coding standards that have some similarities, as well as some distinguishable differences, with MPEG-x standard. Each frame in an MPEG image stream is encoded using one of three schemes: I-frame, P-frame and B-frame. While in H.261 compression Standard, there are two types of frames: I-frame and P-frame. P-frames are predicated from previous frame P or I (forward prediction) so that the bit rate of H.261 is lower than that of MPEG.

The H.263 has been developed as a modification of the H. 261 but with some features of MPEG to achieve very low bit rates. In the H.263 encoding, the P-frame is predicted from the last I-frame while the B-frame is predicted from both the last valid frame (P or I) and succeeding P-frame as shown in Figure 2.2. This prediction technique is known as PB-frame or PB mode, which enhances compression ratio but suffers from latency and requires higher processing cost.

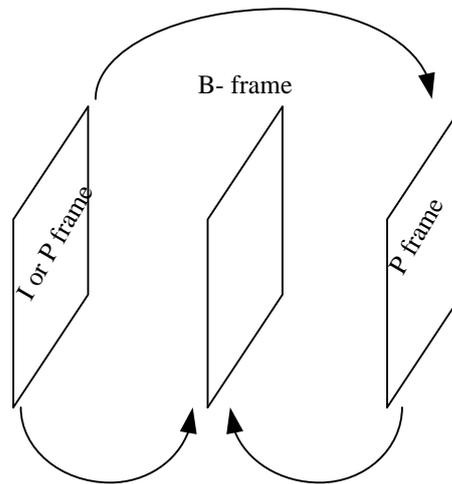


Figure 2.2 PB Mode

The H.264 is another video coding standard in the H.26x that is based on the use of Hadamard and DCT transforms. It is similar to H.263 in terms of encoding and manipulations of the P, I and B-frames. It is designed to improve the motion compensation technique of previous coding, to facilitate real-time video communication and to provide lower latency as well as better image quality for higher latency. (Richardson 2011) (Wiegand, et al. 2003) (Vetrivel, Suba and Athisha 2010) (Abomhara, et al. 2010).

#### 2.1.2.1.1 *Flash Video Technology*

The Flash video technology derives from the H.263 standard but modified according to the Sorenson's Spark codec which replaces PB-frames with prediction D frames. A D frame is based on previous P or I frame (forward prediction), while PB frames use both forward and backward prediction. The D frames do not need to re-sync to the next I frame and the video coded streams are sequences of frames like this I-D-P-D-P-D-P. Hence, the D frames codec reduces the compression efficiency but achieves real-time

processing for video streaming. Therefore, the FLV is used to deliver video and flash movies over internet (Sonnati and Sinergia 2004).

### 2.1.2.2 *JPEG2000 and MJ2*

JPEG image compression is based on the DCT transform for reducing spatial redundancies in the input image. When high compression is required too much information will be discarded and low quality image is recovered. The JPEG2000 outperforms JPEG in many ways: high compression efficiency with bit rate less than 0.25 Bit per Pixel (BPP) and highly detailed image. It is also able to decompose whole image or selected parts of image (region of interest) with maximum quality and resolution, multi-resolution representation by exploiting the properties of the wavelet transformation. JPEG2000 replaces the transform DCT and Huffman encoder used in JPEG by DWT transform and binary arithmetic encoder called MQ-coder.

The fundamental building blocks of a JPEG 2000 encoder are shown in Figure 2.3. These components include pre-processing, DWT, quantization, arithmetic coding and bit stream organization. The first step in pre-processing is converting the pixels of input image from unsigned to signed values. Then divide the input image into non-overlapping rectangular tiles of equal size. The tile size can be arbitrary up to and include the entire size of the raw image. Choosing smaller tile size will reduce the compression efficiency compared to the larger tile size, and if the tile size is too large, it requires larger memory buffers for application either by software or hardware. Every tile is compressed in four stages: (1) wavelet decompose the tile; (2) individually quantize the frequency sub-bands; (3) each sub-band is broken into blocks and these blocks are encoded through adaptive binary arithmetic coder (MQ-encoder); (4) output bits of encoder by organised in packets with headers that contain all the information such as resolution level, quality level, and which parts of code stream is secured. The header information is essential in decoding packets.

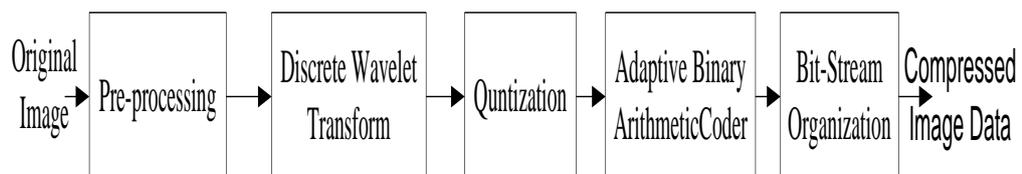


Figure 2.3 JPEG 2000 fundamental building blocks

The MQ binary-coder input differs from the arithmetic binary encoding in terms of calculating the probability of occurrence of symbols. In arithmetic encoding the probability of symbols (0 or 1) is known in advance while in MQ-coder the probability of symbols is determined by dynamic decision. The main idea of MQ-coder is to classify the input symbol (bit) as more probable symbol (MPS) or less probable symbol (LPS), if we assumed that (1) is MPS and the input symbol was (1) so the decision is MPS, if symbol does not match with it, symbol is LPS. The LPS probability of the next symbol is estimated from probability estimation standard table.

Motion JPEG2000 (also known as MJ2) is not concerned with inter-frame coding as is MPEG. In MJ2 each frame is coded independently using JPEG 2000 for still image (Acharya and Tsai 2005) (Impoco 2004) (Miyamoto 2004).

### 2.1.2.3 *Inter-frame Compression Techniques*

In most applications, the video frames include the same objects over a numbers of frames within a short period of time resulting in what is referred to as temporal redundancy. Inter-frame compression exploits the similarities between adjacent frames to reduce the temporal redundancy between video frames. The simplest inter-frame compression technique is frame sub-sampling, where only every other frame is transmitted and others are dropped. This method produces a compression factor of 2, and the receiver duplicates each received frame. Another technique is based on difference coding in which each frame of video is compared with its previous frame and only pixels which change significantly are transmitted. If the number of pixels to be transmitted is large, then this compression is infeasible. A third technique is called Block Differencing Coding. In this technique the frames are divided into blocks of pixels, and each block in the frame is compared to the corresponding block in the preceding frame. If the blocks difference is by more than a certain threshold, then this block is processed and transmitted. If the frame contains a lot of motion, many pixels will change and many mismatched blocks are produced from the encoder. So this method is almost impractical when there is a lot of motion.

In the inter-frame compression, motion compensation prediction is mostly used to reduce the temporal redundancy between video frames. This technique divides non-Reference Frames (n-RF) into blocks (Target Blocks (B)). For each B, the error distance (the average of the absolute difference between pixels in B and candidate block in

reference frame) is measured between B and the selected candidate blocks in Reference Frame (RF) to find the nearest matching block to B in the search area  $((b+2dx)$   $(b+2dy))$  of RF, see Figure 2.4. The position (vector motion) of the matched block which obtained from RF is encoded including the position of B and sent to the decoder.

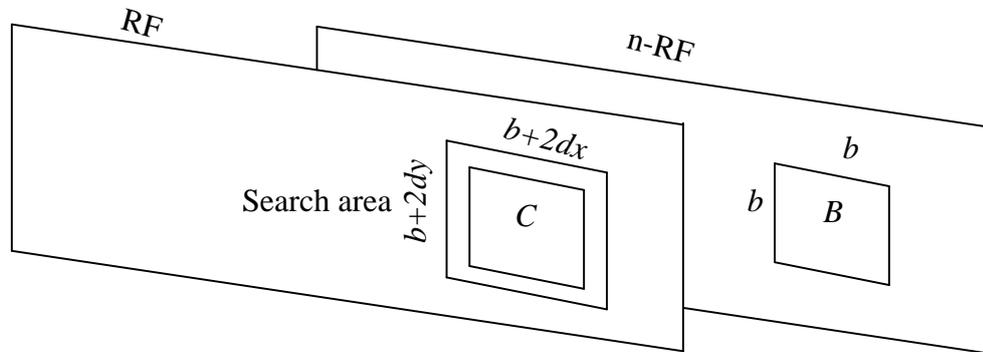


Figure 2.4 the search area

The decoder will recover the matched blocks by copying the block from RF at position determined from motion vector information (see chapter 5 for more details).

The block matching search in the motion compensation is the most time consuming part of the inter-frame encoding. Therefore, a lot of research has been done to optimize the search methods to find suitable matches for B block with fewer overlapping candidate blocks in the search area. Jain and Jain (Jain and Jain 1981) introduced The Quadrant Monotonic Search (QMS) method which is a locality-Based search. The principle of locality suggests that better matches are expected to be found near other good matches. Firstly, QMS measures the distortion between the target block B and sparse blocks C within search area which contains overlapping blocks. The distortion value increases when we move away from the best matched candidate block with B. The second step may predict where the best match is likely to be found in the neighbourhood of the best match in the first step.

The Hierarchical Motion Estimation (HME) is another motion compensation technique that attempt to improve efficiency using a multi-level refinement process whereby the motion vectors are first coarsely estimated on a sub-sampled picture. To reduce the computational burden further and overcome real-time constrains, Urban and Nezan (Urban, D forges and Nezan 2012) suggest a parallel implementation of the HME technique with motion estimation at pixel level performance by DSP. The results show there is a slight decrease in compression ratio.

Acharjee and Chaudhuri describe a new motion vector estimation algorithm which defines the motion vector for every pixel rather than blocks. It is assumed that the pixel can move around its location, at most, up to three pixels position. This assumption is based on the human visual system which has problems with recognizing fast moving objects. Consequently, the search area of RF is restricted up to three pixels on all four sides of the corresponding pixels of n-RF. Then the difference between the search area pixels and the n-RF pixel is computed, if the difference is within the threshold (intensity change is insensitive to human vision), the pixel of RF is the best match and the motion vector of that pixel is transmitted. Therefore, to get the decision that the pixel has not moved, this process will require the maximum number of iterations (= 18424) to get that pixel as a zero vector (Acharjee and Chaudhuri 2012).

H.264/AVC provides high quality encoding and decoding for streaming video frames in real-time and HDTV broadcast. There are three kinds of transformation which are used in H264/AVC codec; 4x4 integers transform, 4x4 Hadamard transform and 2x2 Hadamard transform. H264/AVC added 8x8 integers transform into high profile for higher resolution video data over High Definition (HD). Chang and Cho described a unified transform circuit to perform all inverse transform of; 8x8 and 4x4 integer inverse transform, 4x4 and 2x2 Hadamard transform. The proposed circuit exploits the similarities between transforms of H.264 decoder to achieve high performance of inverse transform (Chang and Cho 2010).

The High Efficiency Video Coding (HEVC) is a video compression derived from the H.264 standard. It has a double data compression ratio compared with H.264/AVC at the same quality. The intra-prediction modes of HEVC use a different Coding Tree Unit (CTU) size for achieving high coding efficiency. The selection of CTU size is based on the Rate Distortion (RD) optimized method. This method causes a high computational complexity in the encoder. In order to reduce the encoding time of HEVC, Kim and Younhee in (Kim, et al. 2013) proposed a fast intra-prediction method by using the difference between the minimum and second minimum of RD cost estimation based on the Hadamard transform. The results of this proposal show a 32% reduction in the intra-prediction compared with HEVC.

Finally, Zhang and Shunliang described an image/video coding based on combining CS theory into DCT to enhance compression efficiency of any algorithm which employed DCT coding like JPEG, MPEG and H263. However, the computational complexity of this implementation of CS technique has been shown not to be suitable for video real-

time transmission (Zhang, et al. 2008). This is an area of potential research that could prove to be of great benefits to image/video compression as well as analysis.

## **2.2 Image and Video Encryption**

Many data encryption algorithms have been developed and deployed throughout the centuries to protect transmitted/stored data and information. Over the last century an increasing number of ciphers have been developed to meet for the protection of digital data and communications. There are different methods that have been adopted for image encryption depending on the domain, format of signal and the expected level of security. These methods vary in their complexity and security. Among the most widely available and tested ciphers are the DES, AES, RSA and 3DES. High computational cost of such block ciphers is a major obstacle for real-time video encryption. Encrypting online video streams, of no fixed duration, imposes some restrictions on type of ciphers and/or encryption keys. Stream ciphers (e.g. LFSR's and chaotic map ciphers) rather than block ciphers are therefore more appropriate for encrypting video streams and GSM signals.

But, do we need to encrypt the entire image/video data to be assured of the security of transmission? The concept of selective encryption method that has been proposed for still image encryption, works by only encrypting selected significant coefficients from crucial transformed parts of compressed data (Uhl and Pommer 2005). The coefficients that are less important will not undermine the security of the images if not encrypted. Consequently, for video encryption, selective encryption is preferable because it reduces the encryption time dramatically without compromising security.

Generally, there are two main approaches for combined image/video compression and encryption: Sequential Compression and Encryption (SqCE) and the Joint Compression and Encryption (JnCE). We shall review both approaches later but now we review the stream ciphers that are suitable for image/video encryption.

Note that, most of the following sections deal with greyscale image and videos. However, when dealing with colour images (e.g. RGB) many researchers have proposed encrypting individual colour channels. And we should follow this tradition, although exploiting differences in human vision sensitivity to different colours may provide a mean of improving efficiency without undermining security.

## 2.2.1 Stream ciphers

The main and most important component of such ciphers is a random key generator. Linear Feedback Shift Register (LFSR) is the simplest method of generating a random key stream of any length using an initial fixed length initial secret register, a primitive polynomial and an iterative procedure that outputs one bit at a time. The generated bit stream is used to encrypt the significant parts of images/video bit stream by XORing.

### 2.2.1.1 *Linear Feedback Shift Register (LFSR)*

The Linear Feedback Shift Register (LFSR) consists of clocked storage elements (known as flip flop) and feedback paths. The LFSR is successively connected flip flop configuration with feedback from contains some flip flops output (taps) that XOR together and result is feedback into register input as shown in Figure 2.5.

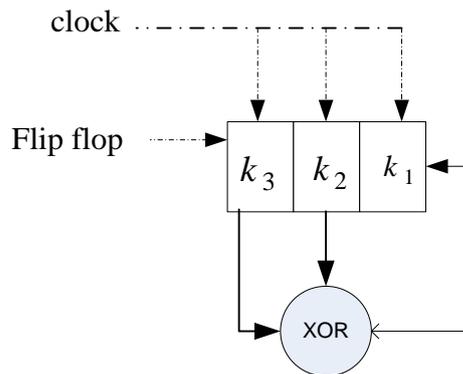


Figure 2.5 LFSR of degree 3

The length of register and positions of taps depend on primitive polynomial. For instance, if the primitive polynomial was  $x^3 + x^2 + 1$ , then the register will be composition from 3 ( the highest exponential of primitive polynomial ) flip flops and the positions of taps will be 3 and 2 in register sequence as shown in figure above. Usually, there are  $(2^n - 1)$  possible binary states produces from LFSR until the start set (called the seed of LFSR) repeats, where n is the length of LFSR.

In stream cipher, the LFSR is seeded with random binary seed call the secret key ( $k_i$ ) where  $i=1,2,\dots n$ . According to the above primitive polynomial, the  $k_3$  will XOR with  $k_2$  and the result output will be input to LFSR. Therefore, the right content of LFSR is shifted one bit to the left. Assume the secret key is  $k_3 = 0, k_2 = 1, k_1 = 0$ . The table 2.2 gives the sequence state of LFSR after clocking. (Paar and Pelzl 2009)

Initial state Secret key	clock	$k_3$	$k_2$	$k_1$
	0	0	1	0
	1	1	0	1
	2	0	1	1
	3	1	1	1
	4	1	1	0
	5	1	0	0
	6	0	0	1
	7	0	1	0

Table 2. 2 Sequence states of LFSR of degree

The main drawback of LFSR is its linearity (weakness that the each bit in a LFSR sequence is linearly related to the initial state, i.e. the initial state is easy deduced from some of the later bits in the LFSR bits sequence), and is thus vulnerable to algebraic and correlation attacks. The A5 stream cipher, which uses 3 LFSR's of pairwise coprime lengths, is used to provide confidentiality in GSM. It solves the linearity weakness by applying non-linear clocking to the cryptosystem i.e. the majority function (Chen and Gong 2012). Several other approaches have been developed to overcome the linearity weakness of LFSR. Horan and Guinee (Horan and Guinee 2006) introduced a novel stream cipher, based on five LFSRs; each of them is connected to five dynamic feedback polynomials switching block. The clocking rule of five LFSRs is based on majority function similar to the one used in the A5 cipher. The changing switch of dynamic feedback polynomials block of LFSR is based on minority function i.e. if only one LFSR is not clocked by majority function then its dynamic feedback polynomials switch is changed. Jolfaei and Mirghadri (Jolfaei and Mirghadri 2010) also proposed cryptosystem for image encryption, which is based on nonlinear filter generator to disguise the linearity which is produced by an LFSR, with large secret key, which produced 607 bit binary sequences filtered by nonlinear function and resilient function.

Zakaria (Zakaria, Seman and Abdullah 2011) presented two suggestions to improve the A5 stream cipher of GSM communication. Firstly, the taps of LFSRs of A5 are changed based on a new polynomial. Secondly, two new registers are added to the A5 structure. Therefore, the length of the proposed cipher becomes longer than A5 stream cipher, and the linear complexity of the modified cipher is increased compared with original A5. However, the registers clocking of proposed A5 are still based on majority function. In this thesis, we will improve the clocking rule of A5 based on chaotic logistic map rather than majority function.

Recently, Sathishkumar (Sathishkumar, Ramachandran and Bagan 2012) introduced a new image encryption using random pixel permutation. The pixels of the original image are scrambled using Prime Modulo Multiplicative Linear Congruence Generators (PMMLG), and then the index pixels in the image are permuted based on pseudorandom numbers which are generated from chaotic map. PMMLG sequence is generated from the input initial secret key.

Chaos theory is another source of random number generation. It has been utilized in cryptography and data encryption due to its sensitivity to initial conditions and control parameters, i.e. a small change in the input of nonlinear chaos system results in large differences in the chaos output (Kocarev and Lian 2011). In fact, chaotic maps have been widely used in image/video encryption. Li and Yu in (Li, et al. 2002) described a video encryption technique based on multiple digital chaotic systems, called the Chaotic Video Encryption Scheme (CVES). CVES is independent of any video compression and is used to achieve real-time video encryption. In this approach, a number of chaotic maps are used to generate pseudo-random signals to mask the video, and then the masked video is permuted based on the chaotic map.

The key space of chaotic logistic map is not large enough to make brute-force attack infeasible. In order to increase key space and the security level of chaotic logistic map Chen and Zhang (Chen, Zhang and Zhou 2012) proposed a new image encryption based on combining a chaotic logistic map with a sine map. The results show that the proposed approach has better chaotic behaviour than traditional chaotic logistic map, because the control parameter interval of combined map is larger than the interval of traditional chaotic. As a result, this approach will increase the complexity against the brute-force attack. In chapter 6 of this thesis, we will be investigating the behaviour of the aforementioned combination system and we will adopt it in our proposal to video encryption.

Liansheng and Wang proposed an encryption scheme based on chaotic logistic map, where two grayscale images are formed by using two different logistic maps. Firstly, one dimensional chaotic map is used to constitute random grayscale image from original image. Next, two dimensional logistic map used to convert the randomized image into two random grayscale images. Finally, these randomized images are combined with original image (Liansheng, et al. 2014).

Rohith and Bhat (Rohith, Bhat and Sharma 2014) proposed chaotic key sequence generated by sequence of logistic map and sequence of state of LFSR to image encryption, where the logistic map is iterated to the size of raw image. Then, the chaotic sequence is multiplied by 255 and converted into 8-bits word. Next, the output data of LFSR are XORed with binary data of chaotic logistic map to form key sequence. Lastly, the binary image pixels are XORed with key space. The computational time of this approach is high because it does not use the selective encryption concept.

For increased security, a cryptosystem for image encryption that uses two chaotic maps logistics and Tent map was proposed (Gopalakrishnan, Ramakrishnan and Balakumar 2014). The Tent map defined as follow:

$$y_{n+1} = \begin{cases} \mu y_n & \text{for } y_n < 1/2 \\ \mu(1 - y_n) & \text{for } y_n \geq 1/2 \end{cases}$$

Where  $y_n$  and  $\mu$  is the initial condition and control parameter respectively.

The encryption is performed in three steps; mixing, permutation and diffusion. In mixing, the chaotic map will iterate for the size of the plain image and random values of logistic map will be XORed with plain image pixel values. The indexes of mixing pixels are subsequently permuted using the iteration of Tent map. In the last round, the permuted pixels are XORed with random bits generated from both Tent and logistic map.

To secure medical images, Dridi and Bouallegue ( (Dridi, Bouallegue and Mtibaa 2014)) utilized chaotic Arnold cat map defined as follow:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab + 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \text{ mod } N$$

Where  $N$  is the number of pixels in one column / row, ‘ $a$ ’ and ‘ $b$ ’ are the control parameters. In this method, the plain image is sub-divided in two sub-images and then these images are sub-divided into blocks. The DCT, quantisation and RLE are applied to each block. Next, the S-box (Substitution-box, the input bits to S-box will transform to another output bits based on the lookup table), in the AES cipher, is mixed based on Arnold cat map. These S-boxes are used to scrambling values of RLE.

### **2.2.2 Sequential Compression and Encryption (SqCE)**

Encrypting compressed image/video streams is influenced by the adopted compression tool and standard. The format of the compressed image/video files must be taken into account, when selecting ciphers. It is worth noting, that while security of encryption depends on destroying any correlation in the input data, the success of compression procedures depends on the presence of correlation in the input data. Therefore, most SqCE schemes first compress the image/frame and then encrypt.

Zhou and Panetta (Zhou, Panetta and Agaian 2009) presented an image encryption based on edge information. The basic principle is to separate the original image into two images: one includes the edges, and the other image is without edges. Each one of them will be encrypted individually using a different encryption method. Finally, the encrypted results will be combined to form the encrypted image. This cipher can be classified as selective encryption but it is not clear why do they encrypt the non-edge image.

The lightweight encryption technique for video surveillance system proposed in (Dufaux and Ebrahimi 2006), performs video analysis module which identifies the Region of Interest (ROI). Then the video is compressed for efficient storage and transmission. At the same time, scrambling is applied in ROI. The scrambling is implemented in the transform domain by pseudo randomly flipping the sign of transform coefficients during encoding. This method can be applied to all existing video coding standards such as MPEG-4, Motion JPEG200 and H.264.

### **2.2.3 Joint Compression and Encryption (JnCE)**

During the last decade, numerous algorithms of image/video compression and encryption have been proposed based on JnCE. Pommer and Uhl in (Pommer and Uhl 2003) proposed an image compression and encryption, whereby compression is based on a wavelet packet rather than pyramidal compression schemes in order to provide confidentiality, and AES cipher is used to encrypt the header information of the wavelet packet. This method of encryption reduces the amount of data to be encrypted, because it is based on the small header information only.

The most obvious approach to JnCE is to only encrypt the coding table. For example, the outcome from the arithmetic entropy coding (AE), or the Huffman tree, can be

scrambled. Generally such an approach can also be classified as SqCE as encryption comes after the final task in compression is completed. We shall first review few such schemes.

Pande and Zambreno in (Pande, Zambreno and Mohapatra 2011) presented simultaneous coding and encryption based on arithmetic coding and Piece-Wise Linear Chaotic Maps (PWLCM), known as Chaotic Arithmetic Coding (CAC). The PWLCM is simple, has regular invariant density and good correlation function, confusion, and determinacy. It is suitable for cryptography and pseudo-random generator. It is defined by the iterative formula (Hu, Zhu and Wang 2014):

$$x_{n+1} = F(x_n, P) = \begin{cases} \frac{x_n}{q} & x_n \in [0, q) \\ \frac{x_n - q}{0.5 - q} & x_n \in [q, 0.5) \\ F(1 - x_n, q) & x_n \in (0.5, 1) \end{cases}$$

Where  $x_n \in (0, 1)$ , control parameter  $q \in (0, 0.5)$ , and  $q$  can be served as a secret key.

The AE involves recursive partitioning of the range  $[0, 1)$  in accordance with the relative probabilities of the input symbols occurrence. The CAC has the effect of scrambling the intervals without making any changes to the width of the interval in which the code word must be laid, thereby allowing encryption without losing any coding efficiency. This can be applied to most compression standards such as JPEG2000, MPEG-4/ H.264 AVC and SVC standards (Pande, Zambreno and Mohapatra 2011).

SubhamastanRao and Ravthi (SubhamastanRao, et al. 2011) has proposed a method to shuffle nodes in original Huffman tree. This is efficient and is suitable for any algorithm which employs Huffman coding like JPEG, MPEG and H264.

Unterweger and Uhl have proposed a more complex JnCE encryption method for JPEG compressed images which retains the compression efficiency. The encryption is based on three independent scrambling using AES cipher. First, scramble the order of code word value (RLE) then toggling value bits of Huffman encoding. Finally, the orders of all blocks which use the same Huffman table are scrambled. (Unterweger and Uhl 2012).

Wong and Lin in (Wong, Lin and Chen 2010) have proposed a simultaneous compression and encryption that rely on iteratively applying a piecewise linear chaotic map to generate random key stream to encrypt the arithmetic coding stream. At the same time the compressed image stream is encrypted by another pseudorandom key

stream generated from another chaotic map. Thus, are achieved in this approach, one through encoding and another through masked encoding. The proposal achieved compression/encryption speed between 1.2 to 3.4 MB/sec. The authors claim that the two level encryption increases compression ratio, is suitable for real-time processing, and is applicable for any entropy coding of MPEG-x, H.26x and JPEG.

The video codec H.264/AVC supports two types of entropy coding. First is the Context Adaptive Variable Length Coding (CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC). Shahid and Chaumont (Shahid, Chaumont and Puech 2011) proposed an approach for protection of H.264/AVC by selective encryption of CAVLC and CABAC for Intra (I) and Inter (P) frames of video. The encryption is performed by using AES with cipher feedback mode. For CAVLC, encryption is performed on equal length code words from variable length tables. The CABAC encryption is done on equal length bit strings. The encryption is performed simultaneously with entropy coding of video codec of H.264/AVC. The result shows there was an increase in computation time for encoder, less than 0.4% for both CAVLC and CABAC encryption. Therefore, the proposal can be used for real-time multimedia streaming over networks.

Alpha rooting is a function when applied to any image transformed domain increases the magnitudes of coefficients (not the phase) and produces a blurred and degraded image. Wharton in (Wharton, Panetta and Aghaian 2008) use the “inverse” of the Alpha rooting function to reduce the magnitudes of and leaving the phase of the DCT coefficients for joint encryption/compression of JPEG scheme. Encryption works by reversing the “expected” image enhancement. This encryption is easy to implement but it is vulnerable to the statistical attack as shown in the presented histograms of their encrypted images.

Simultaneous fusion of compression and encryption of multiple images was suggested in (Jridi and AlFalou 2010). The compression is based on DCT and special filtering, followed by two levels of encryption. The first level of encryption is based on grouping of DCTs of multiple images in the spectral domain and one of the input images is used as encryption key. The second level of encryption is created from the quantization of filtered DCT coefficients. The proposed method achieved PSNR as 21.718 on Lena image compared to that of JPEG as 20.69 at the very height compression ratio of 98%.

Generally, video codec H.264/AVC employed for real-time applications, such as video conferencing, use multiple optional modes for predicting luminance and chrominance

blocks of intra frame mode. Khelif (Khelif, Damak, et al., A very efficient encryption scheme for the H.264/AVC CODEC adopted in Intra prediction mode 2014) proposed an encryption scheme to secure H.264/AVC by permuting the optional modes prediction using chaotic logistic map. The experimental results show that the histograms of encrypted frames are not uniformly distributed indicating weakness against statistical attack.

Another video selective encryption technique, known as Index Chaotic Sequence based Selective Encryption of Compressed Video (ICSSECV) is described in (Batham, Yadav and Mallik 2014) whereby the indexes of compressed blocks from the (I) and (P) frames encoding are permuted before applying the entropy encoding. The random permutation is generated using chaotic logistic map. This can be applied to MPEG-x stream.

Shen and Zhuo (Shen, Zhuo and Zhao 2014) proposed a selective encryption for H.264 video streams that uses Motion Reference Ratio (MRR) of macroblock of non-Reference Frame (non-RF). The MRR is the total number of pixels in non-RFs which are replaced with pixels in Reference Frame (RF), as motion vector prediction. The MRR is divided by the total number  $T$  of macroblock pixels and compared with Average Value of MRR (AVM) of the RF. If  $MRR/T < AVM$ , then motion vector is classified as significant and the sign bits of all non-zero coefficients of significant macroblock are encrypted with AES.

Another selective encryption method for secure H.264 video frames was proposed by Khelif (Khelif, Damak, et al., Motion vectors signs encryption for H.264/AVC 2014) by changing the sign motion vectors of inter prediction frames (non-RF) only. The decision of change the sign depend on binary sequence generated from chaotic logistic map. The experimental results of this proposal did not include any security analysis.

Fadil and Yaakob (Fadil, Yaakob and Ahmad 2014) designed an encryption system for secured MPEG-2 video transmission over wireless channels. The proposed encryption was based on chaotic logistic map and neural network combined, referred to as the Chaotic Neural Network (CNN). It is applied to each motion vector produced from inter frame encoding of MPEG-2 algorithm.

The transcoding is the process of converting a media file format to another. Annop and George proposed an approach for a secure video transcoding based on correlation preserving sorting algorithm whereby, the RF and the non-RF frames which have a correlation coefficient less than a specified threshold are transmitted via secure channel

and the sorted video frame which are equal or greater than threshold are given to video coder such as MPEG and H.264. The encryption in secure channel preformed in two stages. The first stage using block shuffling in DCT domain based on chaotic logistic map then randomized arithmetic coding (Anoop, George and Deepthi 2014).

#### **2.2.4 Homomorphic Encryption**

In recent years, this new class of encryption schemes has been designed primarily to protect sensitive information in storage for privacy protection. Although, this is not related to our objectives but we include a description of it for completeness of discussion. Homomorphic Encryption (HE) refers to encryption techniques that allow computation of certain functions of the encrypted data without the need to decrypt the cipher text. It provides data security in many scenarios when users have permission to extract some information about encrypted data without decrypting. HE permits specific types of computations on encrypted data and the outcome remains encrypted. Therefore, in general the HE allows specific mathematical operations to be performed on the encrypted data without compromising the encryption. The HE can provide the same results as if it has been performed on the original data. (Lu, Varna and Wu 2014) (Fontaine and Galand 2007).

HE is useful, when sensitive images are securely transferred to a remote server such as in the case of Cloud Computing. But in order to preform specific computation on the encrypted image in the server this will require the image to be decrypted. This makes this sensitive image vulnerable to an unauthorised access.

#### **2.2.5 Security for JPEG-2000 (JPSEC)**

JPSEC is part of JPEG2000 of ISO standards. JPSEC specifies methods of employing security to JPEG2000. There are three types of JPSEC security tools: template tools, registration authority tools and user defined tools. These tools are used to implement security function. The template tools have an identifier that determines which cryptographic method is used, such as AES, DES and RSA. The registration tools have a registration authority, a unique identification number (ID) that is specified in the syntax. The user defined tools are defined by user application.

### 2.2.5.1 JPSEC framework

#### 2.2.5.1.1 JPSEC system

JPSEC consists of JPSEC creator (encryption) and consumer (decryption) as shown in Figure 2.6. The raw image is encoded by JPEG2000, and JPSEC performs security service by implementing the security tools to produce JPSEC stream. The inverse processing will be applied on JPSEC stream to recover the original image.

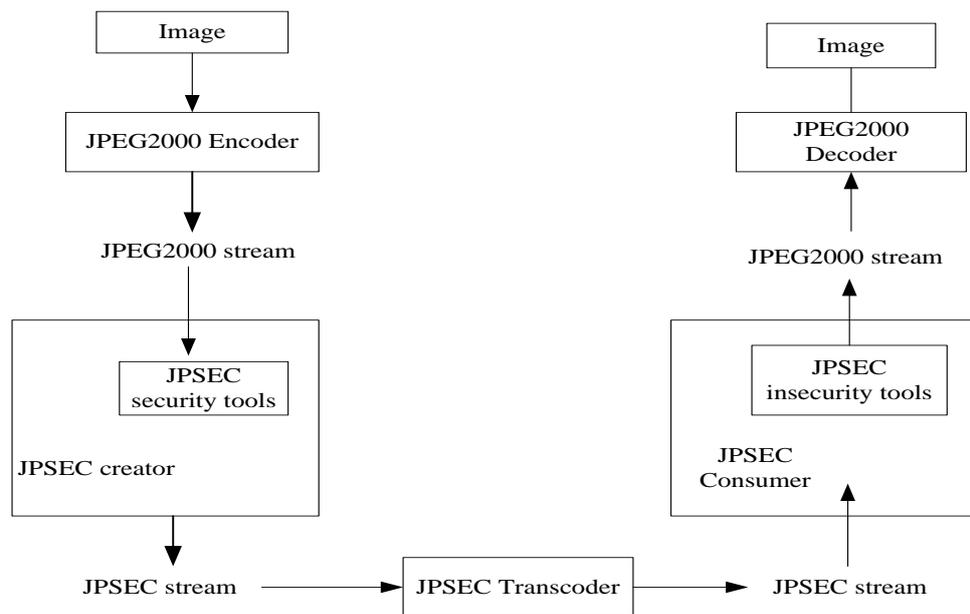


Figure 2.6 The JPSEC creator and consumer

#### 2.2.5.1.2 JPSEC stream

The JPSEC stream structure as shown in Figure 2.7 described as follows:

1. Start of code-stream (SOC).
2. Header and contents.
3. Tile size marker (SIZ) to indicate which tiles of image is protected.
4. Marker segment (SEC).
5. SEC indicates the JPSEC security tools which are used to secure the image, and it contains the Zone of Influence of protection tools (ZOI). The ZOI tool can be used to describe which part of encoded image is protected such as resolution level, components, and quality layers, region of interest and packet indices.

6. Coding style (COD) which contains information related on coding parameters such as code-size block.
7. Quantization default (QCD), contains quantize step size information in the JPEG2000 encoder.
8. Start of tile (SOT).
9. Start of data (SOD).
10. Protected image data.
11. End of code-stream marker (EOC).

SOC
SIZ
SEC
COD
QCD
SOT
SOD
Protected Data
EOC

*Figure 2.7 The structure of JPSEC stream*

JPSEC tools can be applied to protect the JPEG2000 stream. There are four domains in images encoded by JPEG2000 that need to be taken into account when encrypting. These domains are: Pixel domain, wavelet coefficients domain, the quantized domain and the code stream domain. If the wavelet or quantized domains are where JPSEC is being applied, it can either on the sign bit (by inversing the signs or randomizing) or the most significant bits. If the code-stream domain is chosen to be applied, it can either be applied on both the packet header and body or only on the packet body (Apostolopoulos, et al. 2006) (Schelkens, Skodras and Ebrahimi 2009).

In Engel and Uhl approach, only four least significant bits of byte in packet body data of encoded stream are encrypted (Engel, et al and Uhl 2009). Norcen and Uhl described an approach for JPEG2000 coded image, in which AES is used to encrypt 20% of packet body and packet header data. They evaluated that encryption is sufficient to provide a high level of security (Norcen and Uhl 2003) .

## **2.3 An over view of thesis investigations and approach.**

From the above review, we can conclude that some compression techniques have been combining the DCT and DWT transforms but in a very limited ways and for limited purposes. In this thesis we shall investigate other ways of combining those two de-correlating transforms in order to efficiently locate and map the significant coefficients in high frequency wavelet sub-bands. Also recognising, the importance of significant coefficients and edge detection for compression our investigations attempt to use knowledge about the statistical distribution of high frequency sub-bands to develop less time consuming procedures to locate significant and edge-related coefficients, as well as reduce the high frequency sub-bands into a sparse blocks. We shall also borrow and adapt the concept of phase modulation widely used in data transmission over wireless communication system to develop an innovative scheme of mapping the sparse blocks output from the various de-correlation procedures.

In the next two chapters we focus our investigations on developing and testing the performance of still image compression and encryption schemes for use on video Reference Frame (RF). Although the outcome can have their merit for use as still image compression and encryption, but these were designed specifically for processing reference frames of videos. Moreover, for practical reasons relating to our main objectives we shall not consider HD images/videos. Moreover, we delay any investigations of different size frames and colour videos to the last part of the thesis.

## Chapter 3

### Analysis of Still image Compression and Encryption

A raw image contains a massive amount of data, which requires a large storage space and a long time for transmitting over network channels that do not have sufficient bandwidth. Data compression techniques benefit from several signal/image transformations to remove spatial and/or frequency redundancies and hence are used as kind of pre-processing prior to encoding and compression. In this chapter we initiate our first attempt to combine the DCT and DWT frequency domain transforms for image compression. We shall demonstrate how to use the multi-resolution property of the DWT's to develop a very simple selective encryption scheme for still images to complement the compression.

We shall begin by illustrating the image compression based on DWT and DCT in section 3.1 and 3.2 respectively. In Section 3.3, we shall describe our proposal for the enhancement of the EZW technique by incorporating image compression and encryption.

#### 3.1 Image compression using wavelet transform

The Wavelet Transform (WT) is a frequency domain transform designed to analyse and decompose finite-energy signals at multi-resolutions. WTs differ from the Fourier transform in that they provide simultaneous spatial and frequency support. The Discrete Wavelet Transform (DWT) is a special case of the WT and provides a compact efficient representation of a signal in time and frequency. The DWT is used to decompose a signal into frequency sub-bands at different scales. The level 1 wavelet transform of images works by first transforming each row, into its low and high frequency sub-bands, followed by transforming the two resulting sub-bands column-wise. Therefore, the wavelet transform of an image partitions it into four different frequency sub-bands, namely Low-Low (LL) sub-band, Low-High (LH), High-Low (HL) and High-High (HH) sub-band. The output can be decomposed in different ways to subsequent levels. The pyramid scheme is the most commonly used decomposition, where at each subsequent level only the current level LL sub-band is analysed creating a multi-resolution frequency analysis of the input image. At resolution depth of  $k > 1$ , the

pyramidal scheme decomposes an image  $I$  into  $3k + 1$  sub-bands,  $\{LL_k, LH_k, HL_k, HH_k, LH_{k-1}, HL_{k-1} \dots LH_1, HL_1\}$ , with  $LL_k$  being the lowest-pass sub-band, (see Figure 3.1 below).

There are large number wavelet filters that are used for a variety of image processing, that differ from each other in terms of filter length and according to whether they are “orthogonal” or “bi-orthogonal”. Their use for image compression is influenced by efficiency consideration as well as effect on image quality (see Jinming Ma, 2002). For simplicity of implementation and efficiency purposes, we shall only use the Haar (db1) filter for wavelet transformation. However, in chapter 6 we shall investigate other filters and compare performances.

There are several algorithms for image compression based on wavelet transform, such as, Embedded Zero Tree Wavelet (EZW), Set Partitioning in Hierarchical Tree (SPIHT), Wavelet Difference Reduction (WDR) and Adapting Scanned Wavelet Difference Reduction, in this section the EZW algorithms is described.

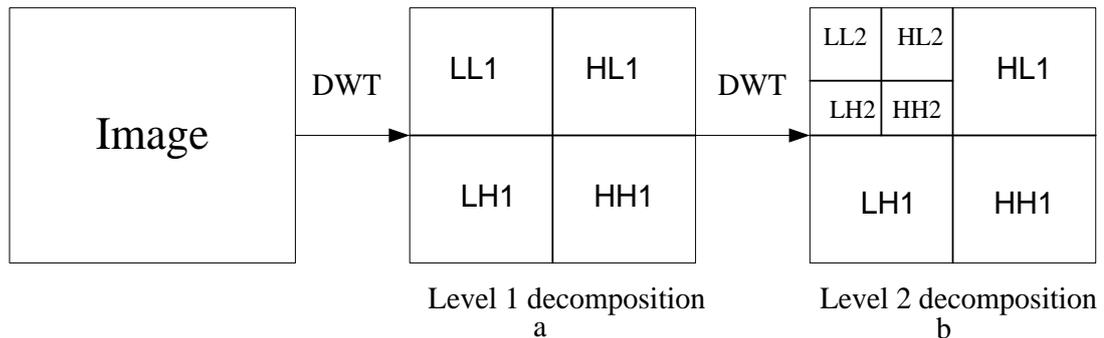


Figure 3.1 WT image decomposition : (a) first-level and (b) second-level.

The EZW algorithmic is progressive encoding scheme to compress an image into a bit-stream with high accuracy in capturing the most significant features that persist at different frequency resolution. EZW exploits an important correlation property between the wavelet coefficients of the multi-resolution signal analysis. First of all the wavelet coefficients in the LL sub-band are large compared with coefficients in the higher frequency sub-bands. Secondly, each coefficient in a low level non-LL sub-band has four descendants in the next higher sub-band resolution whose significance are inter-related to that of the parent, as shown in Figure 3.2a. As shown in figure,  $X_2/HL_2$  has four descendants  $X_3/HL_1, X_4/HL_1, X_7/HL_1$  and  $X_8/HL_1$  (quad-tree) in the next higher

sub-band L1. The zero-tree method, extracts the significant coefficients from this quad-tree structure for image compression at low bit rate.

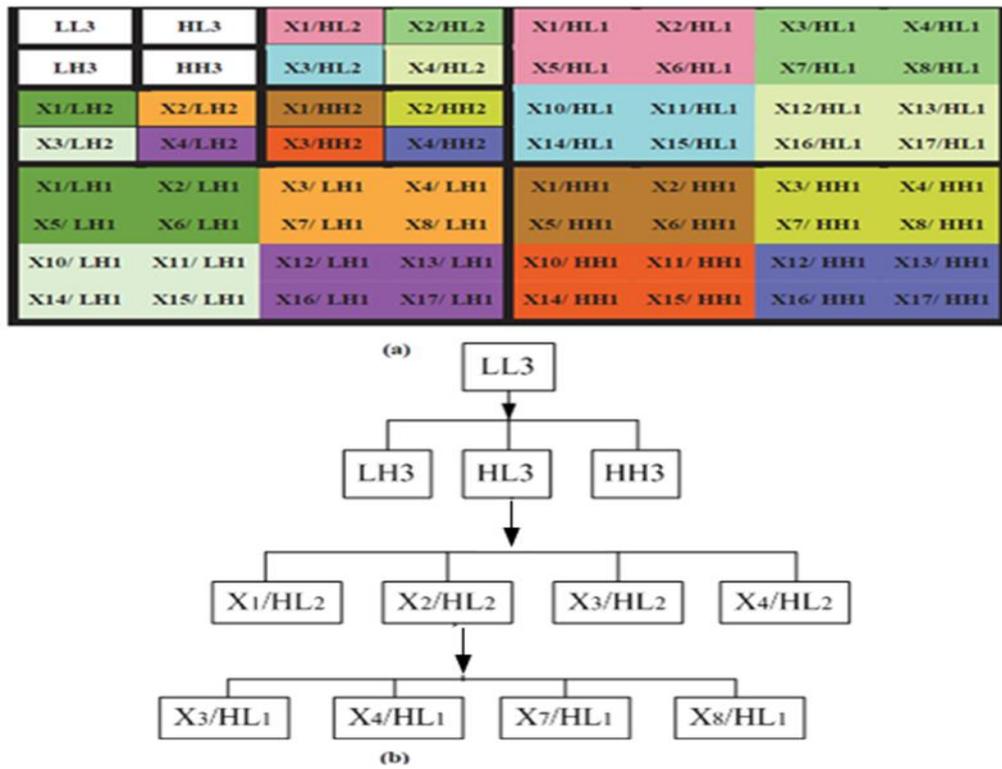


Figure 3. 2 (a) the sub-bands of WT for 8x8 image (b) the quad-tree for each coefficient HL3 and X2/HL2 sub-bands.

Therefore, by Morton scan (see Figure 3.3), the zero-tree is constructed based on the relationships between the wavelet coefficients in different sub-bands. The coefficients in LL sub-band are called parents and the other coefficients are called children, see Figure 3. 2.

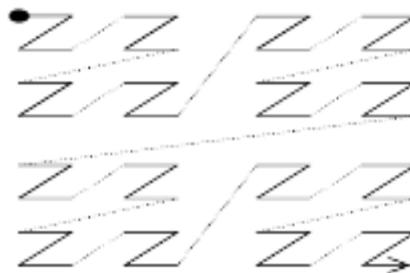


Figure 3.3 Morton scan

The EZW algorithms encode the coefficients of the zero-tree in descending order, in multiple passes, and for every pass a threshold is chosen against all wavelets coefficients measured. If coefficient is larger than threshold, it is encoded and removed

from the transformed image; otherwise it is left for next pass. In the next scan the previous value of threshold is halved.

The main steps of the EZW algorithms are as follows:

**Step 1: Initialize threshold;** choose the initial threshold,  $T = T_0$  such that all coefficient value are less than  $T_0$  and at least one coefficient is equal to or greater than  $T_0/2$ , and calculated by applying this formula

$$T_0 = 2^{\lceil \log_2 (\max(|w|)) \rceil} \quad 3.1$$

Where,  $w$  is the wavelet coefficient.

**Step 2 Update threshold:** let  $T_k = T_{k-1}/2$  ;  $k=0, 1, 2, \dots$ , iteration If  $k=0$  skip this step

**Step 3 Dominant pass (Significance pass):** Morton Scan through coefficient values and  $m$  is the scanning index. Test each value  $w(m)$  as follows;

the wavelet coefficient  $w(m)$  is compared with threshold  $T_k$ , if it is larger than  $T_k$ , it is encoded as significance coefficient and assigned quantization value  $w_Q(m)$  equal to value of threshold, and also assigned symbol (p) to significance coefficient if its value positive or symbol (n) if its value negative and these significance coefficient are removed from transformed image. If the absolute value of coefficient is less than threshold  $T_k$  and all its descendants are less than  $T_k$ , then assign symbol (R) zero-tree root, in this case only symbol R is sent to the decoder and the descendants are not encoded. If at least one of descendants is larger than threshold  $T_k$  then symbol (I) is assigned which means isolated zero, and the descendants will be encoded.

**Step 4 Refinement pass (subordinate pass):** scan through significant values assigned in step3 with the higher threshold value  $T_j$ , for  $j < k$  (if  $k=1$  skip this step). For each significance value  $w(m)$ , do the following

If  $|w(m)| \in [w_Q(m), w_Q(m) + T_k)$ , then Output bit 0

Else if  $|w(m)| \in [w_Q(m) + T_k, w_Q(m) + 2T_k)$ , then output 1

Replace value of  $w_Q(m)$  by  $w_Q(m) + T_k$ .

In refinement pass or known subordinate pass, the significance coefficients which are specified in significance pass will be quantized by quantization transform, in which all significance coefficients are scanned and (0) bits will be assigned for the coefficients  $w(m)$  which are located within intervals  $[w_Q(m), w_Q(m) + T_k)$ , and binary 1 for

coefficients that are located within intervals  $[w_Q(m) + T_k, w_Q(m) + 2T_k)$ . Therefore, in each refinement pass the quantization value of significance coefficient is approaching to the real value of coefficient, see Figure 3.4.

