

7-24-2003

# High performance shift invariant motion estimation and compensation in wavelet domain video compression

Weiting Cai

*Florida International University*

**DOI:** 10.25148/etd.FI14052529

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

---

## Recommended Citation

Cai, Weiting, "High performance shift invariant motion estimation and compensation in wavelet domain video compression" (2003).  
*FIU Electronic Theses and Dissertations*. 1965.  
<https://digitalcommons.fiu.edu/etd/1965>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact [dcc@fiu.edu](mailto:dcc@fiu.edu).

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

HIGH PERFORMANCE SHIFT INVARIANT MOTION ESTIMATION  
AND COMPENSATION IN WAVELET DOMAIN VIDEO COMPRESSION

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Weiting Cai

2003

To: Dean Vish Prasad  
College of Engineering

This dissertation, written by Weiting Cai, and entitled High Performance Shift Invariant Motion Estimation and Compensation in Wavelet Domain Video Compression, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

---

Dr. Prashanna Jayakar

---

Dr. Armando Barreto

---

Dr. Subbarao V. Wunnava

---

Dr. Malek Adjouadi, Major Professor

Date of Defense: July 24, 2003

The dissertation of Weiting Cai is approved.

---

Dean Vish Prasad  
College of Engineering

---

Dean Douglas Wartzok  
University Graduate School

Florida International University, 2003

© Copyright 2003 by Weiting Cai

All rights reserved.

DEDICATION

to

My Parents!

## ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my Major Professor, Dr. Malek Adjouadi, for his patient guidance and strong support during the course of my doctoral studies. This research work has been carried out by his constructive advice and detailed comments. Besides the academic content I learned from him, I'm also very impressed by his strict research attitude and great working enthusiasm!

I want to thank my committee members, Dr. Subbarao V. Wunnava, Dr. Armando Barreto and Dr. Prasanna Jayakar, for their helpful suggestions.

I gratefully acknowledge the support from the National Science Foundation under grant EIA-9906600 and the Office of Naval Research under grant N00014-99-1-0952.

I would thank Dr. Melvin Ayala, Mark Rossman, Nuannuan Zong, Andres Herrera, Mike Valdes and all the friends in the CATE Lab. I also want to thank the Electrical and Computer Department, FIU. Special thanks to Pat Brammer.

I would say my deepest appreciation to my father, my mother and my sister who are the inspirations to me, for their constant encouragement and unreserved love!

ABSTRACT OF THE DISSERTATION

HIGH PERFORMANCE SHIFT INVARIANT MOTION ESTIMATION  
AND COMPENSATION IN WAVELET DOMAIN VIDEO COMPRESSION

by

Weiting Cai

Florida International University, 2003

Miami, Florida

Professor Malek Adjouadi, Major Professor

The contributions of this dissertation are in the development of two new interrelated approaches to video data compression:

- 1) A level-refined motion estimation and subband compensation method for the effective motion estimation and motion compensation.
- 2) A shift-invariant sub-decimation decomposition method in order to overcome the deficiency of the decimation process in estimating motion due to its shift-invariant property of wavelet transform.

The enormous data generated by digital videos call for an intense need of efficient video compression techniques to conserve storage space and minimize bandwidth utilization. The main idea of video compression is to reduce the interpixel redundancies inside and between the video frames by applying motion estimation and motion compensation (MEMC) in combination with spatial transform coding. To locate the global minimum of the matching criterion function reasonably, hierarchical motion estimation by coarse to fine resolution refinements using discrete wavelet transform is applied due to its intrinsic multiresolution and scalability natures.

Due to the fact that most of the energies are concentrated in the low resolution subbands while decreased in the high resolution subbands, a new approach called level-refined motion estimation and subband compensation (LRSC) method is proposed. It realizes the possible intrablocks in the subbands for lower entropy coding while keeping the low computational loads of motion estimation as the level-refined method, thus to achieve both temporal compression quality and computational simplicity.

Since circular convolution is applied in wavelet transform to obtain the decomposed subframes without coefficient expansion, symmetric-extended wavelet transform is designed on the finite length frame signals for more accurate motion estimation without discontinuous boundary distortions.

Although wavelet transformed coefficients still contain spatial domain information, motion estimation in wavelet domain is not as straightforward as in spatial domain due to the shift variance property of the decimation process of the wavelet transform. A new approach called sub-decimation decomposition method is proposed, which maintains the motion consistency between the original frame and the decomposed subframes, improving as a consequence the wavelet domain video compressions by shift invariant motion estimation and compensation.



# TABLE OF CONTENTS

CHAPTER	PAGE
CHAPTER 1 Introduction.....	1
1.1 Motivation behind the research.....	1
1.2 Objective of the research .....	1
1.3 Methodologies of the research .....	2
CHAPTER 2 Video Compression system and standards .....	5
2.1 Video compression standards .....	5
2.2 Compression schemes .....	5
2.2.1 Psychovisual compression .....	5
2.2.2 Spatial compression .....	8
2.2.2.1 Discrete cosine transform .....	8
2.2.2.2 Wavelet transform.....	8
2.2.3 Temporal compression.....	9
2.2.4 Coding compression.....	10
2.3 Fidelity criteria.....	11
2.4 Video compression system .....	12
CHAPTER 3 Wavelet Multiresolution Analysis and Compression .....	15
3.1 A retrospective on wavelet transform .....	15
3.2 Multiresolution analysis.....	19
3.3 Fast wavelet transform.....	21
3.4 Orthogonal wavelet filters.....	26
3.5 Biorthogonal wavelet filters.....	30
3.5.1 Biorthogonal wavelet system.....	30
3.5.2 Biorthogonal 9/7 filter.....	36
3.6 Wavelet image compression .....	41
CHAPTER 4 Motion Estimation and Motion Compensation Methods.....	48
4.1 Block-based full search method.....	48
4.2 Fast search methods .....	56
4.2.1 Step search method .....	57
4.2.2 Block-based gradient descent search method .....	60
4.3 Multiresolution motion estimation.....	62
4.4 Video codec simulation and compression results .....	63
CHAPTER 5 Wavelet-Based Video Compression .....	70
5.1 Wavelet-based video codec.....	70
5.2 Wavelet domain multiresolution motion estimation and compensation.....	73
5.3 Entropy coding.....	78
5.3.1 Motion vector coding.....	80
5.3.2 Coefficient coding.....	81

5.4	Simplified wavelet domain motion estimation and compensation methods.....	88
5.4.1	Level-refined motion estimation.....	88
5.4.2	Subband-refined motion estimation.....	91
5.5	Level-refined motion estimation and subband compensation method .....	95
CHAPTER 6 Symmetric-Extended Wavelet Transform .....		102
6.1	Symmetric-extended signal.....	102
6.2	Symmetric-extended wavelet transform .....	104
6.3	Symmetric-extended wavelet transform on image compression .....	108
6.4	Symmetric-extended wavelet transform on video compression .....	112
CHAPTER 7 Wavelet Domain Sub-decimation Decomposition Method .....		115
7.1	Sub-decimation decomposition method.....	115
7.2	Sub-decimation decomposition motion estimation and compensation.....	119
7.3	Sub-decimation decomposition motion estimation and compensation by symmetric-extended wavelet transform .....	129
CHAPTER 8 Conclusions.....		136
References.....		140
Vita.....		147

## LIST OF TABLES

TABLE	PAGE
Table 2.1. Quantization table for luminance channel .....	7
Table 2.2. Quantization table for chrominance channels.....	7
Table 4.1. Computational load comparison using full search and step search method ....	60
Table 5.1. Motion vector refinements between the subbands in wavelet domain    motion estimation.....	76
Table 5.2. Differential motion vector coding in wavelet domain .....	81
Table 5.3. Motion vector refinements and coding .....	90
Table 5.4. Motion vector refinements and coding in subband-refined .....	93
Table 5.5. Computational load comparison of a current block motion estimation using wavelet domain methods.....	97
Table 6.1. Image symmetric extension modes using biorthogonal 9/7 filter.....	109
Table 7.1. Two-level sub-decimation decomposition.....	118
Table 7.2. Computational load comparison between full search method .....	126
Table 7.3. One-dimensional signal extension modes after SWT by biorthogonal 9/7 filter.....	129
Table 7.4. Two-dimensional signal extension modes after SWT by biorthogonal 9/7 filter.....	130

## LIST OF FIGURES

FIGURE	PAGE
Figure 2.1. Example of the importance of color space transformation.....	6
Figure 2.2. Video encoding system.....	13
Figure 2.3. Video decoding system.....	14
Figure 3.1. DWT and IDWT computational structure .....	25
Figure 3.2. Multilevel wavelet decomposition and reconstruction.....	26
Figure 3.3. Daubechies wavelet filter family.....	29
Figure 3.4. Biorthogonal 9/7 filter coefficients .....	37
Figure 3.5. Biorthogonal 9/7 analysis and synthesis lowpass filters .....	38
Figure 3.6. Biorthogonal 9/7 analysis and synthesis highpass filters .....	39
Figure 3.7. The scaling function $\phi(t)$ of the causal biorthogonal 9/7 filter .....	40
Figure 3.8. The scaling function $\tilde{\phi}(t)$ of the causal biorthogonal 9/7 filter.....	41
Figure 3.9. Two-dimensional DWT and IDWT.....	43
Figure 3.10. Multilevel wavelet decomposition on image “azalea” .....	44
Figure 3.11. The corresponding wavelet domain decomposed subbands.....	45
Figure 3.12. The frequency spectrums of the wavelet decomposed subbands .....	46
Figure 4.1. Block-based motion estimation .....	49
Figure 4.2. Block-matching method .....	50
Figure 4.3. Reference-current frame pair from test video sequence “football” .....	51
Figure 4.4. Matching function distributions over the search area.....	52
Figure 4.5. Motion vector fields by FS method .....	53
Figure 4.6. Intermode and intramode compression in motion estimation .....	54

Figure 4.7. Y component motion compensation after FS motion estimation .....	55
Figure 4.8. $C_r$ component motion compensation after FS motion estimation .....	55
Figure 4.9. $C_b$ component motion compensation after FS motion estimation.....	55
Figure 4.10. Step search algorithm .....	58
Figure 4.11. Motion estimation and compensation from step search method .....	59
Figure 4.12. Block-based gradient descent search algorithm .....	61
Figure 4.13. Motion estimation and compensation from BBGDS method .....	61
Figure 4.14. Video encoder kernel.....	64
Figure 4.15. Video decoder kernel.....	65
Figure 4.16. Reference-current frame pair from test video sequence “foreman” .....	65
Figure 4.17. Video compression and reconstruction for sequence “football” .....	66
Figure 4.18. Video compression and reconstruction for sequence “foreman” .....	67
Figure 4.19. Bit rate vs. PSNR for sequence “football” .....	68
Figure 4.20. Bit rate vs. PSNR for sequence “foreman” .....	68
Figure 4.21. Low bit rate video compression .....	69
Figure 5.1. Wavelet video encoder .....	72
Figure 5.2. Wavelet video decoder .....	73
Figure 5.3. Multiresolution wavelet motion estimation.....	75
Figure 5.4. Wavelet-based video encoder kernel.....	77
Figure 5.5. Wavelet-based video decoder kernel.....	77
Figure 5.6. Wavelet decomposition of reference frame and current frame .....	82
Figure 5.7. Motion vector fields of wavelet domain motion estimation.....	82

Figure 5.8. $Y, C_r, C_b$ prediction error frame histograms before and after quantization ...	83
Figure 5.9. Reconstruction frame "football" from wavelet domain.....	84
Figure 5.10. Wavelet domain compression on frame "trover" using multiresolution motion estimation and compensation.....	85
Figure 5.11. Wavelet domain compression on frame "carphone" using multiresolution motion estimation and compensation.....	86
Figure 5.12. Wavelet domain search and spatial domain FS MEMC on frame "trover" .	87
Figure 5.13. Wavelet domain search and spatial domain FS MEMC.....	87
Figure 5.14. Frequency coverage of the lowpass and highpass subbands between levels	88
Figure 5.15. Motion field refinements in level-refined motion estimation method.....	89
Figure 5.16. Wavelet domain compression using level-refined .....	91
Figure 5.17. Wavelet domain subband-refined motion estimation.....	93
Figure 5.18. Wavelet domain compression using subband-refined .....	95
Figure 5.19. Computational loads of motion estimation.....	98
Figure 5.20. Wavelet domain compression using level-refined motion estimation and subband compensation method .....	99
Figure 5.21. Wavelet domain MEMC comparisons on frame "football" .....	100
Figure 5.22. Compression comparison of sequence "carphone" by wavelet domain MEMC methods.....	101
Figure 6.1. W extension of the finite-length signal .....	103
Figure 6.2. H extension of the finite-length signal .....	103
Figure 6.3. (1,2) mode extension of the finite-length signal.....	104
Figure 6.4. (2,1) mode extension of the finite-length signal.....	104
Figure 6.5. Half decimation on W symmetric-extended signal .....	106
Figure 6.6. Half decimation on H symmetric-extended signal .....	106

Figure 6.7. One dimensional SWT and inverse SWT using biorthogonal 9/7 filter.....	108
Figure 6.8. Image compression comparisons using JPEG, PWT and SWT .....	110
Figure 6.9. JPEG, PWT and SWT performances by bit rate vs. PSNR for “azalea”.....	111
Figure 6.10. JPEG, PWT and SWT performances by bit rate vs. PSNR for “parrots”...	111
Figure 6.11. Video compression on sequence "foreman" using SWT method.....	113
Figure 6.12. Wavelet-based video compression on sequence "foreman" .....	114
Figure 7.1. A branch of the discrete wavelet transform.....	115
Figure 7.2. Decimation patterns on one-dimensional wavelet decomposition .....	116
Figure 7.3. Sub-decimation decomposition in wavelet filter bank .....	118
Figure 7.4. Two-dimensional sub-decimation decomposition method.....	119
Figure 7.5. Sub-decimation decomposition method .....	121
Figure 7.6. Motion estimation between the reference-current frame and subframe pairs .....	124
Figure 7.7. Video compression using sub-decimated decomposition method.....	127
Figure 7.8. Compression comparison between FS, PWS and SDD_PWS methods.....	128
Figure 7.9. Compression comparison between FS, PWS and SDD_PWS methods.....	128
Figure 7.10. One-dimensional sub-decimation decomposition .....	130
Figure 7.11. Video compression using sub-decimation decomposition method by symmetric-extended wavelet transform with biorthogonal 9/7 filters on frame "football" .....	132
Figure 7.12. Compression comparison of SDD_SWS and SDD_PWS method.....	133
Figure 7.13. Compression comparison of SDD_SWS and SDD_PWS method.....	133
Figure 7.14. Compression comparison for frame "football" at $CR = 8$ .....	134
Figure 7.15. Compression comparison for frame "carphone" at $CR = 24$ .....	135

# **CHAPTER 1**

## **Introduction**

### **1.1 Motivation behind the research**

Various types of video applications in this information and telecommunication era are emerging around us, from the high definition television to low bit rate video telephony communications over the Internet and the radio systems. The amount of data generated by the information resources may be so large that it can overwhelm the practical cost and bandwidth limitations, as well as tax the demand for processing and transmission operations. The widespread usage of digital videos calls for an intense need of video compression technology to conserve storage space and minimize bandwidth utilization.

Video compression techniques are mainly characterized by applying motion estimation and motion compensation (MEMC) in combination with spatial transform coding. The discrete wavelet transform (DWT) has manifested its intrinsic multiresolution and scalability advantages on image compressions without the annoying blocking artifacts. Thus, video compression techniques have been infused with exciting prospects for new efficient approaches by the beneficial integration of motion estimation and wavelet transformation.

### **1.2 Objective of the research**

The purpose of this dissertation research is to find a new approach to improve the existing motion estimation and compensation methods. The proposed algorithm is used to



compensate for the limitations imposed by the shift variant property of the discrete wavelet transform in wavelet domain motion estimation and compensation. The goal is, thus, to improve video compression efficiency and accuracy through the development of a new algorithm that has the inherent property of shift invariance.

### **1.3 Methodologies of the research**

This dissertation explores a new research avenue that integrates key aspects of motion vision with the mathematics of wavelet theory and the MPEG basis in estimating and compensating the matching blocks of data. The major contributions of this dissertation can be summarized as follows:

- (a) Established new mathematical derivation on the basis of the wavelet theory to improve the search method using motion estimation and motion compensation;
- (b) Successfully made use of the symmetric-extended wavelet transform to reduce the boundary distortion experienced in multiresolution search; and
- (c) Modified the decimation process of the Wavelet to make it shift invariant for motion estimation.

The video compression system is introduced in chapter 2 regarding several aspects necessary to reduce video redundancies. The mathematical theory of wavelet multiresolution analysis and its two-dimensional applications on image compression are presented in chapter 3 in order to provide the theoretical support for the wavelet domain video compression.

Since the full motion estimation places excessive computational loads on the video encoder, fast search methods are required. However, the performance of the fast search algorithm is degraded if the matching criteria function converges to the local rather than to the global minimum. The multiresolution nature of wavelet transform can realize hierarchical motion estimation and compensation by the mechanism of reaching the desired region and converging to the global minimum, which is the main subject of chapter 4.

Conventional discrete cosine transform (DCT)-based codec has limitations such as blocking artifacts at low bit rate applications. Wavelet transform is expected to have good performance, based on the results of experiments on image compression at high compression ratios. In addition, the entropy is even lower in the prediction error subframes, which generates more efficient compressions. The wavelet video codecs are implemented in chapter 5 including the encoder and the decoder.

Critical subsampling from wavelet filter convolution under perfect reconstruction constraint requires circular convolution instead of linear convolution to make the minimum half-sample decomposition possible. However, circular convolution generates border distortion because of the discontinuous periodic extension on the two ends of the signal. As more wavelet decomposition levels are involved, border distortions on the low-resolution subframes after convolution with the wavelet filters become more apparent. Proper boundary extensions on the finite length signal are necessary for subframe motion estimation without boundary discontinuities and for perfect reconstruction without coefficient expansion. Chapter 6 presents the symmetric-extended wavelet transform to

improve image compression and MEMC performance by appropriate boundary handlings.

Wavelet transform is shift variant, translation in the spatial domain does not correspond to translation of coefficients in the wavelet domain due to the decimation operations. This deficiency by taking only one another samples breaks the motion consistency between the decomposition layers, except that the shifts in the spatial domain are multiples of the sampling period. The important objective in chapter 7 is to compensate for the shift variant property of DWT in motion estimation and compensation, so that more accurate and efficient video compression can be achieved, making DWT effectively shift invariant.

Chapter 8 provides concluding remarks in the context of the new developments in terms of algorithm design, the results achieved, and the main contributions attained through this dissertation work.

## **CHAPTER 2**

### **Video Compression system and standards**

#### **2.1 Video compression standards**

Moving picture expert group (MPEG) [1] defined by International Organization of Standard (ISO) is currently the most popular compression standard family. MPEG-1 was developed to produce reasonable quality of images and sound at about 1.5Mbps on digital storage media. MPEG-2 was designed to realize higher bit rate around 10Mbps for high definition digital market. MPEG-4 [2, 3] is targeted at very low bit rate multimedia applications between 4.8~64kbps. MPEG-7 is a content representation standard for information search through a video library [4].

#### **2.2 Compression schemes**

The underlying basis of video compression is to uncorrelate the data and reduce the redundancies inside and between the frames by means of spatial, temporal, psychovisual and coding compressions, so that less data are used to represent the same amount of information.

##### **2.2.1 Psychovisual compression**

Certain information in an image is psychovisually redundant because the human visual system does not respond with equal sensitivities to all visual components. Color space transform is used to decorrelate the color data by transforming the regular RGB color space into the  $YC_bC_b$  luminance-chrominance color space by:

$$Y = 0.257R + 0.504G + 0.098B \quad (2.1)$$

$$C_r = 0.439R - 0.368G - 0.071B \quad (2.2)$$

$$C_b = -0.148R - 0.291G + 0.439B \quad (2.3)$$

where  $Y$  is the luminance channel,  $C_r$  and  $C_b$  are the chrominance channels. The significant energy from the luminance channel can be illustrated by the similar results of the original frame and an MPEG 4:2:0 format frame in which the chrominance channels are half subsampled in both horizontal and vertical resolutions, which indicates the importance of the color space transform. An example of such color space transformation is shown for visual appreciation in Figure 2.1.

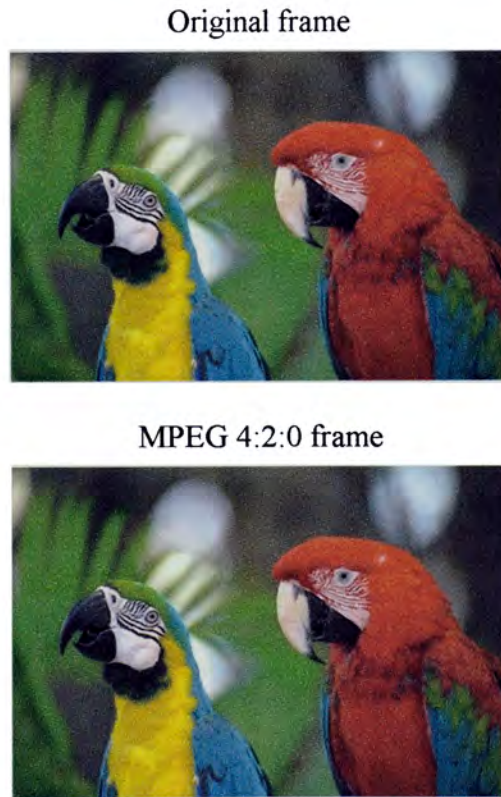


Figure 2.1. Example of the importance of color space transformation

The inverse color space transform recovers the luminance-chrominance components back into the RGB color components as:

$$R = 1.1641Y + 1.5958C_r - 0.0018C_b \quad (2.4)$$

$$G = 1.1641Y - 0.8135C_r - 0.3914C_b \quad (2.5)$$

$$B = 1.1641Y - 0.0012C_r + 2.0178C_b \quad (2.6)$$

In general, the human perception cannot involve quantitative analysis of each pixel of the image. Thus, quantization is another feasible way to eliminate psychovisual redundancy by mapping the pixel values into restricted values of data that carry the most information after dequantization. Combined with color characteristics, the quantization tables for luminance and chrominances as shown in Table 2.1 and Table 2.2, respectively, are used to allocate more bits for the coefficients that are visually more important.

Table 2.1. Quantization table for luminance channel

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	120	120	101
72	92	95	98	112	100	103	99

Table 2.2. Quantization table for chrominance channels

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

### **2.2.2 Spatial compression**

Two-dimensional pixel array can be transformed into a more uncorrelated data format to reduce the spatial redundancies. Although the domain transformation of image pixels does not produce any compression, most energy has been packed into minority of the transformed coefficients that need to be maintained. DCT-based and wavelet-based transformations are applied in the coders to convert the spatial information into frequency or scale domain [5].

#### **2.2.2.1 Discrete cosine transform**

Joint Photographic Expert Group (JPEG) perhaps is the most common color compression scheme defined by ISO/ITU-T standard (International Organization for Standardization/International Telecommunication Union-Telecommunication Standard Sector) in the late 1980s [6]. The image is divided into  $8 \times 8$  pixel blocks as the processing unit. Each block is transformed by the DCT and then quantized by a frequency dependant quantization table for luminance and chrominance components. After being scanned in zigzag order, the blocks are encoded according to the well-known Huffman tables [7].

#### **2.2.2.2 Wavelet transform**

For the last few years, the JPEG committee has been working towards the establishment of a new image compression standard, JPEG 2000 [6, 8, 9, 10, 11], which is based on discrete wavelet transform (DWT).

Wavelet transform can approximate non-stationary image signals by its flexible support basis functions. Because of the multiresolution nature, scalability and progressive transmission are feasible by wavelet transform to support variable bandwidth applications such as image archive, printer and World Wide Web. Scalability refers to the ability to transmit only a certain part of the bitstream as necessary. Signal to noise (SNR) scalability generates video at different qualities. Spatial scalability provides video at different spatial resolutions by size and format of interest. Unlike the sequentially processed DCT in a single resolution, progressive transmission allows the bit stream to be stopped in the middle with acceptable accuracy and resolution in both the encoder and decoder. This feature is important since intermediate images can be displayed hierarchically as being transferred under limited channel rate. In addition, global wavelet transform overcomes the major drawback of the DCT on image compression, called the blocking artifact [12], which is the visible discontinuity along the transformed block boundaries after quantization.

A disadvantage of wavelet transform is the shift variance property in that the transformed coefficients of a shifted signal are not simply shifted. Further investigations are thus still needed in the case of applying wavelet transform on translational motions for video compression.

### **2.2.3 Temporal compression**

Digital video processing uses multi-frame compression to deal with the significant temporal correlations between consecutive frames. The video sequence is processed by



group of pictures (GOP) as a random access unit containing a predesigned number of frames. They are basically classified as intraframe (I-frame), predicted frame (P-frame) and bi-directional interpolation frame (B-frame) [13]. The coding for I-frame is spatial in nature, and is called intraframe compression. The coding for P-frame and B-frame is on the other hand called interframe compression, because P-frame employs a reference frame from the past and B-frame uses a frame from the future to reduce redundancies between similar frames.

The central processing for interframe compression used in MPEG technique is the motion estimation and motion compensation (MEMC) [14]. Motion estimation searches for the best matching block (BMB) for each current processing block from the reference frame, and motion compensation represents the video frame by the matched blocks from the corresponding motion vectors.

#### 2.2.4 Coding compression

Entropy coding provides lossless compression on information source by the statistical frequency of occurrences. Given the occurring probability of the symbols  $x$  in a data set  $X$  as  $P = \{P(x) : x \in X\}$ , the average information called the entropy of the source is:

$$H(X) = -\sum_{x \in X} P(x) \log_2 P(x) \quad (\text{bits / symbol}) \quad (2.7)$$

If the source is an image, the average number of bits used per pixel (bpp) is called the bit rate. Variable-length coding achieves data compression by assigning fewer bits to the more probable pixels. An ideal coder is capable to encode each symbol using fractional

bits, while the codelength for Huffman coding needs to be integer to simplify the lookup table.