**Step 5: Loop: Repeat** steps 2 through 4.

The output from each scan is like the following stream:

$T_0$ , P, n, n, n, n, p, p, p, p, R, R, R, R, R, I, I, I, R, R, R, for significance pass  
 0, 1, 1, 1, 0, 1, 0, 0, 1 for refinement pass (Rao and Yip 2010) (Valens 1999)

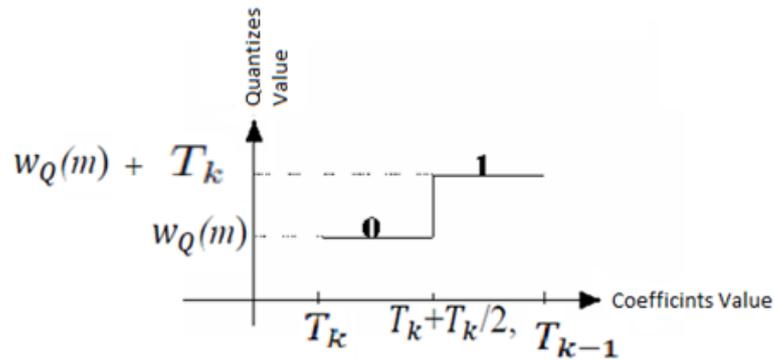
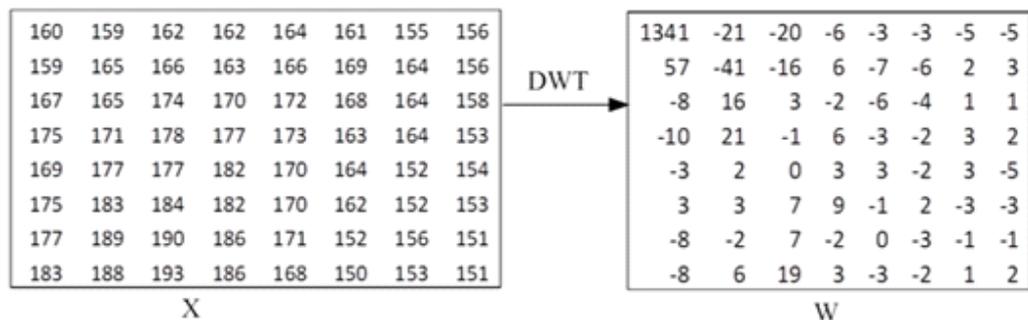


Figure 3.4 Quantization transforms

**Example**

To illustrate EZW methods, we select a block (X) size (8x8) pixels from Lena image. The DWT (Harr, level 3) has been applied to X and the result is W as shown in Figure 3.5a. The coefficients of W are scanned in order index as shown in Figure 3.5 b. The initial threshold  $T_0$  is determined by applying equation 3.1 and the result  $T_0 = 1024$ .



1	2	5	6	17	18	21	22
3	4	7	8	19	20	23	24
9	10	13	14	25	26	29	30
11	12	15	16	27	28	31	32
33	34	37	38	49	50	53	54
35	36	39	40	51	52	55	56
41	42	45	46	57	58	61	62
43	44	47	48	59	60	63	64

A  
B

Figure 3.5 (a) a block 8x8 pixels from Lena image that is DWT (b) Three-level scan order

Based on step 3, The  $W$  coefficients are scanned and a symbol is assigned for every coefficient, if the coefficient is larger than  $T_0$  a 'p' (+) is coded, if the coefficient is smaller than  $-T_0$  a 'n' (-) is coded, if the coefficient and its descendants is the root of a zero-tree (smaller than threshold) then an R is coded and all descendants are not scanned. If the coefficient is smaller than the threshold but it is not the root of a zero-tree (one of descendants is larger than threshold), then I (isolated zero) is coded. Finally, all coefficients that are assigned as significance coefficients (coefficients which are assigned p or n) are taken out and processed in subordinate pass (step-4) and their locations in the block are filled with zeroes.

For instance, in the first iteration of EZW, the  $W_{(1,1)}$  coefficient (1341) is greater than  $T_0$  (1024), therefore only  $W_{(1,1)}$  will be assigned as p and the children:  $W_{(1,2)}, W_{(2,1)}, W_{(2,2)}$  (-21, 57, 41) are assigned as R. Thus, the output symbols of first dominant pass D1 are p, R R,R.  $W_{(1,1)}$  will be processed in subordinate pass and  $W_{Q(1,1)} = 1024$ . As result,  $W_{(1,1)}$  will be in the interval  $[W_{Q(1,1)}, W_{Q(1,1)} + 1024)$  i.e.  $1341 \in [1024, 2048)$  then 0 is output from first subordinate pass ( $S1=0$ ), see Figure 3.6. This iteration is repeated with update threshold until the exact bitrate specified by the user has been reached. It can be seen from Figure 3.6 that all descendants of (R) are assigned ( $\times$ ) to indicate they are not scanned.

P	R	x	x	x	x	x	x
R	R	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x

O	R	x	x	x	x	x	x
P	n	x	x	x	x	x	x
R	R	R	R	x	x	x	x
R	R	R	R	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x

A  
B

Figure 3.6 (a) Iteration 1, threshold = 1024. (b) Iteration 6 threshold = 64

## 3.2 Image compression using DCT coding

The DCT is one of the most commonly used transforms for image compression and it is baseline algorithm of the JPEG coding standard. In this section we will briefly describe the image compression based on DCT transformation.

### 3.2.1 The encoding process

The block diagram of image based DCT coding is shown in Figure 3.7. Firstly, the image data is partitioned into blocks of 8x8 pixels and the DCT is applied to each block. After the DCT is transformed, the transformed blocks are quantized. Finally, the quantized blocks are entropy encoded and output as compressed data.

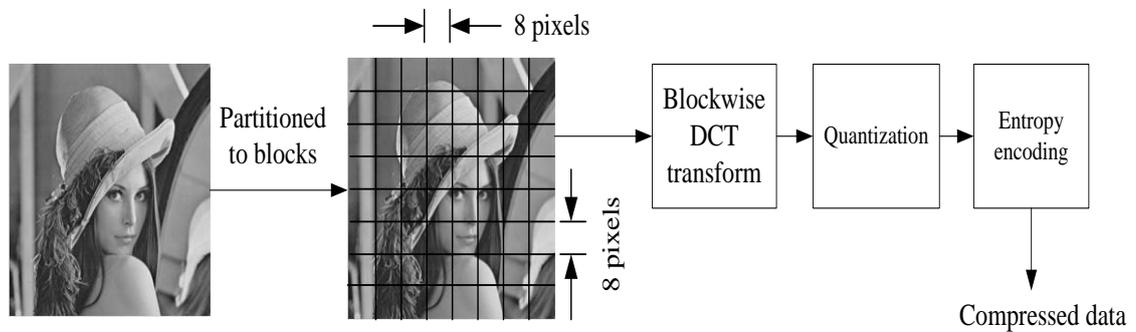


Figure 3.7 Block diagram of image encoding based DCT coding

### 3.2.2 DCT transform

The image is comprised from pixels and the correlation between neighbouring pixels is high. The main work of DCT is de-correlating the neighbouring pixels and concentrating the significance information in low frequency region. The two dimensions (2-D) DCT of 8x8 blocks are defined as follows:

$$S_{uv} = \sqrt{\frac{1}{4}} C_u C_v \sum_{i=0}^7 \sum_{j=0}^7 s_{ij} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} \quad 3.2$$

And the inverse DCT (IDCT)

$$s_{ij} = \sqrt{\frac{1}{4}} \sum_{u=0}^7 \sum_{v=0}^7 C_u C_v S_{uv} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} \quad 3.3$$

$$C_u C_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases}$$

Where

$s_{ij}$  : is the value of the pixel at position ( i, j) in the block.

$S_{uv}$  : is the transformed (u, v) DCT coefficient.

The transform matrix of DCT (DCTM) is driven from equation 3. 2, we ignore  $s_{ij}$  and varying  $u$  and the DCTM will show as fellow:

$$\text{DCTM} = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

The values of the DCTM are called cosine basis functions. The rows of these basis functions are shown in Figure 3.8. It can be seen that the frequency is increasing as we advance down the rows .i.e. the frequency of the DCT coefficients increases as we go from top to the bottom. Consequently, the DCT compact the significance coefficients (low frequency) in the top left triangle of blocks. This property is exploited in quantization processing to achieve lossy compression.

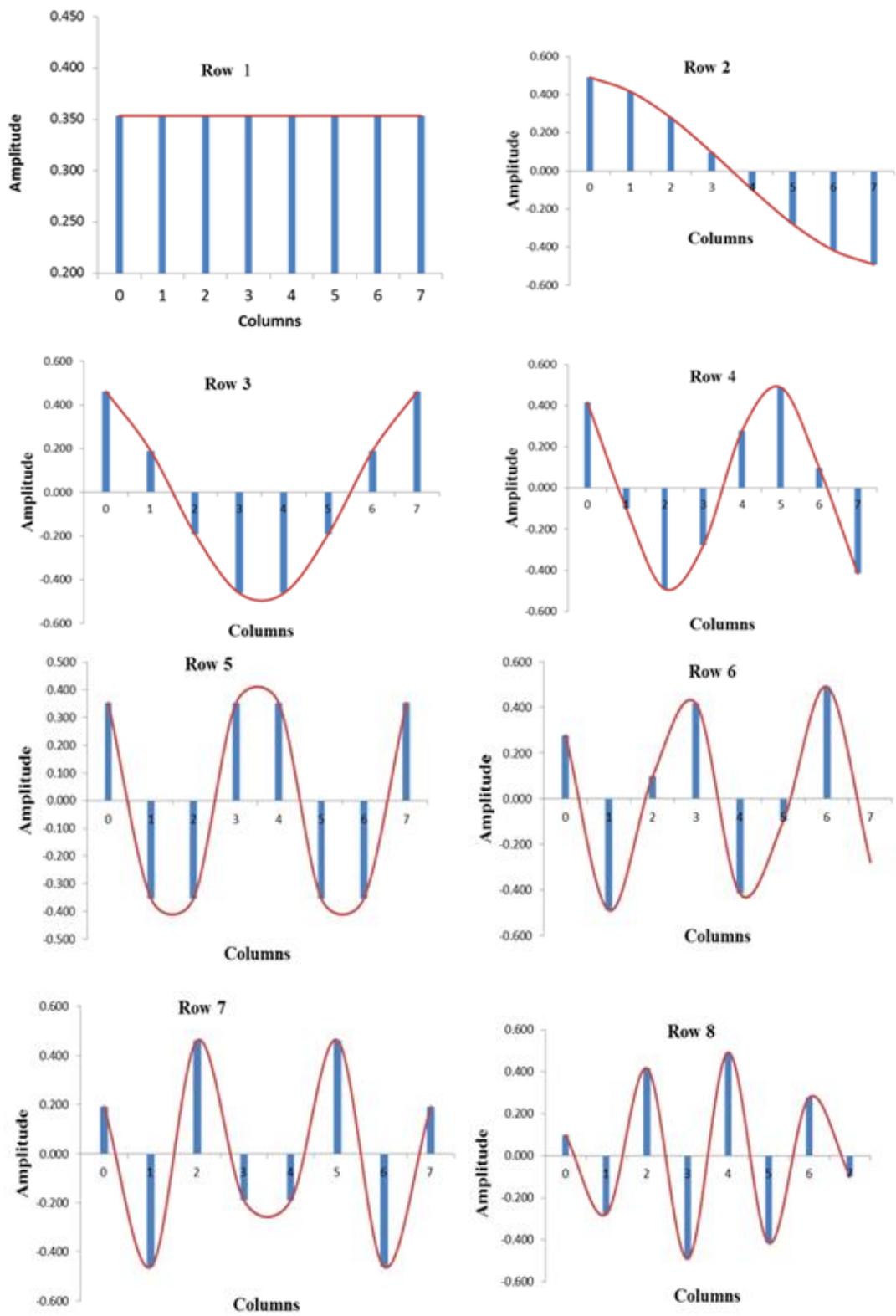


Figure 3.8 the DCT basis function of 8x8 arrays

### 3.2.3 Quantization

After the image transformation, the resulting coefficients have a large numbers; quantisation is used to reduce the number of required for coefficients representation. Small numbers take less space than large ones, so quantization generates compression. Therefore, the transformed blocks are quantized by using an 8x8 quantization table. Each value in the transformed block is divided by the matching value in the quantization table, and the result is rounded to the nearest integer. For instance, two sample quantization tables are shown in Figure 3.9. It can be seen from Figure 3.9 that the quantization factor (step size) generally increases as we progress from the top left corner to bottom right. As a result, the quantization error will be introduced in the high frequency coefficients more than in low frequency coefficients. The decision on the relative size of the step sizes is based on how the human visual system will sense these errors in the quantized coefficients. Furthermore, the quantization table will be specified by the required quality of decompressed image.

3	2	2	3	5	8	10	12
2	2	3	4	5	12	12	11
3	3	3	5	8	11	14	11
3	3	4	6	10	17	16	12
4	4	7	11	14	22	21	15
5	7	11	13	16	12	23	18
10	13	16	17	21	24	24	21
14	18	19	20	22	20	20	20

80	60	50	80	120	200	255	255
55	60	70	95	130	255	255	255
70	65	80	120	200	255	255	255
70	85	110	145	255	255	255	255
90	110	185	255	255	255	255	255
120	175	255	255	255	255	255	255
245	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

A  
B

Figure 3.9 Quantization tables (A) Low compression high quality (B) high compression Low quality

Thus, the quantization process converts many high frequency coefficients (non-significance) to zeroes in the quantized blocks. Finally, the quantized blocks are scanned by zigzag scan where the low-frequency coefficients are read first and the high-frequency coefficients last as shown in Figure 3.10. Therefore, the zeroes value data will be clustered at the end of data sequence. These redundant zeroes are reduced by the run-length-encoding (RLE). The compressed sequence of RLE is encoded by either Huffman or arithmetic encoding (entropy encoding) to form the final compressed data.

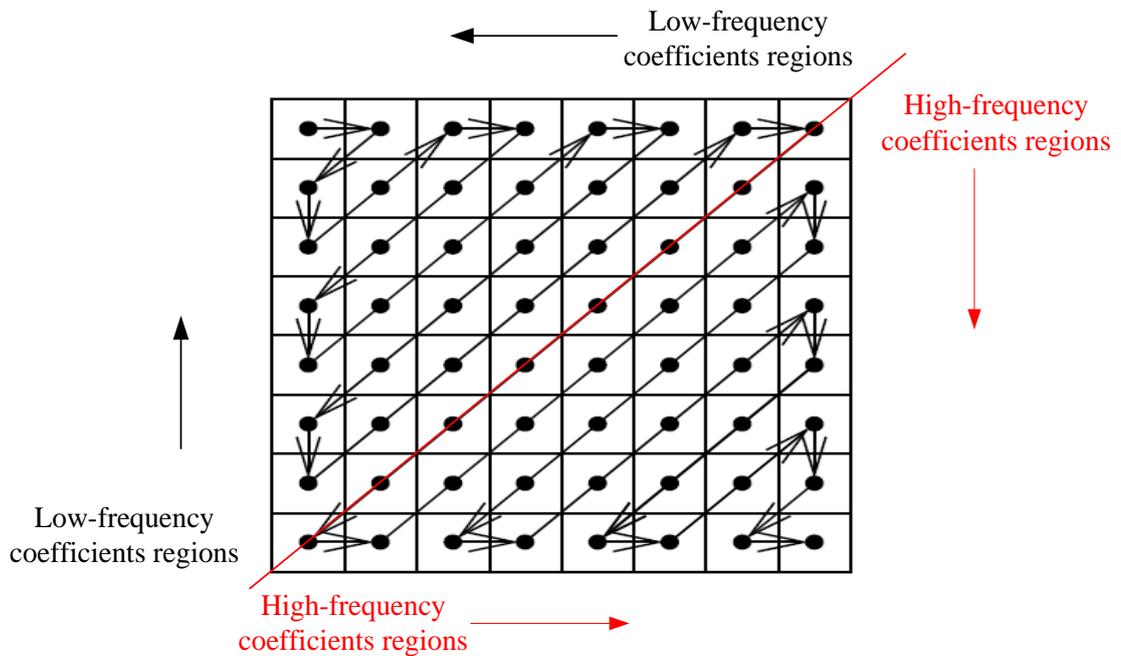


Figure 3.10 Zigzag scanning order of quantized DCT coefficients sequence

### 3.3 Optimized EZW technique for image compression and encryption

In this section we describe a new method for image compression and encryption based on combined EZW and DCT algorithms. Such encoding will improve the compression ratio and reduce the encoding time compared with individual EZW algorithm. The encryption method is based on determining the crucial parts obtained by the compression algorithm and only encrypting these parts.

#### 3.3.1 A Combined DCT and EZW image compression scheme

In order to justify the combination of DCT and EZW we shall use the example given in section 3.1. The EZW algorithm shows that if the coefficient (parents) and corresponding descendants are smaller than the chosen threshold ( $T$ ), the coefficient is coded as zero-tree root (R) and the descendants are not encoded in the current iteration and only R will be sent to the decoder. On the other hand, when the coefficient has significance descendant ( $>T$ ), the coefficient and the descendants are encoded. Thus, when the value of parents and their descendants are not in decreasing order, the compression ratio and encoding time of EZW tend to increased.

Now we go back to the example in section 3.1. In iteration 8, the threshold was  $T_8 = 8$  and there were three descendants of parents  $W_{(4,1)}$  and  $W_{(3,2)}$  are larger than or equal to  $T_8$ . These descendants are  $W_{(6,4)}$ ,  $W_{(7,1)}$  and  $W_{(8,1)}$  as shown below. Therefore, there were 20 symbols output from significance pass

0	0	0	-6	-3	-3	-5	-5
0	0	0	6	-7	-6	2	3
-8	0	3	-2	-6	-4	1	1
-10	0	-1	6	-3	-2	3	2
-3	2	0	3	3	-2	3	-5
3	3	7	9	-1	2	-3	-3
-8	-2	7	-2	0	-3	-1	-1
-8	6	0	3	-3	-2	1	2

The main disadvantage of the EZW is the number of iterated scanning which could reduce the efficiency for many images. On the other hand, in section 3.2.2, the DCT transform tends to concentrate significant information in low frequency components. Our proposed scheme exploits the well-ordered structure of the DCT coefficients to arrange the wavelet coefficients in decreasing order and enhancing the compression efficiency of image compression based on EZW algorithm, i.e. we propose to apply the DCT on the DWT sub-bands. We call this method a Modified EZW (MEZW).

### 3.3.1.1 *The MEZW Compression scheme*

The approach MEZW starts by subdividing the raw image into blocks size (8x8) pixels. Each block is then DWT the block and applies the DCT on the transformed coefficients, which are then quantized. The DCT concentrates the low frequency coefficients in the top left of block which represents the parents and the high frequency coefficients in the right bottom of the block which represents the children. Hence, during all iterations of EZW encoding the parents are mostly larger than descendants. This might be improved through both compression efficiency and encoding time. Finally the EZW algorithm is applied to transformed block, and for further compression we use lossless coding like Huffman coding.

The proposed MEZW compression is described in the following steps:

1. Partition the input image into blocks size (8x8) pixels.

2. Apply DWT and DCT to each block respectively.
3. Quantize each transformed block by using quantization table.
4. Apply embedded zero trees encoding to each block.
5. Apply Huffman encoding to the outputted data from EZW.

Figure 3.11 illustrate the block diagram of the MEZW algorithm mentioned above. The image is reconstructed by reversing the compression procedure.

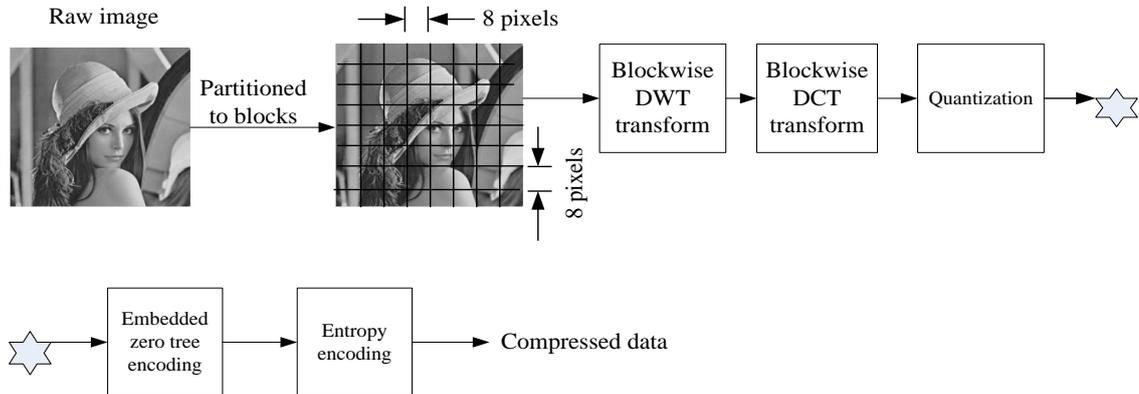


Figure 3.11 The Block diagram of proposal compression system

To illustrate the comparison between our proposal and EZW method, we applied these methods on the same block shown in Figure 3.5a. As illustrated in section 3.2, the data produced from encoder are shown in two forms; symbols(R, I, n and p) which are produced from dominant pass (D) and binary form which are produced from subordinate pass (S). Therefore, we represent each symbol (R, I, n and p) by two bits. The results of this experiment are shown in Table 3.1.

Method	Number of iterations required	size of data produced from D	size of data produced from S	Encoding Time
EZW encoder	11	448	113	0.112
MEZW encoder	5	400	91	0.049

Table 3.1 results of EZW and proposal encoder

From Table 3.1 it can be shown that the number of iterations required in EZW is considerably larger than that needed by MEZW. This algorithm also helps reduce the compressed size. In turn, the consumed time of encoding in MEZW is significantly less than EZW. These results seem to be a consequence of the applied DCT which organizes the wavelet coefficients in descending order.

### 3.3.1.2 *Experimental results*

In this section, we compare the performance of MEZW scheme and EZW encoder using a reasonable size sample of images. For the experimental work, 30 different images have been used for analysis in terms of Compression Ratio (CR), Compressed image Quality and consumed time. The size of the tested images was (258x320) pixels and all images were first converted to grayscale as shown in Figure 3.12. The proposed scheme was implemented using MATLAB V 7.10.0 (R2013a) and achieved on Intel (R) i5 processor 3.2 GHz and RAM 16 GB.

In order to evaluate the performance of MEZW scheme in comparison with EZW standard, the Peak Signal-to-Noise Ratio (PSNR) is used as quality criterion. PSNR is driven from the Mean Square Error (MSE) and defined as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad 3.4$$

MSE between the two images X and Y is thus defined as:

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (Y - X)^2 \quad 3.5$$

$M \times N$  is the image size.

Although the MSE is not a perfect criterion for the human vision system it is still frequently used as a measure of the quality of recovered image.

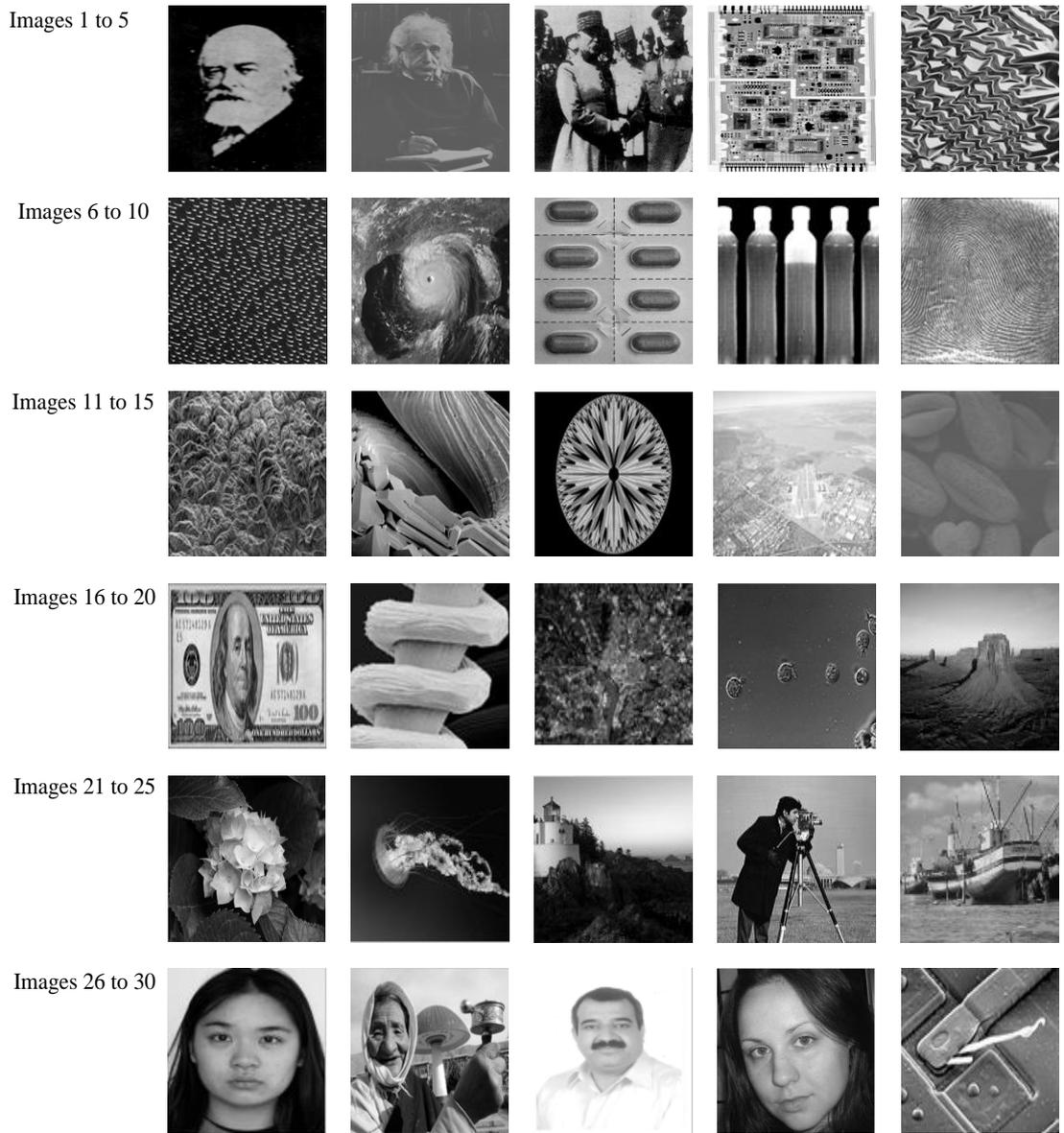


Figure 3.12 Test images

Figure 3.13, compares the PSNR, CR and coding time achieved by the MEZW and EZW technique when applied to these images. It can be seen the PSNR of proposed MEZW is generally improved compared to the EZW method. For instance, the PSNR of recovered image 2 in MEZW method is less than in EZW. On other hand, the CR of image 2 that is achieved in MEZW is better than in EZW as shown in Figure 3.13. According to Figure 3.13, the encoding time of MEZW for all tested image is less than encoding time of EZW. Generally the results demonstrated that the use of DCT combined with DWT tends to improve the CR, quality and consumed time encoded.

Table 3.2 shows the achieved PSNR, CR, time encoding and bit per pixel (bpp) in proposed MEZW and EZW.

MEZW method						EZW method				
Image	PSNR/dB	MSE	bpp	CR	Time/Sec	PSNR/dB	MSE	Bpp	CR	Time/Sec
1	41.473	4.633	1.190	0.149	26.355	40.256	6.131	2.050	0.256	31.272
2	40.319	6.043	2.780	0.347	25.759	41.678	4.419	3.120	0.390	32.338
3	40.042	6.440	2.290	0.286	27.068	36.881	13.336	4.803	0.600	33.273
4	39.980	6.532	3.219	0.402	28.250	36.793	13.607	6.168	0.771	35.869
5	39.853	6.727	2.871	0.359	27.706	36.249	15.424	6.361	0.795	36.018
6	39.906	6.645	2.185	0.273	26.953	36.133	15.842	6.083	0.760	35.519
7	39.908	6.642	2.524	0.315	26.872	35.761	17.260	4.866	0.608	33.343
8	39.855	6.724	3.187	0.398	27.936	36.860	13.400	4.115	0.514	32.501
9	41.114	5.031	2.257	0.282	26.135	36.719	13.841	3.297	0.412	31.555
10	39.843	6.742	3.159	0.395	27.907	34.261	24.379	7.178	0.897	37.660
11	39.847	6.735	2.586	0.323	27.728	35.172	19.763	7.193	0.899	37.324
12	40.114	6.334	2.599	0.325	27.066	35.268	19.334	4.626	0.578	33.085
13	42.018	4.086	1.725	0.216	25.558	38.364	9.478	4.102	0.513	32.410
14	39.926	6.614	3.654	0.457	28.667	38.372	9.461	5.571	0.696	36.715
15	39.906	6.645	2.895	0.362	25.329	37.340	11.997	2.747	0.343	29.375
16	39.923	6.619	3.209	0.401	28.315	36.231	15.489	6.013	0.752	35.081
17	40.004	6.497	2.743	0.343	27.765	38.632	8.909	3.613	0.452	31.480
18	39.869	6.702	2.185	0.273	27.036	35.876	16.805	6.907	0.863	37.599
19	40.466	5.842	2.946	0.368	27.872	39.688	6.988	3.423	0.428	31.220
20	39.907	6.644	2.472	0.309	27.364	38.002	10.302	4.034	0.504	33.768
21	39.916	6.630	1.943	0.243	26.673	37.852	10.663	3.948	0.493	31.728
22	39.946	6.583	1.367	0.171	26.028	39.894	6.663	3.229	0.404	31.365
23	39.993	6.514	2.182	0.273	26.689	38.433	9.328	3.615	0.452	32.128
24	39.832	6.759	2.725	0.341	27.592	35.767	17.232	4.174	0.522	32.372
25	39.849	6.732	2.978	0.372	28.314	36.883	13.330	4.723	0.590	33.424
26	39.982	6.530	2.212	0.276	26.956	37.925	10.486	3.421	0.428	30.701
27	39.875	6.692	2.889	0.361	27.635	37.717	11.000	4.676	0.585	33.373
28	41.091	5.058	3.902	0.488	28.565	42.517	3.643	3.164	0.396	30.941
29	39.775	6.849	2.214	0.277	26.925	38.212	9.816	3.377	0.422	31.091
30	39.855	6.723	2.844	0.356	26.522	35.382	18.832	4.720	0.590	33.458
Mean	40.146	6.332	2.598	0.325	27.185	37.504	12.572	4.511	0.564	33.266
STD	0.547	0.697	0.605	0.076	0.894	1.920	4.878	1.373	0.172	2.243

Table 3.2 Compression results after encoded by MEZW and EZW

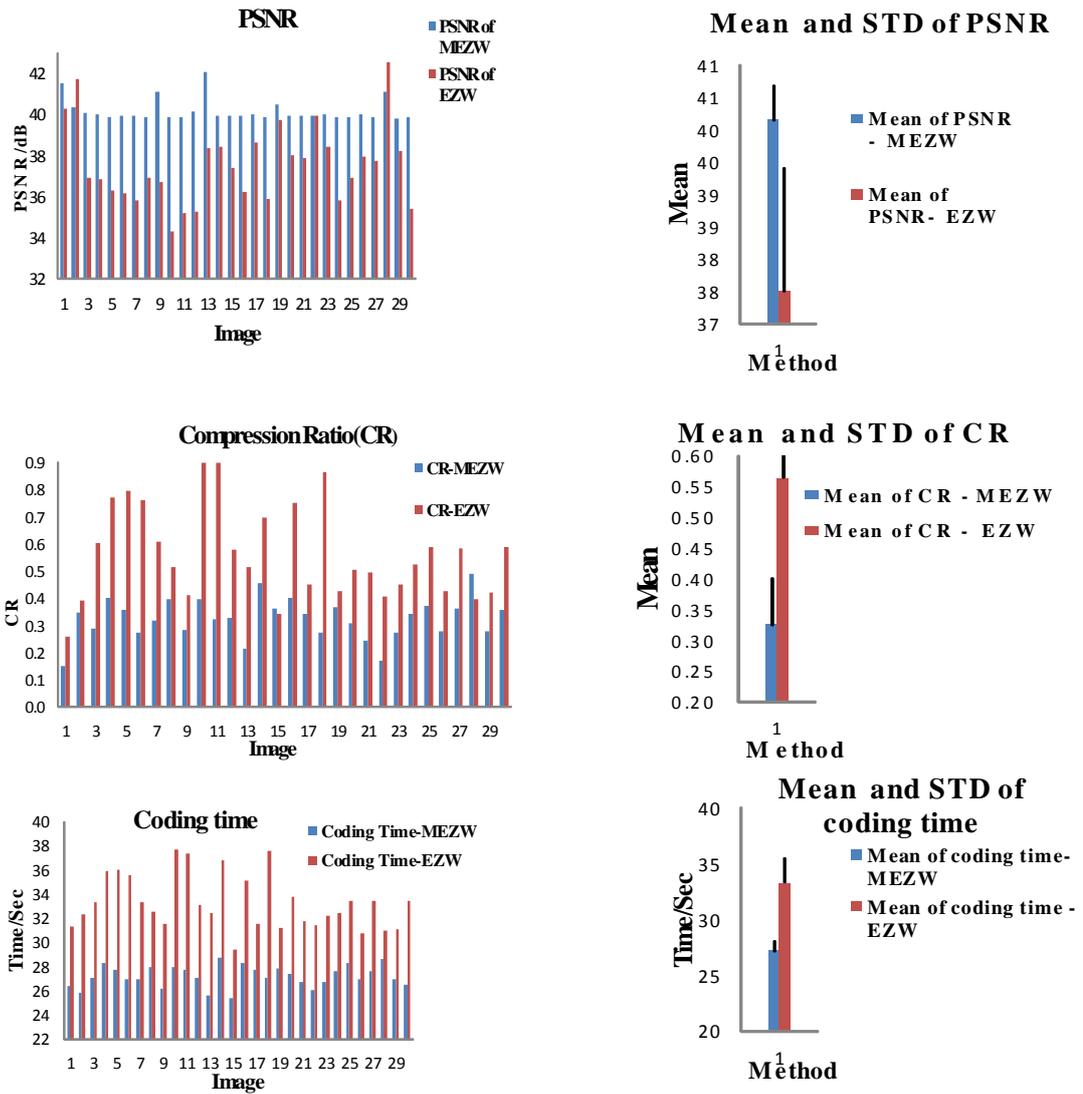


Figure 3.13 the comparison results between MEZW and EZW

### 3.3.2 The Encryption Component of the MEZW

As mentioned earlier, the traditional block ciphers are less efficient/suitable than stream ciphers for image/video encryption in constrained environment. Moreover, instead of whole image encryption, selective encryption is sufficient for image security. In MEZW coder, the decoder can infer the wavelet coefficients from initial threshold and the data stream output from both dominant and refinement pass. Generally, the initial threshold will be sent once for the encoded blocks. Therefore, the initial threshold is exploited in the MEZW Encryption (MEZWE) proposal as a crucial link to the compression algorithm. Thus only the initial threshold will be encrypted by XOR with ten bits which are produced by LFSR as shown in Figure 3.14.

This is the simplest and constant time form of selective encryption in MEZWE instead of encrypting the coefficients. It has similarity with the encryption proposed by (Pommer and Uhl 2003), which only encrypts wavelet packet header.

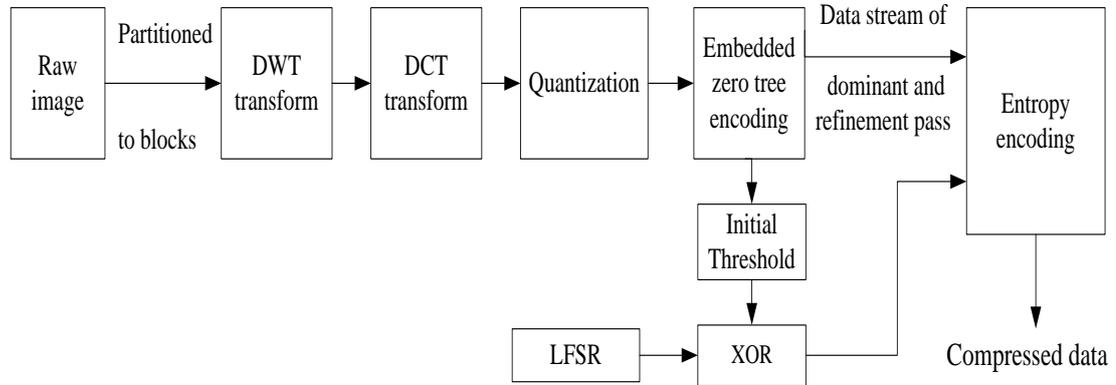


Figure 3.14 The scheme diagram of MEZWE system

### 3.3.2.1 *Experimental Work and Analysis of Results*

The proposed encryption approach was applied to the same images, tested in section 3.3.1.2. The effect of the encryption component on the MEZWE compressed and encrypted images are analysed using histogram and correlation analysis.

#### 3.3.2.1.1 *Histogram analysis*

Histogram analysis is a statistical analysis which reveals the distribution of image pixel values. The attacker may use the histogram analysis to deduce the plain pixels; this kind of attack is known as statistical attack. When the histogram of the encrypted image is approaching a uniform distribution it increases the complexity of statistical attack and reduces its chance of success. Test results show that the histogram of encrypted images by MEZWE tends to be uniform, which increases the difficulty of the statistical attacker to deduce the pixels values from encrypted image. Figure 3.15 shows some typical results.

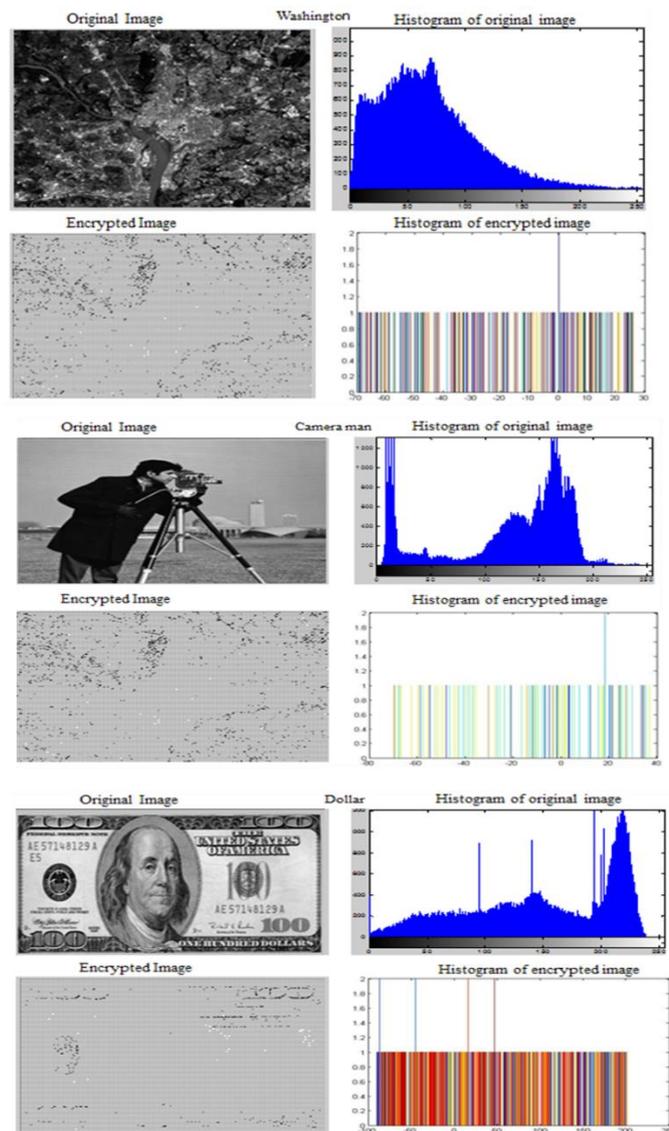


Figure 3.15 Histogram of original and encrypted image

Usually, after encryption is applied, the pixels values of encryption image tend to be change compare to those values in original image. Such change may be irregular. So, the higher change in pixels value, reflect the effective of encryption scheme and hence the quality of encryption. Therefore, the quality of encryption may describe in terms of the total deviation (changes) in pixel values between the original and encrypted image.

The Maximum Deviation Measuring Factor (MDMF) is often used to estimate the quality of encryption scheme. The MDMF maximizes deviation between the histogram of original and that of the ciphered image. Generally, when the histogram of encrypted image is different from the histogram of the unencrypted image, the MDMF tend to increase. Therefore, when the MDMF is higher value, the ciphered image is deviated from the raw image and will increase the complexity analysis of attacker to get significant information about original image.

The MDMF is measured and proved in (H, Kalash and Farag 2007) (El and Abu 2007). The MDMF is measure as follow;

1. Count the number of pixels of each gray scale value in the range 0 to 255 for both original and encrypted image (i.e. calculate the histogram distribution for original and ciphered image).
2. Calculate the absolute difference between the Histogram (HS) of original and ciphered image.
3. The MDMF is given in following equation;

$$MDMF = \frac{HS_0 + HS_{255}}{2} + \sum_{i=1}^{254} HS_i \quad 3.6$$

Where,  $HS_i$  is absolute difference between the histogram of 'i' pixel in original and encrypted image,  $i=1, 2, \dots, 254$ . And  $HS_0, HS_{255}$  is the absolute difference between the histogram of zero and 255 in encrypted and raw image.

We measured the MDMF between the histogram of Figure 3.15. The MDMF for these images was high and equal to (55387, 57652 and 59173) respectively. The high MDMF confirm that the ciphered images are deviated from the raw images, and the histogram of encrypted image does not provide useful information to statistical attack.

### 3.3.2.1.2 *Correlation analysis*

The statistical correlation is a measure of the linear relationship between two variables. It is another measure to test robustness of image encryption schemes against statistical attacks. To demonstrate the correlation between two adjacent pixels in encrypted image, the correlation ( $C_r$ ) between two vertically and horizontally adjacent pixels are calculated. A 1000 pairs of pixels are randomly selected in each direction and ( $C_r$ ) is calculated by using the following formula.

$$C_r = \frac{N \sum_{j=1}^N (x_j \times y_j) - \sum_{j=1}^N x_j \times \sum_{j=1}^N y_j}{\sqrt{(N \sum_{j=1}^N x_j^2 - (\sum_{j=1}^N x_j)^2) \times (N \sum_{j=1}^N y_j^2 - (\sum_{j=1}^N y_j)^2)}} \quad 3.7$$

Where x and y are the value of two adjacent pixels, N is the total number of pixels in the image (Kocarev and Lian 2011). The calculated values are in the real interval [-1,1]. The nearer the absolute value is to 1, are the more spatially correlated the pixels. The correlation coefficients between neighbouring pixels for unencrypted and encrypted images in the horizontal and vertical directions for a number of plaintext images and cipher text images are shown in Table 3.3. From the table it can be inferred that there is

an imperceptible correlation between adjacent pixels in the encrypted images. In contrast, there is a high correlation between adjacent pixels in unencrypted images. Therefore, the encryption images are thoroughly uncorrelated with the original images. This confirms that the encrypted images are robust against statistical attack.

Image	Original image		Encrypted image	
	Correlation coefficients		Correlation coefficients	
	Horizontal	Vertical	Horizontal	Vertical
Images-1	0.578911	0.9309	-0.00552	-0.0037
Images-2	-1	-1	-0.00242	-0.0026
Images-3	-1	-1	-0.00063	-0.0011
Images-4	-0.9503	-1	-0.00199	-0.0016
Images-5	-1	-1	-0.00122	-0.0015
Images-6	-1	-0.9997	-0.01235	-0.0139
Images-7	0.7646	1	-0.00389	-0.0017
Images-8	-1	-1	-0.00230	-0.0036
Images-9	-1	-1	-0.00602	-0.0011
Images-10	-1	-1	-0.00315	-0.0016
Images-11	-1	-0.9998	-0.00262	-0.0023
Images-12	-1	-1	-0.00283	-0.0027
Images-13	0.800438	0.9826	-0.00663	-0.0022
Images-14	-1	-1	-0.00134	-0.0013
Images-15	-1	-1	-0.00275	-0.0018
Images-16	0.74123	1	-0.00216	-0.0012
Images-17	-1	0.9083	-0.00341	-0.0009
Images-18	-0.99924	-0.9990	-0.00533	-0.0017
Images-19	-0.9990	-1	-0.00266	-0.0009
Images-20	0.6175	0.7398	-0.00334	-0.0011
Images-21	-0.9206	-0.9985	-0.00151	-0.0014
Images-22	-1	-1	-0.00247	-0.0008
Images-23	0.8081	1	-0.00231	-0.0011
Images-24	0.89205	0.6231	-0.00213	-0.0012
Images-25	-1	-1	-0.00209	-0.0011
Images-26	-0.99995	0.9999	-0.00223	-0.0015
Images-27	-1	-0.8775	-0.00267	-0.0013
Images-28	0.6411	0.5487	-0.00359	-0.0012
Images-29	-0.99965	-1.0001	-0.00214	-0.0015
Images-30	0.84279	-1.0000	-0.00096	-0.0018

*Table3.3 The correlation coefficient between two adjacent pixels in original and encrypted image.*

### 3.3.3 Conclusion

In this chapter, we introduced a hybrid EZW and DCT algorithm for joint image compression and encryption. We found that the DCT reduced the number of iterated loop scans of the EZW encoding. The experimental results show that the performance of MEZW method is significantly better than EZW standalone in consuming time processing and compression efficiency. The encryption has been performed during the compression using the initial threshold of MEZW scheme encoding as a critical part in the compression algorithms and only this part is encrypted using a simple LSFR scheme. Security analysis shows that the encryption scheme is secure against the statistical and the frequency attacks.

The proposed MEZW compression and encryption is suitable for secured storage and offline secured transmission. On other hand, the computational time of this scheme is relatively high and need to be improved in any video streaming process. In the next chapter, we shall present two methods to refine this scheme and reduce the computational cost of processing even further. The first method is based on Joint DWT, DCT and Compressive Vector Quantization (JDWCT-CVQ). The last component replaces the EZW. In the second method the compression is based on edges extraction and JDWCT-CVQ. This step will form the RF processing in the ultimate secure video compression scheme researched for this thesis.

## Chapter 4

### Optimizing still Image compression & Encryption

In the previous chapter, we developed the joint DWT and DCT to improve the EZW coding algorithm and to add encryption within the coding process. Although, the MEZWE scheme has superior performance on all essential factors, we nevertheless found that it is not fast enough for a still images and by implication would not be sufficiently helpful in our effort to develop a secure video compression scheme. In this chapter, we propose two approaches to optimize the MEZWE image compression and encryption scheme. In the MEZWE scheme, no consideration is given to similarities between different “transformed” blocks within the image. The first optimisation approach is therefore based on incorporating Compressive Vector Quantization (CVQ) into Joint DWT and DCT. We shall call this the JDWCT-CVQ method. In a second approach, we exploit the statistical properties of DWT to extract the edges and combine with JDWCT-CVQ; we call this method the JDWCT-CVQ-Edge method. This second approach is designed to spend less effort on the smoother regions than those that include significantly higher level of texture and edges.

In section 4.1 we describe the JDWCT-CVQ method and present experimental results on its performance. Section 4.2 describes the JDWCT-CVQ-Edge method together with the experimental work and analyses of the results.

#### 4.1 The JDWCT-CVQ simultaneous compression and Encryption

The image compression, in this algorithm, will only compress the coefficients in the wavelet high frequency sub-bands while encryption is applied to the wavelet low frequency sub-band coefficients. This should allow both compression and encryption to be accomplished simultaneously as shown in Figure 4.1. Compression is based on Joint DWT and DCT followed by Compressive Vector Quantization (JDWCT-CVQ), while the encryption is achieved by scrambling the low frequency sub band based on two LFSRs with two different secret keys. Thus the computational time of this method is much lower than that achieved by the MEZWE and will be shown later in section 4.1. We shall first describe the CVQ system.

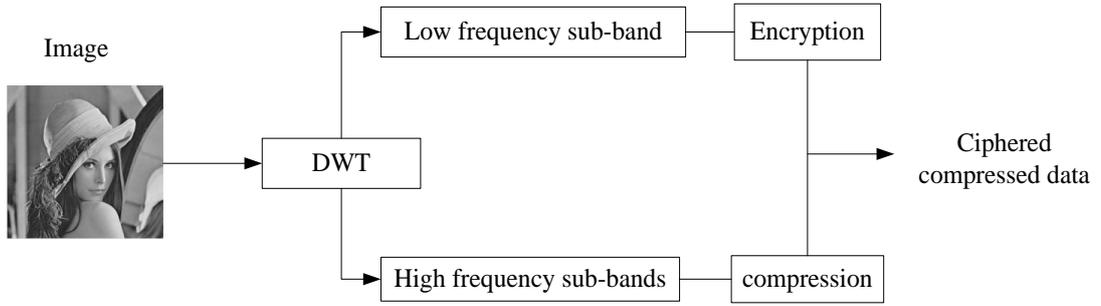


Figure 4.1 Parallel Image compression and encryption

#### 4.1.1 Compressive Vector quantization (CVQ)

Generally, there are two types of quantization: Scalar Quantization (SQ) and Compressive Vector Quantization (CVQ). In SQ, each sample number (pixels, coefficient) is quantized individually to reduce the numbers of bits that are required to represent an integer number as described in subsection 3.2.3. In CVQ the samples are quantized as groups/blocks.