The entropy coding is fully invertible at the decoding end. Fundamental transform coding principles are explained in [15]. Codings on the transformed domain are modified for DCT [16, 17], and combined with wavelet structure for very low bit rate video codecs [18].

### 2.3 Fidelity criteria

The compression ratio of a true color frame (8 bits for each color) is:

$$\text{Compression Ratio} = \frac{24}{\text{bit rate}} \quad (2.8)$$

where

$$\text{bit rate} = \frac{B}{N_1 \cdot N_2} \quad (bpp) \quad (2.9)$$

and  $B$  is the number of bits of the size  $N_1 \times N_2$  frame after compression. The compression performance can be evaluated by two methods: subjective and objective fidelity criteria [19]. Peak signal to noise ratio (PSNR) is an objective measurement that estimates the accuracy of the reconstructed frame in terms of the luminance components:

$$PSNR = 20 * \lg \frac{2^b - 1}{\sqrt{\frac{1}{N_1 N_2} \sum_{N_1, N_2} [f(i, j) - f_r(i, j)]^2}} \quad (dB) \quad (2.10)$$

$b$  is the bit resolution of the original frame, usually 8 bits/pixel.  $f$  is the original image frame and  $f_r$  is the reconstructed frame.

Other major concerns about compression performance are complexity and functionality. Computational loads are evaluated mathematically for complexity regardless of optimization, implementation method and target applications.

## **2.4 Video compression system**

A video codec (coder/decoder or compression/decompression) is a system implemented in hardware or software to compress and decompress the video sequence.

Video frames captured at the source are compressed into bit stream by the video encoder. The basic structure of video encoder is shown in Figure 2.2. A frame is buffered as a reference for the compression of the current processing frame. After applying color space transformation (CST) and mean removal on the luminance and chrominance components of both frames, motion estimation is performed block by block. If a best matching block is found, the current block after motion compensation is represented by a motion vector and a prediction error block. All the motion vectors and the quantized prediction errors go for the entropy coding. The compressed data as well as the side information including the frame size, filters used, transformation information and the Huffman table are sent to the decoder.

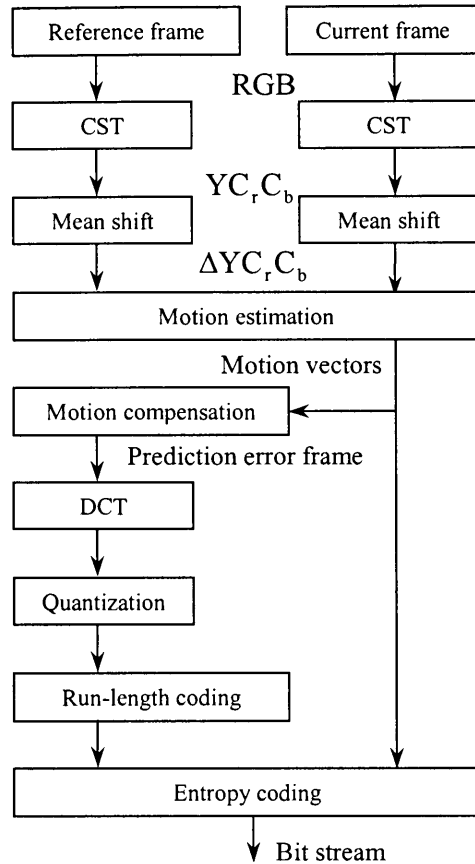


Figure 2.2. Video encoding system

The decoder shown in Figure 2.3 recovers the stream data back into video frames. The prediction error data, motion vectors and Huffman table are extracted by entropy decoding. The current frame is reconstructed by the motion vectors and the prediction error blocks obtained after dequantization and inverse DCT (IDCT). Inverse color space transform (ICST) is performed at last to convert the frame from  $YC_rC_b$  back into the regular RGB format.

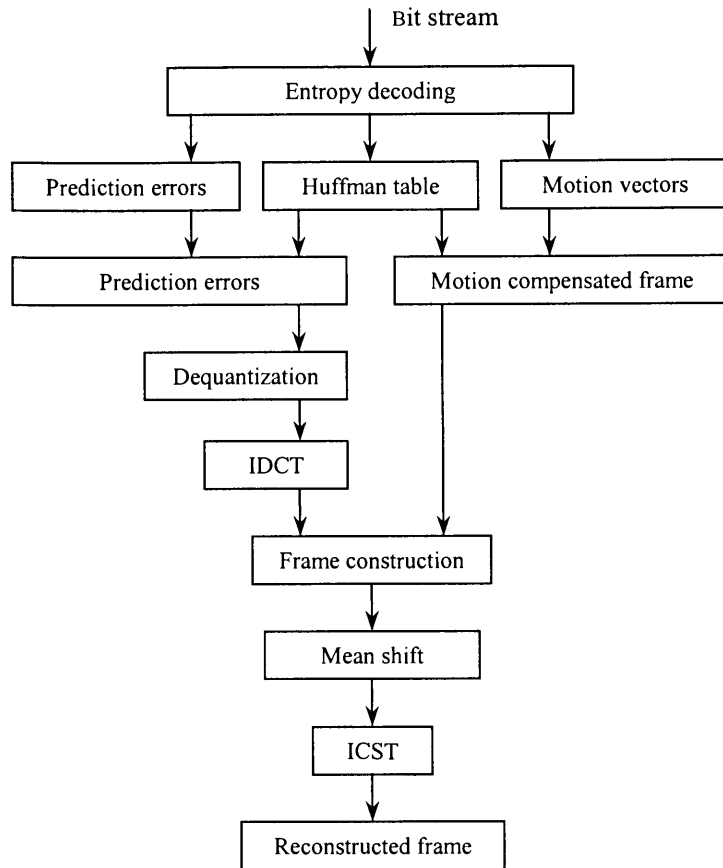


Figure 2.3. Video decoding system

## CHAPTER 3

### Wavelet Multiresolution Analysis and Compression

#### 3.1 A retrospective on wavelet transform

The Fourier transform of a finite-energy signal yields the signal spectrum  $F(w)$  by the inner product of the signal  $f(t)$  with sinusoidal basis functions over the whole real time duration:

$$F(w) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (3.1)$$

This definition indicates that any part of the function in time domain will spread out on the entire frequency domain due to the global support of the basis function. It gives perfect representation in the frequency domain, but no time or spatial domain information is provided. When local frequency content of a signal is of interest, frequency analysis using Fourier transform becomes inadequate. Time-frequency analysis [20, 21] is therefore a more suitable approach to analyze the local characteristics of non-stationary signals. By shifting a window function  $\varphi(t)$  on  $f(t)$  around  $t = b$  before taking Fourier transform, short-time Fourier transform (STFT) can be specified on the two-dimensional time-frequency plane as in the following relation:

$$G(b, w) = \int_{-\infty}^{\infty} f(t)\overline{\varphi(t-b)}e^{-j\omega t} dt \quad (3.2)$$

When changing the location parameter  $b$ , frequency information at different time intervals of the signal can be extracted over the nonzero duration of the window function. Although the frequency of the sinusoidal wave varies inside the envelope of the window function, the time-frequency resolution of the analysis is fixed at all locations since a

single window is used for all frequencies [22]. When a predetermined resolution may not be appropriate for a signal with wide range of dominant frequencies, multiresolution analysis is needed to improve accuracy and efficiency in processing the signal.

Heisenberg's uncertainty principle [23] states that the time-bandwidth product of the base function is lower bounded; that is to say, the window width of the base function is inversely proportional to the spectrum window width. Thus, localization cannot be good in the time domain and the frequency domain simultaneously. However, a signal under analysis is generally composed of high frequency components on short duration and low frequency components on long duration. Therefore, wavelet window function with support adaptability is used to overcome the resolution problem of STFT [24]. When a contracted, high frequency version of the wavelet function is applied, temporal analysis is performed to detect low frequency contents of the signal; on the other hand, when a dilated, low-frequency version of the same wavelet is applied, frequency analysis is performed to locate high frequency contents of the signal.

Continuous wavelet transform (CWT) [25, 26, 27] convolves the signal  $f(t)$  with wavelet base function  $\psi_{b,a}(t)$  obtained from the mother wavelet function  $\psi(t)$  via translations and dilations in scale  $a$  and position  $b$  :

$$W(b, a) = \int_{-\infty}^{\infty} f(t) \overline{\psi_{b,a}(t)} dt \quad (3.3)$$

where

$$\psi_{b,a}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (a > 0) \quad (3.4)$$

$W(b, a)$  is defined as the wavelet coefficient.

Unlike the sinusoidal base function predefined in the Fourier transform,  $\psi(t)$  is not specified. Instead of filling the window function with different frequencies of oscillations in STFT, wavelet transform keeps the number of oscillations constant and modifies the width of the self-similar function to change the frequency. When the scale factor  $a$  increases, the support of the wavelet function increases, the time resolution decreases, while the frequency support decreases and shifts to the low frequency band with increased frequency resolution. On the contrary, when  $a$  decreases, the support of the wavelet function decreases, the frequency support shifts to the high frequency band with decreased frequency resolution. This is the so-called constant relative bandwidth analysis since the frequency support width is proportional to its center. In this way, higher frequencies are better resolved in time, and lower frequencies are better resolved in frequency.

In continuous wavelet transform, in addition to the time domain function, dilation as well as translation parameters are changed continuously. These continuously varying parameters in both time and scale domain produces redundancies in signal analysis and synthesis. This problem can be resolved by critically subsampling  $a$  and  $b$  on a dyadic grid of the time-scale plane as:

$$\begin{aligned} a &= 2^{-s} \\ b &= k \cdot 2^{-s} \quad (s, k \in \mathbb{Z}) \end{aligned} \tag{3.5}$$



Thus, the wavelet function is redefined as:

$$\psi_{s,k}(t) = 2^{\frac{s}{2}} \psi(2^s t - k) \quad (3.6)$$

by the discretized parameters of integer scale  $s$  and shift  $k$ , and the wavelet transform is turned into wavelet series expression as:

$$w_{s,k} = \int_{-\infty}^{\infty} f(t) \overline{\psi_{s,k}(t)} dt = \langle f(t), \psi_{s,k}(t) \rangle \quad (3.7)$$

Although CWT allows us to use very general wavelet functions to satisfy the admissibility condition for function reconstruction, more restrictive conditions such as orthogonality are required after the discretization. If  $\{\psi_{s,k}(t)\}$  forms an orthonormal basis of the finite energy functional space  $L^2(R)$  [28], as:

$$\langle \psi_{s,k}, \psi_{l,m} \rangle = \delta_{s,l} \delta_{k,m} \quad (3.8)$$

then any function  $f(t) \in L^2(R)$  can be synthesized by the linear combinations of these wavelet bases in different sizes via dilations and in different positions via translations:

$$f(t) = \sum_s \sum_k w_{s,k} \psi_{s,k}(t) \quad (3.9)$$

Otherwise, we need the dual wavelet function set  $\{\tilde{\psi}_{s,k}(t)\}$ :

$$\langle \psi_{s,k}, \tilde{\psi}_{l,m} \rangle = \delta_{s,l} \delta_{k,m} \quad (3.10)$$

$$w_{s,k} = \langle f(t), \tilde{\psi}_{s,k}(t) \rangle \quad (3.11)$$

Thus, a square integratable signal can be decomposed and reconstructed partially as approximate representation or fully as perfect reconstruction from these elementary building blocks of wavelet bases at different scales.