Figure 4.2 shows the block diagram of the CVQ scheme. Firstly, the image to be compressed is partitioned into non-overlapping blocks by raster scan, each block is referred to as a vector and these vectors are organised in a list called a codebook. Then, we start with the first vector as a reference vector and search will start to find vectors that are similar to it among all other ones. Similarity is determined by a matching criterion, or a distance function. Here two such functions are defined on pairs of vectors  $B_1 = (b_{11}, b_{12}, b_{13}, \dots, b_{1n})$  and  $B_2 = (b_{21}, b_{22}, b_{23}, \dots, b_{2n})$ :

1. The so called supreme distance function:

$$\text{Dist}(B_1, B_2) = \text{MAX}_{i=1}^n (b_{1i} - b_{2i}) \quad 4.1$$

2. The Euclidean distance function:

$$\text{Dist}(B_1, B_2) = \sum_{i=1}^n (b_{1i} - b_{2i})^2 \quad 4.2$$

The Dist will be compared with a pre-set threshold depending on the required CR and image quality factor. Here we assume that  $B_1$  is a reference vector. If the Dist is smaller than threshold, the  $B_2$  vector is considered to be matched with reference block, otherwise it is labelled to be a mismatch vector that would become a new reference vector. Similar vectors in the codebook are grouped together and their indices will be recorded in the codebook together with reference vector. The reference block and the

indices of matched vectors will be sent to the decoder and removed from the codebook. This process is repeated until all blocks are labelled.

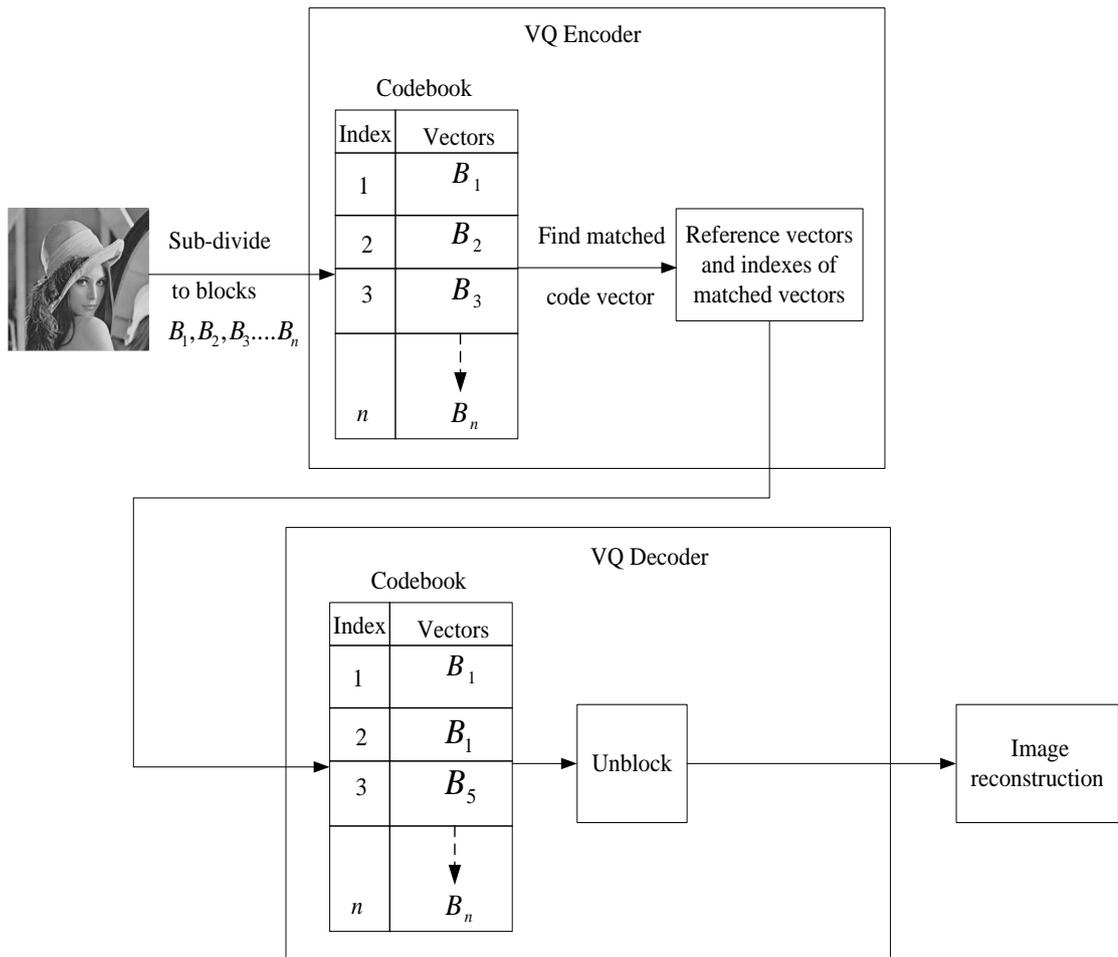


Figure 4.2 Vector quantization coding and decoding

Consider for instance, 4x4 pixels in Figure 4.3 as an image to be encoded. The image is sub-divided into blocks of 2x2 pixels by raster scan, and each block is considered to be vector and these vectors will be construct the codebook. The error distance between the first vectors  $B_1$  and other vectors ( $B_2, B_3$  and  $B_4$ ) is measured by applying equation 4.1 and then compared with threshold ( $TH=30$ ). As result of comparison,  $B_2$  and  $B_3$  are matched with  $B_1$ . So that the vectors  $B_1, B_2$  and  $B_3$  are removed from a codebook. The reference block  $B_1$  with indexes of  $B_1, B_2$  and  $B_3$  are sent to the decoder. Finally,  $B_4$  and its index are sent to the decoder, as seen in Figure 4.3.

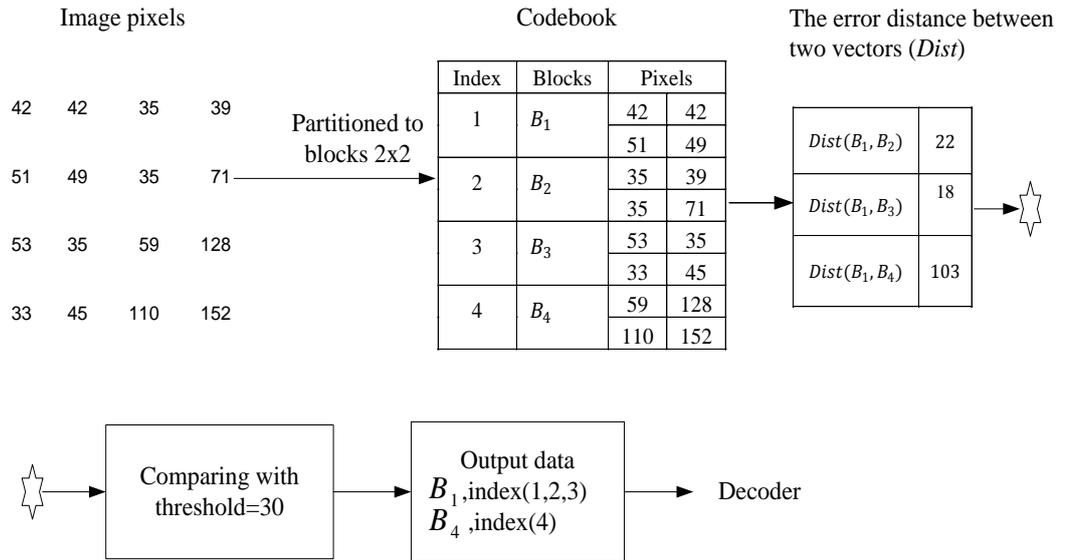


Figure 4.3 example of CVQ encoding

In case of grayscale images, each pixel is represented by 8 bits, and a 4x4 image is represented by (8x16=128) bits. Since there are 4 vectors, each index requires 2 bits in binary representation, while each vector contains 4x8=32 bits. Therefore, the output bit stream of CVQ encoder is equal to summation of bits that were produced from:  $B_1$ , index of matched block,  $B_4$  and  $B_4$  index=32+6+32+2=72 bits.

Consequently, the CR achieved by CVQ compression is equal to (72/128 = 0.56). (Sayood 2012) (Richardson 2011) (David, Motta and Bryant 2007).

#### 4.1.2 The proposed JDWCT-CVQ scheme

Using the CVQ as a replacement for the scalar quantisation scheme in the scheme developed in the last chapter is not the only proposed modification. In the new proposed JDWCT-CVQ compression will be confined to the matched blocks in high frequency sub-bands of the decomposed image wavelet, while the input to the encryption will be applied on the low frequency sub-band coefficients without compression.

##### 4.1.2.1 JDWCT-CVQ Compression scheme

Figure 4.4 illustrates the compression component of the JDWCT-CVQ scheme. Certain steps of this scheme are inspired by and adopted from JPEG2000. Firstly, image pixels are converted from unsigned to signed value by subtraction 128 from each pixel value. Then the image is DWT transform. Next, each frequency sub-bands are quantized individually by a different step size  $d$ , i.e. wavelet coefficient  $t$  is mapped to:

$$q(t) = \text{sgn}(t) \left\lfloor \frac{|t|}{d} \right\rfloor.$$

The step size  $d$  is determined by the JPEG2000 approach as follows:

$$T = 2^{R-C+i} \left(1 + \frac{f}{2^{11}}\right)$$

Where

$T$ : is the base step size.

$R$ : is the number of bits needed to represent the original intensity.

$C$ : is the number of bits needed to represent the exponent of wavelet coefficients.

$f$ : is the number of bits needed to represent mantissa of wavelet coefficients.

$i$ : the level of DWT decomposition

$$d = \begin{cases} \frac{T}{2^i} & \text{if LL} \\ \frac{T}{2^{k-1}} & \text{if HL or LH} \\ \frac{T}{2^{k-2}} & \text{if HH} \end{cases}$$

Where  $k$ : 1, 2, etc. is the levels of wavelet decomposition.

The signs of coefficients are extracted and wavelet coefficients are converted back into unsigned integer values. Each high frequency sub-band is divided into small blocks size 16x16 coefficients by raster scan as and input to the CVQ procedure (see) Figure 4.4.



space and encryption complexity a second LFSR with different secret key is used for the blocks scrambling of  $LL$  sub-band. Finally, the encrypted  $LL$  sub-band is sub-divided into block  $16 \times 16$  and sent to the decoder.

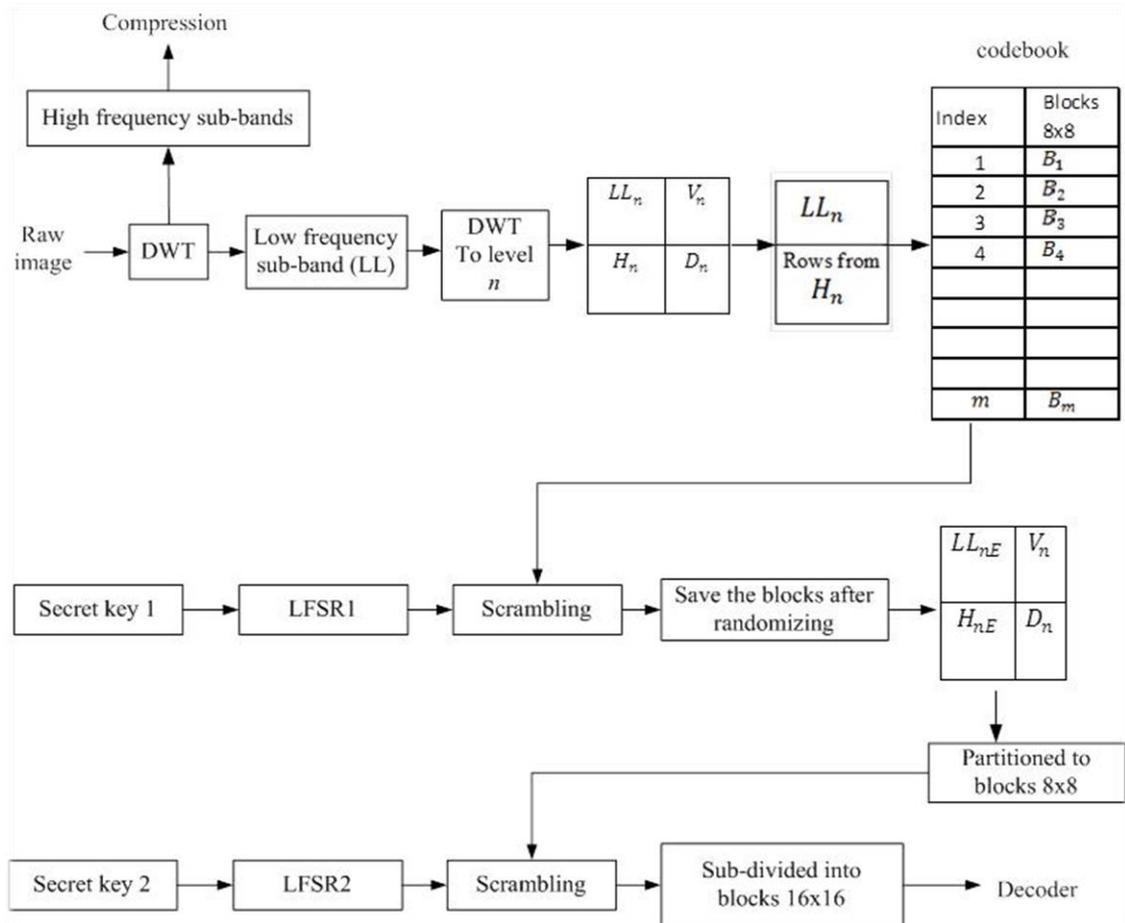


Figure 4.5 Block diagram of the proposed encryption

### 4.1.3 Experimental and analysis results

We evaluated the performance of proposed JDWCT-CVQ for image compression and encryption using the same set of still images which are tested in chapter 3. In these experiments, MATLAB V 7.10 (R2013a) has been used for scheme implementation on the same machine used in chapter 3.

#### 4.1.3.1 Compression analysis

We tested the performance of the JDWCT-CVQ scheme on image quality of reconstructed images, measured the PSNR and Histogram Intersection (HI) with the original image as a reference. The experiments tested the effect of CR and processing

time when using different threshold (TH) of CVQ with three levels of wavelet decomposition. Recall that we use the Haar filter for wavelet analysis.

Figure 4.6 below show the performance of the scheme in terms of the evaluation factors CR, HI, PSNR and the processing time required for the set of test images compression and encryption, when the images were WT decomposed to level 2 with thresholds in range 1 to 30.

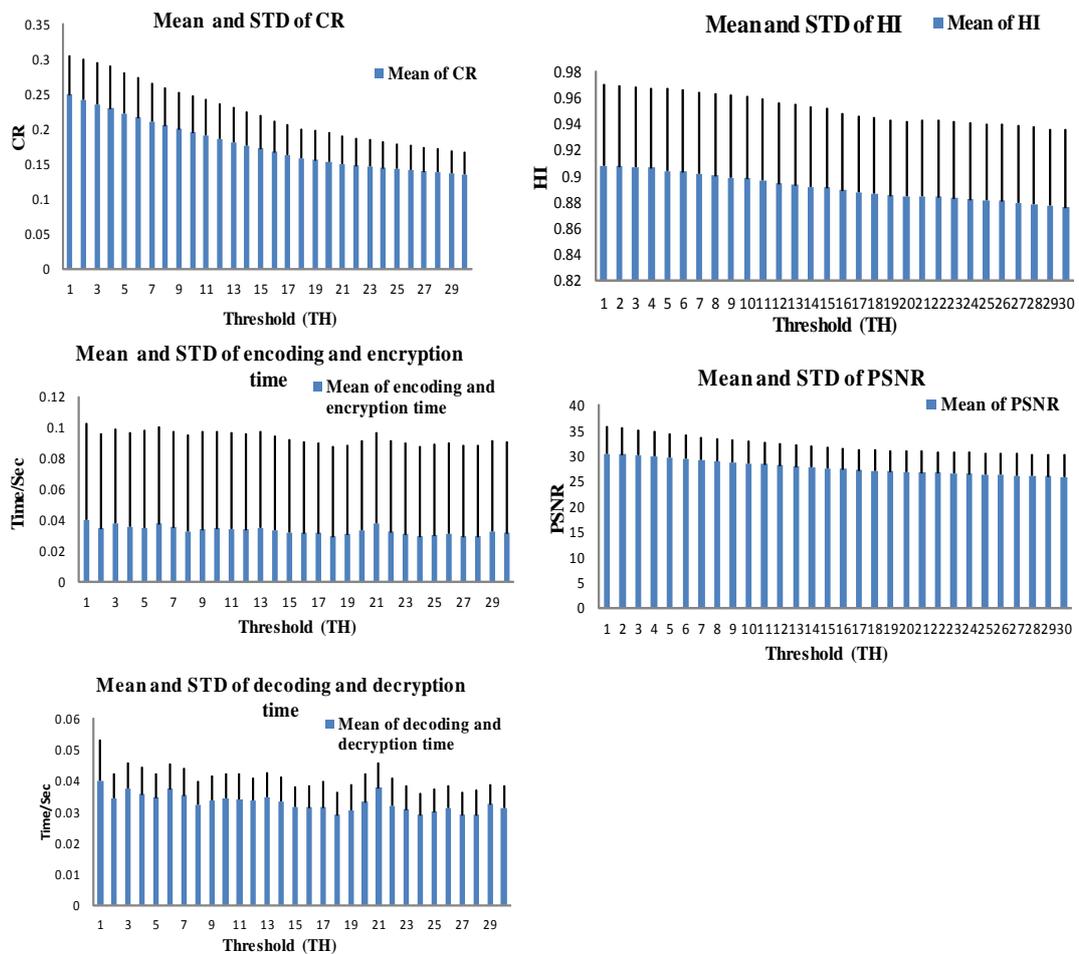


Figure 4.6 Compression results after WT to level 2

To determine the effect of decomposition level on the various performance parameters, the above experiment setting were applied but images decomposed to level 3. The achieved CR, PSNR, HI and the execution time for compression and encryption after WT to level 2 and 3 are shown in Figure 4.7.

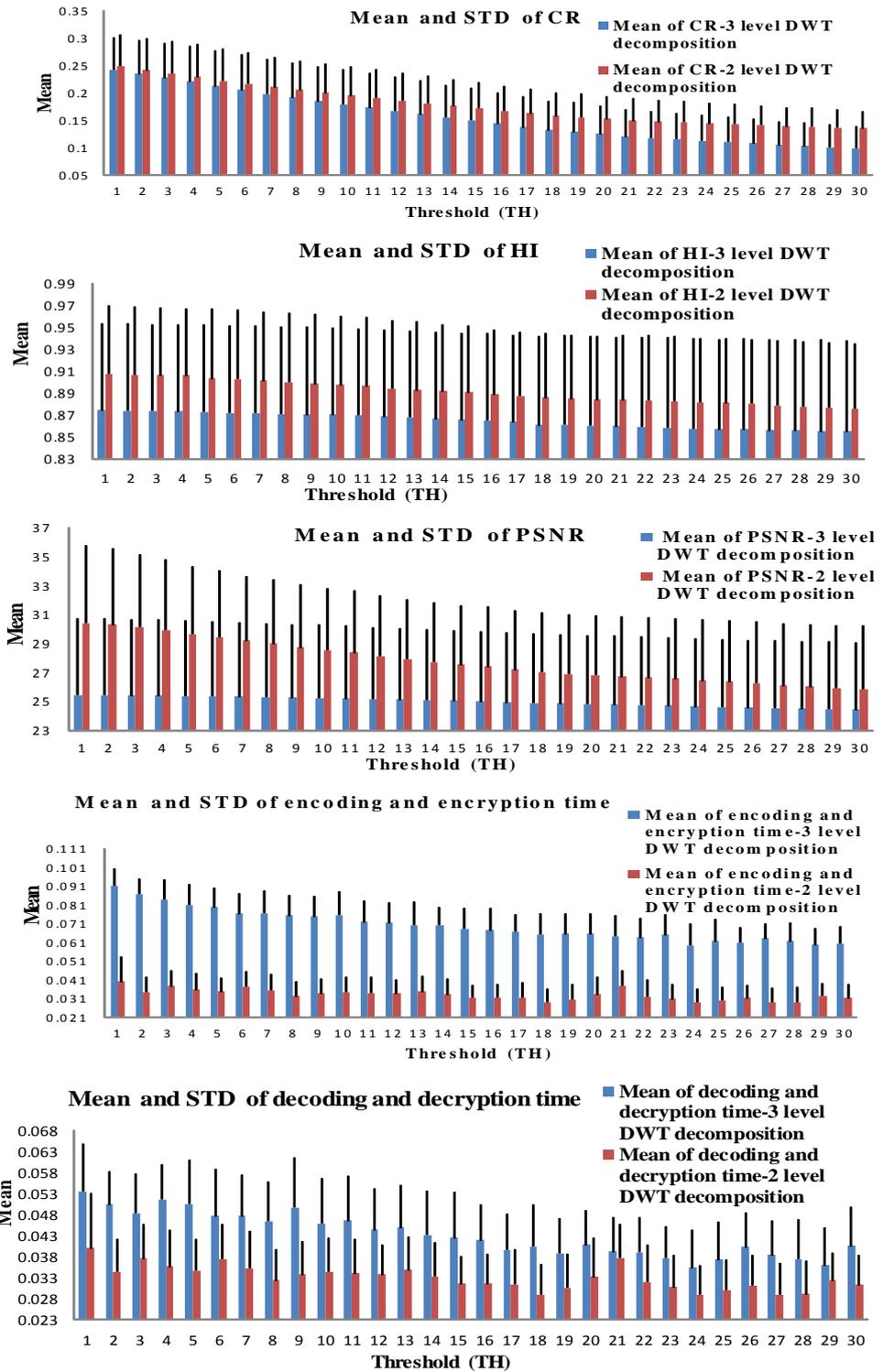


Figure 4.7 The mean and STD of Compression results after WT to level-3 and level-2

In Figure 4.7 it can be seen that CR is improved the higher the level of decomposition and the improvement increases as the threshold increases. But this improvement is at the expense of the increased compressing time and lower PSNR and HI when level 3 decomposition is applied. Increased processing time cannot all be attributed to the cost of level 3 decomposition. In order to identify other factors that contribute to this

performance we calculate the number of blocks produced from compressing a codebook  $k$  at constant threshold  $TH=5$  for doth decomposition levels of 2 and 3 using different images. The results are shown Table 4.1.

From the table, below, it can be seen that the number of blocks that were produced after applying JDWCT-CVQ on level 1 and 2 is considerably less than the original number of blocks. On other hand, at level 3 of decomposition there was a little difference between the number of compressed and codebook blocks. Moreover, the increase in the number of mismatched blocks at level 3 requires longer time for CVQ encoding. Thus, the proposed JDWCT-CVQ compression method is not yet feasible for relatively fast video processing when images are WT decomposed to level 3. Consequently, in this thesis we shall apply the compression method on high frequency sub-bands level 1 and 2 only and the encryption method will be applied on level 3.

Image	Level 1 Block size=16x16	Level 2 Block size=16x16	Level 3 Block size=8x8
	No. of blocks after compression out of 240 blocks	No. of blocks after compression out of 60 blocks	No. of blocks after compression out of 60 blocks
1	1	17	50
2	1	19	36
3	2	44	60
4	3	56	60
5	2	55	60
6	3	46	60
7	2	42	60
8	2	28	58
9	3	40	52
10	2	56	60
11	2	60	60
12	2	39	60
13	2	48	56
14	2	43	54
15	1	6	30
16	2	58	60
17	2	38	57
18	2	60	60
19	2	34	48
20	2	35	44
21	2	36	47
22	2	35	47
23	2	35	48
24	2	33	46
25	2	35	48
26	2	34	48
27	2	35	49
28	2	35	49
29	2	33	45
30	2	45	60

Table 4.1: Number of blocks post compression, at decomposition level 1, 2 and 3.

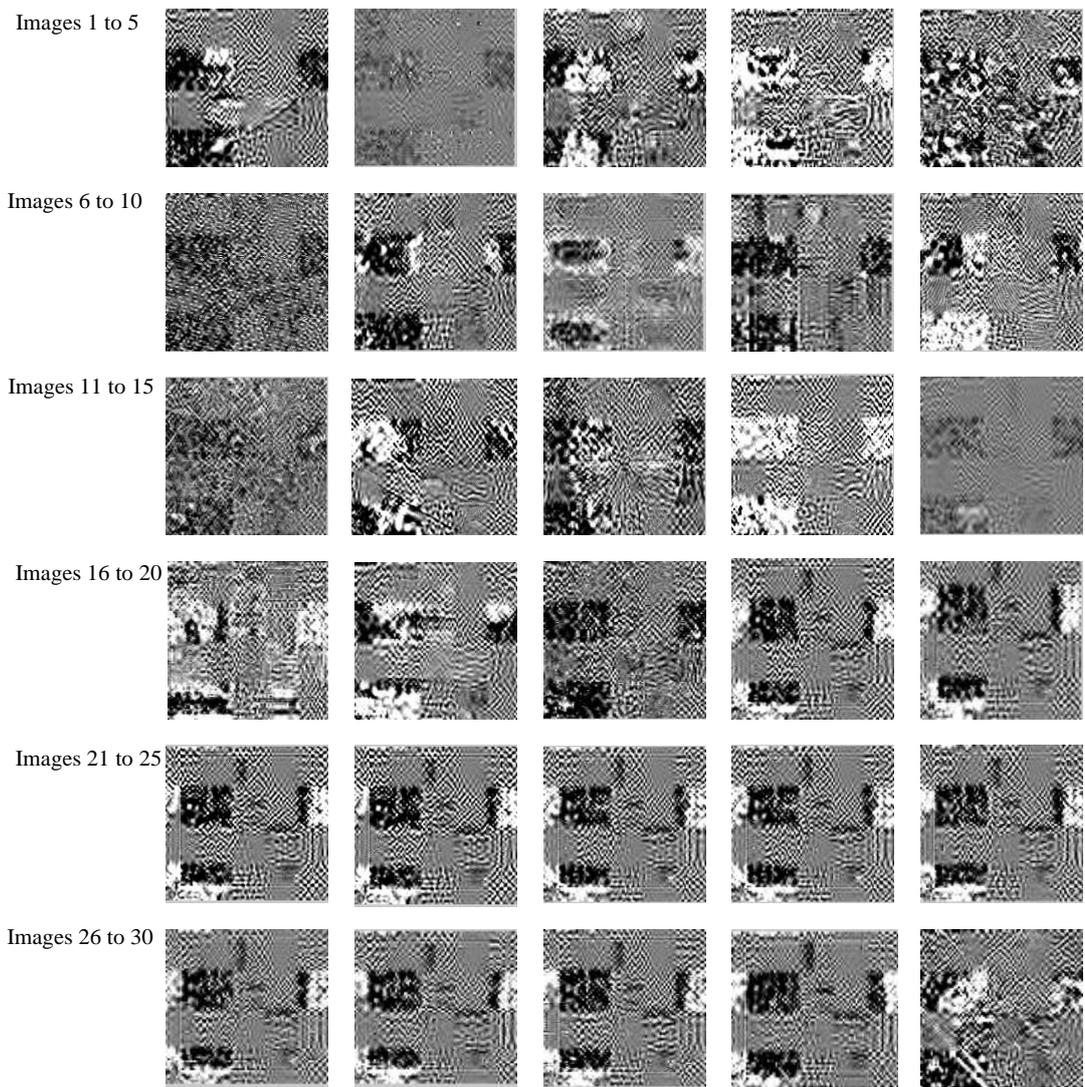
Finally, the results of the JDWCT-CVQ scheme are compared below with the performance of the MEZW scheme introduced in chapter 3. Table 4.2 shows that better quality is achieved by MEZW. The cause of this performance is due to error produced from matching criterion that used to quantify the similarity between blocks in CVQ. On the other hand, the results show that the JDWCT-CVQ produced better CR as compared to MEZW. Moreover, the execution time of JDWCT-CVQ is significantly lower than that of MEZW. Therefore, JDWCT-CVQ is significantly better than MEZW in coding time, but the price for that is worse quality

Method	MEZW	JDWCT-CVQ
Mean of CR	0.325	0.248
STD of CR	0.076	0.056
Mean of PSNR/dB	40.146	30.422
STD of PSNR	0.547	5.336
Mean of coding and encryption time/Sec	27.185	0.040
STD of coding and encryption time	0.894	0.013

*Table 4.2 the comparison results between MEZW and JDWCT-CVQ*

#### 4.1.3.2 ***Encryption analysis***

The proposed encryption scheme was then tested for robustness to statistical and frequency attack. Figure 4.8 shows the test images after encryption indicating how infeasible to gain information on images content. Robustness in terms of the histogram, correlation and PSNR analysis were tested and in Figure 4.9 we show some examples.



*Figure 4.8 Encrypted Images*

Figure 4.9 shows the histogram of images 1, 7 and 30 before and after encryption. Obviously, the histograms of encrypted images are completely different from the original images. Moreover, when the MDMF is computed for these histograms using equation 3.6 (chapter 3) as shown in Table 4.3.

image	1	7	30
HS <sub>i</sub>	57023	49140	41341
HS <sub>0</sub>	7815	29018	28359
HS <sub>255</sub>	1912	2744	13986
MDMF	61887	65021	62514

*Table 4. 3 the MDFM of encryption scheme*

The table shows that the MDMF values are high and hence the scheme is more robust against statistical attack. Therefore, inferring the secret key from the ciphered images is infeasible for a statistical attack. In addition, the correlation between adjacent horizontal and vertical pixels are calculated by equation 3.7, and shown in Table 4.4. The table shows that the correlation coefficients in encrypted images tend to be zero. So, predicting the relationship between encrypted pixels is again infeasible for the frequency attacks.

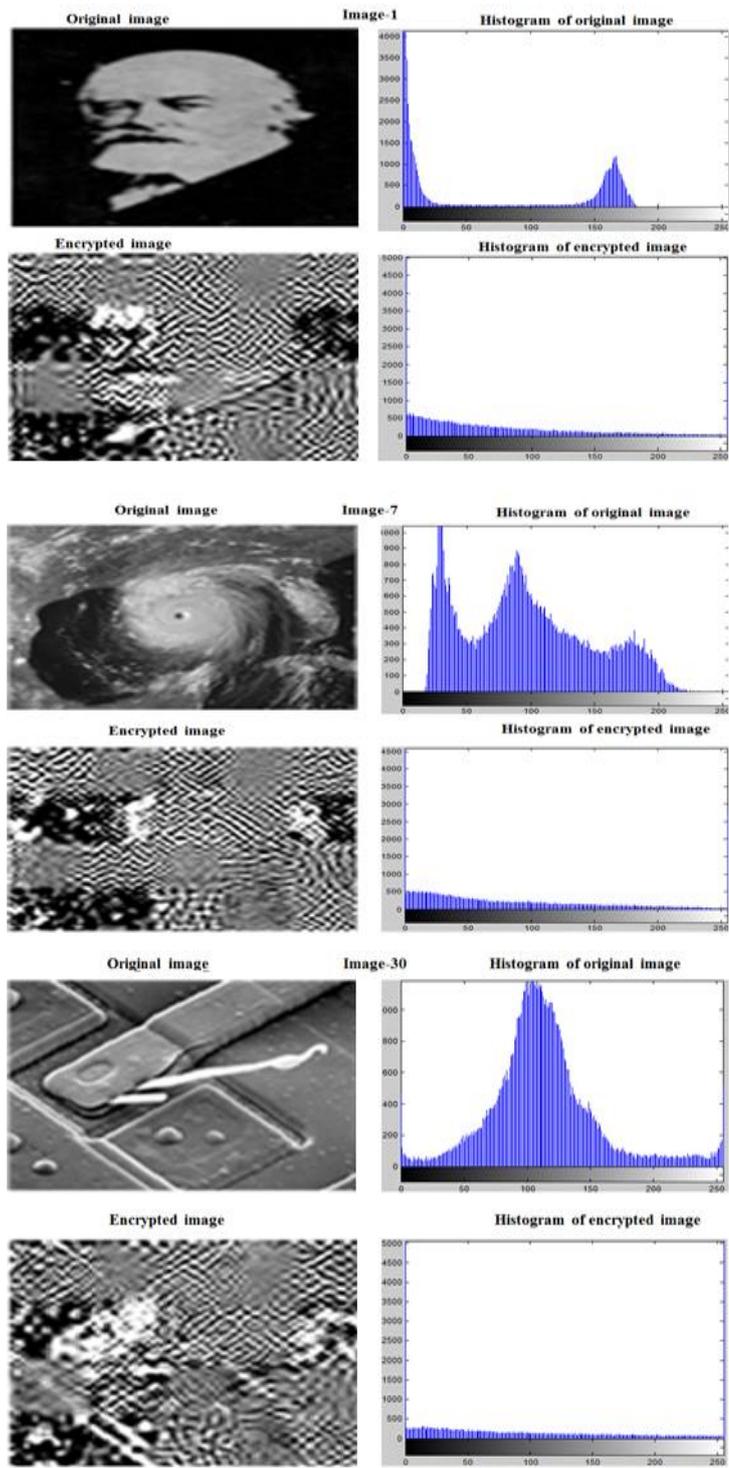


Figure 4.9 Histogram of original and encrypted image

Image	Original image		Encrypted image	
	Correlation coefficients		Correlation coefficients	
	Horizontal	Vertical	Horizontal	Vertical
Images-1	0.578911	0.9309	-0.00552	-0.0037
Images-2	-1	-1	-0.00242	-0.0026
Images-3	-1	-1	-0.00063	-0.0011
Images-4	-0.9503	-1	-0.00199	-0.0016
Images-5	-1	-1	-0.00122	-0.0015
Images-6	-1	-0.9997	-0.01235	-0.0139
Images-7	0.7646	1	-0.00389	-0.0017
Images-8	-1	-1	-0.00230	-0.0036
Images-9	-1	-1	-0.00602	-0.0011
Images-10	-1	-1	-0.00315	-0.0016
Images-11	-1	-0.9998	-0.00262	-0.0023
Images-12	-1	-1	-0.00283	-0.0027
Images-13	0.800438	0.9826	-0.00663	-0.0022
Images-14	-1	-1	-0.00134	-0.0013
Images-15	-1	-1	-0.00275	-0.0018
Images-16	0.74123	1	-0.00216	-0.0012
Images-17	-1	0.9083	-0.00341	-0.0009
Images-18	-0.99924	-0.9990	-0.00533	-0.0017
Images-19	-0.9990	-1	-0.00266	-0.0009
Images-20	-1	-1	-0.00334	-0.0011
Images-21	-1	-1	-0.00151	-0.0014
Images-22	-1	-1	-0.00247	-0.0008
Images-23	-1	-1	-0.00231	-0.0011
Images-24	-1	-1	-0.00213	-0.0012
Images-25	-1	-1	-0.00209	-0.0011
Images-26	-1	-1	-0.00223	-0.0015
Images-27	-1	-1	-0.00267	-0.0013
Images-28	-1	-1	-0.00359	-0.0012
Images-29	-1	-1	-0.00214	-0.0015
Images-30	-1	-1	-0.00096	-0.0018

*Table 4.4 Correlation coefficients analysis*

The PSNR is widely used as objective image quality metric. Normally, when PSNR > 30dB, the quality of recovered image is estimated as being of reasonable quality (Huang and Sakurai 2011). The PSNR values of encrypted images are shown in Figure 4.10. The average of PSNR value for all encrypted images is 6.96 dB which makes it evident that the encryption algorithm does successfully scramble and conceal the content in original images.

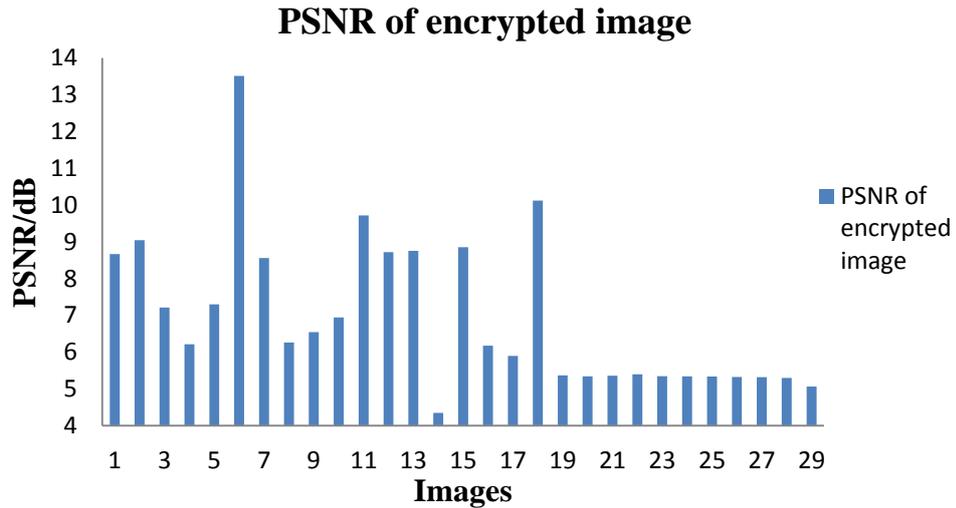


Figure 4.10 PSNR of encrypted images

## 4.2 JDWCT-CVQ-Edge compression

Despite the very good results obtained above, our intended secure video compression could benefit greatly from a higher still image compression while maintaining quality. For this we can exploit the link between the statistical parameters of the Laplacian distributions of wavelet coefficients in the high frequency {HL, HH, LH} subbands and the horizontal, vertical and diagonal image features and edges (see section 3.1). We shall use this fact to detect the significant image features (such as edges and corners) from these sub-bands by discarding all insignificant coefficients that are further away from the mean coefficients. Next we shall describe the new simultaneous image compression and encryption, JDWCT-CVQ-Edge scheme. Similarly to the JDWCT-CVQ, compression will be applied on the high frequency sub-bands at level 1 and level 2.

### 4.2.1 JDWCT-CVQ-Edge scheme

The new proposed JDWCT-CVQ-Edge compression scheme works in steps as follows:

1. Apply two levels DWT to raw image.
2. The signs of wavelet coefficients are extracted and apply the quantization method as mentioned in section 4.1.2.1
3. Calculate the STD for high frequency sub-bands(V1, H1, D1, V2, H2 and D2)
4. Measure  $THR = STD \times m$ . ( $m$  is a real number that determine the compression ratio (CR) and the quality of recovered image).

5. Compare the coefficients of each high frequency sub-band with THR and convert the coefficients (non- significant) that less than THR to zero.
6. Partition the high frequency sub-bands of level 1 and two into blocks size (16x16) coefficients.
7. Apply DCT to the high frequency sub-bands each block and construct a codebook.
8. Apply CVQ as mentioned in section 4.1.1.

Figure 4.11 summarises the main compression steps mentioned above. The image is recovered by inverting compression processing.

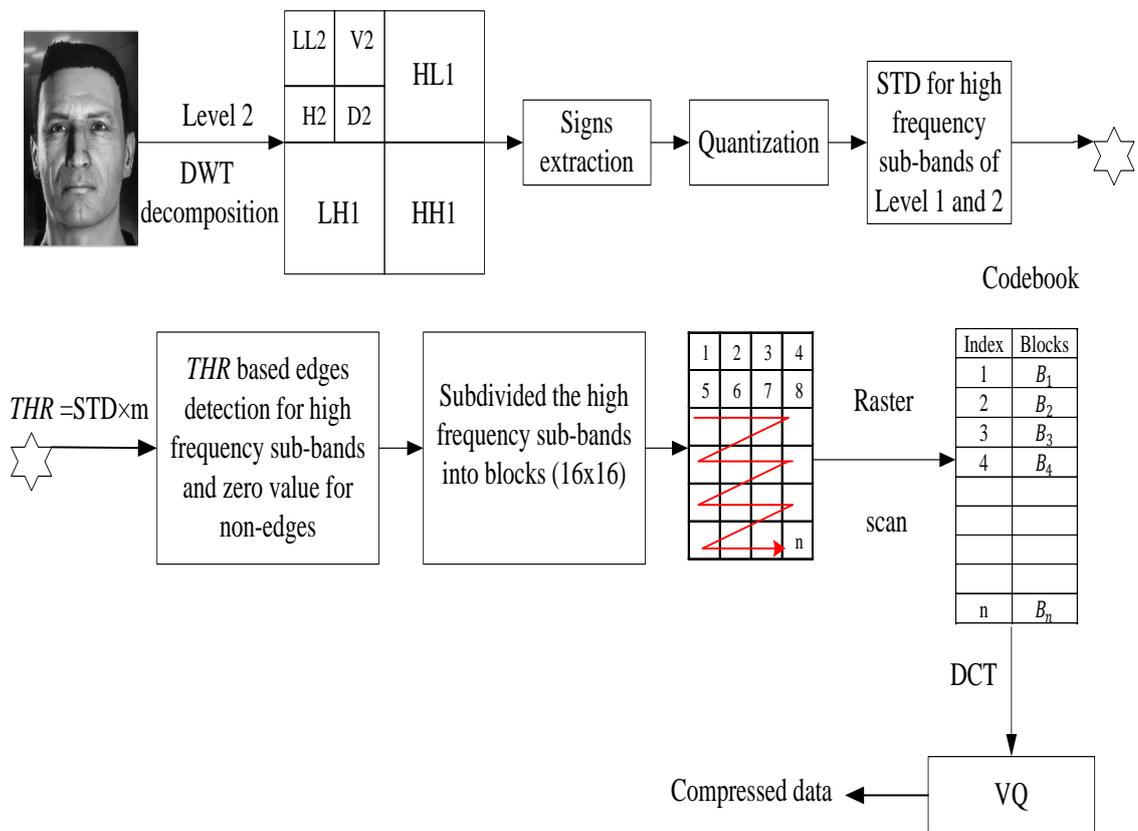


Figure 4.11 Illustrated steps for image compression scheme

#### 4.2.2 Experimental work and analysis of results

As before, various experiments are carried out to demonstrate the benefits of the JDWCT-CVQ-Edge image compression scheme. We test the performance of our proposed schemes using the test images used in chapter 3, section 3.3.1, and are

designed to test the performance of JDWCT-CVQ-Edge method at multiple values of THR, where:  $THR = STD \times m$  as mentioned in section 4.2.1 and to show the effect of implementing DCT and CVQ on the compression efficiency. The values of THR is driven from the well-known statistical rule which states that for a “normal” distribution about 68 percent of the values will be within one STD ( $\sigma$ ) of the mean, 95 percent lie within two STD ( $2\sigma$ ) and 99.7 percent within three STD ( $3\sigma$ ) (Bluman 1996). In fact this is roughly true for any distribution and it is used as a test for normality. Figure 4.12, below, illustrate this rule for the Gaussian distribution. Note that all high frequency sub-bands have a Laplacian distribution which are otherwise called Generalised Gaussian.

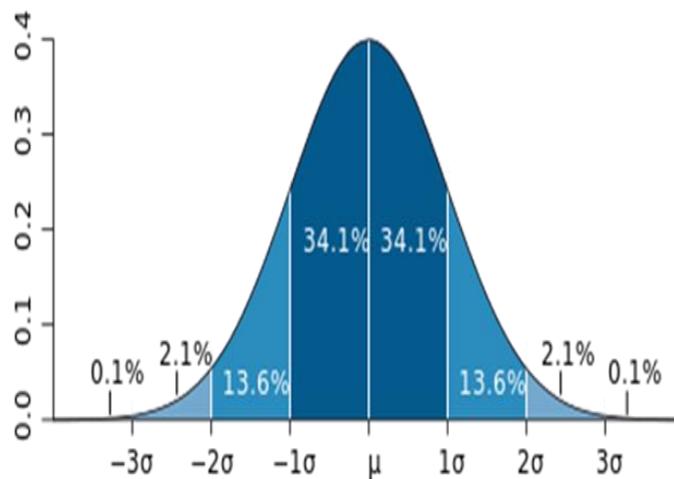


Figure 4.12 Illustration of the Empirical Rule

These statistical properties have been exploited by many researchers (Al-jawad 2009) (Ma 2002) for a basic image and video compression without applying other than simple quantisation, The first test, we conduct below, illustrates the performance of such image compression scheme through edges extraction in the high frequency sub-bands of level 1 without using DCT and CVQ method. Figure 4.13 illustrates the block diagram of the testing procedure. The quantization method mentioned in section 4.1.2.1 has been used in this test.

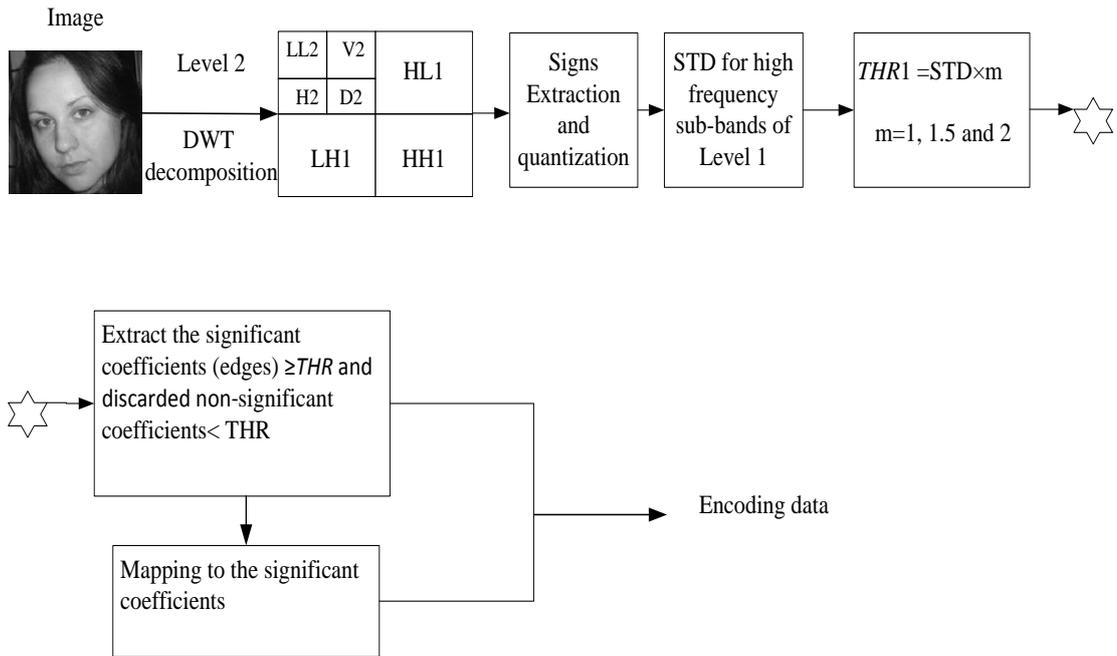


Figure 4.13 Experiment scheme to image compression through edges extraction for level 1 wavelet decomposition

Figure 4.14 displays the achieved CR and the quality of compressed high frequency sub-bands of level 1 (V1, H1, D1) at threshold (THR) of level 1 when ‘m’ equal to 1, 1.5 and 2. It can be seen that the CR and quality seem to be inversely proportional to the THR value, but the PSNR values are less affected by increased threshold. This trend apply to all images but the performance is dependent on the amount of texture present in the image (compare the CR results for images 1, 2, 17, and 26 that involve relatively more texture than the other images). These results allow us to increase the CR, if need be, without significant loss of image quality.

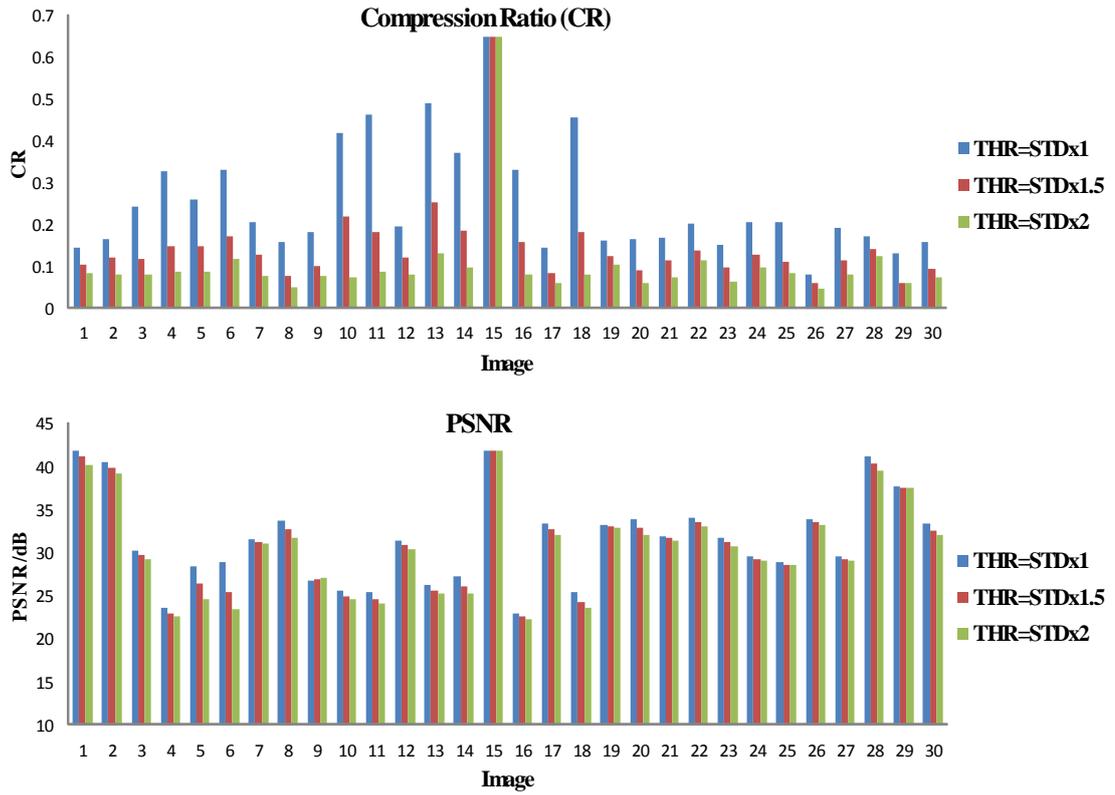


Figure 4.14 CR and PSNR from level 1 sub-band (V1, H1, and D1).