### 3.2 Multiresolution analysis

Multiresolution analysis (MRA) [29, 30, 31] is a framework used to approximate a function  $f(t) \in L^2(R)$  by a sequence of closed subspaces  $A_i$  ( $i \in Z$ ) with the following properties:

1. nesting:  $A_{i-1} \subset A_i$
  2. closure:  $\overline{\bigcup_{i \in Z} A_i} = L^2(R)$
  3. shrinking:  $\bigcap_{i \in Z} A_i = \{0\}$
  4. scaling:  $f(t) \in A_i \Leftrightarrow f(2t) \in A_{i+1}$
  5. shifting:  $f(t) \in A_i \Leftrightarrow f(t + 2^{-i}k) \in A_i$  ( $k \in Z$ )
- (3.12)

Since the union of subspace set  $\{A_i\}$  completes the function space  $L^2(R)$ , successive approximation of function  $f(t)$  can be constructed from these nested subspaces until the infinite resolution  $L^2(R)$ ,

$$\{0\} \leftarrow \cdots \subset A_{-1} \subset A_0 \subset A_1 \subset \cdots \rightarrow \{L^2\} \quad (3.13)$$

The difference between two successive approximation subspaces  $A_{s+1}$  and  $A_s$  is the detail subspace  $W_s$  expressed by wavelet functions at scale  $s$ , which is complementary to the approximation subspace  $A_s$  as:

$$A_s \cap W_s = \{0\} \quad (3.14)$$

$$A_{s+1} = A_s \oplus W_s \quad (3.15)$$

$\oplus$  is the direct-sum decomposition operator. It indicates that the higher resolution representation can be completely expressed by the lower scale approximation and the

wavelet difference from the same scale. Assuming the basis set of  $A_s$  is the scaling function  $\{\phi_{s,k}(t) = 2^{\frac{s}{2}} \phi(2^s t - k)\}$ , defined by the shift orthogonal property that translations at this scale are found to be mutually orthogonal, satisfying the condition:

$$\langle \phi_{s,m}(t), \phi_{s,n}(t) \rangle = \delta_{m,n} \quad (3.16)$$

In this case, the inner product of the base function with itself is unity, while the inner products with its shifts are zero. And the basis function set of  $W_s$  is the wavelet function

$\{\psi_{s,k}(t) = 2^{\frac{s}{2}} \psi(2^s t - k)\}$  by the condition that

$$\langle \psi_{s,m}(t), \psi_{s,n}(t) \rangle = \delta_{m,n} \quad (3.17)$$

Therefore, subspace  $A_0$  is spanned by scaling function  $\phi(t - k)$  with integer translations;  $W_0$  is spanned by wavelet function  $\psi(t - k)$ ; subspace  $A_1$  is spanned by  $\phi(2t - k)$ ;  $W_1$  is spanned by  $\psi(2t - k)$  and so on. Since  $\phi(t) \in A_0 \subset A_1$ ,  $\phi(t)$  can be expressed by a weighted sum of the scaling function  $\phi(2t)$  from  $A_1$  with shifts,

$$\phi(t) = \sum_k g_0[k] \phi(2t - k) \quad (3.18)$$

The same nested relation can be applied to the wavelet function in that  $\psi(t) \in W_0 \subset A_1$ , then

$$\psi(t) = \sum_k g_1[k] \phi(2t - k) \quad (3.19)$$

$g_0$  and  $g_1$  are the scaling and wavelet function coefficient sequences. Equations (3.18) and (3.19) are called the two-scale dilation equations. It states that the scaling function and wavelet function at a certain scale can be expressed in terms of the translated scaling

functions from the next higher scale. Obviously, the connections between any two successive scales are fixed for a specific multiresolution system. Using equation (3.18), and taking the Fourier transform on both sides, yields:

$$\hat{\phi}(w) = \frac{1}{2} \sum_k g_0[k] e^{-jk \frac{w}{2}} \hat{\phi}\left(\frac{w}{2}\right) \quad (3.20)$$

and based on the approximation property of the scaling function,  $g_0$  sequence shows a lowpass filter characteristic on the DC frequency:

$$G_0(w) |_{w=0} = \sum_k g_0[k] = 2 \quad (3.21)$$

The nonzero coefficient of  $g_0$  represents the weight of the corresponding shift. The finite length of the sequence defines a compact support base function set from an analog point of view, and a finite impulse response (FIR) filter [32] from a digital point of view.

### 3.3 Fast wavelet transform

The complementary condition between  $A_s$  and  $W_s$  in (3.14) can be realized by orthogonal decomposition  $A_s \perp W_s$  in that any function from the approximation subspace will be orthogonal to those from the wavelet subspace at the same scale. It can be simply realized by the orthogonality between the base functions:

$$\langle \phi(t), \psi(t-l) \rangle = 0 \quad (3.22)$$

assuming  $f_{s+1}(t) \in A_{s+1}$ ,

$$f_{s+1}(t) = \sum_l a_{s+1,l} \phi_{s+1,l}(t) = 2^{\frac{s+1}{2}} \sum_l a_{s+1,l} \phi(2^{s+1}t - l) \quad (3.23)$$

Changing the basis due to the established relations in (3.16), (3.17) and (3.22), and

applying the two-scale equations of

$$\phi(2^s t - l) = \sum_k g_0[k] \phi(2^{s+1} t - 2l - k) \quad (3.24)$$

$$\psi(2^s t - l) = \sum_k g_1[k] \phi(2^{s+1} t - 2l - k) \quad (3.25)$$

so that the higher resolution base function is projected into the scaling and wavelet subspaces in lower resolution, the following equation is attained:

$$\begin{aligned} \phi_{s+1,l}(t) &= \sum_k \langle \phi_{s+1,l}(t), \phi_{s,k}(t) \rangle \phi_{s,k}(t) + \sum_k \langle \phi_{s+1,l}(t), \psi_{s,k}(t) \rangle \psi_{s,k}(t) \\ &= \sum_k \frac{1}{\sqrt{2}} g_0[l-2k] \phi_{s,k}(t) + \sum_k \frac{1}{\sqrt{2}} g_1[l-2k] \psi_{s,k}(t) \end{aligned} \quad (3.26)$$

Constant  $\frac{1}{\sqrt{2}}$  could be dropped from the sequences for simplicity,

$$\frac{1}{\sqrt{2}} g_0 \rightarrow g_0 \quad (3.27)$$

$$\frac{1}{\sqrt{2}} g_1 \rightarrow g_1 \quad (3.28)$$

Equation (3.23) can be written as:

$$f_{s+1}(t) = \sum_k \left\{ \sum_l g_0[l-2k] a_{s+1,l} \phi_{s,k}(t) + \sum_l g_1[l-2k] a_{s+1,l} \psi_{s,k}(t) \right\} \quad (3.29)$$

On the other hand, with  $f_s(t) \in A_s$  and  $d_s(t) \in W_s$ ,

$$f_s(t) = \sum_k a_{s,k} \phi_{s,k}(t) \quad (3.30)$$

$$d_s(t) = \sum_k w_{s,k} \psi_{s,k}(t) \quad (3.31)$$

and from the subspace decomposition in (3.15), we have

$$f_{s+1}(t) = f_s(t) + d_s(t) \quad (3.32)$$

Substitute in (3.32) equations (3.30) and (3.31), then compare the correspondent coefficients of  $\phi_{s,k}(t)$  and  $\psi_{s,k}(t)$  with (3.29), it shows the convolution relationships of the scaling and wavelet coefficients between two successive scales:

$$a_{s,k} = \sum_l g_0[l-2k]a_{s+1,l} \quad (3.33)$$

$$w_{s,k} = \sum_l g_1[l-2k]a_{s+1,l} \quad (3.34)$$

Observe that if filter  $h_0$  and  $h_1$  have the impulse response as time inverse of  $g_0$  and  $g_1$  sequences as:

$$h_0[k] = g_0[m-k] \quad (3.35)$$

$$h_1[k] = g_1[m-k] \quad (3.36)$$

$m$  is used to make  $h_0$  and  $h_1$  causal filters, and their z transformations in the relations of

$$H_0(z) = z^{-m}G_0(z^{-1}) \quad (3.37)$$

$$H_1(z) = z^{-m}G_1(z^{-1}) \quad (3.38)$$

(3.33) and (3.34) become

$$a_{s,k} = \sum_l h_0[2k-l]a_{s+1,l} = \{h_0 * a_{s+1}\}_{\downarrow 2} \quad (3.39)$$

$$w_{s,k} = \sum_l h_1[2k-l]a_{s+1,l} = \{h_1 * a_{s+1}\}_{\downarrow 2} \quad (3.40)$$

This is the discrete wavelet transform (DWT) or fast wavelet transform (FWT). The

computation is performed as a filter convolution followed by half downsampling ( $\downarrow 2$ ) shown as block 1 and 2 in Figure 3.1.

Applying the same subspace expansion and combination method for reconstruction as above, the inverse discrete wavelet transform (IDWT) is attained as follows:

$$\begin{aligned}
\sum_l a_{s+1,l} \phi_{s+1,l}(t) &= \sum_l a_{s+1,l} 2^{\frac{s+1}{2}} \phi(2^{s+1}t - l) \\
&= \sum_k a_{s,k} \phi_{s,k}(t) + \sum_k w_{s,k} \psi_{s,k}(t) \\
&= \sum_k a_{s,k} 2^{\frac{s}{2}} \sum_m g_0[m] \phi(2^{s+1}t - 2k - m) \\
&\quad + \sum_k w_{s,k} 2^{\frac{s}{2}} \sum_m g_1[m] \phi(2^{s+1}t - 2k - m) \\
&= \sum_k a_{s,k} 2^{\frac{s}{2}} \sum_l g_0[l - 2k] \phi(2^{s+1}t - l) + \sum_k w_{s,k} 2^{\frac{s}{2}} \sum_l g_1[l - 2k] \phi(2^{s+1}t - l)
\end{aligned}$$

therefore,

$$\begin{aligned}
a_{s+1,l} &= \frac{1}{\sqrt{2}} \sum_k \{g_0[l - 2k]a_{s,k} + g_1[l - 2k]w_{s,k}\} \\
&= g_0 * \{a_s\}_{\uparrow 2} + g_1 * \{w_s\}_{\uparrow 2}
\end{aligned} \tag{3.41}$$

The scaling and wavelet coefficients are upsampled by 2 ( $\uparrow 2$ ) before filter convolution as shown in block 3 of Figure 3.1.

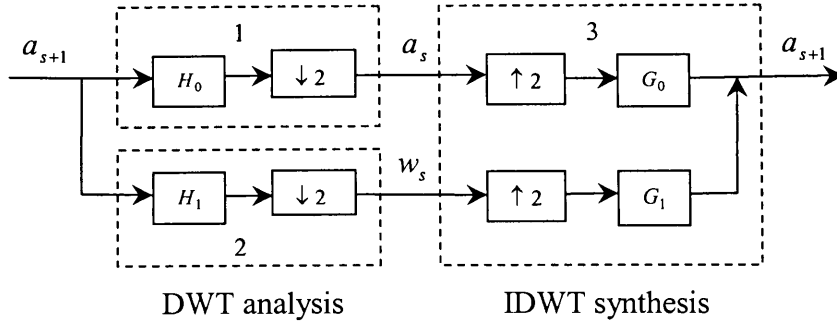


Figure 3.1. DWT and IDWT computational structure

The relations of  $a_{s,k}$ ,  $w_{s,k}$  with  $a_{s+1,k}$  can fast calculate the CWT coefficients of each level once the function is projected into the approximation subspace [33]. It illustrates that the lower resolution coefficients of the scaling and wavelet function can be obtained from the higher resolution coefficients.

As far as the computational structure is concerned, it is a two-channel wavelet filter bank to process digital signals. The decomposition part of the filter bank performs DWT for signal analysis, and the reconstruction part performs IDWT for signal synthesis. When the input is received, the full bandwidth of the signal is split into low and high spectrum subbands by the lowpass filter  $h_0$  and highpass filter  $h_1$ . After being downsampled by keeping only every other sample in each branch, these decimated outputs constitute the approximation and detail signals in octave subbands. In reconstruction, the approximation and wavelet output from the two branches are upsampled by inserting zeros between every other sample, and then passed through the lowpass and highpass synthesis filters  $g_0$  and  $g_1$ . The sum is the perfect reconstruction of the original input to the filter bank.



Hierarchical decomposition can be involved in multi-stage wavelet filter bank in a pyramid structure as shown in Figure 3.2 for one-dimensional signal decomposition. It expresses that a discrete signal is separated into multiresolution coarse approximation and the added details.

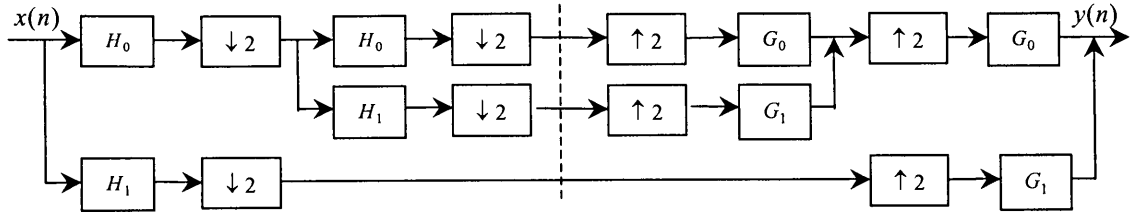


Figure 3.2. Multilevel wavelet decomposition and reconstruction for one-dimensional signal

For two-dimensional image signals, wavelet decomposition performs a multiresolution representation on an image by a series of subbands with the most important information in the approximation and better detail information in the high frequency subbands.