To explain the above trend, Figure 4.15 displays the number of significant coefficients (i.e. surviving the THR's filtering) in the non-LL sub-bands at level 1 only. It is clear that a large percentage of non-significant coefficients will not survive the filtering when THR increases.

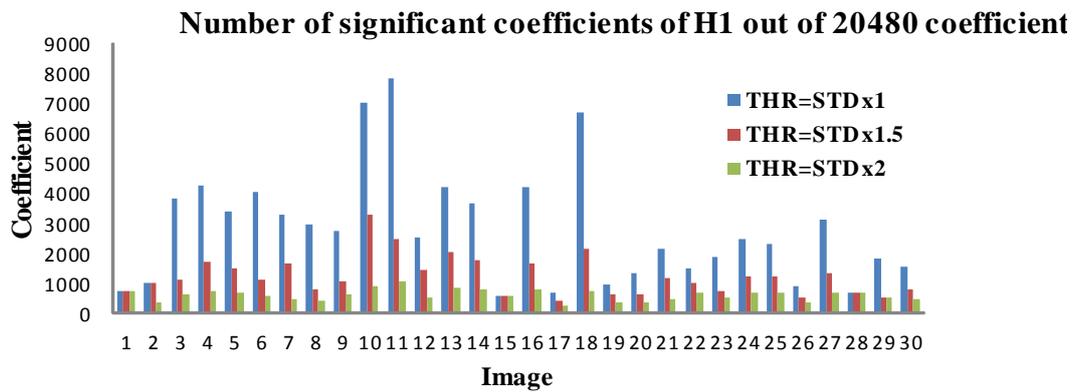
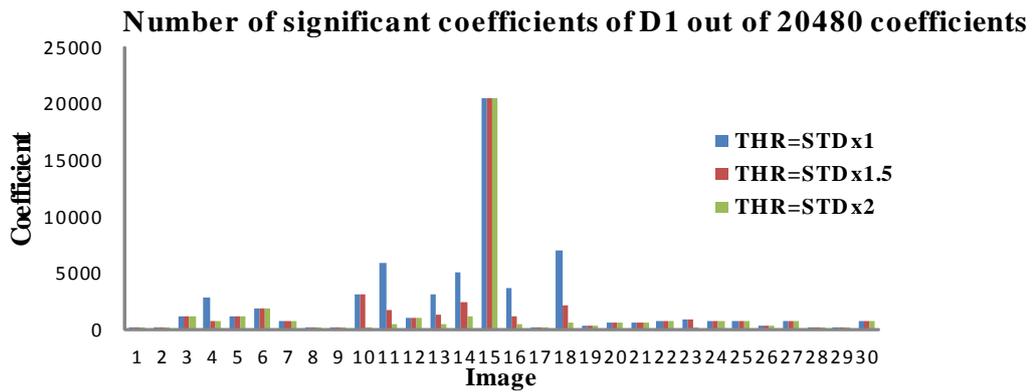
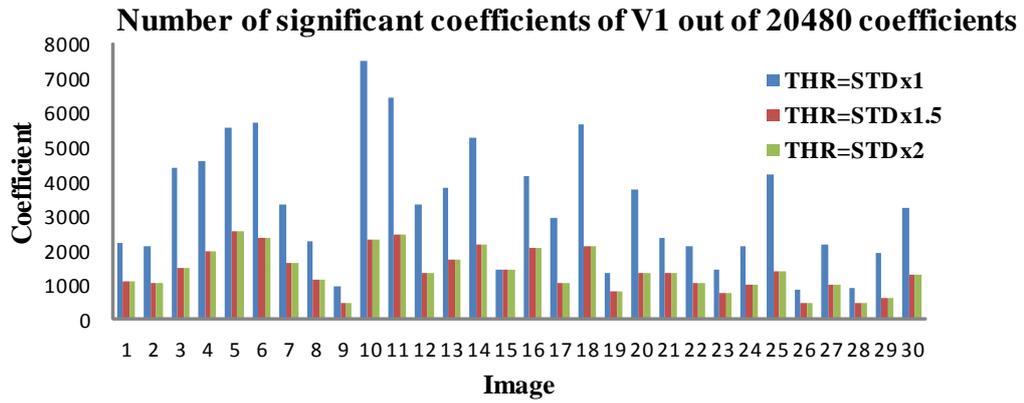


Figure 4.15 Number of significant coefficients of the sub-bands (V1, H1 D1) for THR=1, 1.5, 2.

The experiment settings applied in Figure 4.13 were also applied on both high frequency sub-bands of level 1 and level 2 with 'm' equal to 1, 1.5 and 2. In the first set of experiments, only wavelet transforms applied without using DCT or CVQ. The charts in Figure 4.16 show the CR and PSNR of compressed high frequency sub-bands of level 1 and 2 wavelet decomposition. Thus, the experimental results show again that the CR and PSNR are correlated with THR values. And at the same time, the show slightly improved CR without loss of quality.

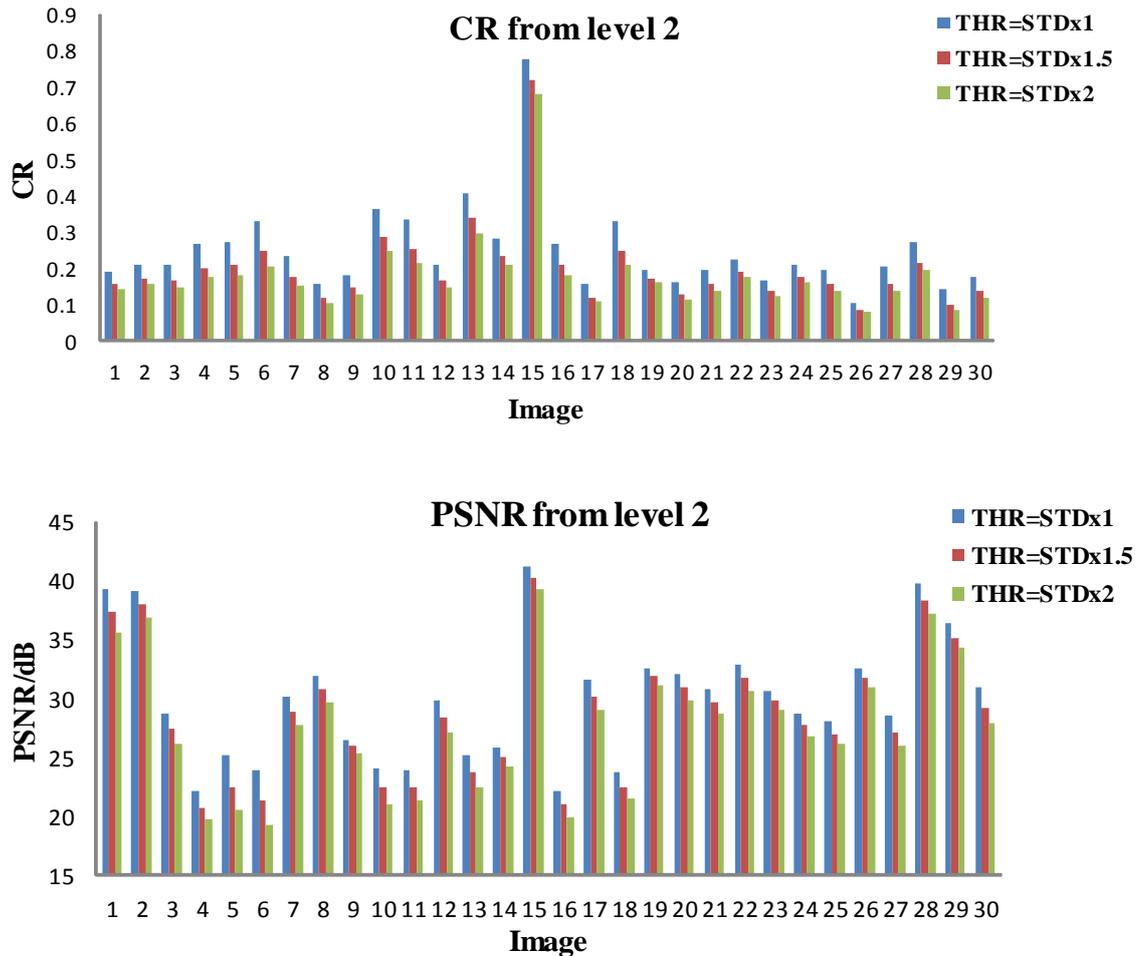


Figure 4.16 CR and PSNR of JDWCT-CVQ-Edge encoded images for different thresholds.

The next set of experiments tested the effect of using joint DCT and CVQ with edges detection (JDWCT-CVQ-Edge) on CR and PSNR. Figure 4.17 show the comparison between the results (CR and PSNR) of compression based on edges extraction and JDWCT-CVQ-Edge. The data from the experiment shows that both CR and PSNR for JDWCT-CVQ-Edge method improved as compared with CR and PSNR of edges extraction method alone because the DCT compact the signals energy in low frequency region, and JDWCT-CVQ-Edge method does not require mapping the significant coefficients. As can be seen in Figure 4.17, the quality is proportional with CR.

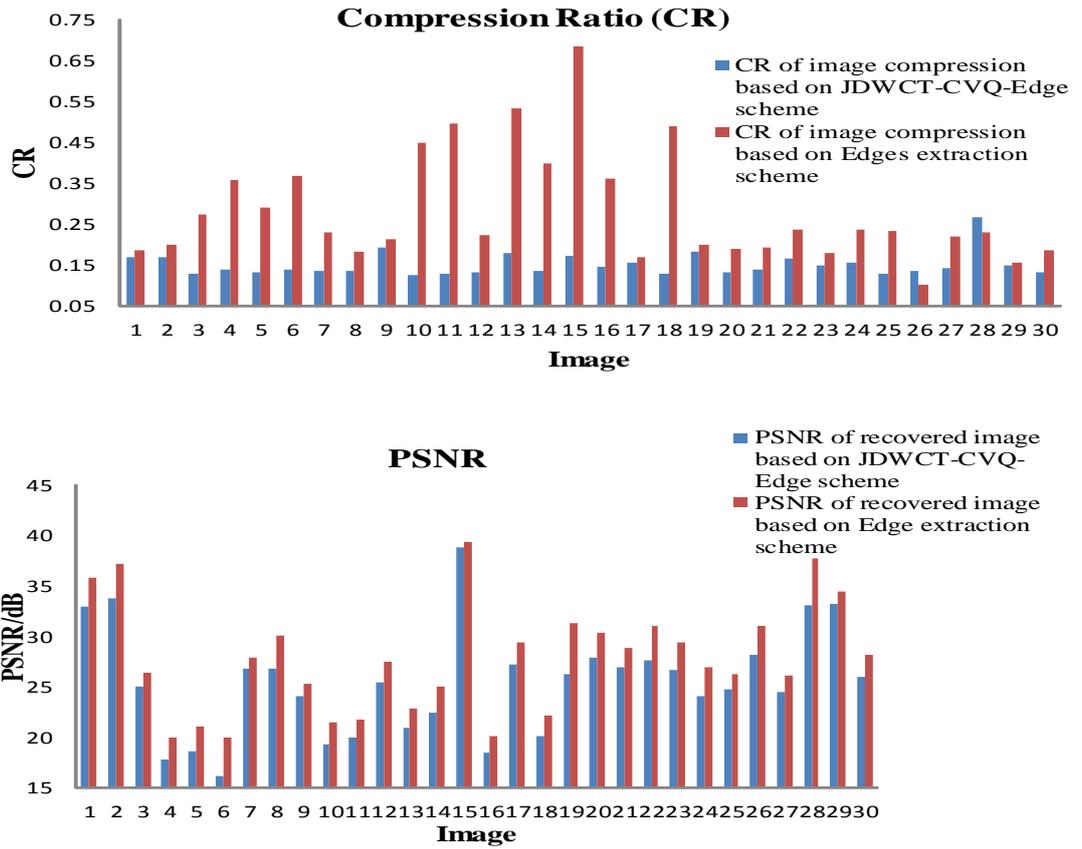


Figure 4.17 represent the comparison results between compression based on hybrid method and edges extraction

The results of this approach are compared with MEZW and JDWCT-CVQ scheme as shown in Table 4. 5 The MEZW scheme achieved better quality than other schemes. On the other hand, the table shows that JDWCT-CVQ and JDWCT-CVQ-Edge produced better CR when compared to MEZW. In addition, the coding time of JDWCT-CVQ and JDWCT-CVQ-Edge scheme is significantly lower than that of MEZW. Therefore, we are proposing to use JDWCT-CVQ and JDWCT-CVQ-Edge schemes for video compression as presented in chapter 5 and 6 respectively.

Method	MEZW	JDWCT-CVQ	JDWCT-CVQ-Edge
Mean of CR	0.325	0.248	0.1499
STD of CR	0.076	0.056	0.0287
Mean of PSNR/dB	40.146	30.422	25.424
STD of PSNR	0.547	5.336	5.3255
Mean of coding time/Sec	27.185	0.040	0.0625
STD of coding time	0.894	0.013	0.0007

Table 4. 5 the comparison results between MEZW, JDWCT-CVQ and JDWCT-CVQ-Edge

Although, the average image quality in terms of PSNR, but the image quality can be improved by adjusting the threshold for controlling the level of significant coefficients obtained from the various high frequency sub-bands as well as the threshold that control block similarity. Surely, this will results in increasing further the processing time which is somewhat worse than the JDWCT-CVQ time. However, for compressing videos this could be compensated for by avoiding the use of DCT for non-reference frames. In fact this approach will be investigated in the next chapters.

### 4.3 Conclusion

In this chapter two image compression and encryption schemes have been proposed. Compression was applied on high frequency sub-bands and encryption was applied on the low frequency sub-bands of image wavelet decomposition. The (JDWCT-CVQ) compression works block-by-block by combining DWT, DCT and CVQ. The algorithm has a relatively high speed processing time with high compression ratio and reasonable quality. We have found that the compression on high frequency sub-band of level 3 is ineffective and degrades the quality. The encryption scheme is applied on the low frequency sub-bands with extra appended information selected from high frequency sub-band by using two LFSRs. Our experimental work demonstrated that the encryption method provides a good security against statistical and frequency attack. The proposed compression and encryption scheme can be used simultaneously. Therefore, our intended video compression method will be implemented on the high frequency sub-bands of level 1 and 2 and the encryption on the low frequency sub-bands of level 3.

Moreover, in this section, an image compression through hybrid edges detection, DCT and CVQ has been investigated. We have used the statistical properties of the high frequency sub-bands of wavelet decomposed images. These statistical properties are exploited in image compression to extract the significant coefficients (edges). Our experimental work demonstrated that the mapping of these significant coefficients is cost effective compression algorithm. The CVQ (JDWCT-CVQ-Edge) was designed a block based similarity by combining DWT, edges sensing, and DCT to achieve better compression ratios and image quality compared with edges standalone wavelet-based compression. Furthermore, we investigated the effect of applying different thresholds, which is driven from STD of high frequency sub-bands, to preserve the significant coefficients. The results show that the compression ratio and image quality are proportional to the threshold value. We also tested the performance of this algorithm

without using the DCT and CVQ. The results will be exploited to investigate a version of this algorithm without the DCT for compressing non-reference frames. The CVQ provides an appropriate mechanism to design a video coding. Therefore, in the next chapter we shall use the JDWCT-CVQ for video compression and encryption simultaneously at low computational cost.

## Chapter 5

### CVQ for Secured Video compression

Video and data communication is an essential component in developing a solution to the specific application described in Chapter 1, whereby we assume mobility but constrained computational power for sensitive video transmission over wireless information of limited bandwidth channel. In this chapter we shall describe a new technique for video compression and encryption. The compression builds on and benefits from the work of the last two chapters on the Joint DWT, DCT and CVQ procedures for encoding the reference frame as well as subsequent frames by exploiting intra-frame block similarity. The encryption algorithm utilizes two LFSRs seeded with three secret keys to scramble the significant wavelet coefficients multiple times. As described in the previous chapter, section 4.1.2, the compression will be applied on high frequency sub-bands and the encryption is to be applied on LL sub-band of image wavelet decomposition at level 3. Both algorithms may be applied simultaneously based on wavelet domain. We shall demonstrate experimentally that the proposed algorithms have the following features; high compression ratio, acceptable image quality, resistance to the statistical and frequency attack, and low computational processing cost. In section 5.1, we shall review the most popular video compression techniques. Section 5.2 describes the video version (JDWCT- CVQ) scheme for video compression and encryption. The experimental results and analysis will be presented and discussed in section 5.3. Finally, the conclusion is presented in section 5.4.

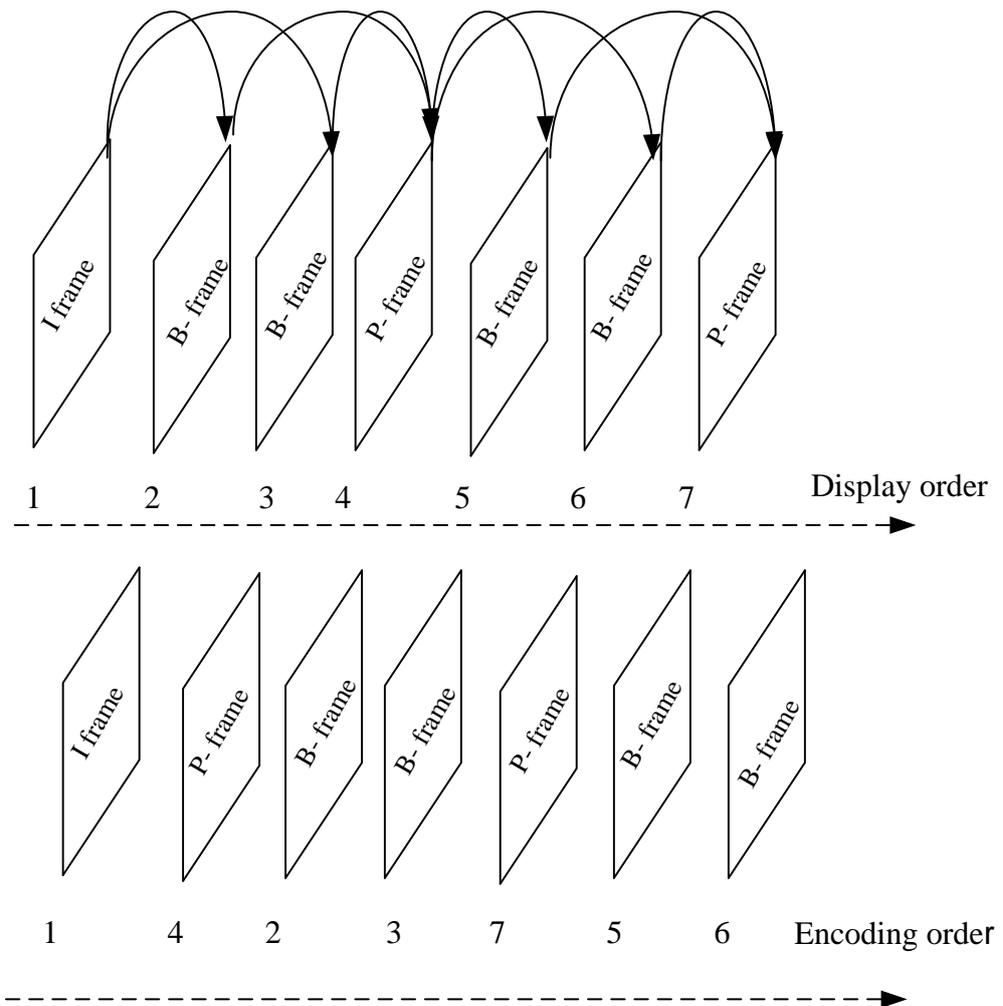
#### 5.1 Existing Video coding techniques

A digital video object is a sequence of images, called frames. Besides the spatial redundancy within each single frame in video sequences, successive frames usually are very similar and this is known as inter-frames redundancy or temporal redundancy. Naturally, existing video compression techniques aim to remove as much as possible intra and inter-frame redundancies within the video data subject to the various constraints on CR, image quality, bandwidth, and time complexity. Usually, video compression starts by compressing the first frame (Intra-frame) using a still image

compression method. Then each successive frame (Inter-frame) is compressed by identifying the differences between the intra and past frame and encoding only these differences (motion vector data) to be transmitted rather than the whole frame again. This inter-frame method is known as temporal encoding. The intra frame encoding is referred to as the 'I' frame or the Reference Frame (RF) and the inter frame encoded frames are referred to as the P (for predictive) frame while an inter frame encoded using both past and next frames is known as B (Bidirectional) frame. In this thesis, we adopt the forward prediction temporal encoding and only I frames (RF) will be used for motion vector prediction to help in reducing the video encoding time.

Figure 5.1 shows an example of video frames encoding, the video contains seven frames: I frame is sequence 1, P frame is sequence 4 and 7, and B frame is sequence 2, 3, 5 and 6. I frame 1 is encoded individually as reference frame. B-frames 2 and 3 are encoded after P-frame 4 is encoded and B-frames 5 and 6 are encoded after encoding P frame-7 which is encoded based on past P frame-4 as shown in Figure 5.1B. This process is applied in MPEG compression to construct the P and the B frames based on I frame. This kind of temporal encoding is time consuming and it is applied mainly in off line video compression.

In the flash player (described in chapter 2, section 2.2.2.1.1), the B frames are replaced with D frames. The D frames use only the recent I or P frame (forward prediction) for motion vector prediction. As a result, the codec D frames achieve less compression ratio compared with MPEG but it improves the real-time processing.



**A** Figure 5.1 example of video frames encoding (A) display order (B) encoding order  
**B**

The block based motion compensation process is the most frequently used in the inter-frame compression. It produces an approximation of an inter frame (B or P frame) by reusing data contained in the RF (intra or I or P frame). The inter-frame compression is accomplished in three stages. Firstly, the inter-frame is divided into non overlapping blocks (target blocks), and then each Target Block (TB) in the inter-frame is compared to its counterpart(C) in the RF frame to determine the matching block within the search area. Block matching is the most time consuming part of the video encoding process. The search area is defined by the maximum displacement parameters  $dx$  and  $dy$ . If TB is a square block with dimension  $b$ , the search area will include  $(b+2dx)$   $(b+2dy)$  pixels and will contain  $(2dx+1)$   $(2dy+1)$  overlapping blocks, see Figure 5.2. Finally, the position of the matched block (motion vector) in RF is encoded in place of the TB itself. Since fewer bits are required to code motion vector than to code the TB blocks, compression is achieved.

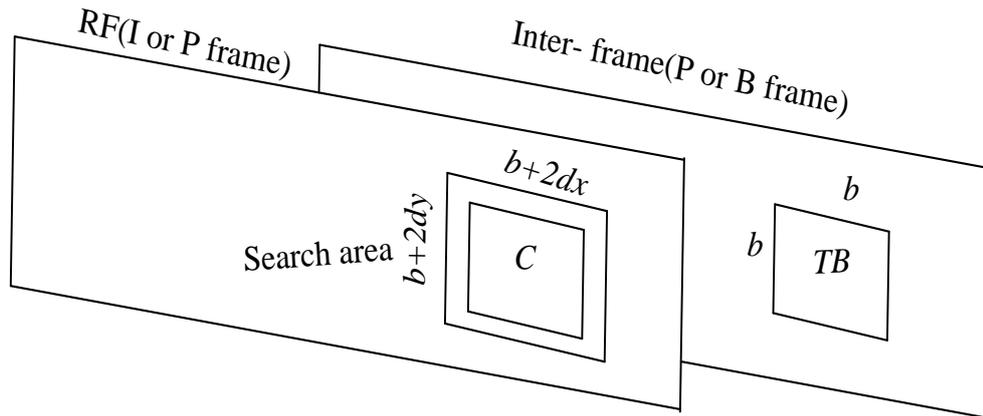


Figure 5.2 displays corresponding blocks from an inter-frame and RF frame and the search area in the RF frame.

During decompression, the decoder uses the motion vectors to find the matching blocks in RF frame (which it has already recovered) and copies the matching blocks from the RF frame into the appropriate positions in the approximation of the inter-frame.

Due to the extensive computation involved in the above two methods (MPEG and Flash player) together with the advances in computation power, many video compression have been proposed to compress each frame independently by Motion JPEG2000 (MJ2) or simple wavelet-based algorithms (Al-jawad 2009) (Ehlers 2008) (Ma 2002). Our proposed will have similarity with the MPEG, but we shall use the forward prediction rather than both backward and forward prediction. Moreover, we exploited knowledge about the spatial and temporal redundancies in terms of statistical parameters of high frequency wavelet sub-bands.

## 5.2 JDWCT- CVQ scheme for video compression and encryption

In this algorithm, all frames are transformed with the Haar filter and work to reduce both spatial redundancies within the blocks of the frames plus the temporal redundancies across neighbouring sequences of frames. The other steps of compression are only applied on the high frequency sub-bands level 1 and 2, whereas encryption is applied on the low frequency sub-band of level 3 (LL3) without compression to achieve relatively faster encryption and effective security rather than using LL2 or LL1. Therefore, this method of video compression and encryption can be used together, as

mentioned in chapter 4, section 4.1, to reduce the computational processing time as demonstrated in Figure 5.3. The high frequency sub-bands are sub-divided into non-overlapping blocks of a certain size and then the DCT is applied to each block. The DCT results in rearranging the significant block coefficients and thereby creating more redundancies and facilitating their removal at the block level in a very efficient manner. Similar blocks, determined by pre-specified thresholds, will be discarded but referenced by only one block. This approach improves the compression ratio. The following sections explain the compression and the encryption in more details.

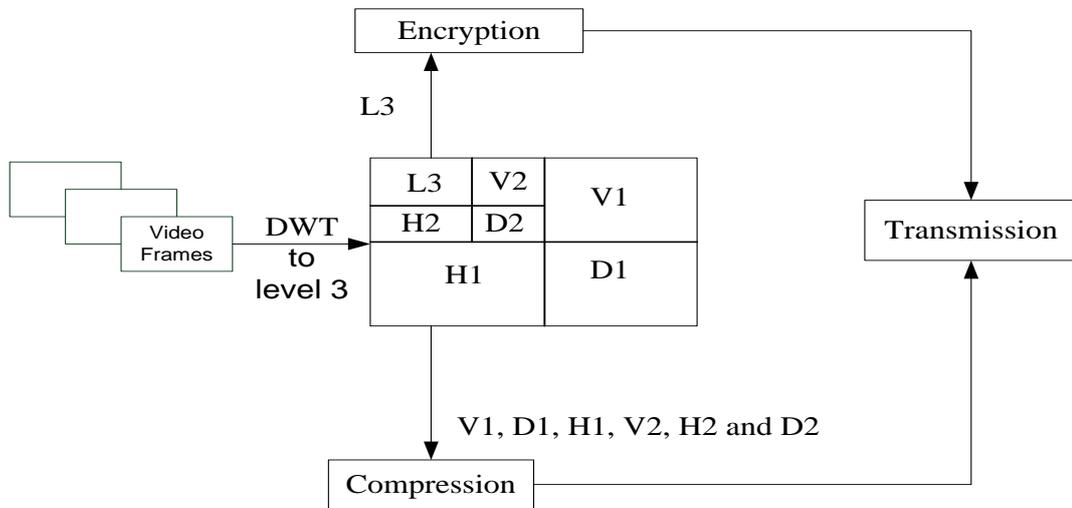


Figure 5.3 parallel video compression and encryption

Note that redundancies among high frequency wavelet sub-band blocks are more common than among the spatial domain image blocks. In fact, similar blocks in high frequency sub-bands do not correspond to similar image blocks. Hence, removing redundancies from high frequency sub-bands is more effective than doing the same in the spatial domain.

### 5.2.1 The compression algorithm

For each frame in the video sequence, a wavelet transform will be applied to level 3, and the coefficients are converted to integer unsigned values. The high frequency sub-bands of level 1 and 2 will be quantized separately, in the same way as applied in the chapter 4, section 4.1.2.1. To provide a high similarity between blocks of adjacent frames we sub-divide the high frequency sub-bands to blocks of 16x16 and apply DCT to each block.

Based on compressive vector quantization (CVQ) method, described in chapter 4, section 4.1.1, each sub-divided block is considered to be a vector and these vectors will be used to construct a codebook. After applying CVQ method the compressed codebook will be transmitted and contains the mismatched blocks and only one copy of any such block with references to the others in its similarity class. This method will be applied to the Reference Frame (RF) only. The RF will be sent every 25 frames to enhance the video quality, but in some cases one may select RFs more frequently or even adaptively depending on network traffic loads. For the nRF frames, the CVQ method will be applied but this time the matching criteria will compare the corresponding blocks of the RF codebook and the Current Frame (CF) codebook. For example, blocks 1, 2, 3, 4 and 5 of RF codebook may be matched with their counterpart of the CF codebook while blocks 6 and 7 are mismatched with the corresponding blocks of CF codebook. Therefore, the matching blocks of the CF are discarded and only the mismatching blocks and their pointers are sent to the transmitter as shown in Figure 5. 4.

The block matching criterion, in this algorithm, is based on a pre-set threshold (THR), calculated in terms of the desired compression ratio and the quality factor as we displayed in chapter 4, section 4.1.3.1. This will be decided in advance or is left to vary adaptively depending on network traffic loads. The distance *Dist* between the any two blocks, B1 and B2, is calculated as shown in the equation below:

$$Dist(B1, B2) = \max(B1 - B2) \quad 5.1$$

This *Dist* will be compared with THR to identify the matching block.

During decompression, the decoder uses the indices of matching blocks to catch the matching blocks in the RF codebook which it has already received, and copies the matching blocks from RF into the CF codebook positions. Additionally, the decoder uses the mismatching blocks and its indices into the CF codebook positions. The frame is reconstructed during the inverse DCT, CVQ and DWT.

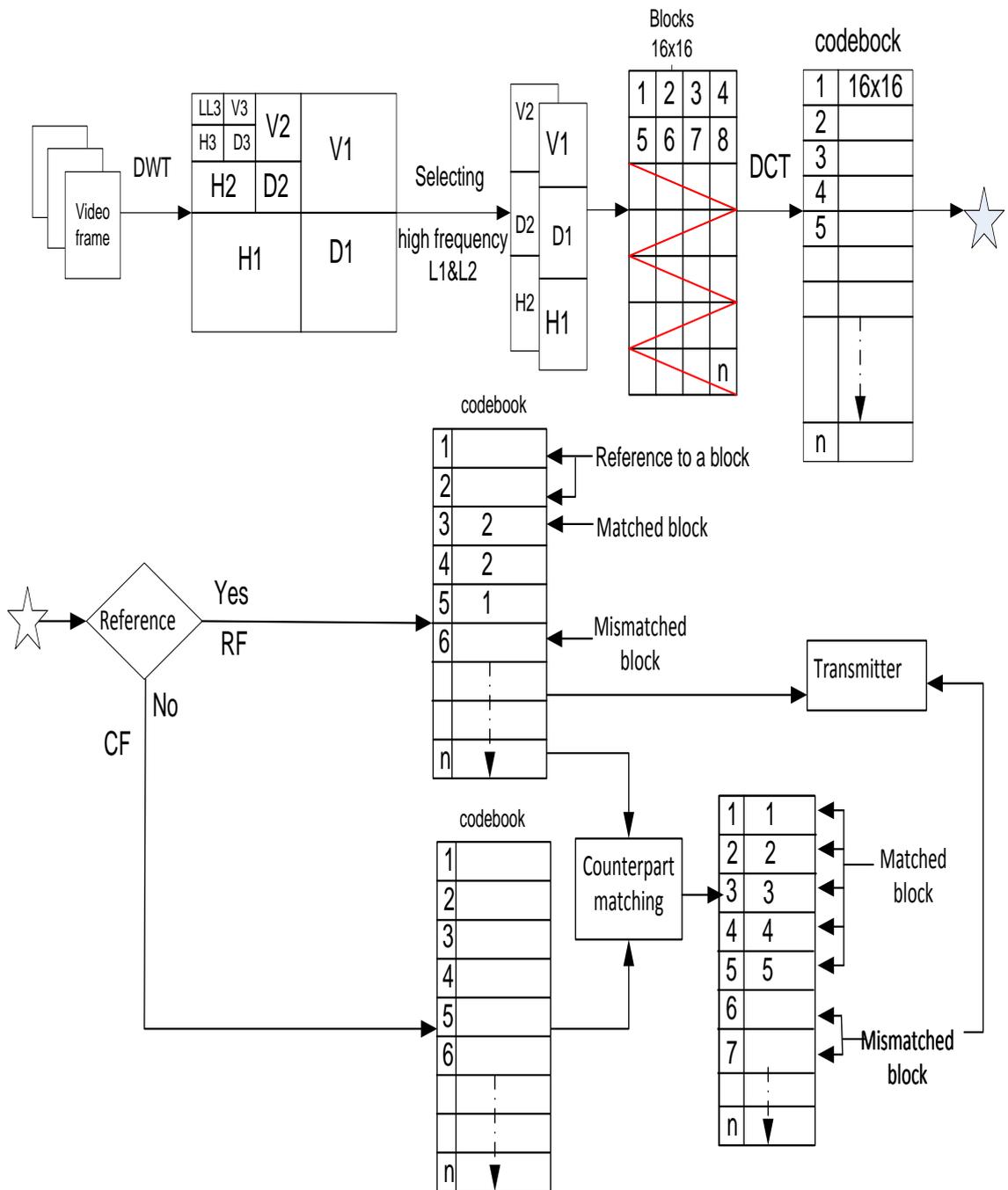


Figure 5. 4 Block diagram of the video compression

### 5.2.2 The encryption algorithm

Considering the linearity weakness of LFSR, here we follow the A5 stream cipher and use a number of LFSRs, with three secret keys applied in three rounds on wavelet sub-bands of level 3. This combination will increase the security level of the used encryption method without significantly increasing processing time. This method is shown in Figure 5.5. In the first round, the encryption focus will be on the low

frequency sub-band of level 3 ( LL3) part padded with extra information added from high frequency sub-band of level 3 (H3). This part will be divided into blocks of 8x8 and then scrambled using a certain LFSR supported by a secret key (Key1). All scrambled blocks will be put back into their original places. The second round will involve the entire level 3 sub-bands by sub-dividing it into 8x8 blocks. In this round all level 3 blocks will be scrambled again using a second LFSR with secret key (Key2). The final round will use a different block sub-division (16x16) with the initial LFSR set to a third secret key (Key3).

This method will shuffle the significant information located in LL3 in different forms and distribute it in the entire level 3. This makes re-shuffling the information of the band without knowing the secret key very time consuming, particularly in the case of video streaming.

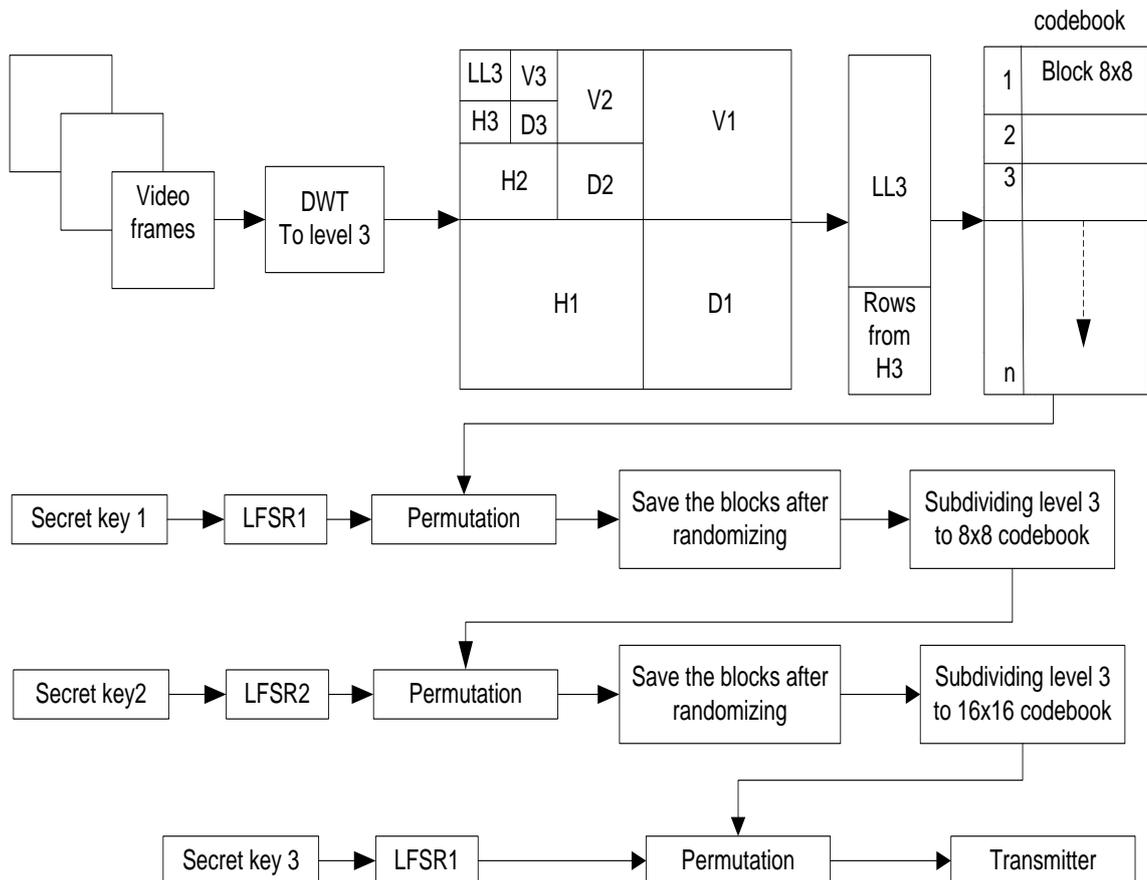
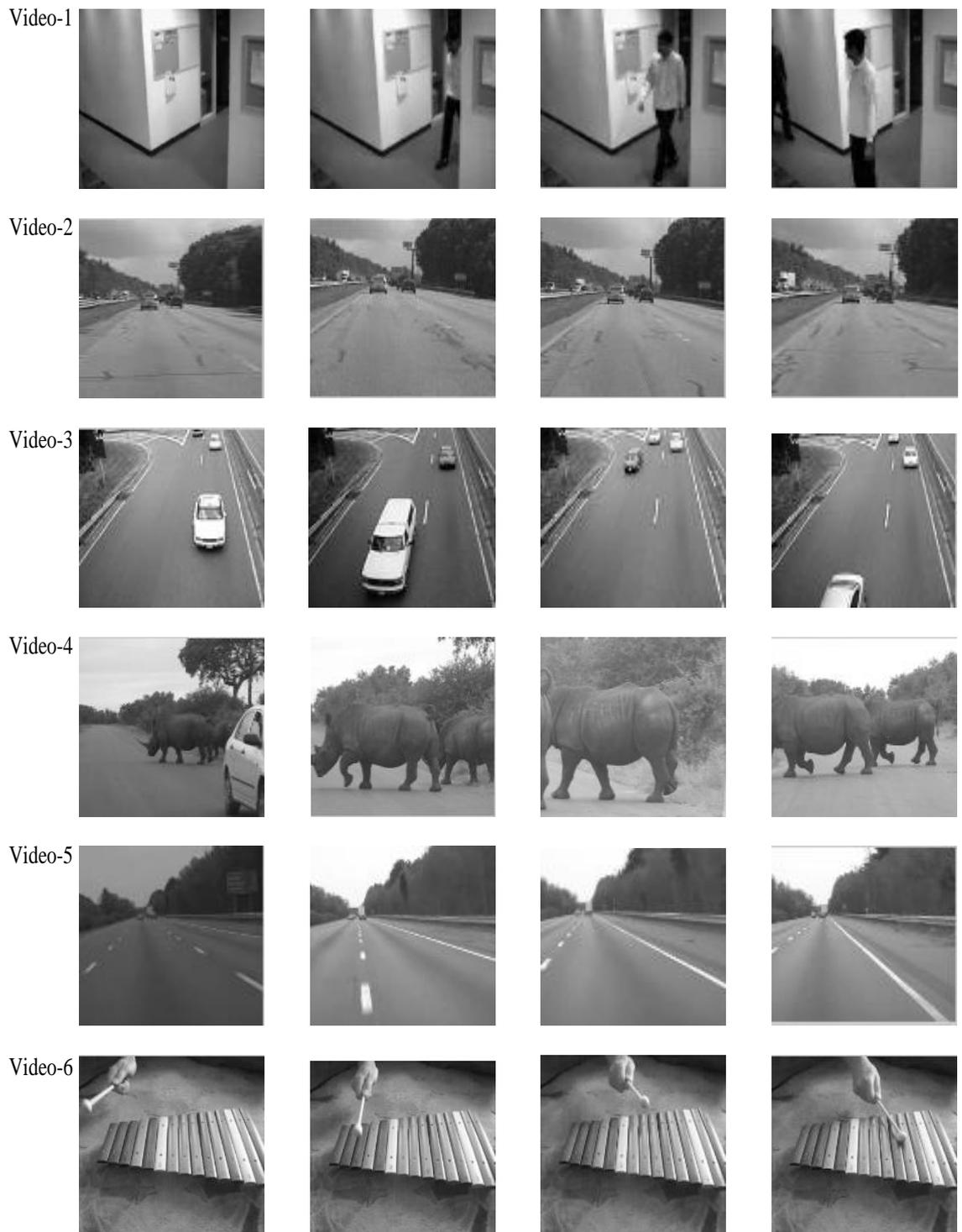


Figure 5.5 Illustrates the Encryption scheme

### **5.3 Experimental work and analysis**

In this section, we shall evaluate the performance of the above JDWCT-CVQ video compression and encryption scheme, and demonstrate the effect of periodic RF on the compression efficiency, quality and the encoding complexity. Then, we shall compare the performance of our approach with other algorithms. The simulation of the proposed scheme is carried out using MATLAB V 7.10 (R2013a) on the same machine revealed in chapter 3.

The proposed method was applied on 6 different videos of different nature, each consisting of 100 frames on average. Few RF frames from these videos are shown in Figure 5.6. The frame size of all these videos is (256 x320) pixels. During our experiments we converted all frames to grayscales

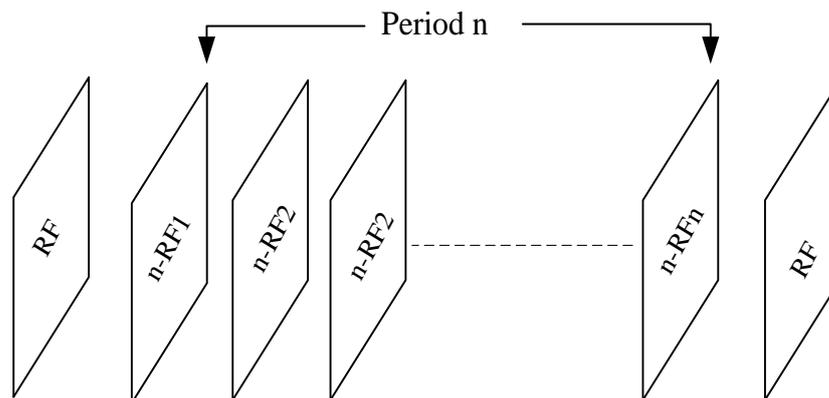


*Figure 5.6 shows sample frames of videos test*

### 5.3.1 The periodicity of reference frames

In this section we shall examine the influence of period of selecting RF on compression method. The RF will be sent to the decoder every  $n$  frames ( $n$  represent the distance between two nearest I-frames) as shown in Figure 5.7. We tested the effect of

periodicity of RF on compression efficiency and quality of recovered frame at various periods of RF. In this test, the periodicity of RF is chosen as  $n= 10, 15, 20, 25, 30, 35, 40, 45$  and  $50$  inter-frames or non-Reference Frame ( $n$ -RF). All tests were performed using the constant threshold  $THR1= 2$  and  $THR2= 8$  for compressing the high frequency sub-bands of level 1 and 2 respectively.



*Figure 5.7 shows a periodic Reference Frame (RF) in video compression proposed scheme*

The results of these experiments are presented in Figure 5.8 and Figure 5.9. As can be seen from Figure 5.8, the Compression Ratio (CR) tends to increase marginally when the period  $n$  of RF increases perhaps because more temporal mismatching blocks will be produced through CVQ processing, especially when the period  $n$  exceeds 25 frames. This trend is evidently similar for all the tested videos, although there are differences between CRs for different videos due to variation in speed of change. Figure 5.9 and Table 5.1 show the mean (M) and standard deviation (STD) of PSNR that were achieved for different period  $n$ . the data of Figures 5.8 and 5.9 shows that there were a trade-off between CR and quality .

From the experimental results it may be estimated that selecting the RF encoding every 25 frames achieves an effective solution for video compression with a good CR, acceptable quality and encoding time which is a suitable for constraint applications.

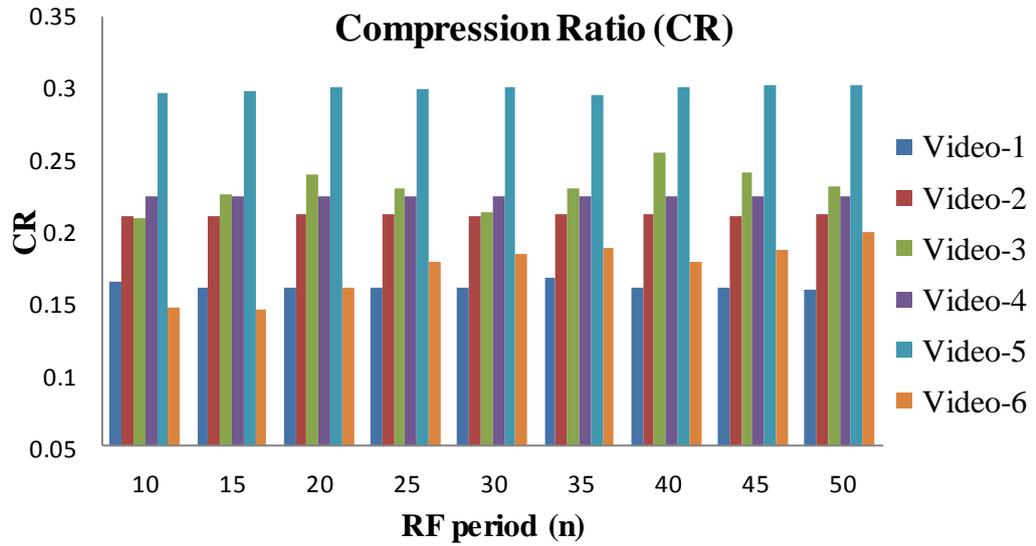


Figure 5.8 the effect of period RF on CR

n	video-1		video-2		video-3		video-4		video-5		video-6	
	Mean	STD										
10	37.776	1.037	33.772	0.870	35.089	3.274	36.002	1.892	36.577	0.844	31.359	0.976
15	37.489	1.077	33.772	0.870	35.846	2.962	36.002	1.892	36.588	0.857	31.091	1.275
20	37.455	1.046	33.789	0.842	36.138	2.901	36.002	1.892	36.662	0.712	31.540	0.921
25	37.191	1.173	33.789	0.842	35.686	3.294	36.002	1.892	36.612	0.844	31.685	1.041
30	37.117	1.291	33.772	0.870	35.214	3.142	36.002	1.892	36.657	0.698	31.805	0.962
35	37.085	1.277	33.789	0.842	35.286	3.679	36.002	1.892	36.540	0.971	31.946	0.643
40	37.023	1.070	33.789	0.842	36.734	2.415	36.002	1.892	36.662	0.712	31.641	1.050
45	36.956	1.104	33.772	0.870	36.301	2.748	36.002	1.892	36.682	0.679	31.732	1.145
50	37.074	1.032	33.789	0.842	35.740	3.209	36.002	1.892	36.682	0.679	32.078	0.577

Table 5.1 The Mean and STD of PSNR for decompression video at different period RF

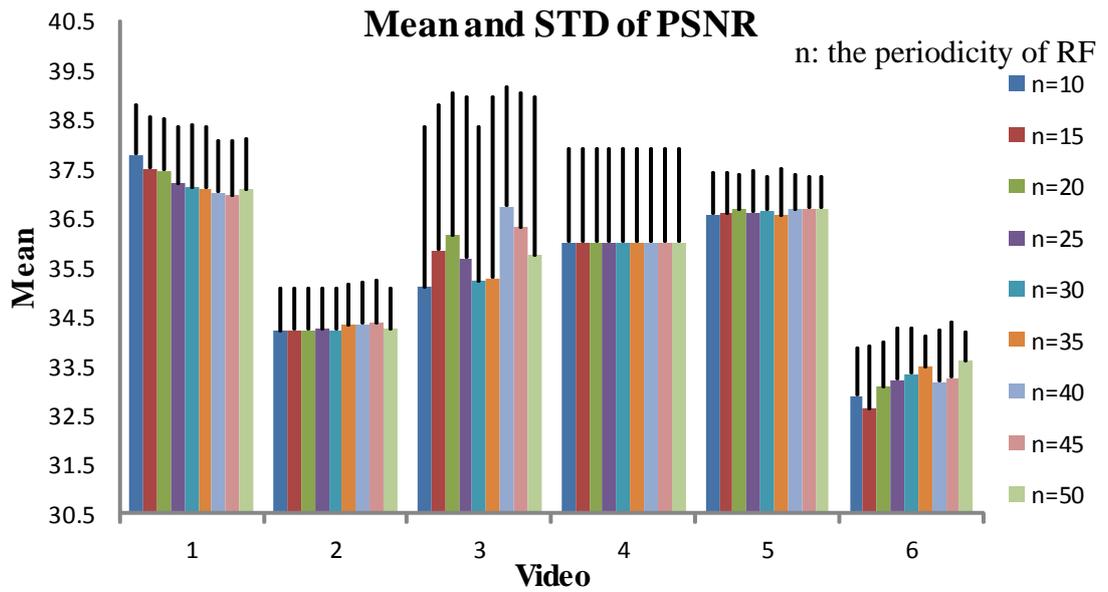


Figure 5.9 the mean and STD of PSNR of tested videos at different RF period

The results in Figure 5.9 confirm that PSNR values are marginally affected by the value of  $n$  in the range 10-50. However, for all  $n$  values the mean PSNR value is influenced by the type of video used. In fact, the lowest PSNR mean is achieved with video 6, which contains moving objects and the second lowest PSNR mean is achieved with video 2 of earthquake, which involve significant temporal variations. Moreover, from Figure 5.8 similar observations can be made about the mean CR values for those two videos. This is also confirmed next.

### 5.3.2 Compression analysis

The achieved compression ratios (CR) of tested videos are shown in Figure 5.10. The compression ratio reflects the video complexity. If the video contains complex objects; many mismatched blocks will be produced during the compression process. Therefore, the compression ratio varies from one video to another.

In order to show the reconstruction quality, the Peak Signal to Noise Ratio (PSNR) is used as quality measurement criterion. All tests were performed using the same compression threshold as mentioned in section 5.2.1. Table 5.2 shows the Mean and Standard Deviation (STD) of PSNR calculated for each frame of video.

It can be seen from Figure 5.10 and Table 5.2 that the quality is proportional to the CR. For instance, video-6 contains objects moving within frames. Therefore, the CR and PSNR are small. In contrast, the objects and camera are moving in video-5, so that the

CR and PSNR are relatively high. Since both CR and image quality are dependent on the block similarity thresholds and therefore the proposed compression should adapt the select these thresholds according to the video complexity.

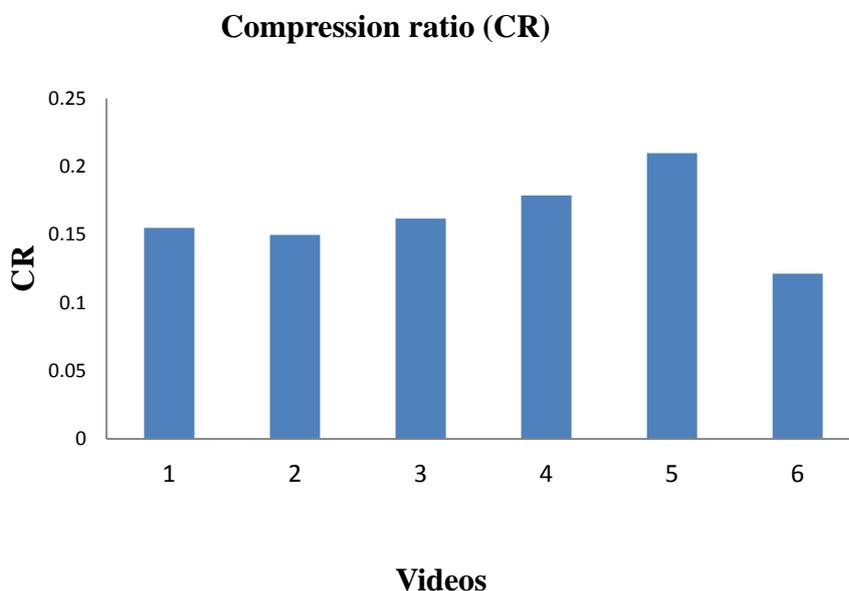


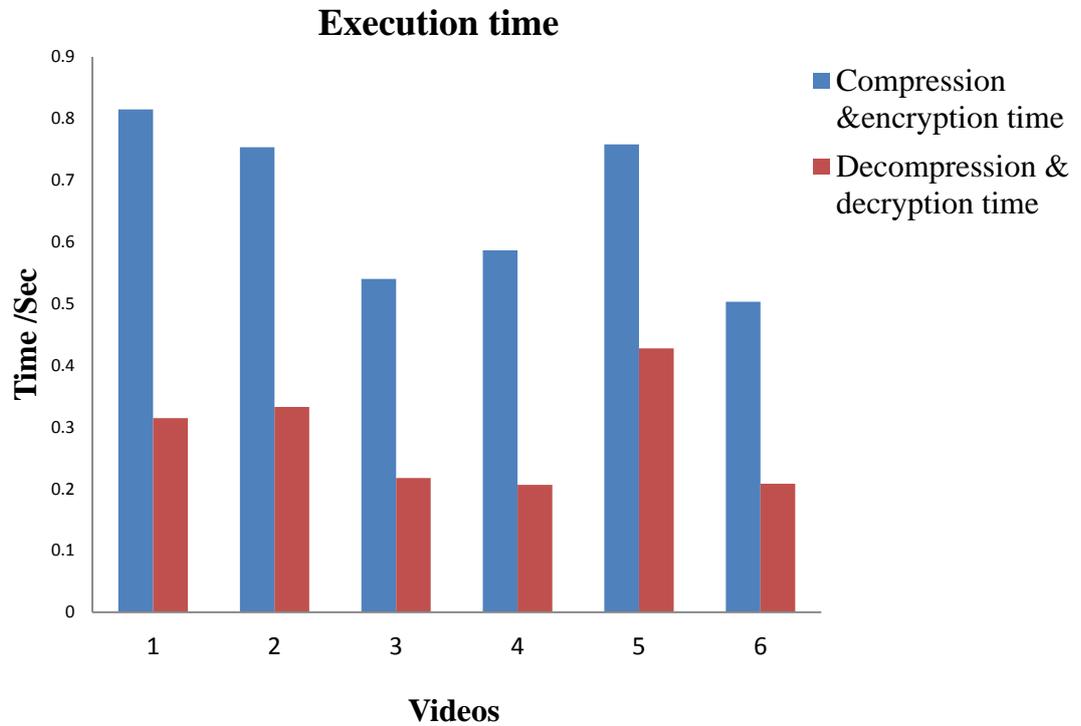
Figure 5.10 displays CR for tested video

Video	Video Complexity	STD	Mean
Video-1	VIP men	0.6335	37.7231
Video-2	Shaky car (earthquake)	0.7835	32.7242
Video-3	Highway car traffic	0.5136	36.1157
Video-4	Slow rhinos with a car	1.6364	34.4655
Video-5	VIP lane departure (highway)	0.6552	36.8360
Video-6	Xylophone	0.2940	31.2958

Table 5.2 the Mean and Standard Deviation (STD) of PSNR for decompression and decryption video

Figure 5.11 shows execution time for compression-encryption and decompression-decryption of 100 frames for each video. The processing time for video-1, including compression and encryption, was 0.82 sec while for video- 3 was 0.54 sec. The

processing time difference reflects the complexity of the videos in terms of the temporal variation



*Figure 5.11 illustrates the execution time for video compression-encryption and decompression-decryption*

The acceptable video streaming assumption is based on the widely agreed fact that the human vision system require a frame rate of the supplied videos be 30 frames/sec. In this case 100 frames need to be processed in 3.33 seconds. Our achieved record on encoding (compression and Encryption) processing time is significantly < 1 second. Admittedly, this may be due to the assumed small frame size and we need to establish acceptable time processing for larger size video frames. In Chapter 7, we shall investigate the performance for larger sizes and for RGB videos.

The performance of this approach, in terms of CR and PSNR, is also compared with other schemes (Shrestha and Wahid 2010) that proposed a hybrid DWT and DCT for biomedical image and video compression. For this comparison we used the same video (endoscopic) that has been used when applying our proposed method. The results show that our method significantly outperforms their method. Note that Shrestha & Wahid method was shown to outperform JPEG and SPIHT schemes by nearly 7% as shown in Table 5. 3. Furthermore, the testing videos and endoscopic video are tested in the

proposed video compression for constant CR at 97 %, where the average PSNR in our method was 30.9 measured for 100 frames as shown in Table 5.4 This improvement in our results was because (Shrestha and Wahid 2010) have discarded all the high-frequency sub-bands, while in our method we applied the DCT to these sub-bands with minimum loss in the quality.

Method	CR	PSNR(dB)
DWT algorithm	97%	19.84
DCT algorithm	97%	14.63
Shrestha & Wahid, 2010)scheme	97%	24.24
Proposal scheme	97.7%	29.979

*Table 5. 3PSNR of reconstructed frame of endoscopic video*

Video	STD	Mean
Video-1	0.3746	33.5896
Video-2	0.6356	30.5988
Video-3	0.2683	28.9361
Video-4	1.3769	31.9616
Video-5	0.7612	32.5053
Video-6	0.6419	29.7121
Endoscopic	0.4774	30.9798

*Table 5.4 the mean and STD of PSNR of tested video at constant CR = 97%.*

Finally, we shall demonstrate in theory the suitability of this algorithm for transmission over a communication channel that uses Wireless Local Area Network (WLAN), is 802.11b with stream data rate of 5.5 M bit/sec (Goldsmith 2005). The required bit rate for this raw video will be calculated as below:

$$256 \times 320 \times 8 \times 100 = 65536000 \text{ bits}$$

The required time to transmit uncompressed 100 frames using the above wireless card, is  $65536000/5500000 \approx 12$  sec. Based on figures 5.10 and 5.11, the achieved CR and the execution time (time needed for encryption and compression) for video-1 was 0.15 and 0.8 second respectively. Hence, the transmission time for a compressed 100 frames of video-1 with this WLAN is equal to  $0.15 \times (65536000/5500000) + 0.8 = 2.587$  second. Thus, 30 frames transmission will require 0.77sec. Furthermore, the mean of the encoding and compression ratio of all tested video is 0.659 sec and 0.163 respectively. So, the average time of transmission 100 frames is equal to  $0.163 \times (65536000/5500000) + 0.0.659 = 1.9422$  second i.e. the proposed algorithm will code these videos at the source video rate within the assumed constraints of processing time.

### 5.3.3 Encryption analysis

The encryption analysis was performed in three different ways; histogram analysis to assess the viability of any statistical attack; correlation analysis to assess the viability of frequency attacks, and the PSNR analysis to assess the availability of significant information left in the clear. These types of analysis have been used in (Huang and Sakurai 2011), (Pareek, Patidar and Sud 2006), and (Sathishkumar, Ramachandran and Bagan 2012).

#### 5.3.3.1 Histogram analysis

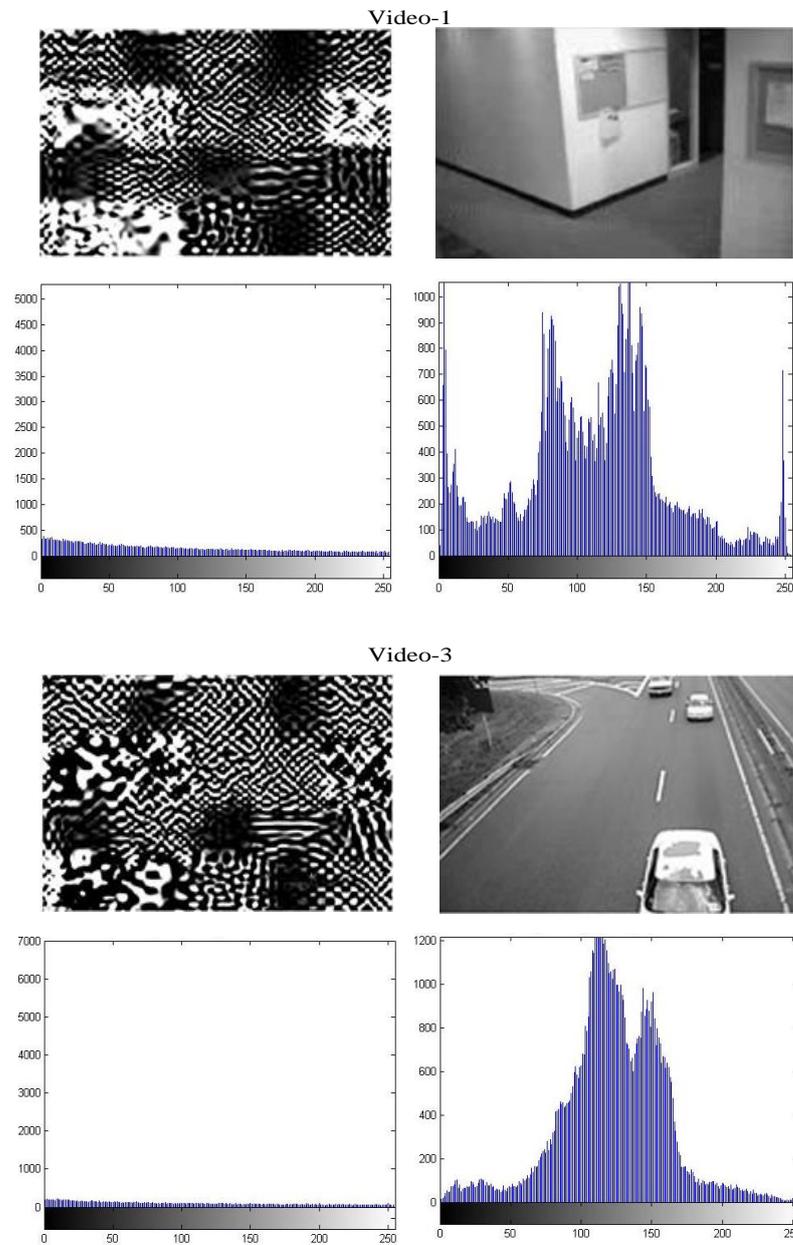
Figure 5.12 and Figure 5.13 shows a frame selected from video 1,3 and 6, the figure shows the frame after encryption and decryption and their corresponding histograms. In addition, we calculate the MDMF between the histogram of encrypted and unencrypted frames by using the equation 3.6 which is given in chapter 3.

video	1	12	25
$HS_i$	42222	41656	43111
$HS_0$	30952	30524	29963
$HS_{255}$	13468	13749	14066
MDMF	64432	63793	65126

Table5. 5 the MDMF of some encrypted video

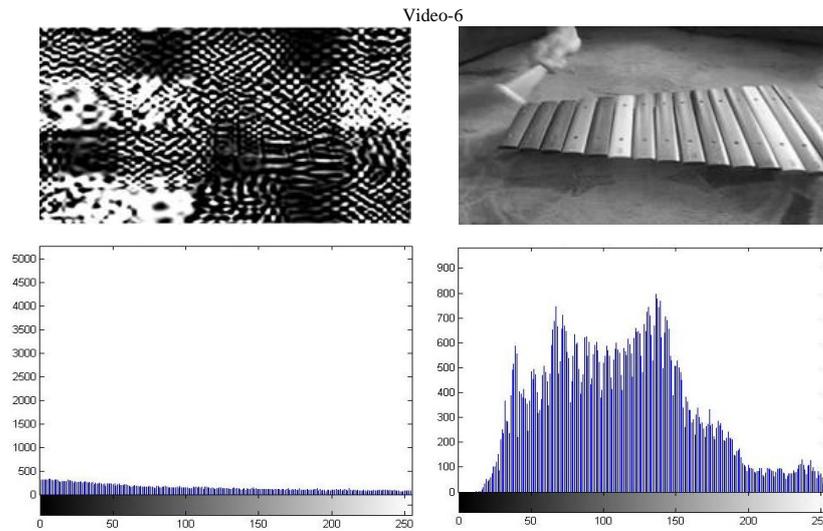
The MDMF for this histogram shows in Table 5.5. The MDMF values showing that the ciphered video frames are different from the deciphered frames. As result, the histogram

of the encrypted frame does not provide any information that can be used for any statistical attack.



A1	A2
B1	B2
C1	C2
D1	D2

*Figure 5.12 (A1, C1) represents encrypted frame, (B1, D1) represents histogram of encrypted frame, (A2, C2) represents decrypted frame, (B2, D2) represents histogram of decrypted frame*



A	B
C	D

 Figure 5.13 (A) represents encrypted frame, (B) represents decrypted frame (C) represents histogram of encrypted frame, (D) represents histogram of decrypted frame

### 5.3.3.2 Correlation analysis

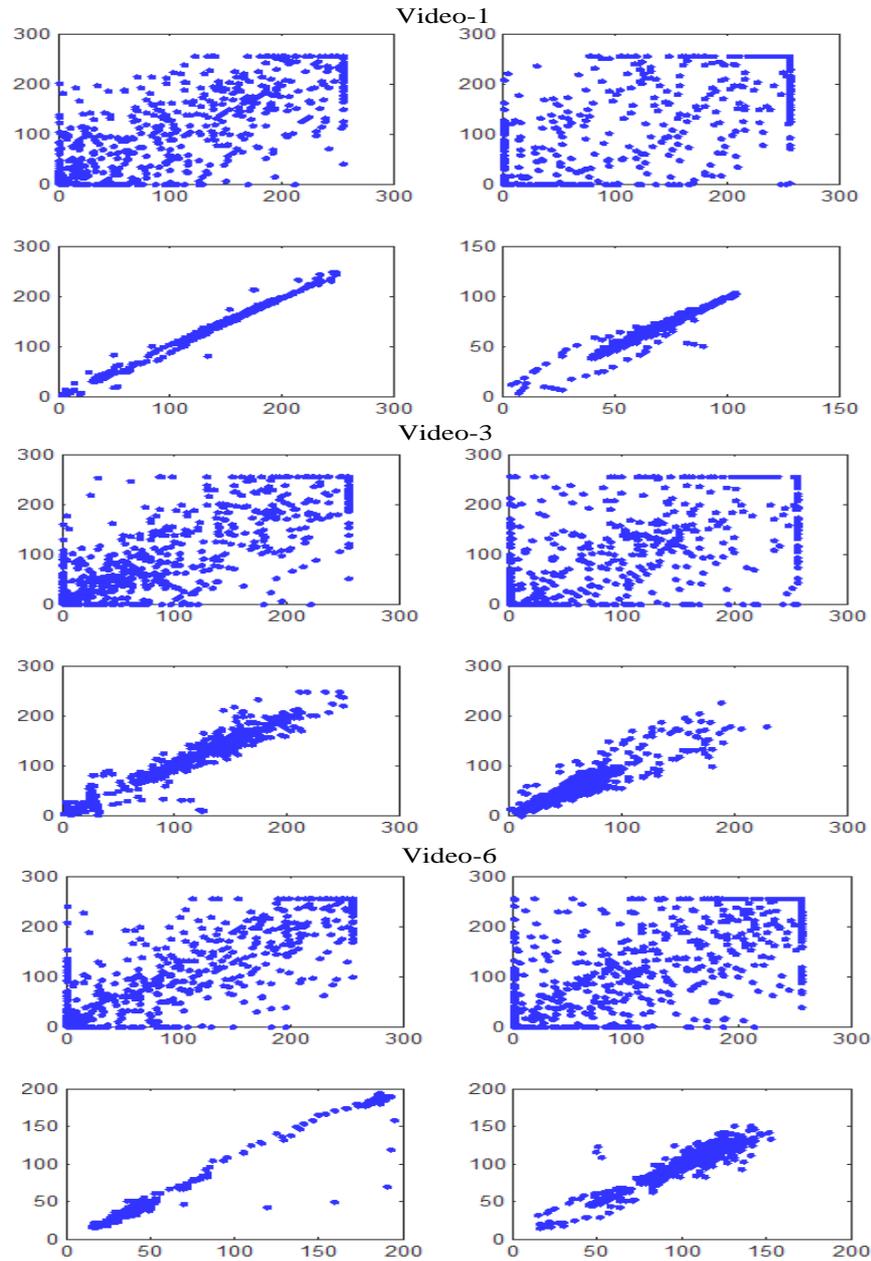
Correlation is a measure of the relation between two variables. So, if the correlation between two variables is close to zero, then predicting their relationship becomes difficult. In order to evaluate the correlation between two adjacent pixels, in vertical and horizontal direction within the same frame, 1000 pairs of horizontally and vertically adjacent pixels are selected randomly from the original and encrypted frames. The correlation of the selected coefficients is then calculated by applying equation 3. 7, given in chapter 3. Table 5.6 shows that the correlation of the original frame is 1 or close to 1. In contrast, the correlation of the encrypted frame is close to 0.

video	Encrypted frame		Original frame	
	Correlation coefficient		Correlation coefficient	
	Horizontal	Vertical	Horizontal	Vertical
1	-0.0014	- 0.018	-1	-1
2	-0.0059	- 0.0052	-1	-1
3	-0.005	- 0.0011	0.999	0.989
4	-0.0066	- 0.0012	-1	-1
5	-0.0061	- 0.00124	-1	-1
6	-0.0022	- 0.0012	0.998	1

Table 5.6 the Correlation coefficient of adjacent pixels

The scattered plots of the correlation of neighbouring pixels in the original and the encrypted frames are shown in Figure 5.14. It can be seen that neighbouring pixels in

the encrypted frame are of low correlation. In contrast, the correlation of neighbouring pixels values in the original frame was around  $45^\circ$  diagonal line, which indicates that two neighbouring pixels are highly correlated.



A1	A2
B1	B2
C1	C2
D1	D2
E1	E2
F1	F2

 Figure 5.14 illustrates the correlation of two adjacent pixels in the encrypted and decrypted frames, (A1, C1, E1) represent the correlation test in horizontal direction of the encrypted frame in video 1,3 and 6 respectively, (A2, C2, E2) represents correlation test of encrypted frame in vertical direction, (B1, D1, F1) and (B2, D2, F2) represents the correlation test for decrypted frame in the horizontal and vertical direction respectively.

### 5.3.3.3 *PSNR analysis*

Peak signal to noise ratio is commonly used as the objective measure to assess intelligibility of reconstructed image. Generally, when  $PSNR > 30dB$ , the quality of reconstruction is estimated as acceptable (Huang and Sakurai 2011), the mean and STD of PSNR for ciphered frames are shown in Table 5.7. The low PSNR values reflect the difficulty in recovering the original frame from the encrypted frame, without knowing the secret key of encryption algorithm.

Video	STD	Mean
Video-1	0.4075	12.9150
Video-2	0.4338	14.7446
Video-3	1.2436	15.1161
Video-4	2.1767	13.1353
Video-5	0.3196	13.2366
Video-6	0.0569	13.2158

Table 5. 7 the Mean and Standard Deviation (STD) of PSNR for encrypted video

### 5.3.3.4 *Attack complexity analysis*

In order to evaluate the encryption security level of our approach in terms of attack complexity, we computed the number of possible combinations of permutations and compared it with AES method with 256 bit keys  $\approx 10^{77}$ . The total number of possible combinations of block permutations constructed from wavelet sub-bands of level is  $3 \approx 10^{161}$  which is greater than the number of possible permutations of AES, thus making a brute force attack on the AES key more efficient than trying to reshuffle the scrambled blocks of our proposal encryption (Unterweger and Uhl 2012).

## 5.4 Conclusion

In this chapter, we adapted the still image JDWCT-CVQ simultaneous compression and encryption for secure video streaming and tested its performance in terms of a variety of compression and security measures. The video compression has been performed using block based similarity by combining DWT , DCT and CVQ (JDWCT-CVQ) to achieve high compression with an acceptable quality compared with non-block based approach. The compression algorithm reduced the computational time for block matching between reference frame and current non-reference frame. In addition, the matching criterion of our proposal, which is used to quantify the similarity between blocks, is very fast because only subtraction operation is included. The selective encryption, based on wavelet coefficient scrambling using two LFSRs.

We have demonstrated that this approach provides a solution for secured video streaming with a relatively high speed processing time while maintain reasonably good quality and high security level with simple stream cipher encryption. We investigated the effect of the periodicity of selecting RF on the compression efficiency, image quality and encoding time. The experimental results show that the period RF for every 25 frames realized high compression efficiency with acceptable quality. The encryption analysis includes histogram analysis, correlation analysis and PSNR. The security analysis shows that a cipher algorithm is secure from statistical analysis attack and frequency analysis attack.

In the next chapter we shall use JDWCT-CVQ-Edge in video compression and encryption to further optimize the computational performance for simultaneous video compression and encryption.

## Chapter 6

# Edges Sensing for Simultaneous Video Compression and Encryption

Applying both compression and encryption techniques on video streaming is one of the challenges where speed, quality and the security are paramount. In the last chapter, we develop and tested the performance of the video JDWCT- CVQ scheme by expanding the use of CVQ for the reduction of the temporal redundancies across sequences of frames between successive RFs. Although the scheme succeeded in meeting the requirements for video streaming under the strict constrains, and high CR, the video quality measures was fluctuating according to the complexity of the video content in terms of temporal variation in texture. In chapter 4, section 4.2 we described the statistical properties of high frequency sub-bands of DWT and we established that these properties provide an easy way of gaining spatial information about significant image features (e.g. edges). This was used to develop the JDWCT-CVQ-Edge image compression scheme which was shown to yield reasonably high image quality by preserving the significant image features including edges. In this chapter, we expand on that work and propose a wavelet based dynamic edge sensing in order to encrypt the low frequency sub-band edges while compressing the high frequency sub-bands. The aim of this work is to improve image quality, maintain high security of encryption and to optimize the computational performance for both encryption and compression. Both encryption and compression are based on edges detected from the wavelet high frequency sub-bands. The DWT provides dynamic edges detection which makes the compression ratio dynamic and provides different approach to a selective encryption. Compression and encryption are achieved individually on different wavelet sub-bands, and thus implemented simultaneously as described in chapter 5, section 5.2.

The compression algorithm includes two major steps, Reference Frame (RF) encoding and non-Reference (n-RF) Frame encoding. As in the last scheme, the compression is applied on high frequency sub-band of level 1 and 2. In the previous chapters, we used LFSR key stream cipher for encryption, and although this may be satisfactory, their main weakness relates to their generation. We modify the encryption algorithm by utilising the chaotic logistic map combined with sine map to scramble the wavelet low frequency sub-band coefficients corresponding to the edges extracted from the low

frequency sub-band. The locations of edges are mapped from the high frequency sub-bands using different thresholds. In the first two sections, below, we describe the chaotic logistic map as well as the sine map, and investigate their combination to provide the desired security. In section 6.3, we shall utilize the statistical properties of high frequency sub-bands of DWT for video compression and encryption. Section 6.4 presents the experimental results to test the performance of our proposed method for video compression and encryption and we shall compare these results with existing approaches used in chapter 5.

## 6.1 Chaotic logistic map for pseudo-random number generation

Traditional LFSR generation use fixed length randomly initialised feedback register using primitive polynomials over finite fields. The generated stream has relatively short length, before repeating itself, which is determined by the length of the initial register, Chaotic random number generation overcomes this problem.

In general, a dynamic system is said to be chaotic if it satisfies the following condition:

1. Sensitive to initial condition, where any change in the initial value will produce unpredictable different paths.
2. Irregular motion in phase space and
3. The periodic points are dense in the  $x$  path, where each point in the space is randomly approached closely by repeated orbits.

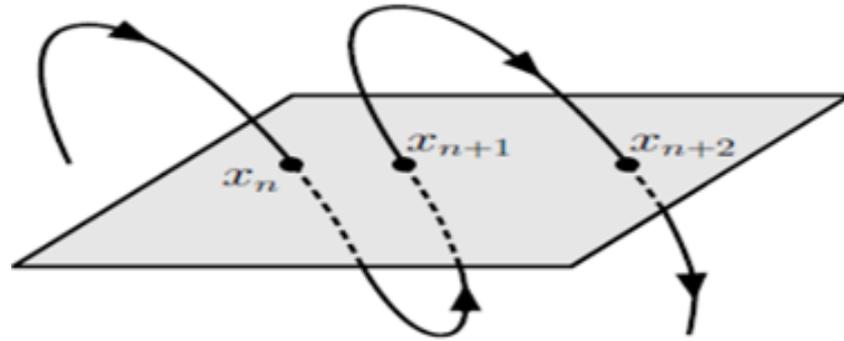
The logistic map is an recursive polynomial function of degree 2 defined as follows

$$x_{n+1} = rx_n(1 - x_n) \quad 6.1$$

Where  $r$  is the control parameter and  $n \in \mathbb{Z}^+$ , if  $n = 0$ ,  $x_0$  is known as initial condition.

The continuous dynamic system of the logistic map is a mapping  $f: x \rightarrow x$  from the state space to itself; see Figure 6.1, defined as follows:

$$x_{n+1} = f(x_n) \quad 6.2$$



*Figure 6.1 represents the curve of the orbit map*

The logistic map can be represented by using a graphical method called a cobweb diagram. The cobweb diagram shows the iterations of control parameter ( $r$ ) and initial condition value ( $x_0$ ) of chaotic logistic map. Figure 6.2 show the cobweb of logistic map at different initial condition and control parameters. From equation 6.1, Figure 6.1 and Figure 6.2 it can be shown that any small error in the initial condition ( $x_0$ ) or control parameter ( $r$ ) of logistic map will correspond to a large difference in  $x_{n+1}$  paths. Therefore, the logistic map satisfied the chaos conditions and Chaotic Logistic Map (CLM) can be used as pseudo random generator by iterate equation 6.1 and used the  $(x_0, r)$  as secret keys. This property can be utilized in video encryption as we proposed in section 6.3 (Kocarev and Lian 2011) and (Mao and Chen 2005).

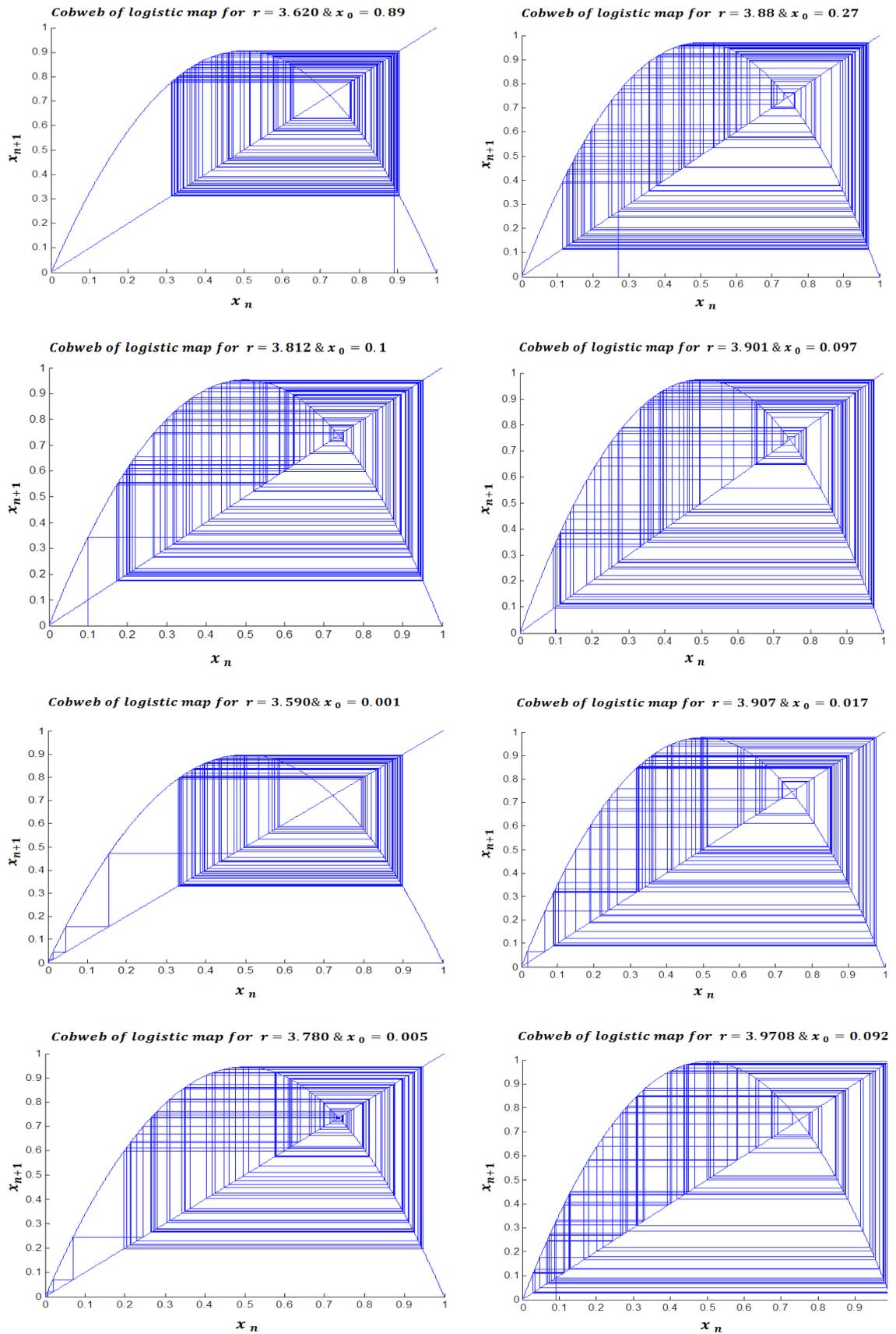


Figure 6.2 A cobweb diagram of the logistic map, showing chaotic behaviour for various values of initial condition ( $x_0$ ) and control parameters ( $r$ ),  $n$  is the number of iterations.

## 6.2 Combined sine map and chaotic logistic map

Existing chaos based encryption techniques are considered to be good for image encryption because of their properties: firstly, the chaos system is sensitive to initial condition and control parameters as described in section 6.1. It has a clear random behaviour and the chaos have aperiodic long-term behaviour in a deterministic system that exhibits sensitive dependence on parameters  $x_0$  and  $r$ .

In order to improve the behaviour of chaotic map, many researchers have suggested a combination of two chaotic maps to achieve higher security level of image encryption. Sarun and Tanachard (Maksuanpan, Veerawadtanapong and San-Um 2014) presented an image encryption scheme based on combined sine and cosine chaotic maps, as shown below, to increase the entropy random-bit sources.

$$x_{n+1} = \cos(b x_n) + \sin(a x_n) \quad 6.3$$

Where ‘ $b$ ’ and ‘ $a$ ’ are frequencies of cosine and sine function and  $x_0$  is the initial condition. Encryption is performed by converting the image pixels into binary form and XORed with bit streams produced by iterating equation 6.3.

A combination of logistic and sine chaotic maps is another scheme that is used for image encryption. Chen and co-workers (Chen, Zhang and Zhou 2012) proposed a nonlinear combination of Logistic Map (LM) and Sine Map (SMa) in which the control parameter  $r$  of LM is driven from SMa as shown below.

The proposed combined system equation can be defined as shown in the equation 6.4.

$$x_{n+1} = r_n x_n (1 - x_n) \quad 6.4$$

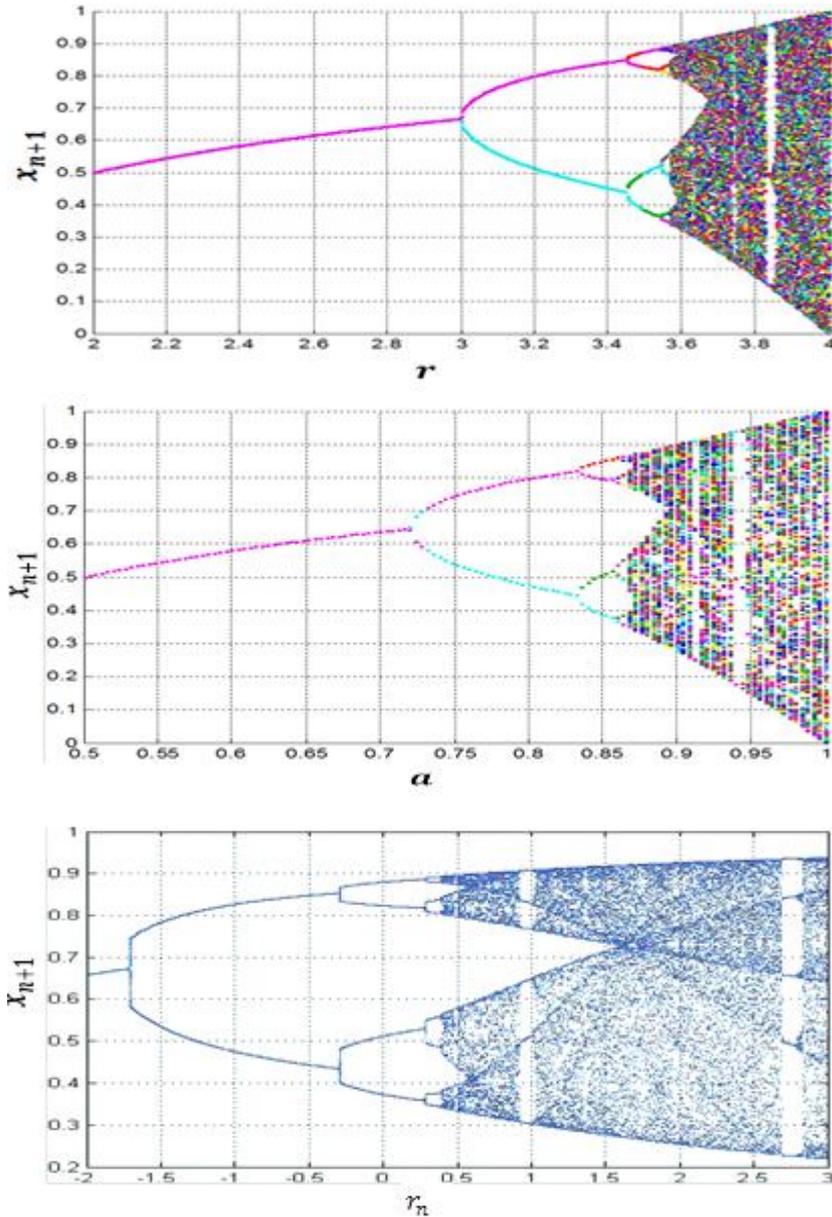
The control parameter  $r_n$  is driven from SMa equation as shown in the equation 6.5 below.

$$r_n = a \cdot \sin(\pi \cdot r_{n-1}) \quad 6.5$$

$a$  : is the control parameter of SMa. The control parameter of the proposed system will be updated using the recurrence equation 6.5.

We investigated the behaviour of the new combination system and compared it with LM and SMa by plotting the bifurcation diagram of these methods as shown in. Figure 6.3 (A, B and C). The bifurcation diagram describes the common phenomenon that occurs in many nonlinear dynamic systems such as vibrated pendulum and switching

converter. Figure 6.3 shows the bifurcation structure between  $r$ ,  $a$  and  $r_n$  where  $r \in [3.55, 4]$ ,  $a \in [0.87, 1]$  and  $r_n \in [1.56, 3]$  respectively. The chaotic region is illustrated in coloured region while the non-chaotic is illustrated in white region. The bifurcation diagram of the combined LM & SMA demonstrates that the new algorithm has larger key space, higher security level and the output chaotic sequence has a more complicated dynamic system compared to the traditional LM and SMA.



A Figure 6.3 bifurcation diagram (A) for logistic map, (B) for sin map (C) for combine  
 B logistic and sin map  
 C

## 6.3 The JDWCT-CVQ-Edge secure for video compression scheme

In this section, we follow and modify the approach followed in chapter 5 for designing the secure video scheme, we organise both compression and encryption independently to be used in parallel as shown in Figure 6.4. Again compression is applied on high frequency sub-bands of level 1 and 2 and encryption is applied on the level 2 low frequency sub-band. However, the modification made in the compression component of the new scheme aims primarily to preserve image quality in the vicinity of the edge features. .

### 6.3.1 A revised Video Compression scheme

The compression of intra-frame (RF) is based on JDWCT-CVQ-Edge image compression scheme, developed in Chapter 4, which encodes the significant coefficients of the high frequency sub-bands that are witnesses to the presence of edge like features. The process starts by subdividing the high frequency sub-bands to non-overlapping blocks of size 16x16 coefficients and then DCT and Compressive Vector Quantization (CVQ) are applied to each block. The revised compression scheme avoids unnecessary computation by not applying the DCT on wavelet blocks of the inter frames (n-RF), i.e. the CVQ method will be applied for the inter-frame (n-RF) blocks without using DCT. The following sections describe the new revised video compression and encryption methods.

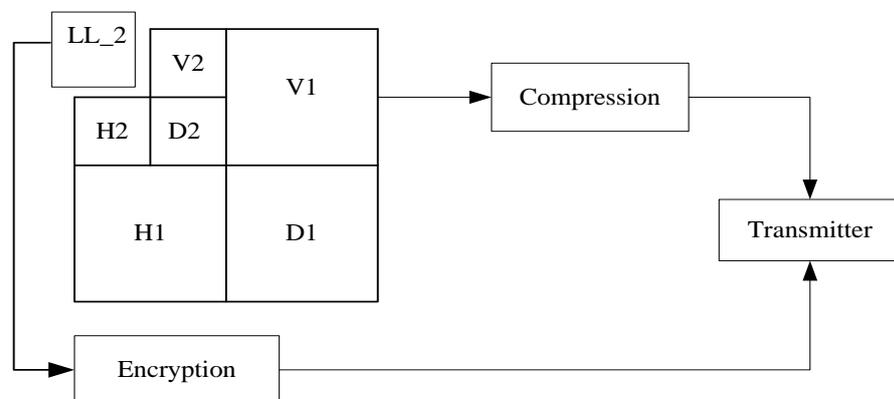


Figure 6.4 The simultaneous compression and encryption scheme

This revised video compression scheme is based on JDWCT-CVQ-Edge. The signs of wavelet coefficients are determined and the coefficients are converted to integer unsigned values. The high frequency sub-bands of level 1 and 2 are quantized individually, in the same manner as they were implemented in chapter 4, section 4.1.2.1. Thus, the significant coefficients (most likely edges) of high frequency sub-bands are determined their Laplace distributions in terms of multiples of the standard deviations. The significant coefficients are furthest away from the '0' mean of each high frequency sub-band. Hence, we use a threshold of the form ( $THR = m * STD$ ) where  $m$  is a real number to filter out non-significant coefficients. The value of  $m$  (usually in the range 0.5-3) is driven from the standard statistic Empirical rule which is demonstrated in chapter 4, section 4.2.2. Moreover, in section 4.2.2, we investigated the effect of applying different thresholds on the compression success parameters, used to control the compression ratio and the quality required, and to sense the amount of significant coefficients needed for the compression and the encryption at the same time. The filtered-out high frequency coefficients are converted to zeroes (i.e. edge detection). Then we sub-divide the high frequency sub-bands into blocks (16x16). To increase the similarity between blocks we applied the DCT on each block of RF. see Figure 6.5.

Recall that the CVQ method defines an equivalence relation on each sub-divided block defined by a similarity function  $Dist$  (equation 4.1.) and a threshold. The equivalence classes (matched) will be represented by one vector. This method will be applied on the RF high frequency sub-band blocks only. Using the conclusions of chapter 5 regarding periodicity of RFs, the RF compressed codebooks will be sent every 25 frames to enhance the video quality. Note that, the thresholds used in determining the equivalence classes of codes are determined from the DCT of the wavelet sub-band blocks. Here, we reduce the computational cost of the CVQ, by using the THR values obtained for the RF blocks to filter out the non-significant coefficients of the n-RF blocks without applying DCT. Therefore, for the n-RFs, the CVQ method will use the THR of the preceding RF, and the matching criterion will be applied on the corresponding blocks of the RF codebook and the n-RF codebook without applying DCT on the n-RF blocks.

Decompression is the inverse of the compression processing; the RF is firstly decompressed, then based on mismatched blocks the n-RF is decompressed and the matched blocks are reproduced from RF.

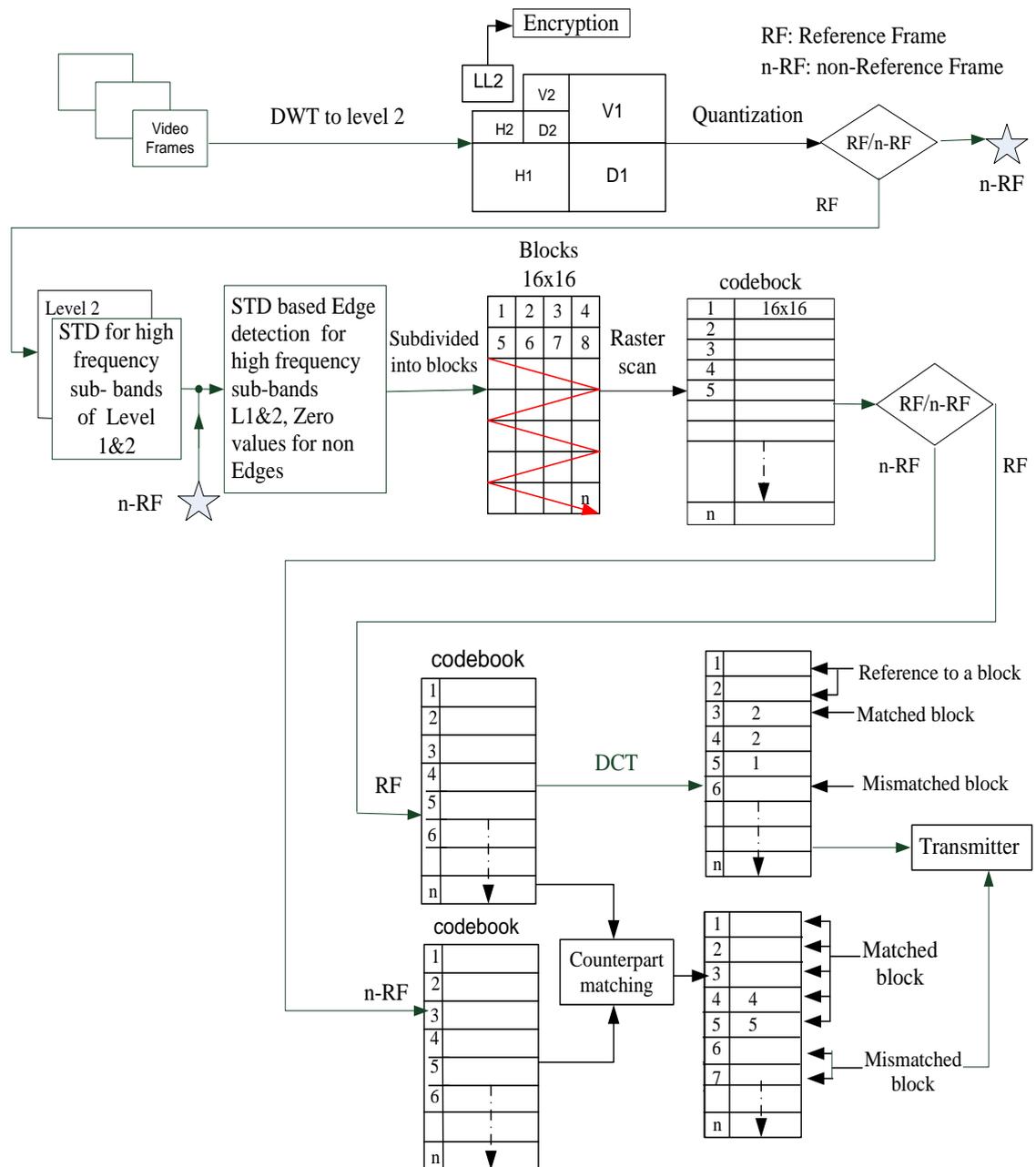


Figure 6.5 diagram illustrates the video compression scheme

### 6.3.2 The Encryption scheme

Here we adopted the approach of (Pommer and Uhl 2003) by selectively encrypting to LL sub-band coefficients corresponding to edge feature detected from the non-LL sub-bands. The locations of such coefficients correspond to those of significant coefficients in the high frequency sub-bands at the same level of decomposition which we already

determined for compression. Hence, encryption is strongly linked to compression without having an overlap in their input.