### 3.4 Orthogonal wavelet filters

By the analysis part of the filter bank, half decimation of the input signal  $x(n)$  results in the output signal

$$y(n) = x(2n) \quad (3.42)$$

and the corresponding Z transform [20] is:

$$Y(Z) = \frac{1}{2} [X(Z^{\frac{1}{2}} + X(-Z^{\frac{1}{2}})] \quad (3.43)$$

In the synthesis part, the output from double upsampling by interpolating zeros is

$$y(n) = \begin{cases} x(\frac{n}{2}) & \text{for } n = 2k \ (k \in \mathbb{Z}) \\ 0 & \text{otherwise} \end{cases} \quad (3.44)$$

with the spectrum given by:

$$Y(Z) = X(Z^2) \quad (3.45)$$

Thus, the output transfer function of the filter bank by the above operations becomes

$$\begin{aligned} Y(Z) = & \frac{1}{2} X(Z) [H_0(Z)G_0(Z) + H_1(Z)G_1(Z)] \\ & + \frac{1}{2} X(-Z) [H_0(-Z)G_0(Z) + H_1(-Z)G_1(Z)] \end{aligned} \quad (3.46)$$

For perfect reconstruction, with the need to eliminate aliasing effect from  $X(-Z)$  caused by decimation, the following relations are assumed:

$$G_0(Z) = H_1(-Z) \quad (3.47)$$

$$G_1(Z) = -H_0(-Z) \quad (3.48)$$

with the corresponding impulse responses given by:

$$h_1[k] = (-1)^k g_0[k] \quad (3.49)$$

$$h_0[k] = -(-1)^k g_1[k] \quad (3.50)$$

From equations (3.35), (3.36), (3.49) and (3.50), it requires that  $m$  being an odd integer and

$$G_1(Z) = z^{-m} G_0(-Z^{-1}) \quad (3.51)$$

which is equivalent to

$$g_1[k] = -(-1)^k g_0[m-k] \quad (3.52)$$

Based on the condition of real coefficient filters,

$$\begin{aligned}
\langle \phi(t), \phi(t-l) \rangle &= \delta_{0,l} \Leftrightarrow \sum_k g_0[k]g_0[k-2l] = \delta_{0,l} \\
&\Leftrightarrow G_0(Z)G_0(Z^{-1}) + G_0(-Z)G_0(-Z^{-1}) = 2
\end{aligned} \tag{3.53}$$

The transfer function of the filter bank is  $Y(Z) = z^{-m}X(Z)$ , which is a perfect reconstruction with only  $m$  samples delay. For practical signal processing, compact support FIR filters are desired for low-complexity implementation. All four filters will have the same even length by the double-shift orthogonality condition in (3.53) and from the impulse response results obtained above. Assume the filters' length is  $L$ . For a least delay of the causal FIR filters,

$$m = L - 1 \tag{3.54}$$

The solution of the filter bank coefficients under perfect reconstruction constraint is thus determined as follows:

$$\begin{cases} \sum_k g_0[k]g_0[k-2l] = \delta_{0,l} \\ g_1[k] = -(-1)^k g_0(L-1-k) \\ h_0[k] = g_0[L-1-k] \\ h_1[k] = g_1[L-1-k] \end{cases} \tag{3.55}$$

Once  $g_0$  is defined, highpass filter  $g_1$  and other filters in the bank are defined. Indeed, the operator  $(-1)^n$  generates the so-called quadrature mirror filter (QMF) in that the lowpass and highpass filters are symmetrical around the point  $\frac{\pi}{2}$  in their magnitude responses. The Daubechies family defines  $g_0$  filter under the regularity constraint that the iterated filter coefficients will converges to a continuous function by a maximum number of zeros of the spectrum  $G_0(w)$  to the  $r^{th}$  derivatives at  $w = \pi$  [33, 34, 35, 36],

$$G_0^r(w)|_{w=\pi} = 0 \Leftrightarrow \sum_k k^r (-1)^k g_0[k] = 0 \quad (3.56)$$

The coefficient matrix calculation of these simultaneous equations (3.21), (3.27), (3.28), (3.53) and (3.56) solves the discrete filters. Daubechies (dB) filters in variable lengths are shown in Figure 3.3 which are the same as those obtained from spectrum factorization method [23, 37].

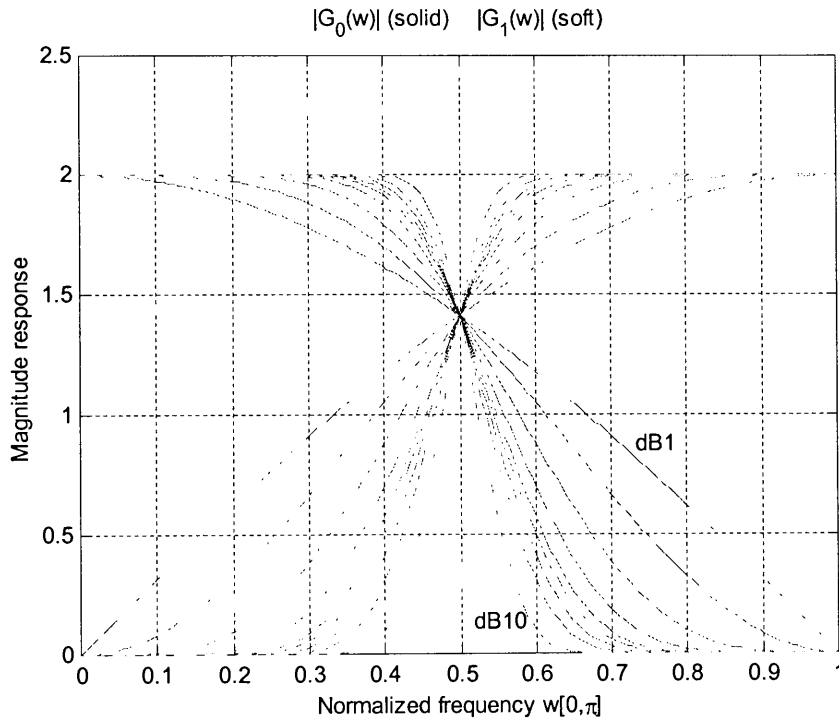


Figure 3.3. Daubechies wavelet filter family

By the fact that the wavelet is a window function in time domain, it's also a window function in frequency domain as a bandpass filter by the difference between two successive lowbands. From the Fourier transform pair  $\psi(2t) \leftrightarrow \frac{1}{2}\hat{\psi}(\frac{w}{2})$ , when the scale goes down, the corresponding spectrum will narrow down by half; when the scale goes

up, the corresponding spectrum will shift to the double passband. It develops an octave filter bank with lowpass and highpass filter branches with respect to the signal before decomposition.

From the magnitude response of the filters, DWT provides perfect signal reconstruction although the wavelet filters are not ideal halfband filters. When filter length is 2, dB1 is a Haar filter. There's no closed-form analytic formula for the compactly supported scaling and wavelet functions except Haar filter, but they can be computed with arbitrary high precision using cascade algorithm [23, 37, 38]. It's a refinement scheme of iteratively solving the dilation equations until the convergence is reached.

### **3.5 Biorthogonal wavelet filters**

When the orthogonality constraint is released, more degrees of freedom can be added into the redundant biorthogonal wavelet system. Compact support biorthogonal perfect reconstruction FIR filters can have linear phase property while orthogonal systems can't, except for Haar wavelet filter [39, 40]. A symmetric filter that has linear phase response is of interest in very low bit rate image coding.

#### **3.5.1 Biorthogonal wavelet system**

Biorthogonal relations are required for complementary subspace decomposition from (3.14) in multiresolution analysis. In a biorthogonal system, there are two sets of basis functions called dual spaces. Each basis is not orthogonal to its own but to its dual. One basis set is used for analysis and the other set is used for synthesis. Thus, we need space

$\tilde{A}_s$  as dual of  $A_s$  and  $\tilde{W}_s$  as dual of  $W_s$  when  $A_s$  is not orthogonal to  $W_s$ . The dual approximation subspace  $\tilde{A}_s$  generates the multiresolution analysis of the functional space as well,

$$\{0\} \leftarrow \cdots \subset \tilde{A}_{-1} \subset \tilde{A}_0 \subset \tilde{A}_1 \subset \cdots \rightarrow \{L^2\} \quad (3.57)$$

The scaling function set  $\{\phi_{s,k}(t)\}$  that span approximation subspace  $A_s$  is the synthesis scaling functions; and its dual, called the analysis scaling function set

$\{\tilde{\phi}_{s,k}(t) = 2^{\frac{s}{2}} \tilde{\phi}(2^s t - k)\}$  span dual approximation subspace  $\tilde{A}_s$ ,

$$\langle \phi_{s,m}(t), \tilde{\phi}_{s,n}(t) \rangle = \delta_{m,n} \quad (3.58)$$

Synthesis wavelet functions  $\{\psi_{s,k}(t)\}$  span detail subspace  $W_s$ ; and its dual, the analysis

wavelet functions  $\{\tilde{\psi}_{s,k}(t) = 2^{\frac{s}{2}} \tilde{\psi}(2^s t - k)\}$  span dual detail subspace  $\tilde{W}_s$ ,

$$\langle \psi_{s,m}(t), \tilde{\psi}_{s,n}(t) \rangle = \delta_{m,n} \quad (3.59)$$

The two-scale dual dilation equations are described by the dual coefficients,

$$\tilde{\phi}(t) = \sum_m \tilde{g}_0[m] \tilde{\phi}(2t - m) \quad (3.60)$$

$$\tilde{\psi}(t) = \sum_m \tilde{g}_1[m] \tilde{\phi}(2t - m) \quad (3.61)$$

where  $W_s$  is the complementary subspace for  $A_s$  in  $A_{s+1}$ ;  $\tilde{W}_s$  is the complementary subspace for  $\tilde{A}_s$  in  $\tilde{A}_{s+1}$ ,

$$\tilde{A}_{s+1} = \tilde{A}_s + \tilde{W}_s \quad (3.62)$$

although they're not orthogonal complement, their duals are orthogonal,

$$A_s \perp \tilde{W}_s \quad (3.63)$$

$$\tilde{A}_s \perp W_s \quad (3.64)$$

so the base functions belong to these subspaces are mutually orthogonal by their translations,

$$\langle \phi_{s,m}(t), \tilde{\psi}_{s,n}(t) \rangle = 0 \quad (3.65)$$

$$\langle \tilde{\phi}_{s,m}(t), \psi_{s,n}(t) \rangle = 0 \quad (3.66)$$

From (3.58), (3.59), (3.65) and (3.66), biorthogonal wavelet filters satisfy double-shift orthogonality constraints:

$$\langle \phi_{s,m}(t), \tilde{\phi}_{s,n}(t) \rangle = \delta_{m,n} \Leftrightarrow \sum_k g_0[k] \tilde{g}_0[k-2l] = \delta_{0,l} \quad (3.67)$$

$$\langle \psi_{s,m}(t), \tilde{\psi}_{s,n}(t) \rangle = \delta_{m,n} \Leftrightarrow \sum_k g_1[k] \tilde{g}_1[k-2l] = \delta_{0,l} \quad (3.68)$$

$$\langle \phi_{s,m}(t), \tilde{\psi}_{s,n}(t) \rangle = 0 \Leftrightarrow \sum_k g_0[k] \tilde{g}_1[k-2l] = 0 \quad (3.69)$$

$$\langle \tilde{\phi}_{s,m}(t), \psi_{s,n}(t) \rangle = 0 \Leftrightarrow \sum_k \tilde{g}_0[k] g_1[k-2l] = 0 \quad (3.70)$$

That is, the filter impulse response is orthogonal to the even translates of its dual. Refer to (3.23) for function approximation, based on the two-scale dilation equation on the current scale,

$$\tilde{\phi}(2^s t - k) = \sum_m \tilde{g}_0[m] \tilde{\phi}(2^{s+1} t - 2k - m) \quad (3.71)$$

$$\tilde{\psi}(2^s t - k) = \sum_m \tilde{g}_1[m] \tilde{\psi}(2^{s+1} t - 2k - m) \quad (3.72)$$

the decomposition is realized by changing the basis from the duals and drop the  $\frac{1}{\sqrt{2}}$

constant, yielding:

$$\begin{aligned}
\phi_{s+1,l}(t) &= \sum_k \langle \phi_{s+1,l}(t), \tilde{\phi}_{s,k}(t) \rangle \phi_{s,k}(t) + \sum_k \langle \phi_{s+1,l}(t), \tilde{\psi}_{s,k}(t) \rangle \psi_{s,k}(t) \\
&= \sum_k \langle \phi_{s+1,l}(t), \sum_m \tilde{g}_0[m] \tilde{\phi}(2^{s+1}t - 2k - m) \rangle \phi_{s,k}(t) \\
&\quad + \sum_k \langle \phi_{s+1,l}(t), \sum_m \tilde{g}_1[m] \tilde{\phi}(2^{s+1}t - 2k - m) \rangle \psi_{s,k}(t) \\
&= \sum_k \tilde{g}_0[l - 2k] \phi_{s,k}(t) + \sum_k \tilde{g}_1[l - 2k] \psi_{s,k}(t)
\end{aligned} \tag{3.73}$$

By comparing the coefficients from (3.30) and (3.31), it turns out that

$$a_{s,k} = \sum_l \tilde{g}_0[l - 2k] a_{s+1,l} \tag{3.74}$$

$$w_{s,k} = \sum_l \tilde{g}_1[l - 2k] a_{s+1,l} \tag{3.75}$$

If  $\tilde{g}_0$  and  $\tilde{g}_1$  are time reversed, set

$$h_0[k] = \tilde{g}_0[m - k] \tag{3.76}$$

$$h_1[k] = \tilde{g}_1[m - k] \tag{3.77}$$

and the spectrum relationships are

$$H_0(z) = z^{-m} \tilde{G}_0(z^{-1}) \tag{3.78}$$

$$H_1(z) = z^{-m} \tilde{G}_1(z^{-1}) \tag{3.79}$$

then the biorthogonal wavelet coefficients actually represent a filter convolution followed by the half downsampling operation.