In summary, encryption is applied to the crucial parts of the low frequency sub-band (LL2), and this means, our encrypting less than 1/16 of the original image size, because our proposed method will not apply the encryption on the entire LL2. The coefficients of the LL2 to be encrypted mapped are identified by the significance coefficients from the high frequency sub-bands of level 2 (see Figure 6.6).

These selected coefficients are scrambled within the LL2 using LFSR supported by a secret key. The secret key of LFSR this time is driven from combining chaotic logistic map and sine map, which is described in section 6.2, as shown in the equation 6.6 below

$$x_{n+1} = u_n \cdot x_n (1 - x_n) \quad 6.6$$

Where  $x_n$  is the initial condition of chaotic logistic map, and  $u_n$  is the control parameter which is defined by the following equation

$$u_n = a \cdot \sin(\pi \cdot u_{n-1}) \quad 6.7$$

Where ‘ $a$ ’ is a real number that controls sine map and ‘ $n$ ’ is an integer representing the iteration index number. The time of iterations is equal to the length of LFSR. Therefore,  $x_0$  and  $u_0$  will be sent to the decoder and the decoder will estimate the secret key of LFSR from  $x_0$  and  $u_0$  for decryption.

After each iteration, the value of  $x_{n+1}$  will be shifted by two digits to the right, truncated to integer value and converted to binary. The secret key will be updated when the RF is changed. We used  $x_1$  and  $x_2$  of previous iteration as the Initial condition  $x_0$  and control parameter  $u_0$  in subsequent secret key update. Finally, the LL2 will be divided into blocks of 16x16 and sent to the transmitter.

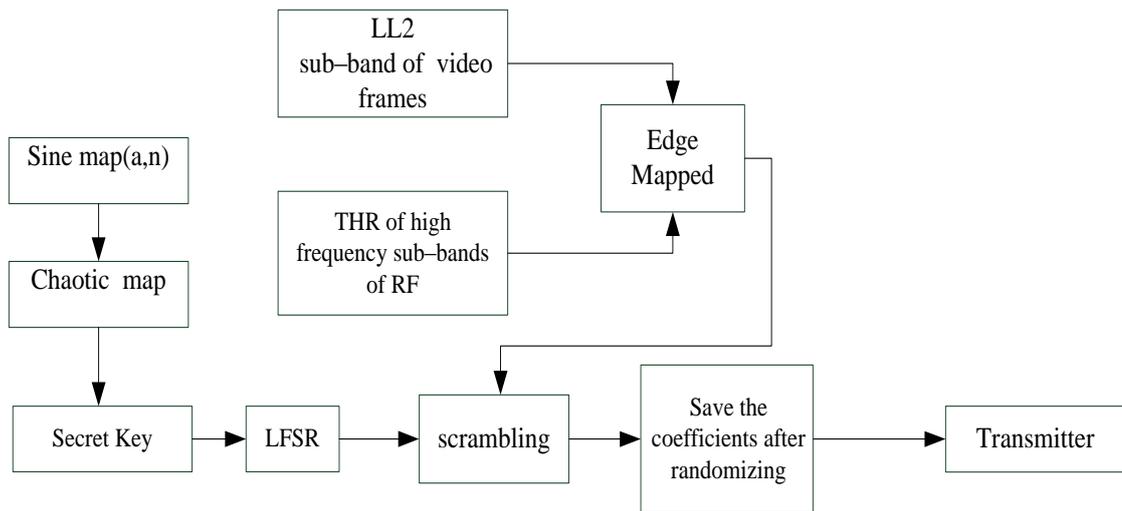


Figure 6.6 represent Encryption scheme

## 6.4 Experimental work and results

In this section, we present experimental work conducted to test the performance of the proposed video compression and encryption based on JDWCT-CVQ-Edge algorithm. Then, we compare the performance of our method with proposed approach of previous chapter and also with other algorithms.

The proposed method was applied on the same videos that have been used in chapter 5. Each video comprises 100 frames that have size (256 x320) pixels in gray level. The proposed scheme demonstrated using MATLAB V 7.10 (R2013a) platform on the same computer mentioned in chapter 3.

### 6.4.1 Compression analysis

The compression ratios (CR) achieved for the tested videos are shown in Figure 6.7. The variation in the compression ratios reflects the variation in the complexity of the videos. If the video contains complex objects; many mismatched blocks will be produced during the compression process. The Mean and STD of PSNR values for the frame of the videos are shown in Table 6.1.

Figure 6.7 and Table 6.1 confirms that the quality is proportional to the CR. For example, video-6 contains objects moving within frames. As result, the CR and PSNR are low compared to the other videos. On other hand, both objects and camera are moving in video-5, so that the CR and PSNR are relatively high.

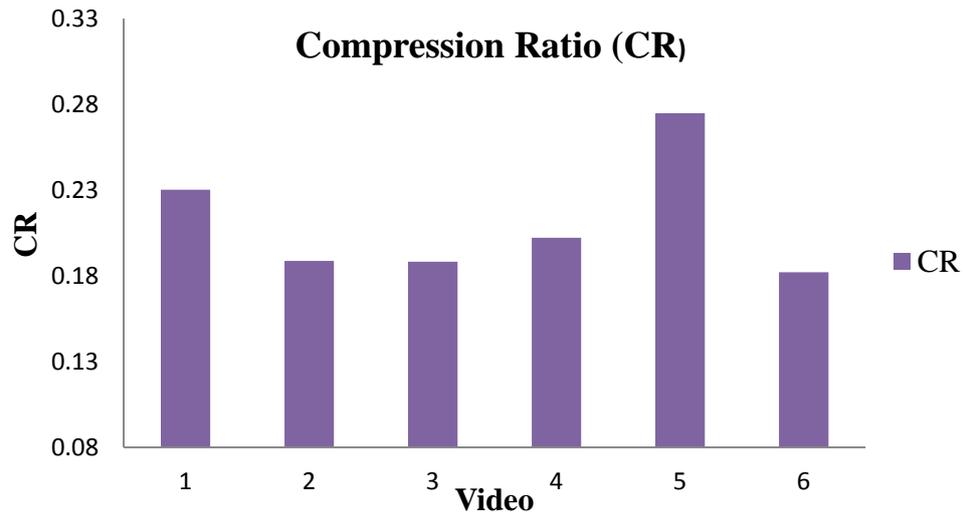


Figure 6.7 CR for tested video

Video	Video Complexity	Mean of PSNR	STD of PSNR
Video-1	VIP men	38.062	0.7480
Video-2	Shaky car (earthquake)	33.110	0.7931
Video-3	Highway car traffic	32.504	0.9231
Video-4	Slow rhinos with a car	34.586	1.7533
Video-5	VIP lane departure (highway)	35.785	0.7945
Video-6	Xylophone	31.465	0.2405

Table 6. 1 The Mean and Standard Deviation (STD) of PSNR for decompression and decryption video

Figure 6.8 shows the execution time for compression-encryption and decompression-decryption for each video. The processing time for the 100 frames of video-1, including compression and encryption, was 0.32 sec, while for video-3 it was 0.38.sec. The time difference in video encoding reflects the complexity of each video which result in variation in number of mismatched blocks

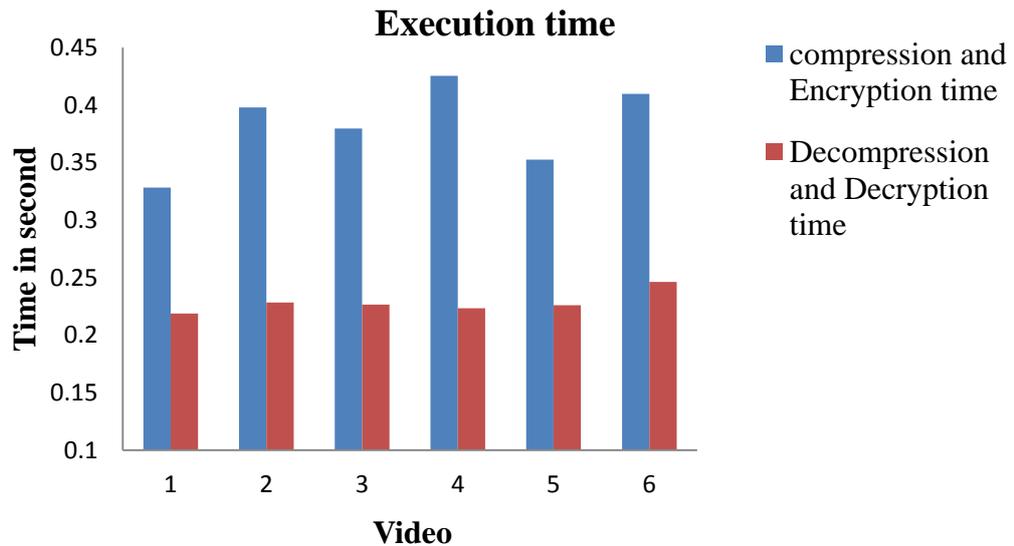


Figure 6.8 The execution time

In order to compare our results with existing standalone DCT and DWT methods, the same video has been used. We have followed the JPEG for DCT algorithm and Set Partitioning in Hierarchical Trees (SPIHT) has been used for DWT algorithm. The quality of reconstruction frame for constant CR at 97 % achieved by our proposed, standalone DCT and DWT is shown in Figure 6.9, from which it may be seen that the quality of our method is better than both DCT and SPIHT methods.

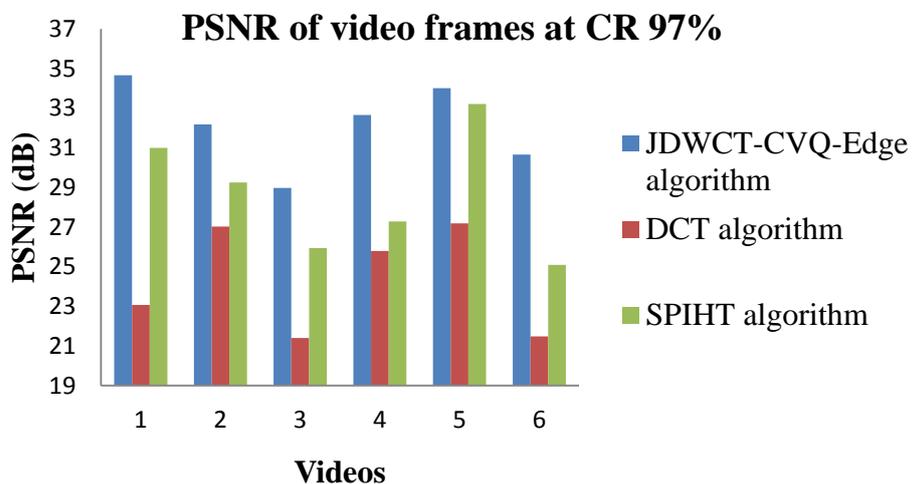


Figure 6.9 illustrates PSNR of video frames at constant CR of 97%

Furthermore, the results of this compression scheme are compared, below, with the performance of the JDWCT-CVQ scheme introduced in chapter 5. For simplicity, we shall denote the method of Chapter 5 as CH5M, while CH6M will refer to our proposed method in this chapter. Table 6.2 shows that except for videos 3 and 5 better quality are achieved by CH6M. However, for videos 3 and 5 better CR is achieved by CH6M. The

cause of this performance is possibly due to discarding the insignificant coefficients in CH6M and the DCT compact the most energy in DC coefficients in CH5M method. On the other hand, the execution time of CH6M is significantly lower than that of CH5M. Therefore, CH6M is significantly better than CH5M in its time processing, but for some videos this is achieved with lower CR and image quality.

Videos		1	2	3	4	5	6
Mean(PSNR)	CH6M	38.062	33.11	32.504	34.586	35.785	31.465
	CH5M	37.723	32.724	36.115	34.465	36.836	31.295
STD(PSNR)	CH6M	0.748	0.793	0.923	1.753	0.794	0.240
	CH5M	0.633	0.783	0.513	1.636	0.655	0.294
CR	CH6M	0.218	0.190	0.160	0.260	0.141	0.292
	CH5M	0.154	0.1497	0.169	0.178	0.209	0.121
Execution time/Sec	CH6M	<b>0.319↓</b>	<b>0.389↓</b>	<b>0.367↓</b>	<b>0.400↓</b>	<b>0.347↓</b>	<b>0.395↓</b>
	CH5M	0.814	0.753	0.540	0.586	0.758	0.503

Table 6.2 The Mean and Standard Deviation (STD) of PSNR for CH6M and CH5M method

#### 6.4.2 Testing the effect of using wavelet different filters

Edge sensing is known, from the literature and the previous work on wavelet decomposing used in video compression (Al-jawad 2009) (Ma 2002), to be influenced by employing different wavelet filter, and the Haar (i.e. db1) filter is known to perform well in detecting sharp edges. Here we complement our experiments in section (6.4.1), we tested the effect of using different wavelet filters on the execution time, compression ratio, and quality based on PSNR. Figure 6.10 shows that on average, the best CR and quality was achieved by “db9” due to the fact that it is the longest filter in the list. In fact, the longer the wavelet filter is the smoother coefficients in the wavelet high frequency sub-bands, and this has a direct effect on the CR and the quality. On the other hands the shortest execution time was achieved when using the shortest wavelet filter, i.e. “db1”. The only exception we had is “Video-6” which has a relatively still background with limited textures compared with the other videos. Taking into account the constrains we are imposing on hardware and communication capability, we recommend to use “db1” for video processing as it expected to assure fast computational time with acceptable quality.

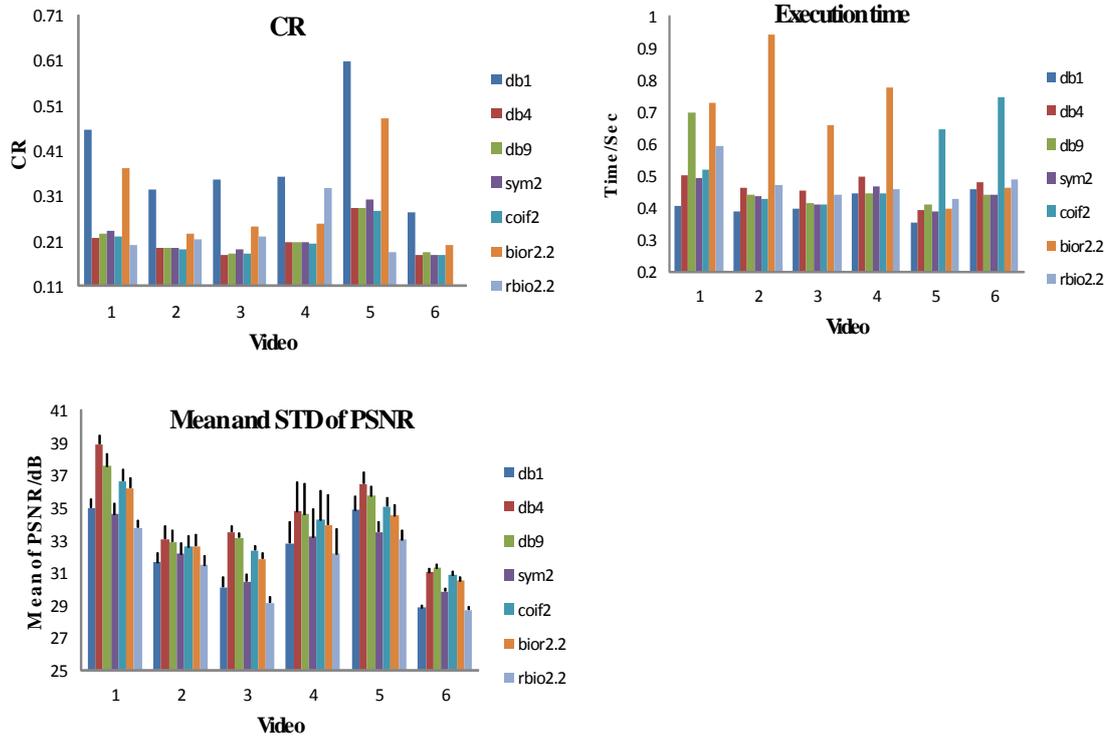


Figure 6.10 CR, execution time and PSNR for different wavelet filters

### 6.4.3 Encryption analysis

In this section, we evaluate the performance of our proposed encryption in three security analyses. The security analysis will include attack complexity, PSNR and histogram analysis.

#### 6.4.3.1 Key-space complexity analysis

The cipher key length of AES is 128,192 or 256 bit. Thus the number of possible combinations  $\approx 10^{77}$  for AES with key length 256 bits (Chown 2002).

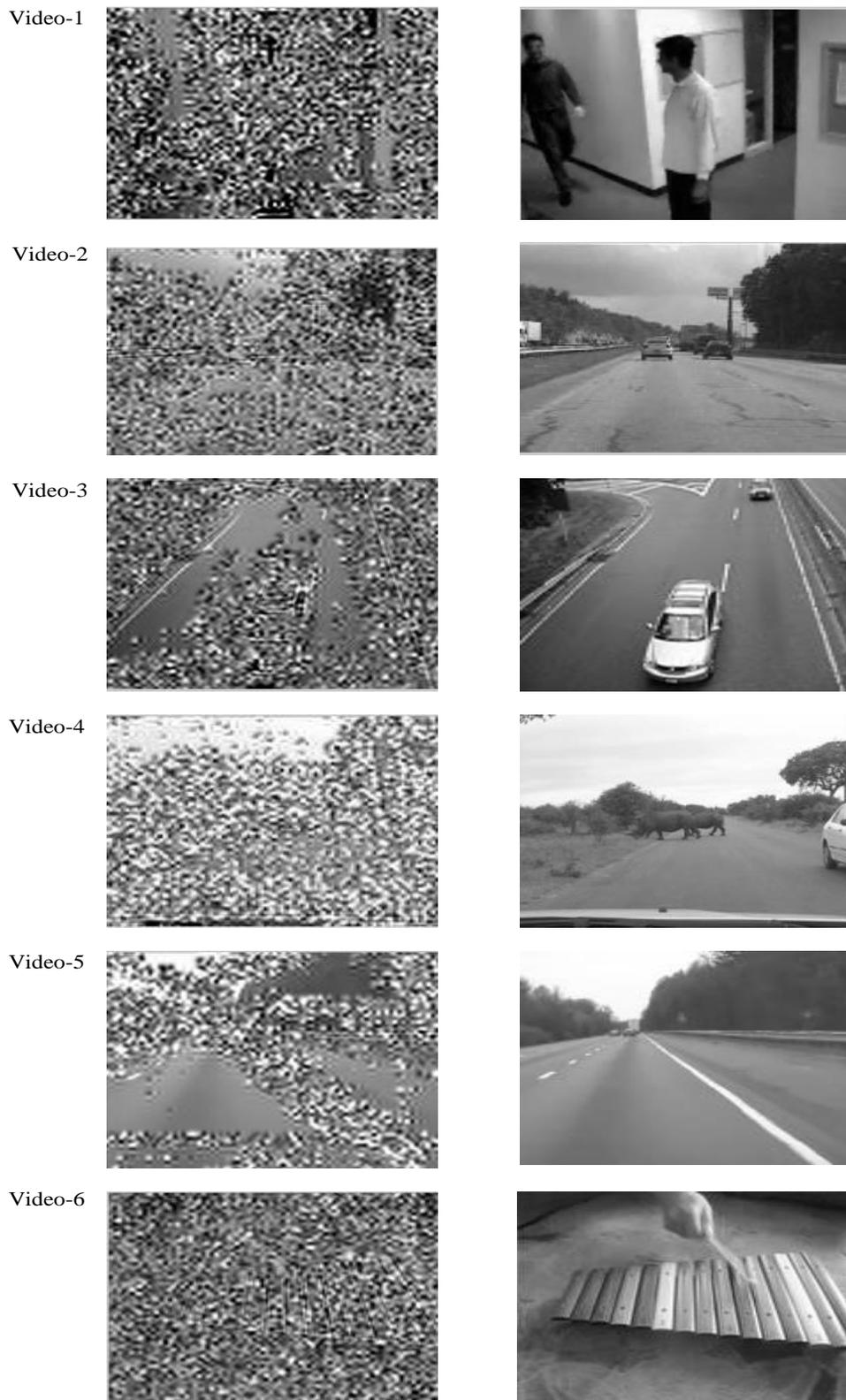
Our proposed encryption is based on scrambling the selected coefficients of low frequency sub-band which are mapped from the significant coefficients of the high frequency sub-bands. We computed the total number of possible combinations of significant coefficients permutations constructed from wavelet sub-bands at level 2 as  $\approx 10^{99}$ , which is greater than the corresponding number of the AES. This make a brute force attack on The AES key more feasible than trying to reshuffle the scrambled blocks of our proposal encryption (Unterweger and Uhl 2012).

### 6.4.3.2 PSNR analysis

The PSNR analysis is commonly used as the objective measure to assess intelligibility of reconstructed image. Generally, when PSNR > 30dB, the quality of reconstruction is estimated as acceptable (Huang and Sakurai 2011). The mean and STD of PSNR for ciphered frames are shown in Table 6.3. The frames of tested videos after encryption-decompression and decompression-decryption are shown in Figure 6.11.

<b>Video</b>	<b>Mean of PSNR</b>	<b>STD of PSNR</b>
Video-1	11.87107	0.443169
Video-2	13.29337	0.398935
Video-3	14.40042	1.282258
Video-4	12.54111	1.755021
Video-5	12.92963	0.607405
Video-6	12.48503	0.176751

Table 6.3 the Mean and Standard Deviation (STD) of PSNR for encrypted video



**A** **B**

Figure 6.11 shows the encrypted and decrypted frames of tested videos (A) encrypted-decompressed frame (B) decrypted-decompressed frame

Note that for these video examples, encryption of the smooth regions (e.g. the road in the video-5) of the image is less successful in concealing them than encrypting the textured regions.

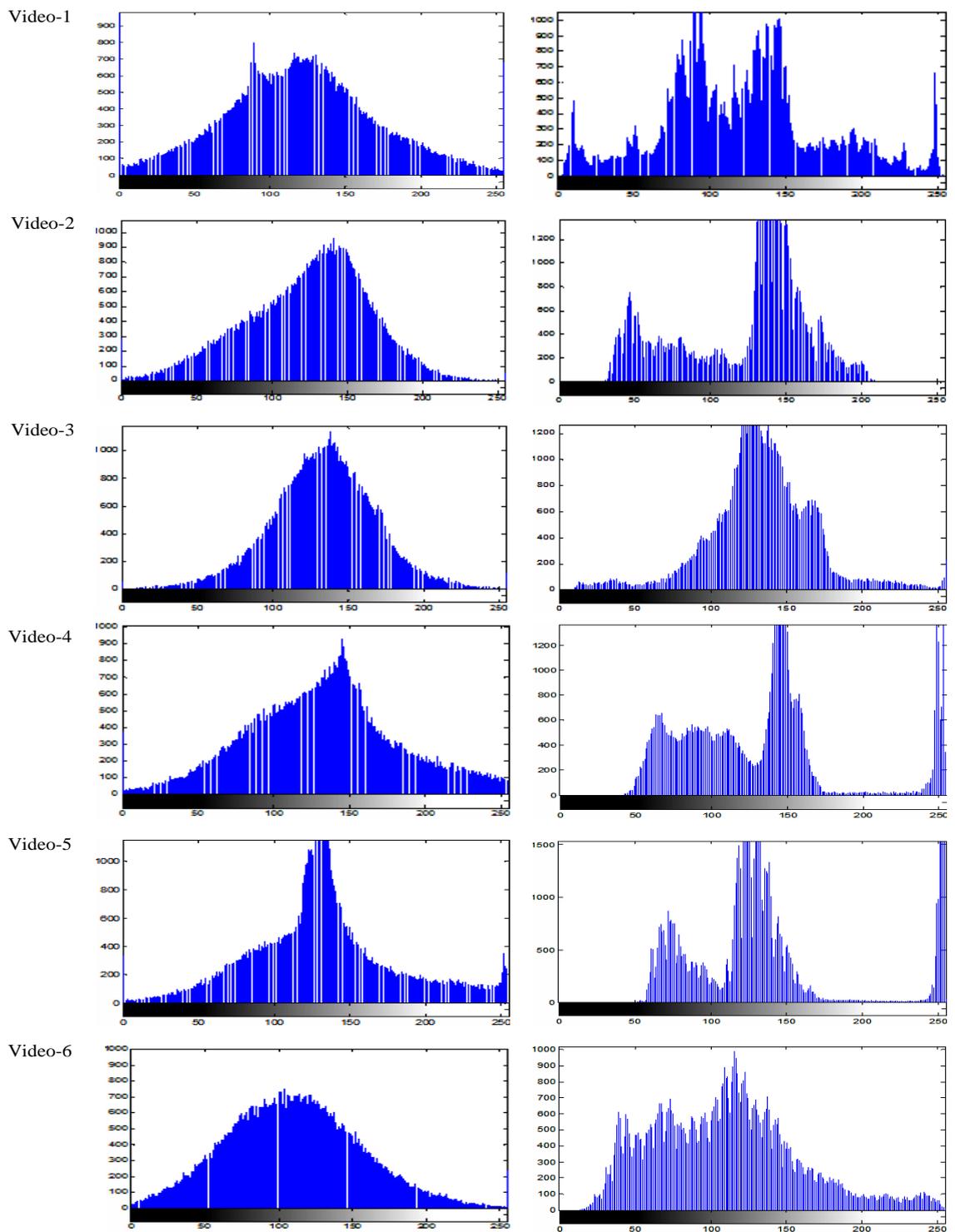
### 6.4.3.3 Histogram analysis

Figure 6.12 shows the histogram of a frame selected from the tested videos after encryption and decryption. The histograms of the encrypted frames are different from the histograms of the unencrypted frames and do not reflect the shape of the original distribution. In addition, the histograms of encrypted frames are not uniformly distributed because the selective encryption is applied only to the frame edges. Therefore, the MDMF values as shown in Table 6.4 that obtained from applying equation 3.6 which is illustrated in chapter 3. The MDMF is relatively high but it is fairly less than MDMF of chapter 5. On other hand, we can make the histogram of the encrypted frame more uniform and increase the MDMF by decreasing THR value as we investigated in chapter 4, section 4.2.2 but it is at the expensive of computational time. The secret key of proposal encryption will change when RF is updated. Therefore, deducing the secret key from the permutations is very difficult for a statistical attack.

Thus our approach will increase the complexity against statistical attack, while the decrypted histogram is nearly similar to the original frame histogram.

video	1	2	3	4	5	6
$HS_i$	50335	503015	50076	50632	51605	515655
$HS_0$	1132	436	103	744	357	641
$HS_{255}$	637	169	37	1443	2261	308
MDMF	51240	50604	50146	51726	52914	516130

Table6. 4 the MDMF of encrypted videos



A B

Figure 6.12 Histogram of encrypted and decrypted frame selected from tested video  
 (A) encrypted frame (B) decrypted frame

## 6.5 Conclusion

In this chapter, a video compression and encryption scheme has been developed that modified the one developed in chapter 5, by focusing to preserve quality in regions of edges and minimise the cost of encryption but increasing key space size using chaotic random generators. This approach has been shown to meet the requirement for video compression and encryption with reasonable quality and effective security level of encryption.

The compression algorithm contains two major steps RF and n-RF encoding. The RF encoding is achieved by using sequence of operations; DWT, wavelet based edge detection, DCT followed by vector quantization. The n-RF will be compressed in terms of RF without applying DCT. Therefore, this approach helps significantly to reduce the computational time compared with video compression method which has been used in chapter 5. For many videos this method marginally improves on compression efficiency and quality in comparison to the JDWCT-CVQ compression schemes of chapter 5 (see section 6.4.1). However, for some other videos, the earlier compression outperforms the current JDWCT-CVQ-edge scheme in terms of CR and PSNR. One can overcome this problem by using adaptive THR values which could in turn slightly increase time complexity.

The encryption analysis includes attack complexity, PSNR and histogram analysis. Experimental results show that the proposed algorithm has the following features; high compression, acceptable quality, and resistance to brute force and statistical attack with low computational processing.

In the next chapter, we shall further refine the JDWCT-CVQ-Edges and the JDWCT-CVQ schemes for optimized video compression efficiency and security level of encryption simultaneously.

## Chapter 7

### Edge and Phase Sensing for Secure Video

#### Compression

In chapter 6, we suggested an approach for video compression and encryption based on edges sensing. In this chapter, we shall improve the compression efficiency of the edge-sensing based method while maintaining the quality. The use of thresholds determined from RFs blocks for encoding the significant coefficients of mismatched blocks of n-RFs is the most likely cause of loss of quality in some videos. The new, and final, scheme attempts to find an efficient alternative mechanism. For this we investigate the use of a new concept, analogous to the concept of phase modulation mostly used for transmission of digital signal/data over wireless communication networks. We use the term “phase sensing” to refer to this concept to be described in the next section and used for optimized encoding of mismatched blocks in n-RFs. To our knowledge, phase sensing has not been used for compression. Its use is motivated by the advances in the recently emerged compressive sensing paradigm.

Both compression and encryption will also be based on Joint DWT, DCT, CVQ, Edges and on phase sensing (JDWCT-CVQ-Edges-phase sensing). The new scheme will include a new encryption scheme based on the A5 cipher for enhanced security of the corresponding simultaneous video compression and encryption scheme. The encryption algorithm combines the A5 cipher with chaotic logistic map. In section 7.1, we shall describe the phase sensing procedure for encoding and decoding square blocks, and in section 7.2, we shall describe the A5 cipher to be combined with the chaotic map for encryption. In section 7.3, we shall present our approach for video compression and encryption. The performance of the JDWCT-CVQ-Edges-phase sensing scheme will be tested, in section 7.4, on video streams with frames of size (512x512), and the bit rate of phase sensing will be compared with other encoding methods. We shall establish that the developed scheme scales up very well to larger frames. And finally, in section 7.5, we shall demonstrate that the AES cryptosystem in selective encryption of video is not suitable in time constrain.

## 7.1 Phase Sensing

In digital communication, Phase Modulation (PM) is commonly used for data transmission. In PM, the phase of a carrier signal (the carrier signal is continuous very high frequency signal usually used to carry data for transmission over wireless communication system) is changed according to the binary input data. Each digital form (symbol) is assigned a particular phase on the carrier signal. This technique is known as Phase Shift Keying (PSK). The receiver will detect the phase of carrier signal and recover the symbol based on phase change in the carrier signal (Schulze and Luders 2005). This provides a possible way of 1-1 mapping of a sparse block of data so that data locations are incorporated into the data values which can be recovered easily i.e. discovering the sparsity pattern. We use the phase sensing to increase the video compression efficiency of method produced in chapter 6 with retain quality, the phase modulation methods exploited in to map the significant coefficients (edges) extracted from high frequency sub-bands.

Phase-sensing is a method that is widely used in signal processing for detecting very small signals in the presence of remarkable noise level and it has not been used in compression. It begins by constructing the Sensing Matrix (SM) the size of which is application dependent. Based on our intended application into video application for mapping the sparse blocks in non RF's, our SM would be of size 16x16 generated by the sinusoidal wave function:

$$y = A \sin(xt + \theta) \quad 7.1$$

Where  $A$  is the amplitude, and  $x = 2\pi f$  is the radial frequency, and  $\theta$  is the phase angle. Here we shall set  $A = 1$ .

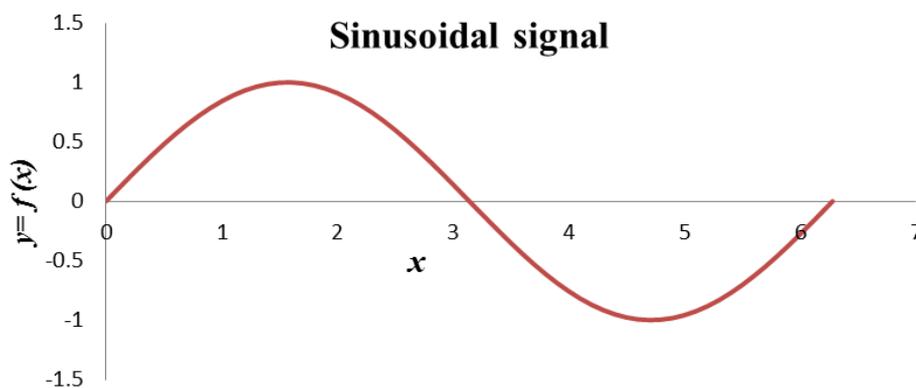


Figure 7. 1 The sinusoidal signal

We will take a quarter of wavelength to avoid symmetrical results (i.e. inforce 1-1 mapping) as shown in Figure 7.1 and calculate the entries of SM using the following equation:

$$y_{(n)} = \sin(a_{(n)} \cdot b) \quad 7.2$$

and  $0 < y_n \leq 1$   $n = 1, 2, 3$ , to the length of the vector, and  $b=\pi/2$

$$a_{(n)} = Z \cdot n \quad 7.3$$

$Z$  is a real number. Let  $\max y_{(n)} = 0.98$ , and the length vector of matrix (16x16) equal to 256. Now substituting equation 7.3 in 7.2 yields:

$$y_n = \sin(Z * 256 * b) = 0.98$$

$$\therefore Z = \frac{\sin^{-1}(0.98)}{256 * b} \approx 0.0031$$

So that equation 7.2 becomes

$$y_{(n)} = \sin(0.0031 * n * b) \quad 7.4$$

Now by iterating the equation 7.4 for  $n = 1, 2, 3, \dots 256$ , we generated the vector

$y = [y_{(1)}, y_{(2)}, \dots y_{(256)}]$  and convert  $y$  to the SM matrix of size (16x16).

The SM matrix is used for phase sensing to extract the significant coefficients in an input mismatched block  $B$ , of an n-RF sub-band, by the 3-step procedure (see Figure 7.2):

1. Calculate the matrix  $B' = SM+B$ .
2. Discard every entry of  $B'$  that is  $< 1$ , and convert the rest to integer values simply by multiplying by 100 and taking the integer part.
3. The output is an integer vector  $x = [x_{(1)}, x_{(2)}, \dots x_{(k)}]$  consisting of only the nonzero values read column by column.

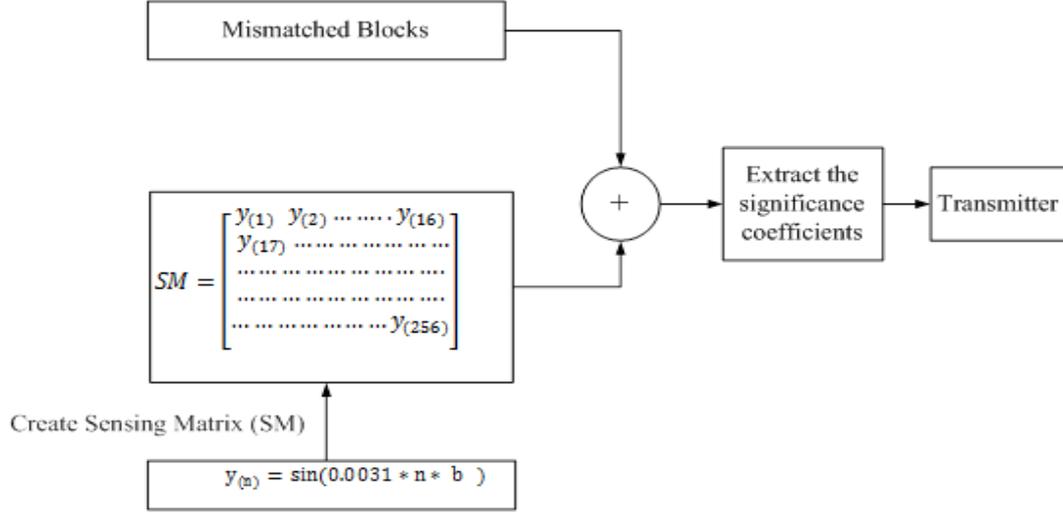


Figure 7. 2 illustrates the mismatched block compression based on phase sensing

Note that, the entries of the vector  $y$  are monotonically increasing in the range 0..1. This property is essential for the decoding procedure (recovering the significant original entries of the encoded block  $B$ ). The decoding of mismatched blocks is given as follows:

Firstly, convert the integer number of compressed vector to floating number by dividing by 100 and separate the integer part from the fractional part. Secondly, we apply equation 7.6 to determine the phase shift of each fractional part. Equation 7.6 is driven as explained below.

$$x_{(n)} = \sin(a_{(n)} * \frac{\pi}{2})$$

$x_{(n)}$  is the fractional part of  $n$  element of the vector.

$$a_{(n)} = 0.0031 * n \tag{7.5}$$

$$\therefore a_{(n)} * \frac{\pi}{2} = \sin^{-1}(x_{(n)})$$

$$a_{(n)} = \frac{\sin^{-1}(x_{(n)})}{\frac{\pi}{2}} \tag{7.6}$$

From equations 7.5 and 7.6 we can determine the value of  $n$  as shown below

$$n = \frac{\sin^{-1}(x_{(n)})}{\frac{\pi}{2} * 0.0031} \tag{7.7}$$

Thus, the phase shift and  $n$  will map the integer part to recover the significant entries of the mismatched blocks. Finally, fill the other entries by 0.

### Example

In this example, we will illustrate how a phase sensing is used to extract significant coefficients from a mismatched block.

Let  $B$  be the mismatched block which has been produced by applying the compression JDWCT-CVQ-Edge method described in chapter 6, section 6.3.1. As a result of edges detection of this method, the  $B$  content can be defined as sparse. For illustration, we will consider the size of  $B$  as (8x8) as shown below.

$$B = \begin{bmatrix} 9 & 5 & 0 & 0 & 150 & 0 & 0 & 13 \\ 4 & 4 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 75 & 0 & 1 & 0 & 11 & 0 & 22 \\ 40 & 0 & 0 & 0 & 8 & 0 & 12 & 0 \\ 0 & 0 & 200 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 70 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 55 & 0 & 0 & 0 & 9 & 0 & 0 & 10 \end{bmatrix}$$

SM is created by iterating this equation 7.4 (with  $Z=0.0136$ ) for  $n=1, 2, 3 \dots 64$ .

$$SM = \begin{bmatrix} 0.02 & 0.03 & 0.05 & 0.06 & 0.08 & 0.09 & 0.11 & 0.12 \\ 0.14 & 0.15 & 0.17 & 0.18 & 0.2 & 0.21 & 0.23 & 0.24 \\ 0.26 & 0.27 & 0.29 & 0.3 & 0.31 & 0.33 & 0.34 & 0.36 \\ 0.37 & 0.39 & 0.4 & 0.41 & 0.43 & 0.44 & 0.45 & 0.47 \\ 0.48 & 0.49 & 0.51 & 0.52 & 0.53 & 0.55 & 0.56 & 0.57 \\ 0.58 & 0.6 & 0.61 & 0.62 & 0.63 & 0.64 & 0.66 & 0.67 \\ 0.68 & 0.69 & 0.7 & 0.71 & 0.72 & 0.73 & 0.74 & 0.75 \\ 0.76 & 0.77 & 0.78 & 0.79 & 0.8 & 0.81 & 0.82 & 0.83 \end{bmatrix}$$

Then,  $B$  is added to SM. Next the coefficients greater than 1 are extracted and all these coefficients are shifted two digits to the right, the outcome of this process is represented by the vector:

$$y = [902 \ 414 \ 4037 \ 5576 \ 503 \ 415 \ 7527 \ 469 \ 20051 \ 618 \ 130 \ 15008 \ 843 \ 980 \ 1133 \ 7064 \ 1245 \ 1312 \ 2236 \ 1083]$$

The bit rate of original block  $B$  is equal to  $8 \times 64 = 512$  bit while for  $y$  it is equal  $16 \times 20 = 320$  bit. Thus, the phase sensing achieves compression ratio = 0.625 for block  $B$  while retaining the quality.

In decoding, we shall generate zeroes matrix ( $Z_r$ ) size (8x8). Then, the  $y$  values are shifted two digits to the left and integer part is separated, which represents the edges,

from the fractional parts which represent the phase shift. From equation 7.7 (with  $Z=0.0136$ )  $n$  is calculated for each fractional part as shown in Table 7.1. The  $n$  will map the edges into  $Z_r$  to recover the mismatched block  $B$ . According to Table 7.1, the element 1 of  $y = 902$  has the significant coefficient 9, which will be in the location  $n=1$  of the  $Z_r$ , and the element 2 of  $y = 414$  has significant coefficient 4 in location 9 of the  $Z_r$ , etc.

$y_n$	Edges	phase shift	n
902	9	0.02	1
414	4	0.14	9
4037	40	0.37	25
5576	55	0.76	57
503	5	0.03	2
415	4	0.15	10
7527	75	0.27	18
469	4	0.69	50
20051	200	0.51	35
618	6	0.18	12
130	1	0.3	20
15008	150	0.08	5
843	8	0.43	29
980	9	0.8	61
1133	11	0.33	22
7064	70	0.64	46
1245	12	0.45	31
1312	13	0.12	8
2236	22	0.36	24
1083	10	0.83	64

Table 7.1 the edges and mapping them in a zeroes matrix

Furthermore, the effect of the SM will not be different if we to select  $\max y_{(n)} = 0.99$  instead of  $\max y_{(n)} = 0.98$ . in fact equation 7.4 becomes

$$y_{(n)} = \sin(0.014224 * n * b )$$

$$SM = \begin{bmatrix} 0.02 & 0.04 & 0.07 & 0.09 & 0.11 & 0.13 & 0.16 & 0.18 \\ 0.2 & 0.22 & 0.24 & 0.26 & 0.29 & 0.31 & 0.33 & 0.35 \\ 0.37 & 0.39 & 0.41 & 0.43 & 0.45 & 0.47 & 0.49 & 0.51 \\ 0.53 & 0.55 & 0.57 & 0.59 & 0.6 & 0.62 & 0.64 & 0.66 \\ 0.67 & 0.69 & 0.7 & 0.72 & 0.74 & 0.75 & 0.77 & 0.78 \\ 0.79 & 0.81 & 0.82 & 0.83 & 0.84 & 0.86 & 0.87 & 0.88 \\ 0.89 & 0.9 & 0.91 & 0.92 & 0.93 & 0.93 & 0.94 & 0.95 \\ 0.96 & 0.96 & 0.97 & 0.97 & 0.98 & 0.98 & 0.99 & 0.99 \end{bmatrix}$$

This SM is completely different from the SM created in the previous example. Therefore, the max  $y_{(n)}$  value can be employed in video encryption method as secret key and it will be sent to the decoder, the decoder will use the max  $y_{(n)}$  to generate the SM and then map the significance coefficients.

In this section, we propose an approach to optimize the video compression and encryption of chapter 6 method while maintaining quality. In our proposal for this improvement, the compression is achieved through JDWCT-CVQ-Edges-phase sensing. The compression is applied on high frequency sub-bands of level 1 and 2. Then the encryption is applied on the low frequency sub-band of level 3 (LL3). In order to reduce the computational cost time, the encryption can be preform simultaneously with compression as shown in Figure 7.3.

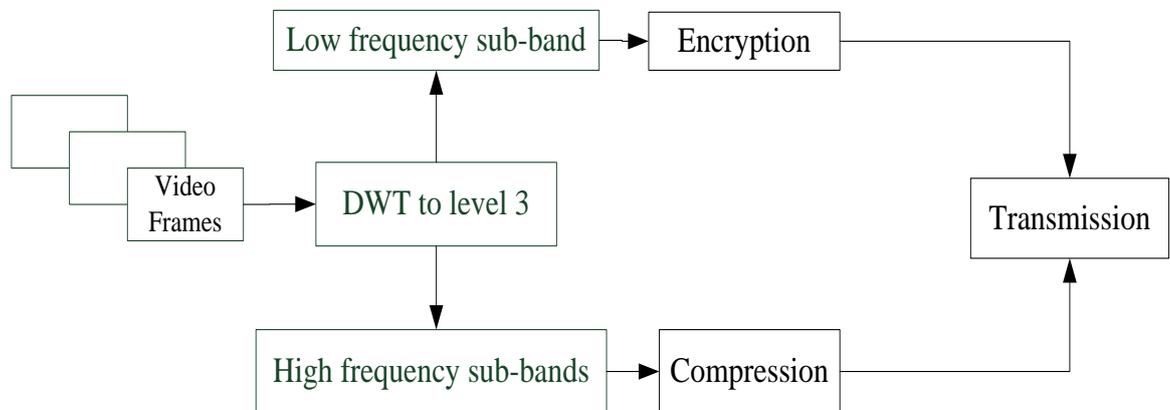


Figure 7.3 simultaneous video compression and encryption

## 7.2 The A5 cipher description

The main weakness of LFSR ciphers is their linearity, which leads to a relatively easy cryptanalysis. A common solution to this linearity weakness of LFSR is to destroy the linearity properties of LFSR. A5 is a stream cipher used in GSM to provide secure communication during conversation via mobile phones. A5 uses three LFSRs with

different taps and length as shown in Figure 7.4. LFSR1 has length 19 bits with four taps: bit 13, 16, 17 and 18, while LFSR2 and LFSR3 have length 22, 23 bits with taps 21, 20, 22 and 21, 20, 7 respectively. The middle bit from each LFSR represents the clocking bit, as shown in red colour in Figure 7.4. When the LFSRs are clocked, their taps will be XORed to produce next Least Significant Bit (LSB).

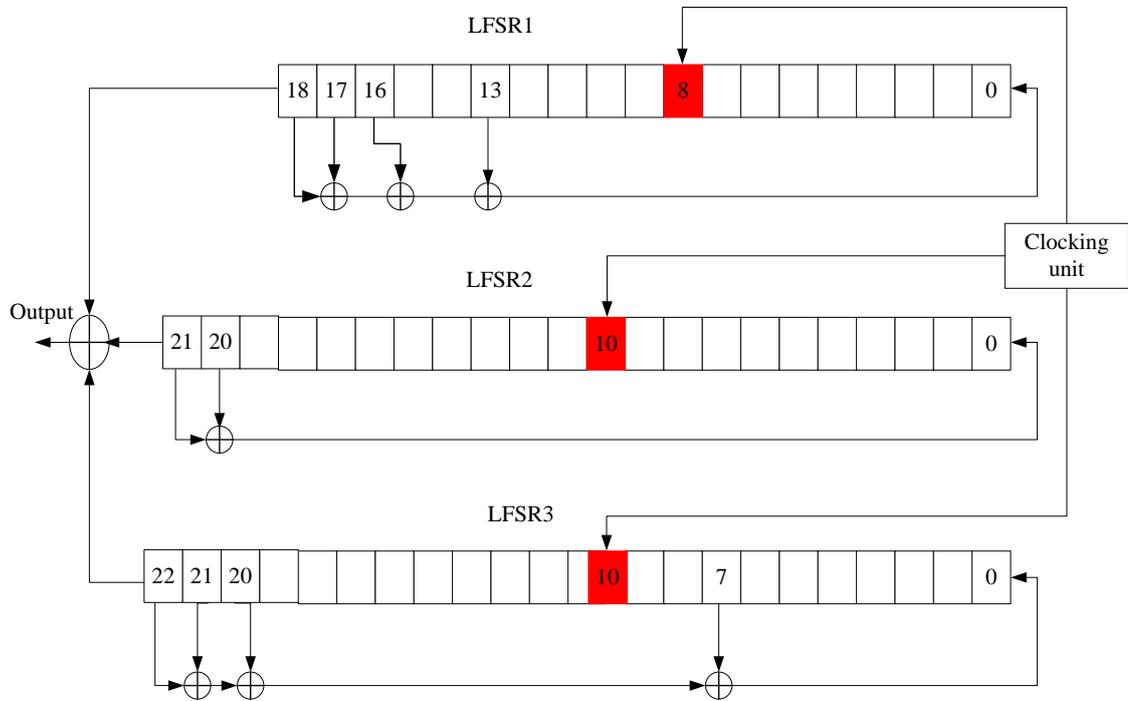


Figure 7.4 illustrates the A5 structure

The A5 cipher solves the linearity weakness by applying irregular clocking of the LFSR to the cryptosystem. The clocking rule of A5 is based on majority function in which the register will be clocked if its middle bit is equal to the majority bit. The majority bit will be 1 if two or more middle bits are 1, else it will be zero, Table 7. 2 summarises the clocking rule of A5. For example, if the middle bit of LFSR1, LFSR2 and LFSR3 was 1, 0 and 1 respectively, the majority function will be 1. Thus, only LFSR1 and LFSR3 will be clocked.

Middle Bit (MB)			Majority	LFSR clocking		
MB of LFSR1(8)	MB of LFSR2(10)	MB of LFSR3(10)		LFSR1	LFSR2	LFSR3
0	0	0	0	clock	clock	clock
0	0	1	0	clock	clock	no clock
0	1	0	0	clock	no clock	clock
0	1	1	1	no clock	clock	clock
1	0	0	0	no clock	clock	clock
1	0	1	1	clock	no clock	clock
1	1	0	1	clock	clock	no clock
1	1	1	1	clock	clock	clock

*Table 7. 2 the clocking rule of A5*

Initially, the LFSRs of A5 are initialised to zeroes. Then, a 64 bits secret key (session key) is seeded to three LFSRs without applying irregular clock. Next, the LFSRs are clocked 22 times (public key), the clocking rule is also ignored in this stage. Subsequently, the LFSRs are clocked 100 times based on majority clocking; at this stage the A5 output is discarded. Finally, the A5 cipher is clocked by using majority clocking to produce 114 bits sequences which are XOR with data voice to produce ciphered data. For more details see (Barkan, Biham and Keller 2003), and (Dubrova, Teslenko and Tenhunen 2008),

### **7.3 JDWCT-CVQ-Edges-phase sensing scheme**

The compression of the JDWCT-CVQ-Edge scheme, developed in chapter 6, utilized the statistical properties of high frequency sub-bands to determine the significant coefficients (edges) and converted all non-significant coefficients to zeroes and then we partitioned the high frequency sub-bands into blocks. Therefore, the blocks contents are sparse. For inter-frame (n-RF) encoding, the block based similarity was used to extract the mismatched blocks which are then sent to the transmitter.

To further improve the compression efficiency, the mismatched blocks need to be encoded into a short bit-string by a more efficient method than so far used which will remember the position of the significant feature. Since the insignificant coefficients are negligible these blocks are sparse and hence susceptible to compressive sensing theory. The compressed sensing (CS) theory, which asserts that certain signals or image blocks

can be recovered from fewer samples or measurements than required by the traditional Shannon-Nyquist sampling theory if this signal or image has sparse representation. Currently, the compressed sensing is still in its early days and focus of research is on reducing the number of measurements. So far the very few CS based image compression schemes tend to require heavy computations. In addition, the quality of recovered image will be reduced because the compressive sensing is lossy compression (Bigot, Boyer and Weiss 2013) (Gan 2007). These observations were sufficient for us not to pursue the compressed sensing theory at this time but perhaps come back to it in the future. Instead we decided to investigate the use the phase sensing procedure, to encode the mismatched sparse high frequency sub-band blocks.

### 7.3.1 The video compression scheme

Wavelet transform to level 3 are applied to the sequence of video frames. The high frequency sub-bands of level 1 and 2 for RF and n-RF will be processed in the same JDWCT-CVQ-Edge method work the threshold is implemented for edges detection. For each mismatched block the nonzero coefficients is significantly less than the actual size of the blocks, i.e. these blocks are sparse, see Figure 7.5. We apply the phase sensing procedure, as described earlier in section 7.1, to encode each mismatched block of n-RF. This procedure outputs a vector of integers, one for each of the significant coefficients; it has a shorter size than that of the block. The way the SM is designed, guarantees that the significant coefficients can be recovered and the decoding procedure remembers the exact positions of these coefficients. Figure 7.5 summarize the main compression steps described above

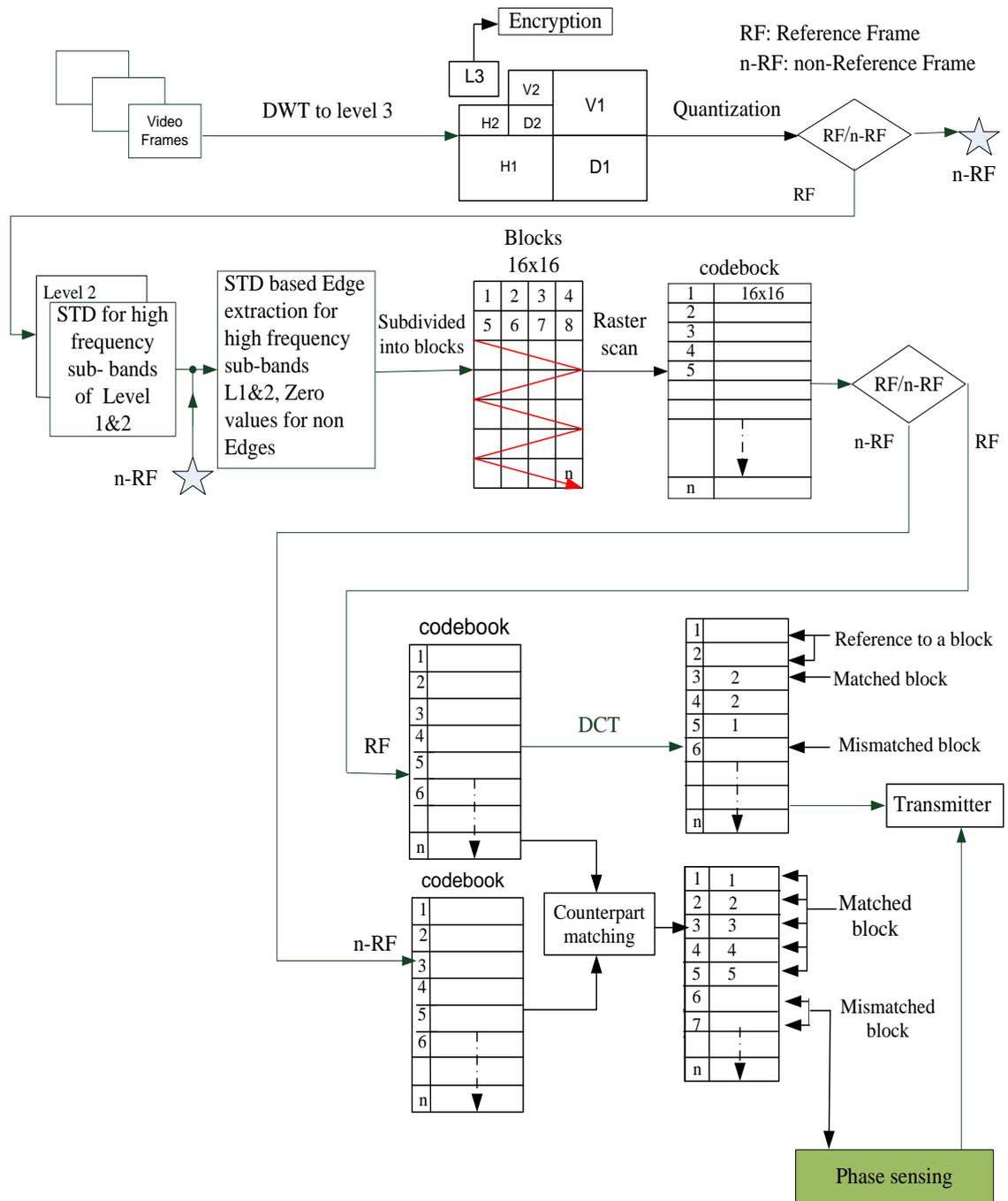


Figure 7.5 represents the process block diagram of the video compression

### 7.3.2 The encryption proposed scheme

Our encryption proposal uses the crucial parts of the low frequency sub-band of level 3 (LL3), but not on the entire level 3, this means that encryption is applied on only 1/64 of the original size. These crucial parts of the low frequency sub-band of level 3 are identified at from the high frequency sub-bands of level 3 using the selected thresholds. This results in sparse blocks to be mapped onto the edge-sensing matrix which were identified. While the decryption will use the same edge-phase sensing matrix to retrieve

all the encrypted data, this will cut the computational time for retrieving the edges dramatically.

As described in the section 7.2, the main problem of the LFSR is the presence of the linear part in the random number generator algorithms. A5 solves the linear weakness of LFSR by using the majority function. However, the majority function is defenceless against correlation attack (Chen and Gong 2012). In our proposed scheme, the linearity weakness in A5 is reduced significantly by improving the clocking rule using the chaotic logistic map rather than the majority function.

The encryption works using the following steps (see the block diagram in Figure 7.6):

- 1- Chose the initial condition  $x_0$  and the control parameter  $r$ . Then, iterate the equation 6.1 to 64 times and truncate the output numbers of iterations to the integer numbers and convert them to binary number (mod 2). These bits will seed the A5 registers.
- 2- Calculate the STD for high frequency sub-bands of level 3 (V3, H3, D3).
- 3- Measure the threshold (THRE=STD $\times$   $\mathbb{R}$ ) where  $\mathbb{R}$  is real number. As illustrated in chapter 4, section 4.2, this THRE will be used to reveal the Significance Coefficients (SC) of high frequency sub-bands.
- 4- Generate the SM by applying equation 7.4 for  $n=1, 2, \dots$  to the size of high frequency sub-band, and then adding MS with high frequency sub-bands.
- 5- Extract the coefficients greater than THRE; these coefficients represent SC of high frequency sub-bands.
- 6- Separate the integer part from fractional part for each SC.
- 7- Apply equation 7.7 to each fractional part of SC, the output  $n$  will be mapped to the significance coefficients of low frequency sub-band of level 3(SC-LL3).
- 8- Iterate the chaotic logistic map to ( $S_k=100+\text{size of SC-LL3}$ ) times and convert the output numbers to integer number  $m$  (mod 3). The  $m$  will be used as clocking rules for A5 and to select one bit  $x_i(m)$  from each SC-LL3. The LFSRs of A5 will be clocked to the size of SC-LL3( $Sz$ )
- 9- XOR the  $x_i(m)$  with output bits stream  $kc(i)$  from modified A5, where  $i = 1, 2, \dots Sz$ .
- 10- The THRE, which is employed assigning the SC-LL3, is ciphered by chaotic logistic map (converted to the binary and XOR with binary representation of chaotic map) and sent to the transmitter. The max  $y_{(n)}$  (which is used in SM generation) is ciphered also by chaotic map and sent to the transmitter.

11- Finally, the encrypted L3 will be separated into blocks of 16x16 and sent to the transmitter. Figure 7.6 summarises the main encryption steps mentioned above.

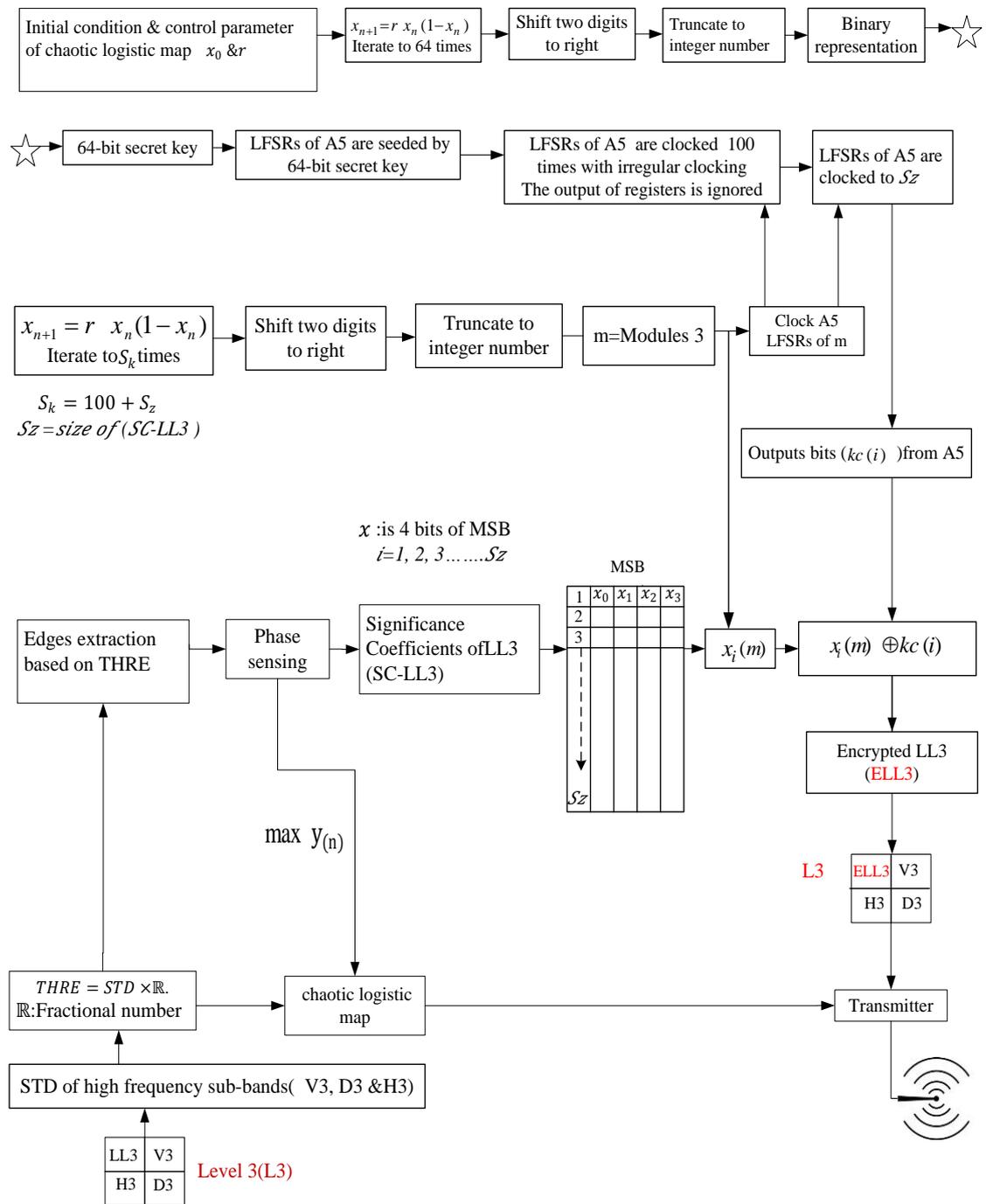


Figure 7.6 shows the encryption scheme

## 7.4 Experimental results

Having developed the final refinement in our investigated scheme, the experimental work will go beyond testing its performance on the same type of videos used in the experiments of the previous chapters. We shall include experiment on gray scale videos of larger size frames as well as RGB colour videos. But we start with the experiments on the same video size considered previously. All the experiments have been performed in MATLAB V 7.10 (R2013a) in the same machine used in chapter 3.

We first tested the performance of the JDWCT-CVQ-Edge-Phase sensing scheme in terms of Compression Ratio (CR), the quality, execution time and the security level. The performance will be compared with the scheme introduced in chapter 6 (CH6M) and other algorithms. Here we expanded the set of videos to include 32 different grayscale videos, each containing 100 frames of size (256x320) pixels. Sample frames from test videos are shown Figure 7.7.



*Figure 7.7 represent sample frames from test videos*

### 7.4.1 Analysis of compression result

Figure 7.8 shows the achieved Compression Ratio (CR) for each tested video. These results in indicate that the CR is somewhat dependent on video complexity. If image objects mostly remain stationary within the video frames, then very few mismatched blocks will be produced during the video encoding processing. On the other hand, if there is a lot of motion or if the camera itself is moving then many mismatched blocks will be shaped through the video encoding. Therefore, Figure 7.8 shows that the CR varies from one video to another.

## Compression Ratio (CR)

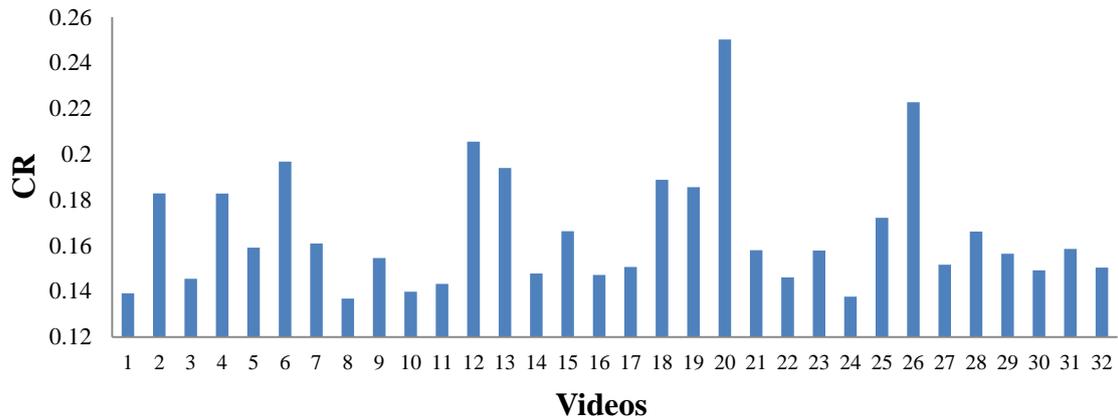


Figure 7. 8 shows the achieved CR for tested video

We will now demonstrate the effect of JDWCT-CVQ-Edges-phase sensing on the video decompression quality represented by PSNR. All tests were performed using the same compression threshold THR which has been described in chapter 4, section 4.2. Figure 7.9 shows the Mean and Standard Deviation (STD) of PSNR calculated for each frame of each video.

## Mean and STD of PSNR

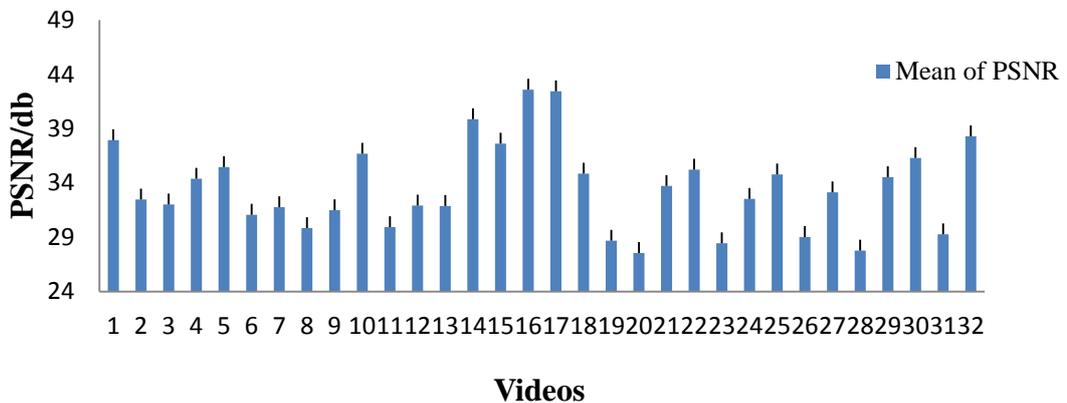


Figure 7. 9 the mean and STD of PSNR for decompression and decryption videos

It can be seen from Figure 7.8 and Figure 7.9 that the relation between quality and CR depends on the nature of the videos. For instance, videos 16 and 17 contain objects moving within frames. Therefore, the CR is small and PSNR is high. In contrast, the objects and camera are moving in videos 4 and 13, so the CR and PSNR are relatively high.

We now attempt to establish that the JDWCT-CVQ-Edges-phase sensing scheme is indeed a good refinement of the previous ones by comparing its performance with that of the previous JDWCT-CVQ-Edge scheme, using the same test videos and thresholds THR have been used here again. Figure 7.10 shows the bit rate of mismatched blocks during video compressing through JDWCT-CVQ-Edge and JDWCT-CVQ-Edges-phase sensing. The results show that the JDWCT-CVQ-Edges-phase sensing produced lower bit rate and better CR as compared to JDWCT-CVQ-Edge as shown in Figure 7.10 and Figure 7.11. The lower bit-rate is the way the phase sensing matrices compactly represent their sparse blocks compared to the way JDWCT-CVQ-Edge do.

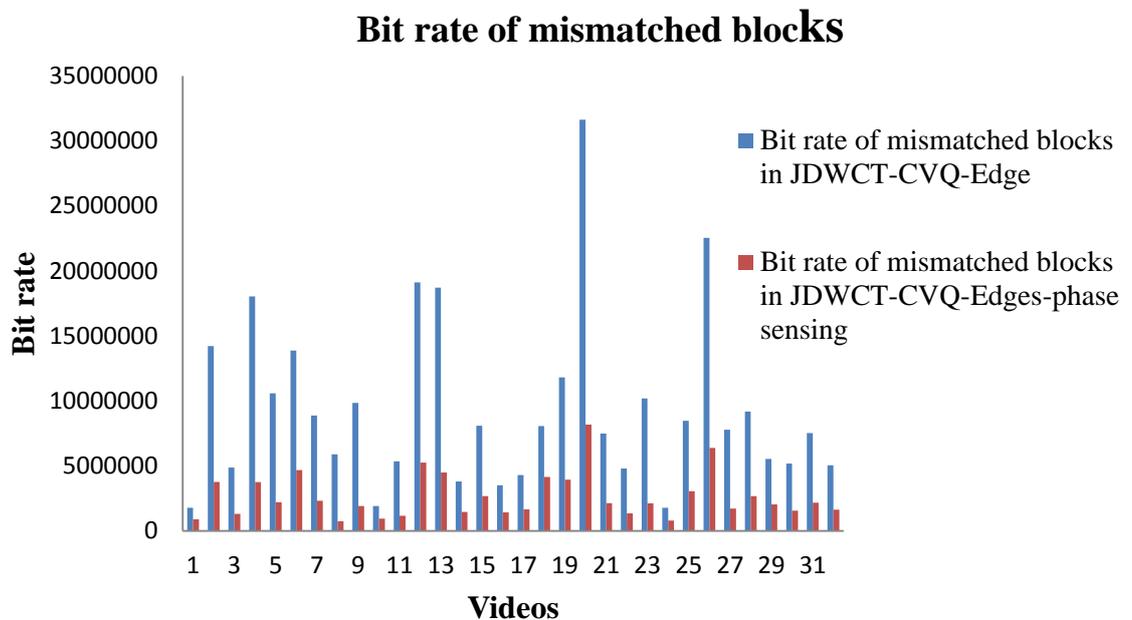


Figure 7.10 shows the bit rate comparison

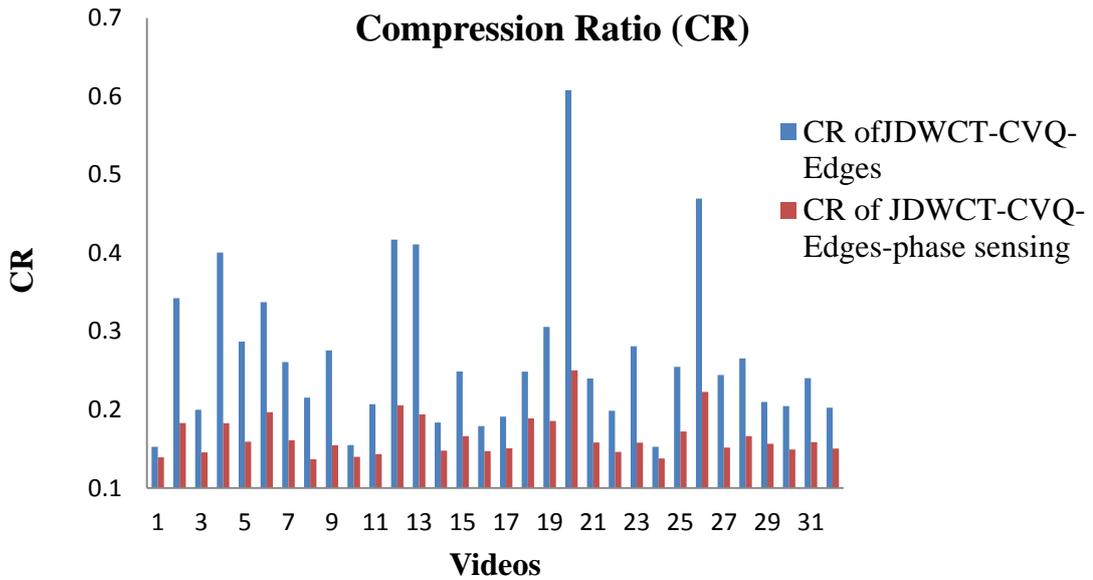


Figure 7.11 CR comparison

Figure 7.12 shows the execution time for compression-encryption encoding and decoding frames for each video (excluding Input Output (IO) operations). The results show that the execution time is proportional with CR, when the CR is relatively small; this means that there were a few mismatched blocks. Hence, the phase sensing and encryption processing time will be reduced. For example, the processing time for video-1, including compression and encryption, was 0.75 sec and it has CR= 0.13, while for video-20 it was 1.63.sec with CR=0.25. Usually, the processing time difference reflects the complexity of each video.

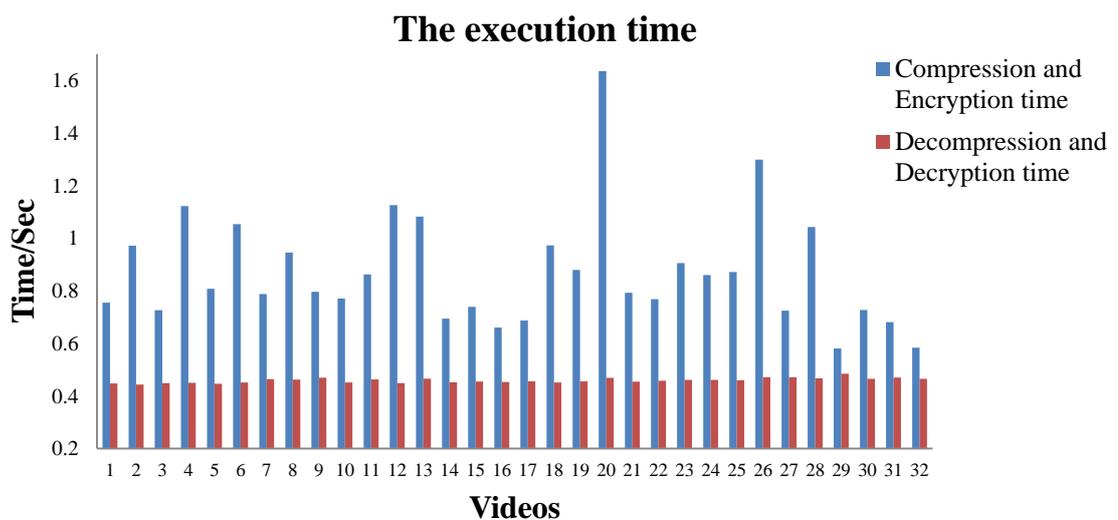


Figure 7.12 shows execution time

The results of JDWCT-CVQ-Edges-phase sensing are also compared, below, with standalone DCT and DWT algorithms; using the same videos above. For a DCT-based

scheme we are using the JPEG and we use the SPIHT as the DWT-based scheme. We used the Central Limit Theorem (CLT) to calculate the mean and STD for all test videos for each algorithm. The results show that JDWCT-CVQ-Edges-phase sensing tends to outperform the other two algorithms with better quality for CR of 97%, where the average PSNR in JDWCT-CVQ-Edges-phase sensing was 31.52 dB measured for 100 frames to each video, as shown in Table 7.3, Figure 7.13 and Figure 7.14. This is due to fact that JDWCT-CVQ-Edges-phase sensing preserves the significant coefficients (image feature) and phase sensing improves the compression efficiency while retain the quality level. .

Method	JDWCT-CVQ-Edges-phase sensing	SPIHT algorithm	DCT algorithm
Mean of PSNR (dB)	31.5208	23.6501	26.9872
STD of PSNR	0.7335	0.5729	0.587

Table 7. 3 shows the mean and STD of PSNR and CR

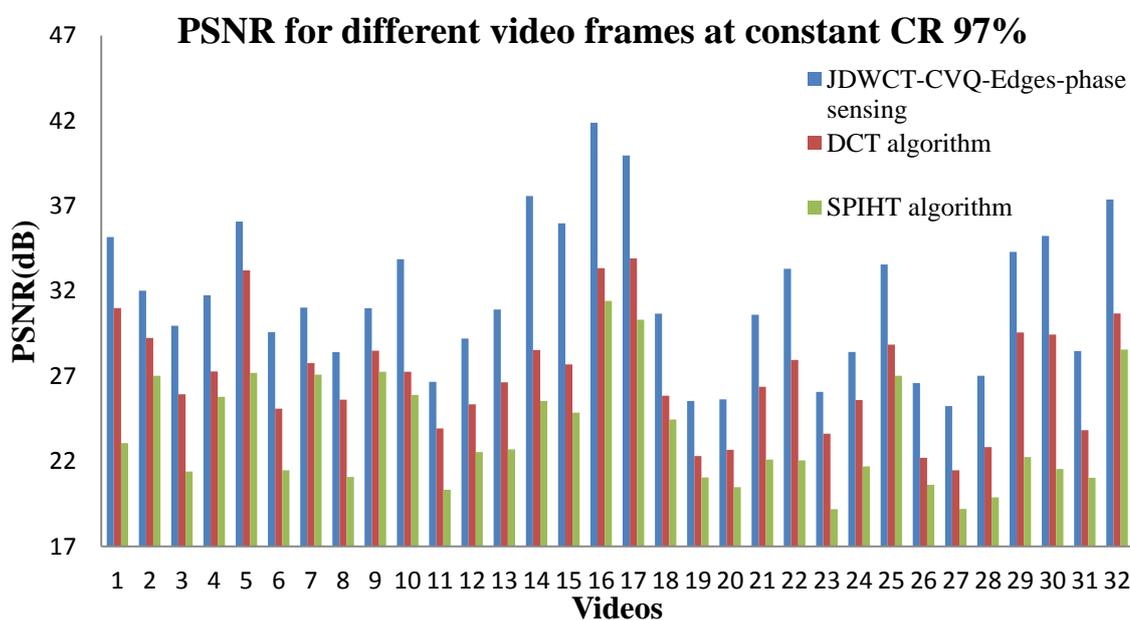


Figure 7.13 PSNR for different video frames at constant CR of 97%



**A B C** *Figure 7.14 sample of recovered frame by three different methods (A) JDWCT-CVQ-Edges-phase sensing (B) DCT based(C) SPIHT*

Moreover, phase sensing procedure has been compared with Run Length Encoding (RLE) followed by Huffman encoding. The RLE followed by Huffman encoding is commonly used in the final step of MPEG-x and H.26x (x=1, 2, 3...etc.) as lossless compression. The comparison is done on mismatched blocks. The result of comparison is shown in Figure 7.15. As can be seen in the Figure 7.15 that the phase sensing achieved CR much better as compared to RLE and Huffman encoding, because the RLE is not efficient with data have not many runs. Therefore, the result shows that the RLE causes an overhead affecting the compression efficiency as shown in Figure 7.15

### Comparison of CR

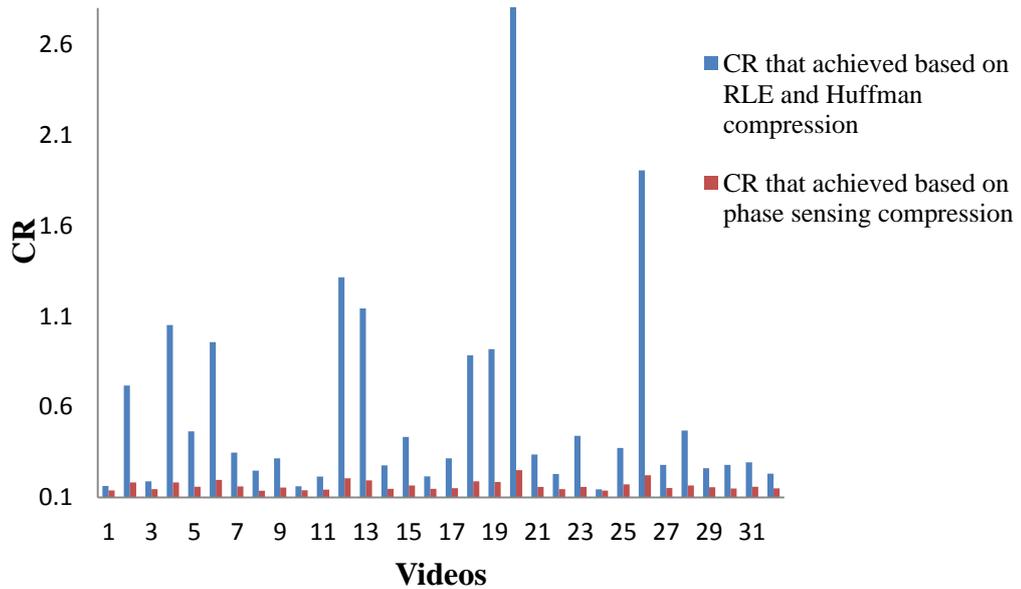


Figure 7.15 Comparison between phase sensing and RLE followed by Huffman encoding

### Performance of the scheme for larger size Videos

Here we conduct experiments to address the following important question:

#### *How scalable is JDWCT-CVQ-Edges-phase sensing scheme?*

We selected a new set of 10 different test videos of frames that have size greater than those tested before. The frame size of these videos is 512x512 pixels in greyscale. Sample frames from these videos are shown in Figure 7.16.



Figure 7.16 represents sample frames size 512x512 pixels from test videos

All tests were performed using the same compression threshold THR which has been described in chapter 6, section 6.3.1. Results of the tests are shown in Figure 7.17. The results show that the proposed method achieved reasonable quality and high CR compared with original size, where the average of PSNR was 35.27dB measured for all videos. Moreover, the results of this compression scheme were compared, below, with the performance of JDWCT-CVQ-Edges-phase sensing when applied on same videos of frames that have same size before i.e.256x320. Table 7.4 shows that the better CR and quality are achieved by large size for all tested videos.

Scalability means that ratio of execution time should be comparable to the ratio of the sizes. Calculating the ratio of the two tested sizes yields a ratio of  $3.2 = 512 \times 512 / 256 \times 320$ . The average execution time of 100 frames for each video in the bigger size is (1.232135 sec) which compares very well with ( $2.90404 = 3.2 \times 0.907512$ ) as shown in Table 7.4. In fact, we can see that our scheme scales up very well.

Mean	Frame size 512x512 pixels	Frame size 256x320 pixels
CR	0.16203	0.179632
Time encoding and encryption/Sec	1.232135	0.907512
Time decoding and decryption /sec	0.64222	0.559365
PSNR/dB	35.27815	33.36175
STD	1.38369	1.103231

*Table 7.4 the comparison results between coding frame size 512x512 and 256x320 pixels*

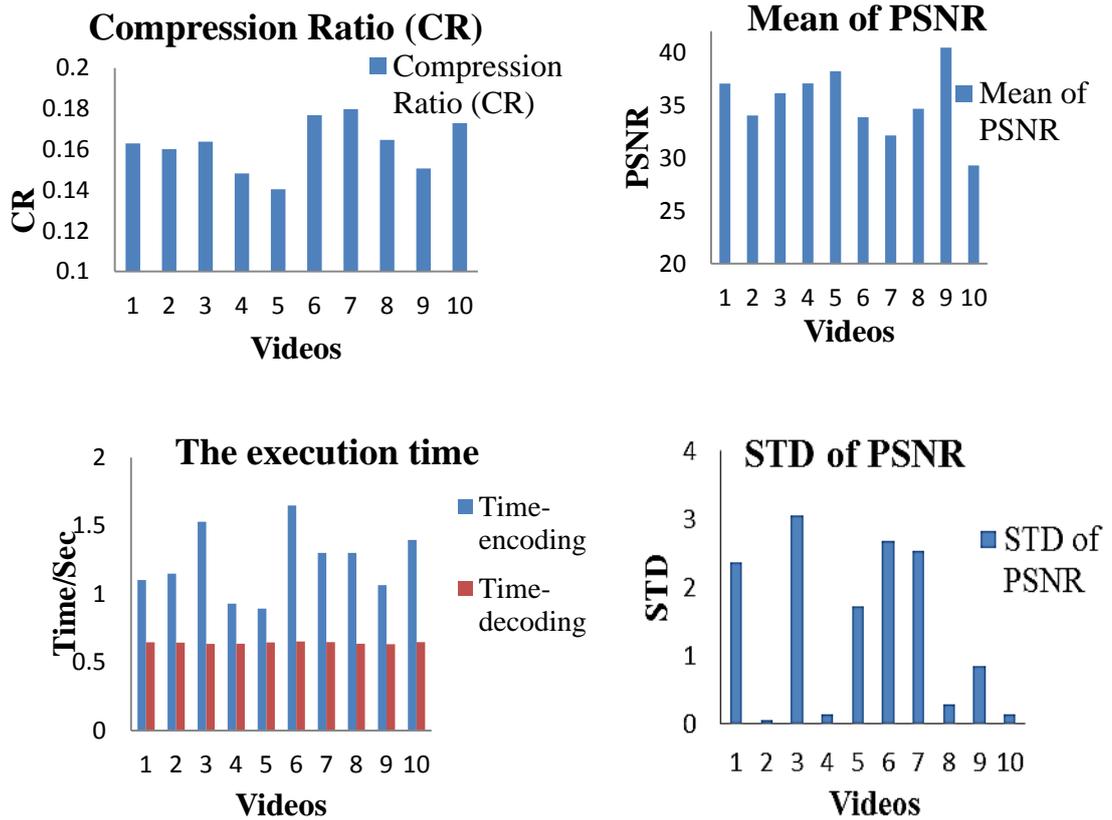


Figure 7.17 illustrates the results of test videos with frame size 512x512 pixels

#### 7.4.1.1 Performance of the scheme for RGB Colour Videos

Generally, the compression of coloured image components of Red, Green and Blue (RGB) are de-correlated by colour transform and each transformed component is compressed individually by image compression algorithm such as JPEG, JPEG2000, etc. Mostly RGB colour space is transformed to  $Y C_b C_r$  colour space. The Y component is the luminance and the two other components ( $C_b C_r$ ) are the chrominance components (Kim and Cho 2014)

Although the coloured videos are not required in the project application, but we are including this test for future development in case it is needed. Therefore, we will apply our latest decoder JDWCT-CVQ-Edges-phase sensing on 10 different colours videos as shown in Figure7.18 to test the effect of using colours on our system performance and based on the achieved frame rate and performance. We have tested 100 frames for each used video, the frame size of all these videos is 512x512x3 pixels. During our experiment, we converted all frames from RGB to  $Y C_b C_r$  components.

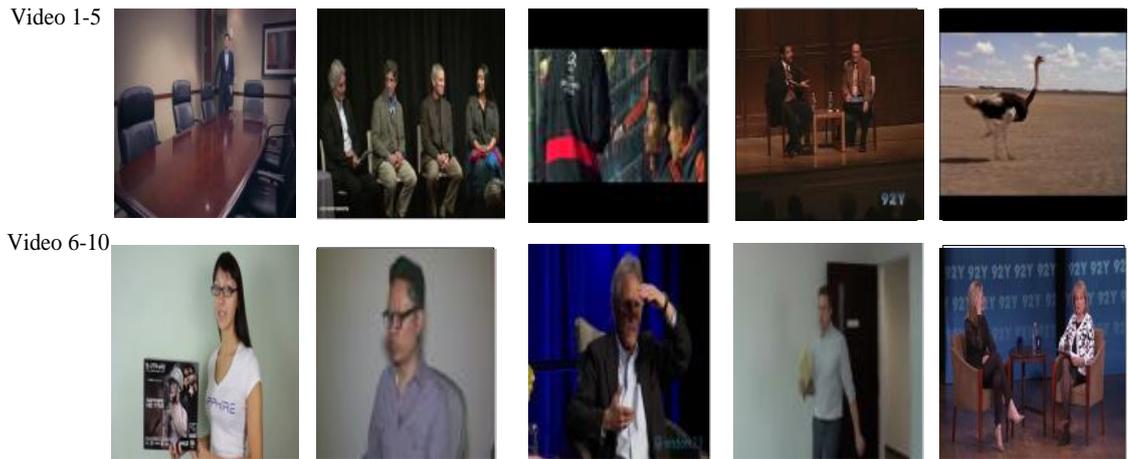


Figure7. 18 represents sample frames size 512x512x3 pixels from test videos

As results of the colour transform, we found that the STD of high frequency sub-bands of the wavelet decomposition became very small and close to zero. Therefore, the significant coefficients (image features) clustered around the zero. Thus, this fact result in increasing the block similarities and decreasing the time consumed for vector quantisation and improved the CR as well as the quality compared with the grayscale videos as shown in Figure 7.19.

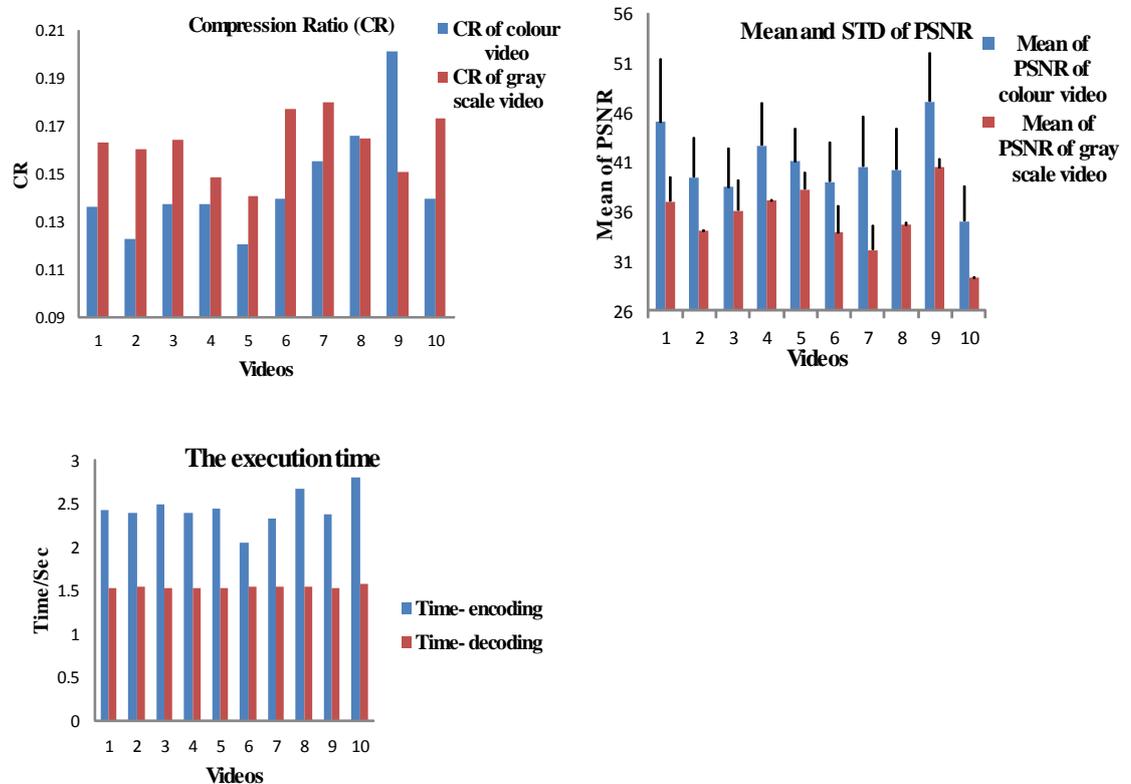


Figure7.19 illustrates the results of test videos with frame size 512x512x3 pixels

The same experiment applied above was applied on another set of coloured videos with frame size of 240x320x3 pixels as shown in Figure 7.20. We considered this size because of its usability in all devices having limited processing speed, capacity, display screen size with limited bandwidth (Zacharovas, Nikolskij and Kuchin 2010). Figure 7.21 shows the Mean, STD of PSNR and coding time calculated for each frame from each video.



Figure 7. 20 represents sample frames size 240x320x3 pixels from test videos

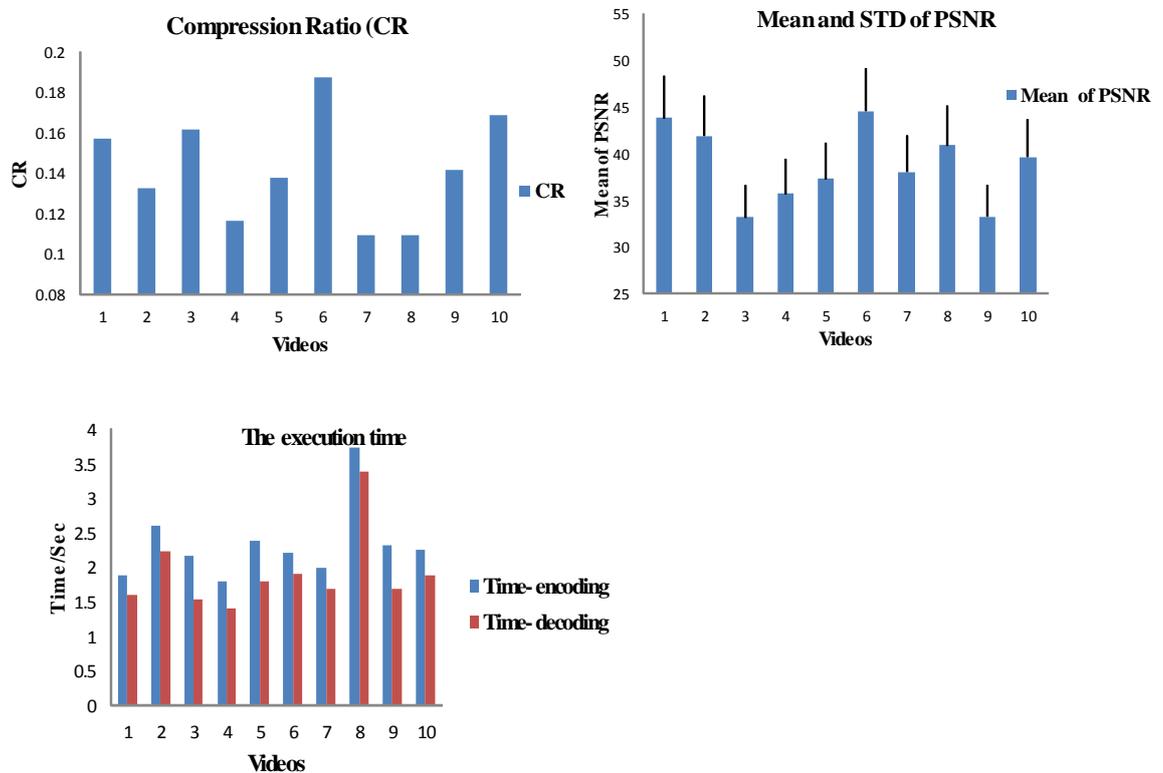


Figure 7. 21 illustrates the results of test videos with frame size 240x320x3 pixels

Based on the above two experiments we noted that the frame rate used in the 512x512x3 frame size is 41 frames per second, and for the videos with 240x320x3 frame size we achieved 43 frame per second. We have to mention that the complexity of the second set of videos is higher than the first set of videos and yet the achievable frame rate is quite accepted for video streaming.