$$a_{s,k} = \sum_l h_0[2k - l] a_{s+1,l} = \{h_0 * a_{s+1}\}_{\downarrow 2}$$



$$w_{s,k} = \sum_l h_l [2k-l] a_{s+1,l} = \{h_1 * a_{s+1}\}_{\downarrow 2}$$

For reconstruction,

$$\begin{aligned} a_{s+1,l} &= \frac{1}{\sqrt{2}} \sum_k \{g_0[l-2k] a_{s,k} + g_1[l-2k] w_{s,k}\} \\ &= g_0 * \{a_s\}_{\uparrow 2} + g_1 * \{w_s\}_{\uparrow 2} \end{aligned} \quad (3.80)$$

As such, we can make use of the same filter bank structure from orthogonal system in figure 3.1 for the biorthogonal filter bank although the relationships between synthesis and analysis filters are different now. From the z transform of the wavelet filter bank structure, the perfect reconstruction constraint demands that

$$\begin{cases} G_0(Z) \tilde{G}_0(Z^{-1}) + G_1(Z) \tilde{G}_1(Z^{-1}) = 2 \\ G_0(Z) \tilde{G}_0(-Z^{-1}) + G_1(Z) \tilde{G}_1(-Z^{-1}) = 0 \end{cases} \quad (3.81)$$

Separate the filters from their duals, yields:

$$\tilde{G}_0(Z) = \frac{-2G_1(-Z^{-1})}{G_1(Z^{-1})G_0(-Z^{-1}) - G_0(Z^{-1})G_1(-Z^{-1})} \quad (3.82)$$

$$\tilde{G}_1(Z) = \frac{2G_0(-Z^{-1})}{G_1(Z^{-1})G_0(-Z^{-1}) - G_0(Z^{-1})G_1(-Z^{-1})} \quad (3.83)$$

and by letting

$$G(Z) = G_1(Z^{-1})G_0(-Z^{-1}) - G_0(Z^{-1})G_1(-Z^{-1}) \quad (3.84)$$

To make  $g_0$ ,  $g_1$ ,  $h_0$  and  $h_1$  all FIR filters, both  $G(Z)$  and  $\frac{1}{G(Z)}$  must have finite terms.

Thus,  $G(Z)$  has one single term,

$$G(Z) = az^{m_0} \quad (a \in R) \quad (3.85)$$

Since  $G(Z) = -G(-Z)$ ,  $m_0$  is an odd integer. Let  $a = -2$ ,

$$\tilde{G}_1(Z) = -z^{-m_0} G_0(-Z^{-1}) \quad (3.86)$$

the impulse responses of  $\tilde{g}_1$  and  $h_1$  are:

$$\tilde{g}_1[k] = (-1)^k g_0[m_0 - k] \quad (3.87)$$

$$h_1[k] = (-1)^{m-k} g_0[m - m_0 + k]$$

by setting  $m_0 = m$ , then

$$h_1[k] = -(-1)^k g_0[k] \quad (3.88)$$

From (3.82),

$$\tilde{G}_0(Z) = z^{-m} G_1(-Z^{-1}) \quad (3.89)$$

the impulse responses of  $\tilde{g}_0$  and  $h_0$  are thus

$$\tilde{g}_0[k] = -(-1)^k g_1[m - k] \quad (3.90)$$

$$h_0[k] = (-1)^k g_1[k] \quad (3.91)$$

The solution to the casual biorthogonal FIR filter bank is

$$\left\{ \begin{array}{l} \sum_k g_0[k] \tilde{g}_0[k - 2l] = \delta_{0,l} \\ g_1[k] = (-1)^k \tilde{g}_0[m - k] \\ h_0[k] = \tilde{g}_0[m - k] \\ h_1[k] = -(-1)^k g_0[k] \end{array} \right. \quad (3.92)$$

with  $m$  being the smallest odd integer for causal FIR filters

In biorthogonal wavelet filter system, the lowpass filter  $g_0$  and  $h_0$  are symmetric; The highpass filter  $g_1$  and  $h_1$  are symmetric when they have odd length and antisymmetric

when they have even length. The reconstruction filters are entirely specified by the decomposition filters of the wavelet filter bank.

### 3.5.2 Biorthogonal 9/7 filter

The wavelet filters are selected by their compression performance and computational complexity [41]. An attractive property for orthogonal filters is the energy conservation that biorthogonal filters don't have. Biorthogonal 9/7 filter is preferred since its filter coefficients are close to be orthogonal [42].

The biorthogonal filter coefficients are solved by the constraints of dual double shift orthogonality and filter frequency response regularity. The double-shift orthogonality can solve  $\frac{N + \tilde{N}}{4}$  variables, where filter  $g_0$  has length  $N$  and its dual filter has length  $\tilde{N}$ .

Same flatness requirements are assigned to  $G_0$  and  $H_0$  filters by considering regularities in their frequency responses equally. When  $N = 7$  and  $\tilde{N} = 9$ , the conditions for biorthogonal 9/7 filters are:

$$\left\{ \begin{array}{l} G_0(w)|_{w=0} = \sqrt{2} \\ \sum_k g_0[k] \tilde{g}_0[k-2l] = \delta_{0,l} \\ G_0^{(r)}(w)|_{w=\pi} = 0 \quad r = 0,2 \\ H_0^{(r)}(w)|_{w=\pi} = 0 \quad r = 0,2 \end{array} \right. \quad (3.93)$$

where  $r$  is the derivative to the filter spectrum function. When  $r = 0$ , the equations of  $G_0^{(r)}(w)$  and  $H_0^{(r)}(w)$  are equivalent to those when  $r = 1$ , that is to say, the value of  $r = 1$  is redundant to solve the simultaneous equations of (3.93). Therefore,  $r = 0$  and  $r = 2$

are applied. After  $g_0$  and  $\tilde{g}_0$  are available, refer to (3.92), the filter coefficients with  $m = 9$  can be solved below as well as illustrated in Figure 3.4:

$$h_0 = [0 \quad 0.0378 \quad -0.0238 \quad -0.1106 \quad 0.3774 \quad 0.8527 \quad 0.3774 \quad -0.1106 \quad -0.0238 \quad 0.0378];$$

$$h_1 = [0 \quad -0.0645 \quad 0.0407 \quad 0.4181 \quad -0.7885 \quad 0.4181 \quad 0.0407 \quad -0.0645 \quad 0 \quad 0];$$

$$g_0 = [0 \quad -0.0645 \quad -0.0407 \quad 0.4181 \quad 0.7885 \quad 0.4181 \quad -0.0407 \quad -0.0645 \quad 0 \quad 0];$$

$$g_1 = [0 \quad -0.0378 \quad -0.0238 \quad 0.1106 \quad 0.3774 \quad -0.8527 \quad 0.3774 \quad 0.1106 \quad -0.0238 \quad -0.0378];$$

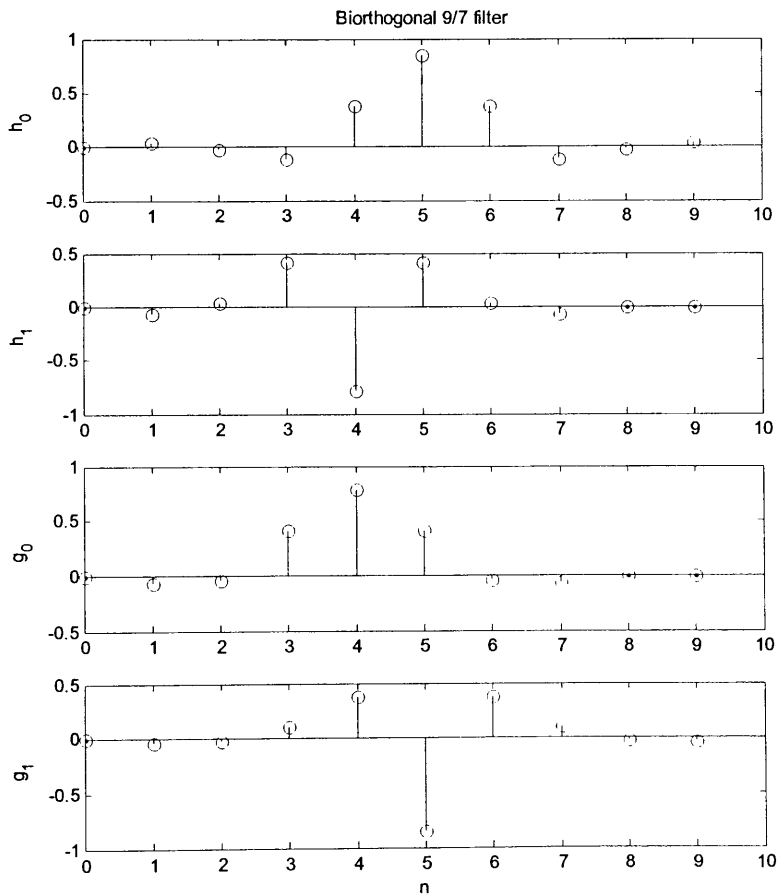


Figure 3.4. Biorthogonal 9/7 filter coefficients

The biorthogonal 9/7 lowpass analysis and synthesis filters with their linear phase frequency responses are shown in Figure 3.5, and the biorthogonal 9/7 highpass analysis and synthesis filters are shown in Figure 3.6.

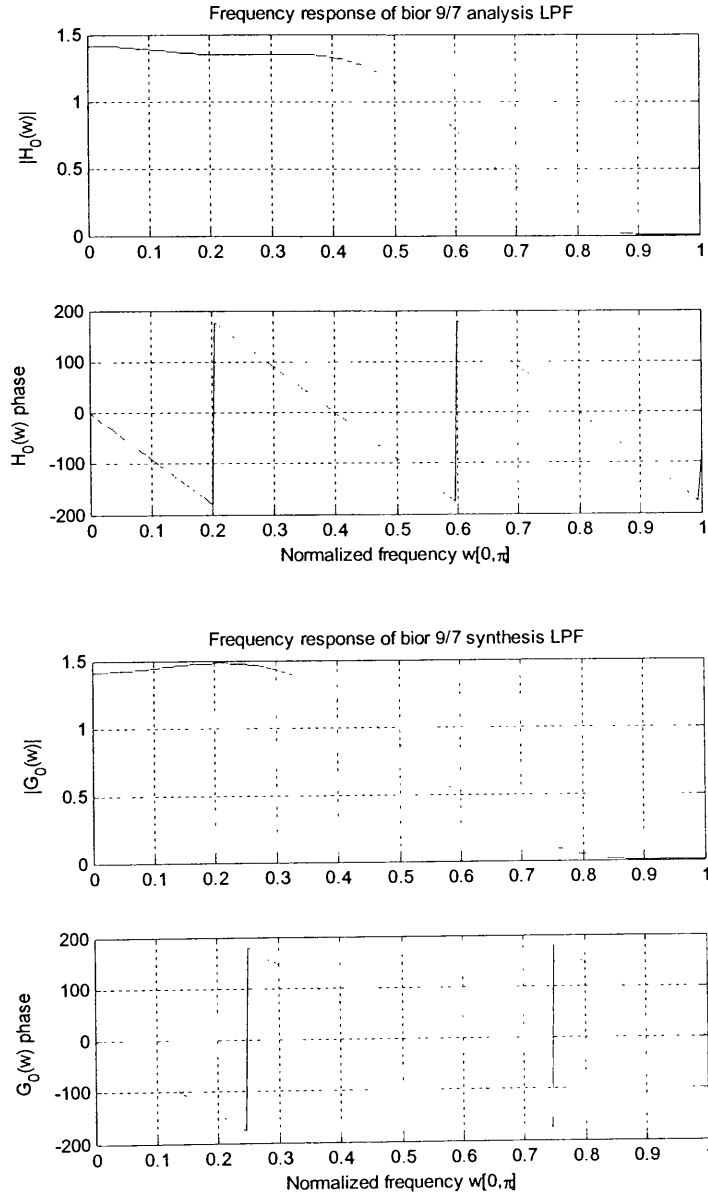


Figure 3.5. Biorthogonal 9/7 analysis and synthesis lowpass filters

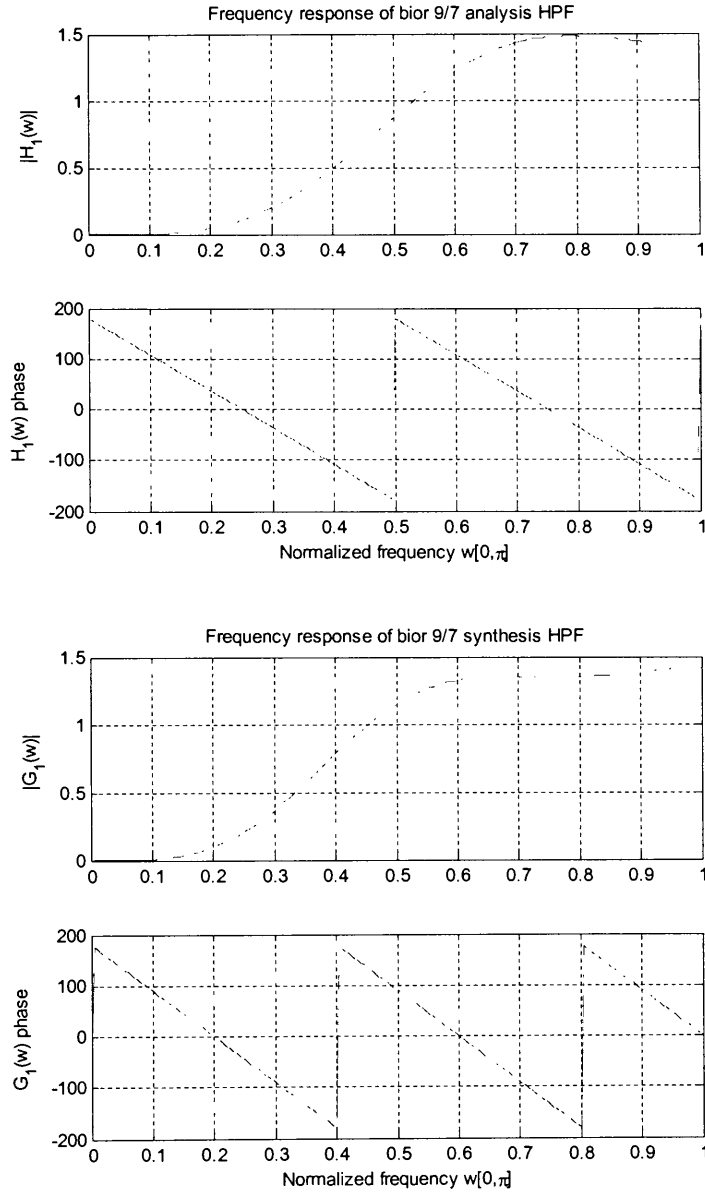


Figure 3.6. Biorthogonal 9/7 analysis and synthesis highpass filters

The scaling function  $\phi(t)$  derived from these discrete filters using cascade algorithm is shown in Figure 3.7 and the dual function  $\tilde{\phi}(t)$  is shown in Figure 3.8.

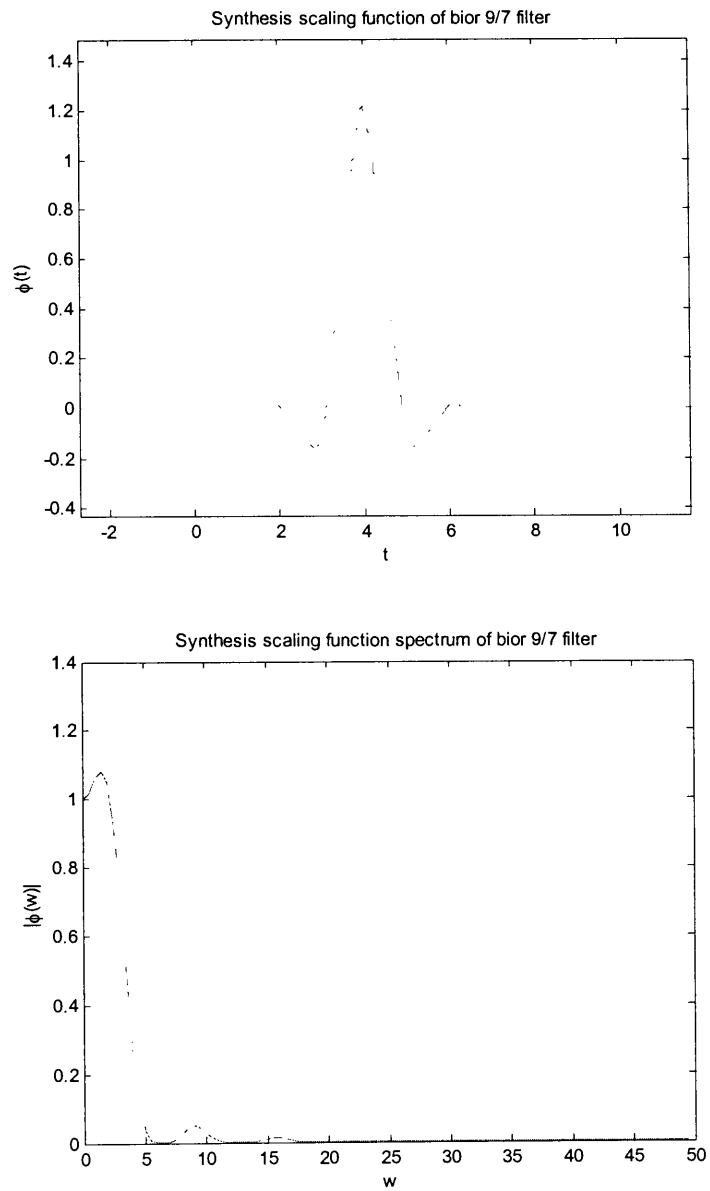


Figure 3.7. The scaling function  $\phi(t)$  of the causal biorthogonal 9/7 filter with its spectrum

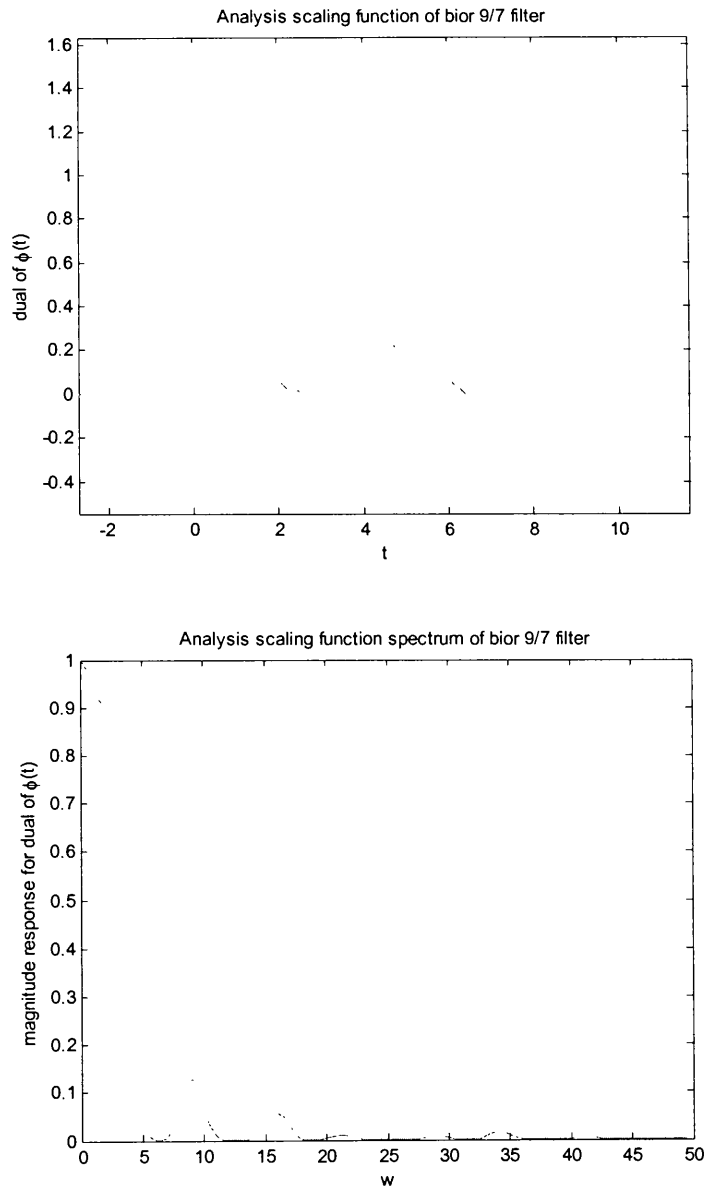


Figure 3.8. The scaling function  $\tilde{\phi}(t)$  of the causal biorthogonal 9/7 filter with the spectrum

### 3.6 Wavelet image compression

Image compression falls into two broad categories: lossless and lossy. The compression ratios of lossless algorithms are not high enough for image and video compression. Lossy



compression is useful for low bit rate applications in which a certain degree of distortion is acceptable as trade-off for higher compression performance. In fact, satisfactory lossy encoding techniques can reproduce virtually indistinguishable images compared to their originals under efficient compressions.

Wavelet transform is competitive with Fourier transform and even takes precedence in image compression such as in fingerprints [43, 44, 45]. The reason is that image signals are intrinsically composed of sharp edges, thus Fourier transform using sinusoidal is inefficient. The wavelet representation provides multiresolution/multifrequency expression of an image with localization in both spatial and frequency domain. Each subband contains certain information of the entire image, from coarse approximation to fine details. Blocking artifact can be avoided from wavelet compression while ringing artifacts appear, but can be improved by choosing high peak to lobe ratio filters.

After preprocessing operations on the color image such as color space transformation and mean shifting, two-dimensional wavelet transform is applied by separable extension of one-dimensional wavelet decompositions [46]. DWT is performed in horizontal direction for each row of the image and then in vertical direction for each column. The two-dimensional inverse DWT is operated on each column of the image and then on each row. Figure 3.9 depicts the filtering and decimation procedures for 2D wavelet transform. Four frequency subbands are generated, namely Low-Low (LL), High-Low (HL), Low-High (LH) and High-High (HH).

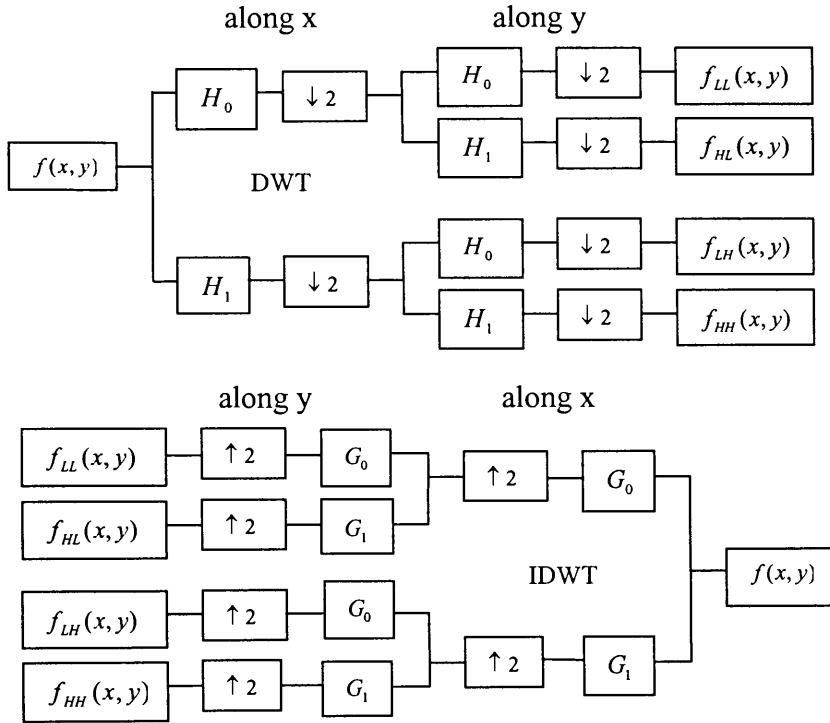


Figure 3.9. Two-dimensional DWT and IDWT

LL subband is the coarse representation of the original image. In other detail images, HL shows the horizontal, LH the vertical and HH the diagonal edges. The size of this wavelet domain image is equal to the original if coefficient expansion problem has been taken care of. For multilevel decomposition,  $N$ -level DWT will finally have  $3N + 1$  different frequency subbands, which include one LL frequency subband and  $3N$  high frequency subbands that vary in scales and orientations. A 2-level wavelet decomposed image is shown in Figure 3.10.

### Wavelet domain

$LL_2$	$HL_2$	$HL_1$
$LH_2$	$HH_2$	
$LH_1$		$HH_1$

### Wavelet decomposition

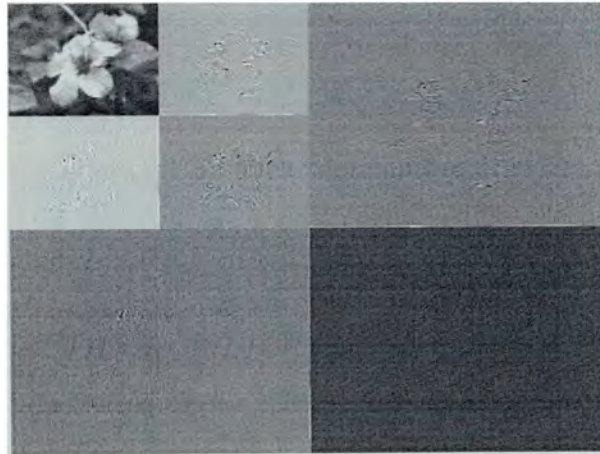


Figure 3.10. Multilevel wavelet decomposition on image “azalea”

The corresponding frequency domain spectrum composed of wavelet transformed subbands is outlined in Figure 3.11, and the center of the frequency domain is the DC frequency. The approximation and detail subband spectrums of the above wavelet decomposed image are shown in Figure 3.12 where the subband boundaries are not sharp since the wavelet filters are not ideal filters.