Moreover, another set of experiments have been conducted using two different computers with different speed and performance. The first computer is of a Relatively High Speed Computer (RHSC) with Intel (R) i5 processor, 3.2 GHz, and 16GB of RAM. The other computer is an Average Speed Computer (ASC) with Intel (R) i5 processor, 2.3 GHz, and 4GB of RAM, as shown in tables 7.5 and 7.6. The experimental results show that the average frame rate in grayscale video with frame size (256x320) pixels was 112 f/sec and 77 f/sec for RHSC and AVC respectively. On the other hand, the frame rate is decreased in the colour video scale for frame size (512x512 x3) and (240x320x3) in both computers. We can conclude from the last set of experimental results (tables 7.5 and 7.6), that the processing time for video coding is acceptable in the two used computers and could easily achieves the objective of our application

Video	RHSC		AVC	
	Coding time /sec	Decoding time /sec	Coding time /sec	Decoding time /sec
1	0.755045	0.447827	1.816173	0.801604
2	0.972067	0.443612	1.390215	0.791459
3	0.726813	0.449341	1.059577	0.796856
4	1.122783	0.450273	1.556734	0.79632
5	0.807869	0.44652	1.162966	0.794285
6	1.053916	0.451404	1.464663	0.794288
7	0.788029	0.464097	1.135136	0.798829
8	0.946053	0.462329	1.327097	0.807558
9	0.796388	0.469806	1.171387	0.801574
10	0.771155	0.451623	1.136642	0.791636
11	0.862889	0.463106	1.221628	0.801015
12	1.125925	0.449307	1.574576	0.793076
13	1.08213	0.46624	1.503654	0.791649
14	0.694558	0.452013	1.011301	0.787387
15	0.739404	0.455246	1.072948	0.789281
16	0.660204	0.453337	0.969383	0.788356
17	0.686888	0.455791	0.989321	0.790681
18	0.972905	0.451753	1.432699	0.788573
19	0.879606	0.455812	1.253283	0.790373
20	<b>1.635308</b>	0.468902	2.348013	0.787822
21	0.792828	0.45473	1.131932	0.790698
22	0.768583	0.458019	1.117677	0.810847
23	0.906007	0.461466	1.305536	0.801741
24	0.860312	0.461253	1.247472	0.793469
25	0.871813	0.460227	1.225979	0.791966
26	1.299187	0.471439	1.816337	0.788248
27	0.724506	0.471511	1.06825	0.798087
28	1.043221	0.467192	1.473943	0.791052
29	0.581063	0.48466	0.891178	0.796913
30	0.727277	0.465686	1.056734	0.787279
<b>Average</b>	<b>0.888491</b>	<b>0.458817</b>	<b>1.297748</b>	<b>0.794431</b>
<b>Frame rate/Sec</b>	<b>112.5504</b>	<b>217.9516</b>	<b>77.05658</b>	<b>125.8763</b>

Table 7. 5 shows the execution time by two different computers for coding 100 frames from 30 different videos with size (256x320) pixels in grayscale

Video	Frame size 512x512x3 pixels				Frame size 240x320x3 pixels			
	RHSC		AVC		RHSC		AVC	
	Coding time/sec	Decoding time /sec	Coding time/sec	Decoding time /sec	Coding time/sec	Decoding time /sec	Coding time/sec	Decoding time /sec
1	2.420	1.520	5.233	2.621	1.874	1.591	4.034	2.229
2	2.385	1.536	4.272	2.533	2.599	2.232	4.081	3.061
3	2.497	1.529	4.718	2.545	2.164	1.538	2.659	2.065
4	2.392	1.529	4.077	2.539	1.793	1.392	3.369	1.817
5	2.437	1.525	4.458	2.527	2.389	1.801	2.935	2.338
6	2.056	1.535	3.389	2.574	2.204	1.908	2.790	2.081
7	2.328	1.535	3.896	2.563	1.999	1.680	2.932	2.131
8	2.669	1.539	4.736	2.608	3.727	3.389	4.809	3.881
9	2.369	1.516	4.658	2.537	2.314	1.687	2.941	2.022
10	2.804	1.571	5.539	2.637	2.244	1.883	2.795	2.216
Average	2.436	1.534	4.498	2.568	2.331	1.910	3.334	2.384
Frame rate/Sec	41.057	65.204	22.234	38.934	42.905	52.351	29.990	41.945

Table 7. 6 shows the execution time by two different computers for coding 100 frames of 10 different coloured videos

## 7.4.2 Security analysis

In this section we evaluate the performance of our encryption method using four different kinds of analysis: histogram, key space, correlation and PSNR analysis to assess statistical attack, brute-force attack, frequency attack and the availability of the significant information left in encrypted image respectively.

### 7.4.2.1 Histogram analysis

The histogram of an image is illustrated by how the pixels of an image are distributed by charting the numbers of pixels intensity values. The attacker can use the histogram to realize the secret key or plain pixel. (Mao and Chen 2005).

The experimental results of histogram analysis for video encryption using the proposed algorithm are presented in Figure 7.22. The figure shows that the frames which are selected from video 1, 12, and 25 after encryption and decryption and their corresponding histograms. The histograms of the encrypted frames are fairly uniform and different from the histograms of unencrypted frames. Moreover, we compute the MDMF for these histograms by using equation 3.6 which is presented in chapter 3 as shown in Table 7.7.

video	1	12	25
$HS_i$	32168	42189	51285
$HS_0$	28837	16643	4684
$HS_{255}$	28225	21555	15183
MDMF	60699	61288	61219

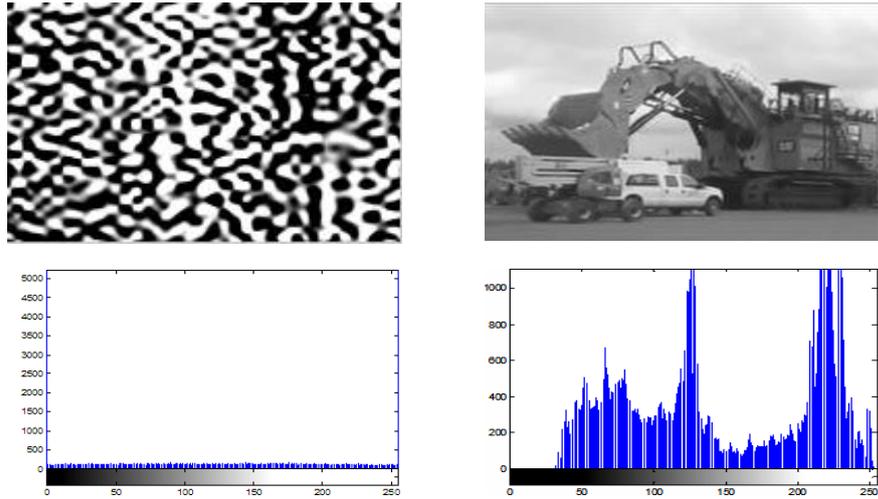
Table 7.7 the MDFM of some encrypted videos

The MDMF shows that the histograms of ciphered frames are differed from original frames. Thus our approach does not provide any information that can be used for any statistical attack, and the decrypted histogram is nearly similar to the original frame histogram.

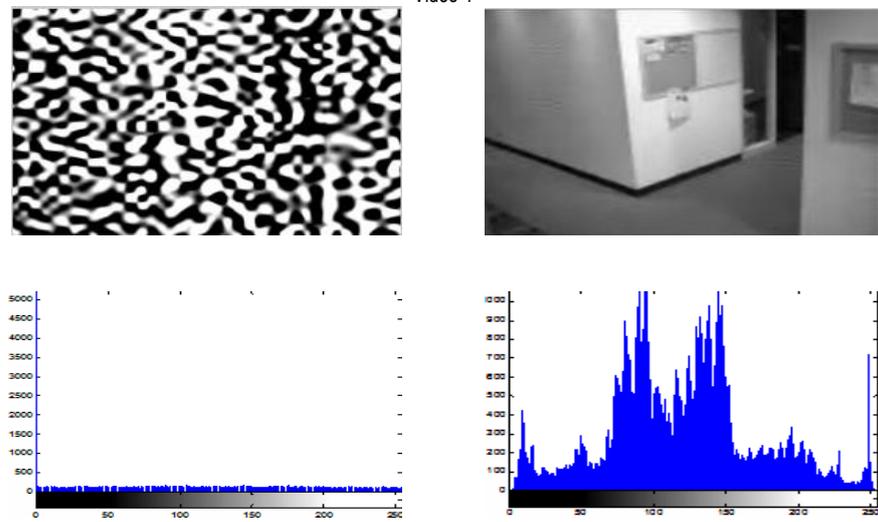
#### 7.4.2.2 Key space analysis

The key space analysis is a set of different keys that can be used to generate the encryption /decryption key. Therefore, the key size should be large enough to make brute-force attack infeasible. The key space should be more than  $10^{30}$  in order to avoid exhaustive search attack (Sathishkumar, Ramachandran and Bagan 2012). Our proposal uses several chaotic logistic maps: firstly to generate the secret key, secondly for clocking the A5 cipher, thirdly to choose a bit from significance low frequency sub bands, fourthly for threshold encryption and finally a map for max  $y_{(n)}$  encryption is used in SM generation. The secret keys used in the proposed encryption are initial condition ( $x_0$ ) and control parameter ( $r$ ) of chaotic logistic map, both are real numbers. The number of possible values of  $x_0$  is  $10^{15}$  as well as  $r$ . As a result, the key space is  $10^{120}$  which is greater than the number of possible permutations of AES, thus preventing a brute force attack on the AES key more efficiently than trying to decrypt the encrypted video of our proposal encryption (Unterweger and Uhl 2012).

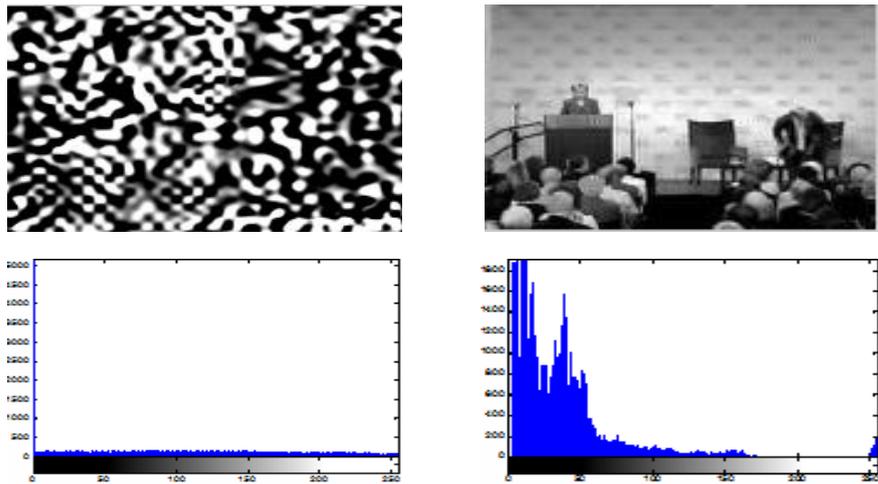
Video-12



Video-1



Video-25



A	B
C	D

Figure 7.22 the histogram of encrypted and decrypted frame (A) encrypted frame (B) decrypted frame (C) the histogram of encrypted frame (D) the histogram of decrypted frame

#### 7.4.2.3 *Correlation coefficients analysis*

The Correlation Coefficient (CC), described in chapter 3, determines the degree of similarity between two variables. If the two variables are completely different, the CC between them will be very low or close to zero. On other hand, the CC becomes large or close to 1 or -1 when there is a relation between variables. To determine the CC between two adjacent pixels, in vertical and horizontal direction, within the same frame, 1000 pairs of two horizontally and vertically adjacent pixels are selected randomly from the original and encrypted frame. The correlation of the selected coefficients is calculated and shown in Figure 7.23. The figure shows that the CC of the original frame is 1 or close to 1. In contrast, the CC of the encrypted frame is close to 0. The result of correlation analysis indicates that the original frames and their corresponding encrypted frames are completely independent of each other.

#### 7.4.2.4 *PSNR analysis*

PSNR is often used as quality measurement between the original image and decompressed image. The PSNR has been defined in chapter 3, section 3.3.1.2, and equation 3.4. From this equation it may be inferred that the Mean Square Error (MSE) is proportionally inverse to PSNR. Thus, the MSE increases with PSNR decreasing and producing more randomness in the recovered image. Generally, when  $PSNR > 30\text{dB}$ , the quality of recovered frames is valued as adequate (Huang and Sakurai 2011). The mean and STD of PSNR for encrypted frames by our proposed encryption are shown in Figure 7.20. The result shows that the average PSNR was 8.56 dB for encrypted videos.

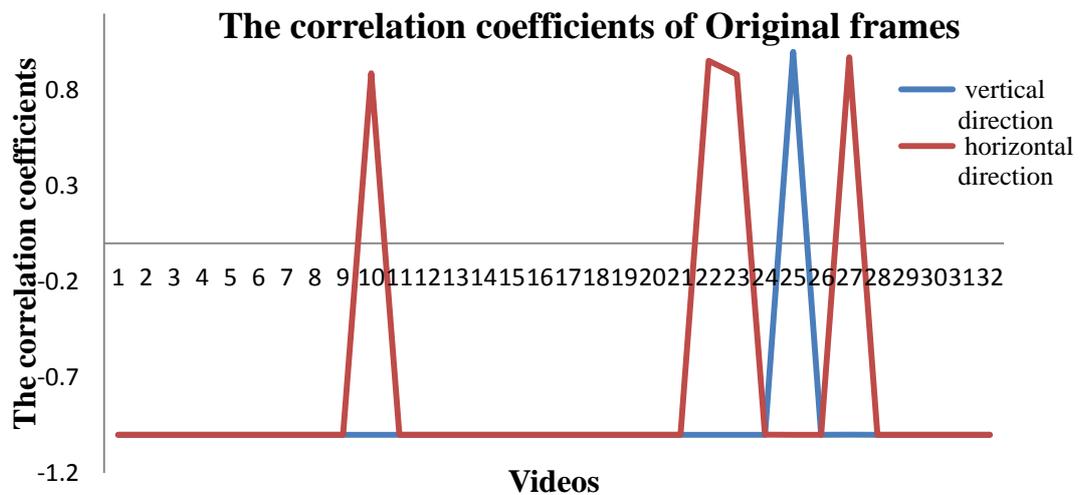
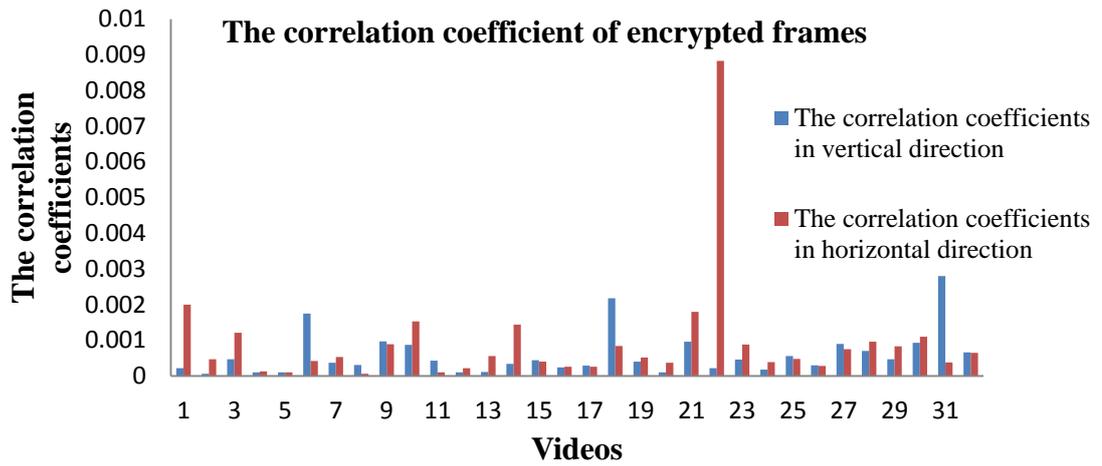


Figure 7.23 the correlation coefficient of adjacent pixels in encrypted and original frames

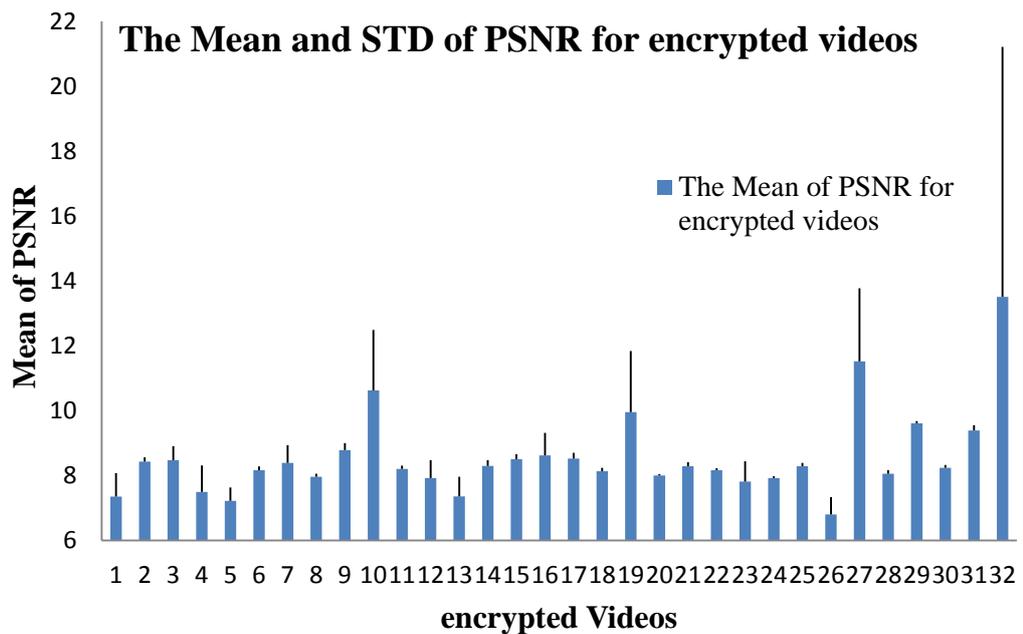
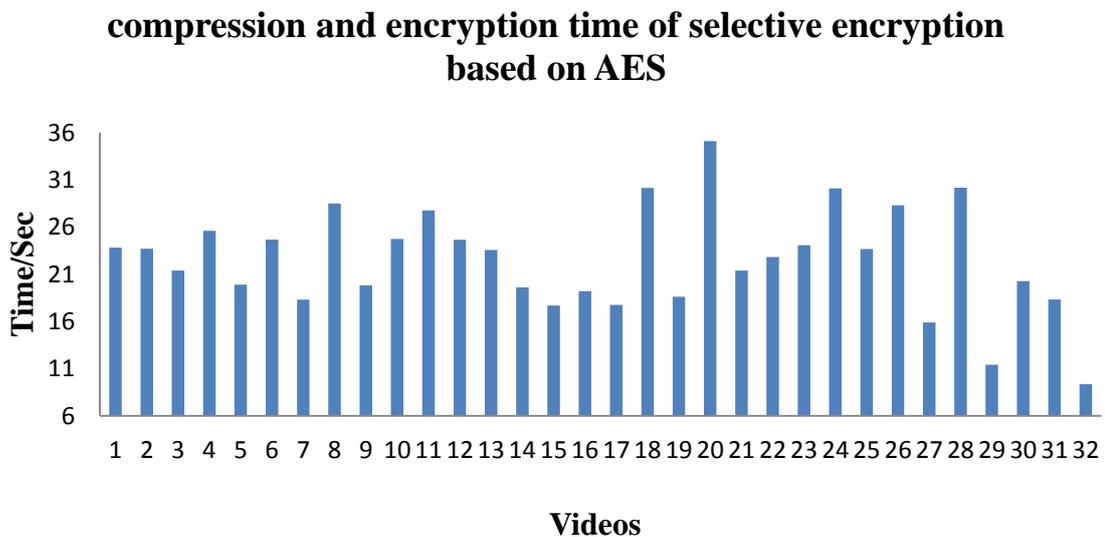


Figure 7.24 the mean and STD of PSNR for encrypted and decrypted videos

## 7.5 Selective encryption based on AES

Selective encryption is desirable in constrained communication, such as real-time networking and limited computational power devices, because it reduces the amount of data to encrypt and provides effective security; this depends on the selected parts identified by our proposed method to encrypt videos. The standard ciphers (AES, Triple DES, RSA, etc.) are known to be more secure but have high computational cost not suitable for very large files encrypting. But it is reasonable to ask whether these ciphers become adequate for selective encryption in video coding.

Here, we shall compare the time complexity of our encryption scheme with that of the traditional AES cipher to encrypt only the significant coefficients (selective encryption) of low frequency sub-band of level 3(LL3). In this experiment, the same test videos and threshold (THR) have been used. Results are shown in Figure 7.25. The results clearly show that the AES cipher is not efficient for video streaming processing although it has been used in selective encryption concepts.



*Figure 7. 25 illustrates the consumed time of selective encryption baser on AES*

## 7.6 Conclusion

In this chapter, another efficient video compression and encryption based on edge-phase sensing has been developed, implemented and its performance in terms of many criteria was tested. This scheme deploys the lossless phase sensing encoding scheme to compress the representation of mismatched high frequency sub-band blocks of n-RFs. The performance testing covered various compression indicators including CR, PSNR and time complexity. Not only this scheme outperformed the earlier schemes, but it has been shown to meet stringent video streaming requirements for secured videos with different size frames.

Test results showed that the proposal efficiently compresses large frame sizes and achieved better CR and quality compared with small frame size. In fact, the scheme is shown to have excellent scalability characteristics. Moreover, the experimental results show that the new scheme performs better than DCT and SPIHT algorithms. In addition, when compared the performance of phase sensing with RLE followed by Huffman encoding, we found that the compression efficiency encoding based on phase sensing outperforms the other two.

The scheme was also shown to reduce video coding processing, good image quality and CR for colour videos of two different sizes even under the stated constrains. The various experiments conducted in this chapter, provide evidence to our confidence that modest relaxation of the stated constrains, will certainly enable our scheme to deal with much higher resolution RGB videos. But we shall deal with that in the future.

The new modified encryption algorithm is based on identifying the significant coefficients of low frequency sub-band of level 3. These coefficients are encrypted by XORing with the bits key stream of a modified A5 cipher. The initial seed (64 bits) of A5 registers is derived from chaotic logistic map. Moreover, the linearity weakness of LFSR stream ciphers has been mitigated using the modified A5 whose majority rule clocking rule was changed to a chaotic logistic map rule. The bit streams of the modified A5 ( $kc_i$ ) are XORed with selected bits  $x_i(m)$  from the significance parts of low frequency sub-band LL3. The selected  $x_i(m)$  is also based on chaotic logistic map. In addition, the threshold value THRE, which is used in allocating the significant coefficients of low frequency sub-band (LL3-SC), is ciphered by chaotic logistic map and sent to the transmitter. In order to increase the security level of encryption, the

control parameter to Sensing Matrix (SM) generation ( $\max y_{(n)}$ ) is also encrypted by chaotic logistic map and sending it to the transmitter.

Finally we have demonstrated that the use of selective encryption based on AES is not-efficient for video encryption.

## Chapter 8

### Conclusion and Future work

The research project conducted during the preparation of this thesis was aimed at investigating the viability of secure video transmission over open and constrained communication network channels. These imposed constraints in this thesis may seem unnecessary in current time characterised by availability of sophisticated powerful computing and communication hardware and infrastructure in the developed world, but it is justified due the huge digital gap that separates them from third world countries and in particular post conflict countries with depleted infrastructure and scarce resources. This project is primarily developed for the latter group of countries, including my own, who desperately need the technology but cannot afford it.

The main challenges in such a task are to meet a number of standard criteria relating to the constrained bandwidth, reasonably high image quality at the receiving, the execution time, and robustness against security attacks. We have developed a number of simultaneous compression and encryption schemes that were tested performances at each stage in terms of above competing criteria.

In general, video compression is based on detecting and removing spatial correlation within the video frames as well as temporal correlations across the video frames. Correlation in this context is measured in terms of similarity between blocks of pixels. Temporal correlation is expected to be more evident across sequences of frames captured within a short period of time (often a fraction of a second). Frequency domain transformation such as DCT and DWT are known for their capability to organise the frequency content (coefficients) in such a way that correlation in the frequency domain is more amenable for efficient detection. The length of the period within which the temporal correlation is high is an indicator of video complexity. Removing spatial/temporal correlation here is based on encoding only one block from each class of equivalent (i.e. similar) blocks and remembering the position of all other block within the equivalence class. Encoding a block here means finding the most compact representation of the block content, which must capture the most significant pixels/coefficient. Encoding scheme must allow the recovery of the significant coefficient together with the block position. Many encoding schemes are available each with their advantages and disadvantages in terms of efficiency, compactness, and the

amount of loss of information when decoding. To develop a video compression scheme with desired properties we need to make a number of decisions, that have an impact on each other, including the choice of the frequency domain transform(s), the size of blocks, how to select similarity threshold(s), how to select the period (number of frames) over which the temporal correlation are encoded, and what encodings can be applied. The success of these choices is dependent on the desired performance in terms of improving on some of the above mentioned compression criteria. Many video compression schemes have been developed simply by using a still image compression scheme on every frame without any consideration of temporal redundancies, i.e. have a period  $n=1$  for temporal correlation encoding. But this approach is not efficient and may involve dropping frames. Efficient video compression will use a period  $n \gg 1$  to encode temporal correlated blocks, and such schemes uses a still image compression scheme on the first reference frame (RF) in the sequence and encodes the next  $(n-1)$  non-reference (n-RF) frames using a special encoding scheme that depends on the encoded RF and detecting the temporal correlation with each other and/or the RF frame before selecting a new RF. The MPEG and H264 schemes all have various implementations of video compression with periods  $n > 1$ , and in this thesis we followed the same strategy. Instead of using one frequency transform, efficiency considerations was the main reason for investigating the use a combination of DWT and DCT, whereby the DCT could only be applied to the non-LL high frequency sub-bands.

Designing video encryption again is influenced by a number of decisions. For image/video encryption, it has long been recognised that the selective encryption is the only viable approach to deal with the overwhelming file size. In theory, selection can be made in the spatial domain or the frequency domain. Simultaneous compression and encryption, is a good reason for opting to apply selective encryption in the frequency domain. For secured video streaming, it is necessary to apply encryption and compression simultaneously and this would be more efficient if the selected data for encryption will be separate from the input data to the compression data. Since the LL sub-bands of wavelet transformed images approximate the original images, while the non-LL high frequency sub-band coefficients are highly correlated then we decided to select the LL-coefficients for encryption. The next decision relates the type of ciphers to be adopted. Again, we followed the common practice of using stream cipher to meet the efficiency requirements of secured video streaming. Key stream generation is the next

decision to make. For that we investigated a number of options and the ultimate choice will depend on robustness to attacks.

We first designed and tested the performance of a hybrid EZW and DCT algorithm for simultaneous still image compression and encryption to obtain the most suitable scheme for encoding and encryption of the RFs within a secure video transmission over open network channels. We established that the application of DCT on the non-LL sub-bands has reduced the iterative loop scan of the EZW encoding. The results confirmed that the performance of hybrid method is better than the standalone EZW in terms of time consumed for coding process and compression efficiency. The encryption was performed on the initial threshold of EZW only which was simply XORed with a 10-bit stream generated by a simple LFSR. The security analysis shows that the encryption scheme is secure against the statistical and frequency attack. However, the computational time of this scheme is perfectly suitable for offline secure video transmission but is relatively high to be suitable in real-time scenario. Consequently, this method may be used.

An alternative to this modified EZW image compression and encryption scheme have been proposed to improve time complexity, compression ratio and security. For the first two objectives, the new compression procedure is limited to the high frequency wavelet sub-bands and exploits the increased chances of similarity between these blocks. The security is improved by encrypting on the entire LL sub-band (low frequency sub-bands of image wavelet decomposition). The compression is performed block-by-block by combining DWT, DCT and Vector Quantization (CVQ). The algorithm has a relatively higher speed and higher compression ratio. We found that compression on high frequency sub-band of level 3 is ineffective and degrades the quality. Therefore, we implement the compression method on high frequency sub-bands of level 1 and 2 and the encryption on low frequency sub-bands of level 3. The encryption scheme is selectively scrambling the low frequency LL3 sub-band appended with few coefficients from the level 3 high frequency sub-bands using two LFSRs. The quality of the decompressed and decrypted image however is dependent on the fixed block similarity threshold, and loss of quality for some images is more noticeable around edge features.

To improve the performance of the above algorithm in terms of quality, we investigated the use of the well-known statistical properties and parameters of high frequency wavelet sub-bands. These properties help fast extraction of the significant coefficients (edges). As a result, we proposed a block based similarity by combining DWT, edges

sensing, DCT and CVQ to achieve better compression and quality compared with edges standalone compression. We investigated the effect of applying variable thresholds, driven from STD of high frequency sub-bands coefficients, to reveal the significant coefficients. The results show that the compression ratio and quality are proportional with the threshold value. The variation in quality for different images can be made less apparent by using adaptive thresholds obtained from different multiples of the STD.

Having developed a suitable still image compression and encryption scheme for use on video RFs, from chapter 5 onwards we then focused on furthering our development of a simultaneous video compression and encryption based on hybrid DWT, DCT and vector quantization. The video compression has been performed using block based similarity by combining DWT and DCT to achieve high compression with an acceptable quality compared with non-block based approach. The compression algorithm reduced the computational time for block matching between reference frame and current frame. In addition, the matching criterion of this approach, which is used to quantify the similarity between blocks, is very fast because only subtraction operation is included. The encryption algorithm utilised two LFSRs seeded with three secret keys to scramble the significant wavelet coefficients multiple times. The encryption analysis includes histogram analysis, correlation analysis and PSNR. The security analysis shows that a cipher algorithm is secure against statistical analysis attack and frequency analysis attack. In addition, we examined the effect of the period Reference Frame (RF) on the compression efficiency, quality and encoding time. The experimental results show that the period RF for every 25 frames realised high compression efficiency with acceptable quality.

The video quality of the above scheme decoder was unfortunately fluctuating according to the complexity of the video content in terms of temporal variation in texture. In chapter 6, we refined this scheme in two ways. Firstly, the n-RF were compressed in terms of RF to reduce temporal correlation but replaced the DCT with the statistically determined thresholds for edge sensing as described in chapter 4. Secondly, the selective encryption algorithm scrambles the wavelet coefficients of the edges extracted from the low frequency sub-band, and we used the chaotic logistic map combined with sine map to generate LFSR secret key. The experimental results show that this proposed algorithm has the following features: high compression, acceptable quality, and resistance to the brute force and statistical attack with low computational processing.

A close consideration of the performance of the compression procedure of last two video scheme shows that the frame quality, compression ratio, and the time complexity is influenced mostly by the way the procedure that detect and locate the significant coefficients in the mismatched blocks which allows the recovery of their sparse version. Realising that recovering sparse signals from smaller set of measurements is what the new emerging sampling paradigm of compressive sensing was a motivation to search for a technique that have the same effect of compressive sensing. In chapter 7, we found that the very efficient phase sensing procedure, commonly used in wireless communication applications, provides a tool which is equivalent to compressive sensing tools. The final refinement of the video scheme employs the phase sensing scheme for the representation of mismatched high frequency sub-band blocks of n-RFs. The performance testing covered various compression indicators including CR, PSNR and time complexity. Not only this scheme outperformed the earlier schemes, but it has been shown to be highly scalable to videos with large size frames and performs the DCT and SPIHT compression schemes.

We have also demonstrated that new scheme works well for colour videos of large sizes, and achieves relatively low time processing, good image quality and CR even under the stated constrains. The various experiments reported in chapter 7, indicate that modest relaxation of the stated constrains, will certainly enable our scheme to deal with much higher resolution RGB videos. Future work will attempt to demonstrate this hypothesis and adapt our scheme for real-time simultaneous compression and encryption of higher resolution videos including HD.

The final scheme also strengthen the security of the encryption algorithm, employed in chapter 6, by encrypting the significant coefficients of low frequency sub-band of level 3 by a new version of the A5 cipher that modifies the clocking rule of A5 using chaotic logistic maps.

The systematic process of refinement we adopted in improving the performance of each of the developed secure image/video compression has helped in meeting the requirements of all stated success criteria. Although, the results reveal that the level of success in terms of image quality and CR fluctuates depending on the level of image complexity in terms of image texture and the speed of motion by image objects. However, these were the result of using the same period of selecting RF's as well as fixing the similarity thresholds throughout the sequence of frames in a period. This can be remedied by making the choice of these factors adaptive in terms of image

complexity. In the future, we plan to investigating measures of video complexity and develop such adaptive schemes.

In the future, we shall also

1. Improve computational time by investigating the use of other than the DCT transform such as Hadamard transforms (Aung, Ng and Shwe 2009).
2. Enhance the compression efficiency by using lossless compression to compress the low frequency sub-band including the use of phase sensing instead of DCT to compress the LL sub-band of the RF frames prior to encryption.
3. Develop an FPGA implementation of such adaptive schemes that should remain scalable to much higher frame size and resolutions than so far tested in this thesis. The performance of such an implementation could be analysed on real hardware for use in secure scenarios such as satellite transmission in conflicts zones.
4. Testing and upgrading this system to be used in real-time wireless video streaming.

## References

- Abomhara, M, OO Khalifa, O Zakana, AA Zaidan, BB Zaidan, and A Rame. "Video Compression Techniques: An Overview." *Journal of Applied Sciences* 10, no. 16 (2010).
- Acharjee, S., and S.S. Chaudhuri. "A new fast motion vector estimation algorithm for video compression." *IEEE*. 2012. 1216-1219.
- Acharya, T., and P.S. Tsai. *JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures*. Wiley-Blackwell, 2005.
- Al-jawad, Naseer. "exploiting statistical properties of wavelet coefficients for image/video processing and analysis tasks ." Ph.D. dissertation, University of Buckingham, 2009.
- Al-Jawad, Naseer, Johan Ehlers, and Sabah Jassim. "An efficient realtime video compression algorithm with high feature preserving capability." *International Society for Optics and Photonics*. 2006. 625002-625002.
- Annadurai, S. *Fundamentals of digital image processing*. Pearson Education India, 2007.
- Anoop, B.N., S.N. George, and P.P. Deepthi. "Secure video transcoders based on correlation preserving sorting algorithm." 2014. 771-775.
- Apostolopoulos, John G, Susie Wee, Frederic Dufaux, Touradj Ebrahimi, Qibin Sun, and Zhishou Zhang. "The emerging JPEG-2000 security (JPSEC) standard." *Citeseer*. 2006.
- Barkan, Elad, Eli Biham, and Nathan Keller. "Instant ciphertext-only cryptanalysis of GSM encrypted communication." In *Advances in Cryptology-CRYPTO 2003*, 600-616. Springer, 2003.
- Batham, S., V.K. Yadav, and A.K. Mallik. "ICSECV: An efficient approach of video encryption." 2014. 425-430.
- Benchikh, Salam, and Michael Corinthios. "A hybrid image compression technique based on DWT and DCT transforms." (IET) 2011.
- Bigot, Jérémie, Claire Boyer, and Pierre Weiss. "An analysis of block sampling strategies in compressed sensing." *arXiv preprint arXiv:1305.4446*, 2013.

- Blelloch, Guy E. "Introduction to data compression." *Computer Science Department, Carnegie Mellon University*, 2001.
- Bluman, Allan G. *Elementary Statistics: A Step-by-Step Approach*. Irwin, 1996.
- Cabeen, K., and P. Gent. "Image compression and the discrete cosine transform." *College of the Redwoods, Mat*, 1998.
- Carreto-Castro, MF, JM Ramirez, JL Ballesteros, and D Baez-Lopez. "Comparison of lossless compression techniques." *IEEE*. 1993. 1268-1270.
- Chang, Hoyoung, and Kyeongsoon Cho. "High-performance inverse transform circuit based on butterfly architecture for H. 264 high profile decoder." *IEEE*. 2010. 394-397.
- Chen, CL, Tong Zhang, and Yicong Zhou. "Image encryption algorithm based on a new combined chaotic system." *IEEE*. 2012. 2500-2504.
- Chen, Lidong, and Guang Gong. *Communications System Security*. CRC Press, 2012.
- Chown, Pete. "Advanced encryption standard (AES) ciphersuites for transport layer security (TLS)." Tech. rep., RFC 3268, June, 2002.
- David, Salomon, G Motta, and D Bryant. "Data compression: The complete reference." *Springer, USA., ISBN 10 (2007): 1846286026*.
- Deng, Chenwei, Weisi Lin, Bu-Sung Lee, and C.T. Lau. "Robust image compression based on compressive sensing." 2010. 462-467.
- Donoho, David L. "Compressed sensing." *Information Theory, IEEE Transactions on (IEEE)* 52, no. 4 (2006): 1289-1306.
- Dridi, M., B. Bouallegue, and A. Mtibaa. "Crypto-compression of medical image based on DCT and chaotic system." 2014. 1-6.
- Dubrova, Elena, Maxim Teslenko, and Hannu Tenhunen. "On analysis and synthesis of (n, k)-non-linear feedback shift registers." *IEEE*. 2008. 1286-1291.
- Dufaux, Frederic, and Touradj Ebrahimi. "Scrambling for video surveillance with privacy." *IEEE*. 2006. 160-160.
- Ehlers, Johan Hendrik. "Definition driven image processing for constrained environments." Ph.D. dissertation, University of Buckingham, 2008.

- El, Nawal F, and Osama M Abu. "Quality of Encryption Measurement of Bitmap Images with RC6, MRC6, and Rijndael Block Cipher Algorithms." *IJ Network Security* 5, no. 3 (2007): 241-251.
- Engel, Dominik, Thomas et al, and Andreas Uhl. "A survey on JPEG2000 encryption." *Multimedia systems* (Springer) 15, no. 4 (2009): 243-270.
- Fadil, T.A., S.N. Yaakob, and B. Ahmad. "A hybrid chaos and neural network cipher encryption algorithm for compressed video signal transmission over wireless channel." 2014. 64-68.
- Fontaine, Caroline, and Fabien Galand. "A survey of homomorphic encryption for nonspecialists." *EURASIP Journal on Information Security* (Hindawi Publishing Corporation) 2007 (2007).
- Gan, Lu. "Block compressed sensing of natural images." *IEEE*. 2007. 403-406.
- Goldsmith, Andrea. "Fundamentals of Wireless Communication." *Fundamentals of Wireless Communication*. USA, Cambridge University Press, 2005.
- Gopalakrishnan, T., S. Ramakrishnan, and M. Balakumar. "An image encryption using chaotic permutation and diffusion." 2014. 1-5.
- Goyal, V.K., A.K. Fletcher, and S. Rangan. "Compressive Sampling and Lossy Compression." *Signal Processing Magazine, IEEE* 25, no. 2 (March 2008): 48-56.
- H, Hossam El-din, Hamdy M Kalash, and Osama S Farag. "Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images." *IEEE*. 2007. 1-7.
- Horan, David, and Richard Guinee. "A novel stream cipher for cryptographic applications." *IEEE*. 2006. 1-5.
- Hu, Yuping, Congxu Zhu, and Zhijian Wang. "An Improved Piecewise Linear Chaotic Map Based Image Encryption Algorithm." *The Scientific World Journal* (Hindawi Publishing Corporation) 2014 (2014).
- Huang, Rong, and Kouichi Sakurai. "A Robust and Compression-Combined Digital Image Encryption Method Based on Compressive Sensing." *IEEE*. 2011. 105-108.
- Impoco, G. "JPEG2000-a short tutorial." *Visual Computing Lab-ISTI-CNR Pisa* (Citeseer), 2004.

- Jain, J., and A. Jain. "Displacement measurement and its application in interframe image coding." *Communications, IEEE Transactions on (IEEE)* 29, no. 12 (1981): 1799-1808.
- Janaki, R., A.D. Tamilarasi, and others. "Still Image Compression by Combining EZW Encoding with Huffman Encoder." *International Journal of Computer Applications (Foundation of Computer Science (FCS))* 13, no. 7 (2011): 1-7.
- Jolfaei, A., and A. Mirghadri. "An Image Encryption Approach Using Chaos and Stream Cipher." *Journal of Theoretical and Applied Information Technology* 19, no. 2 (2010): 117-125.
- Jridi, Maher, and Ayman AlFalou. "A VLSI implementation of a new simultaneous images compression and encryption method." *IEEE*. 2010. 75-79.
- Kenneth, R. "Castleman, Digital image processing." *Castleman, Digital image processing*. Prentice Hall Press, Upper Saddle River, NJ, 1996.
- Khlif, N., T. Damak, F. Kammoun, and N. Masmoudi. "A very efficient encryption scheme for the H.264/AVC CODEC adopted in Intra prediction mode." 2014. 1-7.
- . "Motion vectors signs encryption for H.264/AVC." 2014. 1-6.
- Kim, Seyun, and Nam Ik Cho. "Hierarchical Prediction and Context Adaptive Coding for Lossless Color Image Compression." *Image Processing, IEEE Transactions on* 23, no. 1 (Jan 2014): 445-449.
- Kim, Younhee, DongSan Jun, Soon-heung Jung, Jin Soo Choi, and Jinwoong Kim. "A Fast Intra-Prediction Method in HEVC Using Rate-Distortion Estimation Based on Hadamard Transform." *ETRI Journal* 35, no. 2 (2013).
- Kocarev, Ljupco, and Shiguo Lian. *Chaos-based cryptography*. Springer, 2011.
- Li, J., J. Li, and C.C.J. Kuo. "Embedded DCT Still Image Compression." 1996.
- Li, Shujun, Xuan Zheng, Xuanqin Mou, and Yuanlong Cai. "Chaotic encryption scheme for real-time digital video." *International Society for Optics and Photonics*. 2002. 149-160.
- Liansheng, Sui, Wang Wengang, Duan Kuaikuai, and Zhang Zhiqiang. "A novel grayscale image encryption algorithm based on logistic map." 2014. 222-225.

- Liu, Dong, Xiaoyan Sun, Feng Wu, Shipeng Li, and Ya-Qin Zhang. "Image compression with edge-based inpainting." *Circuits and Systems for Video Technology, IEEE Transactions on (IEEE)* 17, no. 10 (2007): 1273-1287.
- Lu, Wenjun, Avinash L Varna, and Min Wu. "Confidentiality-Preserving Image Search: A Comparative Study between Homomorphic Encryption and Distance-Preserving Randomization." (IEEE) 2014.
- Ma, Jinming. "Wavelet based images/video compression techniques for telemedicine application." Ph.D. dissertation, University of Buckingham, 2002.
- Maksuanpan, S., T. Veerawadtanapong, and W. San-Um. "Robust digital image cryptosystem based on nonlinear dynamics of compound sine and cosine chaotic maps for private data protection." 2014. 418-425.
- Mao, Yaobin, and Guanrong Chen. "Chaos-based image encryption." In *Handbook of Geometric Computing*, 231-265. Springer, 2005.
- Miyamoto, R. "Arithmetic coding/decoding apparatus of MQ-Coder system and renormalization method." *Arithmetic coding/decoding apparatus of MQ-Coder system and renormalization method*. Google Patents, #jul#~20 2004.
- Monro, DM, and GJ Dickson. "Zerotree Coding of DCT coefficients." *IEEE*. 1997. 625-628.
- Niu, Yi, Xiaolin Wu, Guangming Shi, and Xiaotian Wang. "Edge-Based Perceptual Image Coding." *Image Processing, IEEE Transactions on (IEEE)* 21, no. 4 (2012): 1899-1910.
- Norcen, Roland, and Andreas Uhl. "Selective encryption of the JPEG2000 bitstream." In *Communications and Multimedia Security. Advanced Techniques for Network and Data Protection*, 194-204. Springer, 2003.
- Paar, Christof, and Jan Pelzl. *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.
- Pande, Amit, Joseph Zambreno, and Prasant Mohapatra. "Architectures for Simultaneous Coding and Encryption Using Chaotic Maps." 2011. 351-352.
- Pandur, Thiruvallur, and TN Thiruvallur. "Images and Its Compression Techniques--A Review." 2009.

- Pareek, NK, V. Patidar, and KK Sud. "Image encryption using chaotic logistic map." *Image and Vision Computing* (Elsevier) 24, no. 9 (2006): 926-934.
- Pommer, Andreas, and Andreas Uhl. "Selective encryption of wavelet-packet encoded image data: efficiency and security." *Multimedia Systems* (Springer) 9, no. 3 (2003): 279-287.
- Rajagopal, S, and A Shenbagavalli. "A Survey of Video Encryption Algorithms Implemented In Various Stages of Compression." *International Journal of Engineering* 2, no. 2 (2013).
- Rao, Kamisetty Ramam, and Patrick C Yip. *The transform and data compression handbook*. CRC press, 2010.
- Richardson, Iain E. *The H. 264 advanced video compression standard*. John Wiley & Sons, 2011.
- Rohith, S., K.N. Hari Bhat, and A.N. Sharma. "Image encryption and decryption using chaotic key sequence generated by sequence of logistic map and sequence of states of Linear Feedback Shift Register." 2014. 1-6.
- Said, A., and W.A. Pearlman. "A new, fast, and efficient image codec based on set partitioning in hierarchical trees." *Circuits and systems for video technology, IEEE Transactions on* (IEEE) 6, no. 3 (1996): 243-250.
- Sathishkumar, GA, Srinivas Ramachandran, and K Bhoopathy Bagan. "Image encryption using random pixel permutation by chaotic mapping." *IEEE*. 2012. 247-251.
- Sayood, Khalid. *Introduction to data compression*. Newnes, 2012.
- Schelkens, Peter, Athanassios Skodras, and Touradj Ebrahimi. *The JPEG 2000 Suite*. Vol. 15. John Wiley & Sons, 2009.
- Schulze, Henrik, and Christian Luders. *Theory and applications of OFDM and CDMA: Wideband wireless communications*. John Wiley & Sons, 2005.
- Shahid, Zafar, Marc Chaumont, and William Puech. "Fast Protection of H. 264/AVC by Selective Encryption of CAVLC and CABAC for I and P frames." *Circuits and Systems for Video Technology, IEEE Transactions on* (IEEE) 21, no. 5 (2011): 565-576.
- Shen, Haojie, Li Zhuo, and Yingdi Zhao. "An efficient motion reference structure based selective encryption algorithm for H.264 videos." *Information Security, IET* 8, no. 3 (May 2014): 199-206.

- Shi, Yun Q, and Huifang Sun. *Image and video compression for multimedia engineering: fundamentals, algorithms, and standards 2nd*. CRC Press, Inc. Boca Raton, FL, USA ©2008, 2008.
- Shingate, VS, TR Sontakke, and SN Talbar. "Still Image Compression using Embedded Zerotree Wavelet Encoding." *International Journal of Computer Science & Communication* 1, no. 1 (2010): 21-24.
- Shrestha, Suchitra, and Khan Wahid. "Hybrid DWT-DCT algorithm for biomedical image and video compression applications." *IEEE*. 2010. 280-283.
- Sonnati, F., and P. Sinergia. "FLASH VIDEO TECHNOLOGY AND OPTIMIZATIONS." 2004.
- SubhamastanRao, T, M Soujanya, T Hemalatha, and T Revathi. "Simultaneous Data Compression and Encryption." *International Journal of Computer Science and Information Technologies* 2, no. 5 (2011): 2369-2374.
- Tian, J., and R.O. Wells Jr. "A lossy image codec based on index coding." *Citeseer*. 1996.
- Uhl, A., and A. Pommer. *Image and video encryption: from digital rights management to secured personal communication*. Vol. 15. Springer, 2005.
- Unterweger, Andreas, and Andreas Uhl. "Length-preserving Bit-stream-based JPEG Encryption." *ACM*. 2012. 85-90.
- Urban, F., O. D forges, and J.F. Nezan. "Optimization of the motion estimation for parallel embedded systems in the context of new video standards." *International Society for Optics and Photonics*. 2012. 849917-849917.
- Valens, C. "Embedded zerotree wavelet encoding." *perso. orange.fr/polyvalens/clemens/ezw/ezw.html* (Citeseer), 1999.
- Vetrivel, S, K Suba, and G Athisha. "An overview of h. 26x series and its applications." *International Journal of Engineering Science and Technology* 2, no. 9 (2010): 4622-4631.
- Wharton, Eric J, Karen A Panetta, and Sos S Aгаian. "Scalable encryption using alpha rooting." *International Society for Optics and Photonics*. 2008. 69820G--69820G.

- Wiegand, Thomas, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. "Overview of the H.264/AVC video coding standard." *Circuits and Systems for Video Technology, IEEE Transactions on (IEEE)* 13, no. 7 (2003): 560-576.
- Wong, Kwok-Wo, Qiuzhen Lin, and Jianyong Chen. "Simultaneous arithmetic coding and encryption using chaotic maps." *Circuits and Systems II: Express Briefs, IEEE Transactions on (IEEE)* 57, no. 2 (2010): 146-150.
- Yang, Ming, and Nikolaos Bourbakis. "An overview of lossless digital image compression techniques." *IEEE*. 2005. 1099-1102.
- Zacharovas, Stanislovas, Andrej Nikolskij, and Jevgenij Kuchin. "Mobile phone color holography." *International Society for Optics and Photonics*. 2010. 761900--761900.
- Zakaria, Nur Hafiza, Kamaruzzaman Seman, and Ismail Abdullah. "Modified A5/1 Based Stream Cipher For Modified A5/1 Based Stream Cipher For Secured GSM Communication Communication." *IJCSNS* 11, no. 2 (2011): 223.
- Zeng, Jin, Oscar C Au, Wei Dai, Yue Kong, Luheng Jia, and Wenjing Zhu. "A tutorial on image/video coding standards." *IEEE*. 2013. 1-7.
- Zhang, Yifu, Shunliang Mei, Quqing Chen, and Zhibo Chen. "A novel image/video coding method based on compressed sensing theory." *IEEE*. 2008. 1361-1364.
- Zhou, Yicong, Karen Panetta, and Sos Agaian. "Image encryption based on edge information." *International Society for Optics and Photonics*. 2009. 725603-725603.