# Frequency domain

$HH_1$	$HL_1$			$HH_1$
	$HH_2$	$HL_2$	$HH_2$	
$LH_1$	$LH_2$	$LL_2$	$LH_2$	$LH_1$
	$HH_2$	$HL_2$	$HH_2$	
$HH_1$	$HL_1$			$HH_1$

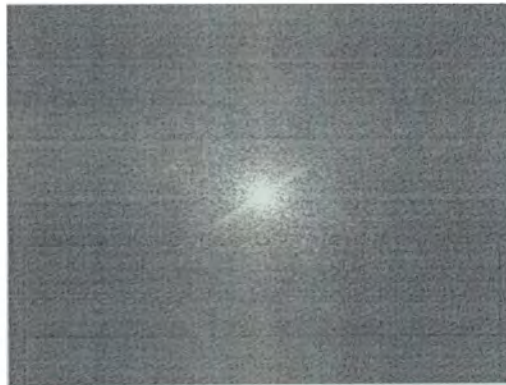


Figure 3.11. The corresponding wavelet domain decomposed subbands in frequency domain

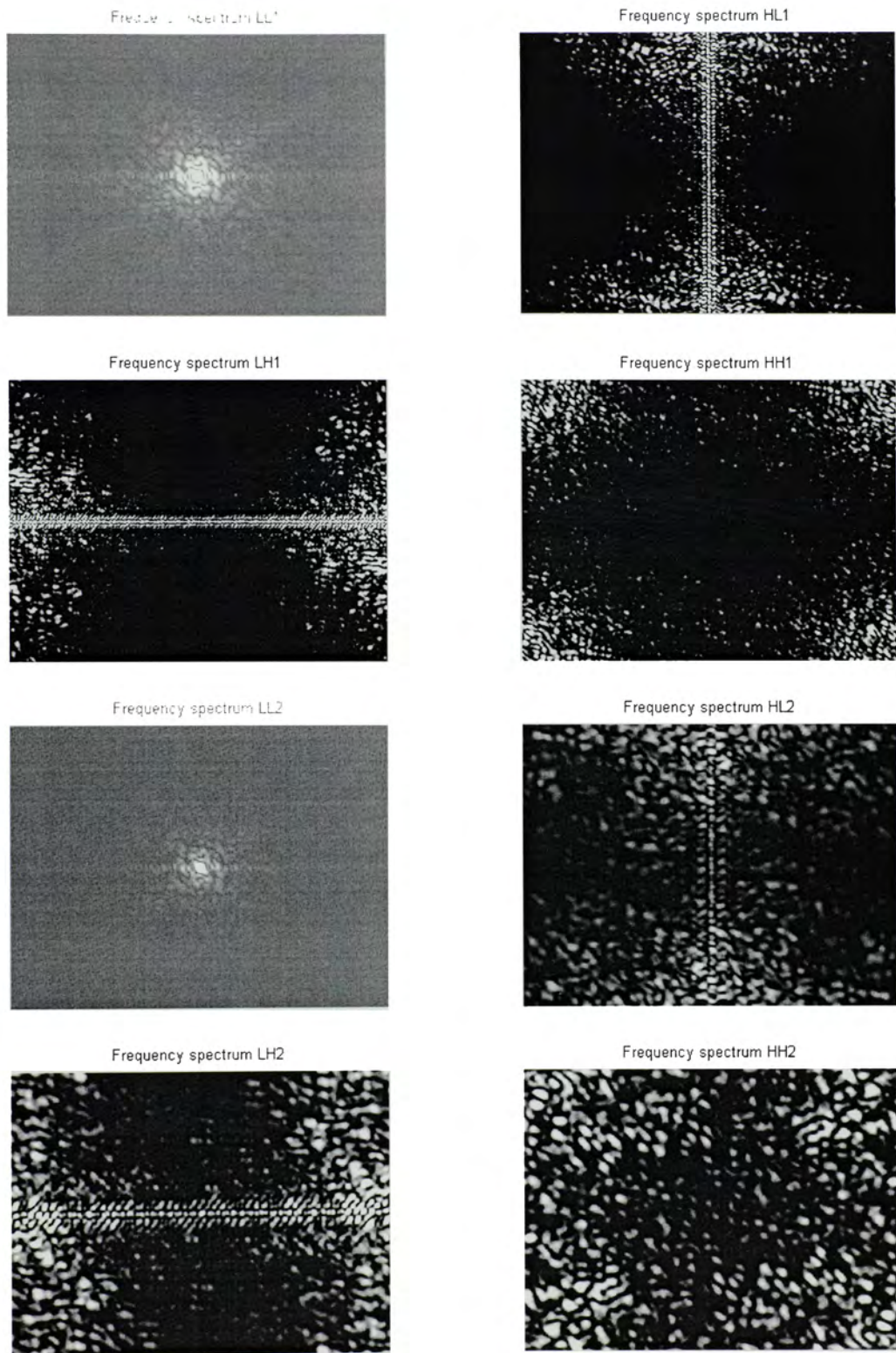


Figure 3.12. The frequency spectrums of the wavelet decomposed subbands

The coefficients are scanned and quantized in the order of LL, HL, LH, HH and from low to high frequency subbands. The bit stream is produced after compressing the image data by entropy coding. The decompression extracts the transformed coefficients, applies IDWT and converts  $YC_bC_b$  format back to RGB, to reconstruct the image.

Image compression techniques are very helpful in video compressions by considering that the frame under processing is composed of differential signals after motion estimation and compensation. Because of the multiresolution representation from wavelet decomposition, video sequence can be handled progressively with scalable resolutions and accuracies on demand.

## CHAPTER 4

### Motion Estimation and Motion Compensation Methods

#### 4.1 Block-based full search method

Reduction of temporal correlation using motion estimation and motion compensation (MEMC) is usually much more efficient than using spatial compression only. The commonly used motion estimation technique in most standard video codecs is the block matching algorithm (BMA), assuming that the objects in some frames are composed of rigid bodies with negligible deformation in translational movements. In reality, the mismatches to these conditions can be well satisfied to a certain degree by simulating the real objects and motions with these piecewise substitutes.

Frequency domain block-based approaches such as phase correlation method [47] estimates the relative shift between two image blocks using cross-correlation function in Fourier domain by the fact that translation in spatial domain results in a linear phase term in frequency domain. Due to the phase wrapping and multiple-object problems, frequency domain motion estimation is not as efficient as spatial search methods.

In block-matching method, the video frame under processing is called the current frame and a frame utilized for interframe compression is called a reference frame as shown in Figure 4.1. The current frame is divided into blocks of size  $N_1 \times N_2$ , called a macroblock. This size is designed not to be too small to balance the bandwidth, and not to be too large to maintain the motion estimation accuracy, normally  $N_1 = N_2 = 16$  as considered to be an appropriate size. The purpose of the search procedure is to obtain a

best matching block from the reference frame for each processing macroblock, or simply called the current block. Motion estimation is functioned on luminance components only, assuming that the color motions can be adequately represented by the motion information from the luminance.

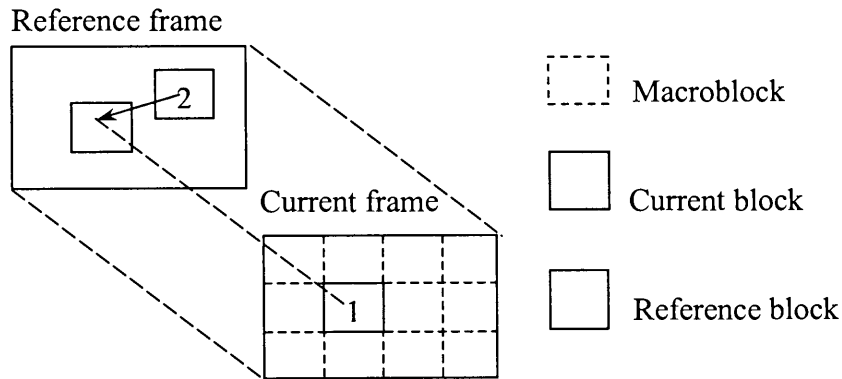


Figure 4.1. Block-based motion estimation

The searching reference block is moving inside a region of the reference frame called a search window around the same location of the current block, defined as a horizontal displacement  $M_1$  and a vertical displacement  $M_2$  as in Figure 4.2. The best matching block is found if it obtains the best result from the matching criteria function. Several matching functions can be applied such as maximum cross-correlation, minimum mean square error and minimum mean absolute difference (MAD). For computational simplicity and efficiency, minimum MAD criterion is used.



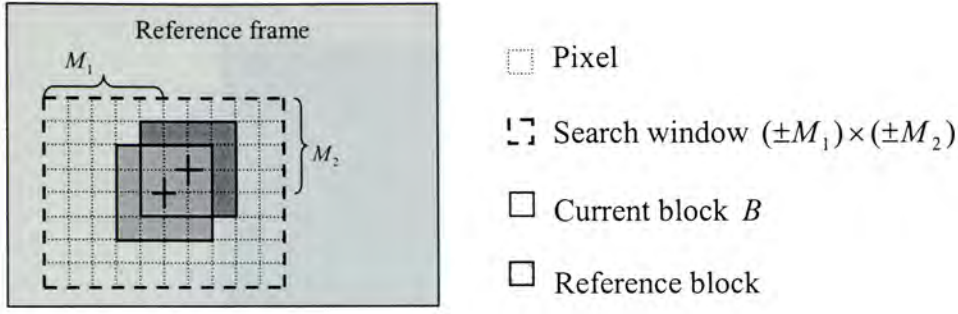


Figure 4.2. Block-matching method

In full search (FS) algorithm, the matching functions for all possible pixel-step displaced reference blocks over the whole search window are computed in order to achieve the global optimum matching block.

$$MAD(d_1, d_2) = \frac{1}{N_1 N_2} \sum_{(n_1, n_2) \in B} |I(n_1, n_2) - I_r(n_1 + d_1, n_2 + d_2)| \quad (4.1)$$

$$-M_1 \leq d_1 \leq M_1, -M_2 \leq d_2 \leq M_2 \quad (d_1, d_2 \in Z)$$

and

$$(MV_x, MV_y) = (d_1, d_2) |_{\min[MAD(d_1, d_2)]} \quad (4.2)$$

$I(x, y)$  is the pixel intensity of the current block and  $I_r(x, y)$  is the pixel intensity in the reference frame.  $MV_x$  and  $MV_y$  are the motion vectors to indicate how far horizontally and vertically the best matching block has been moved.

Since the current frame can take references from its former or future frames in a specific interpolation mode in MPEG (Motion Picture Experts Group) standard, the optimization of motion estimation and compensation can be considered simply as a procedure to improve reference-current frame pair performance. Two frames from test video sequence “football” in Figure 4.3 are used as a reference-current frame pair.

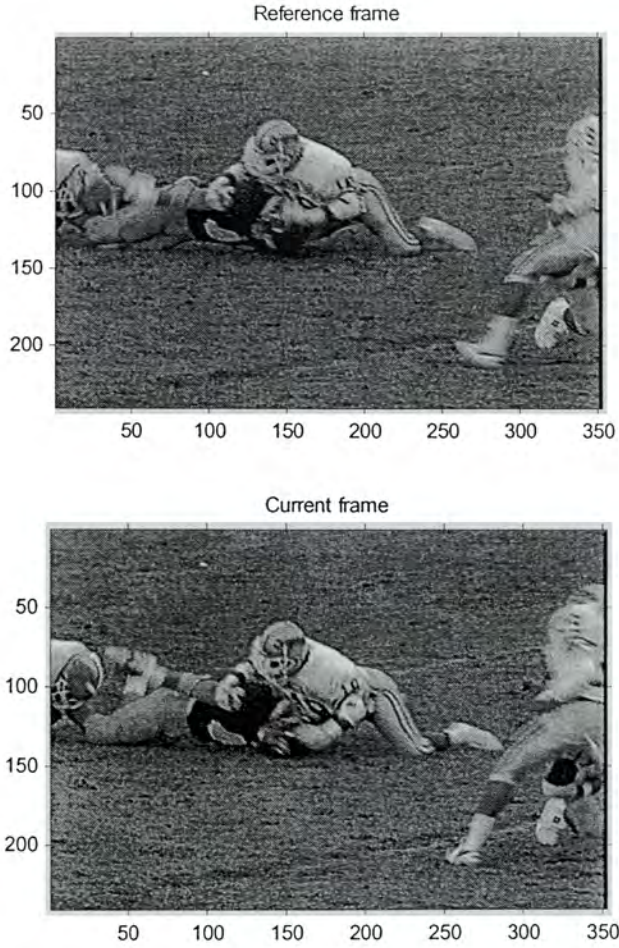


Figure 4.3. Reference-current frame pair from test video sequence “football”

MAD matching function distributions within the search area by maximum displacement of 7 pixels are illustrated in Figure 4.4 for two example blocks. The blocks are represented by  $(r, c)$ , which is in the  $r^{th}$  row and  $c^{th}$  column for pixels in range of  $[16r - 15 \sim 16r, 16c - 15 \sim 16c]$ . The first example shows the matching criterion function of block (8,17), which is in the horizontal pixels of 257~272 and vertical pixels of 113~128. The second example is for block (13,19) in horizontal pixels 289~304 and vertical pixels 193~208. The matching function is displayed by a gray scale image where the darker position is desired to be the center of a better matched block.

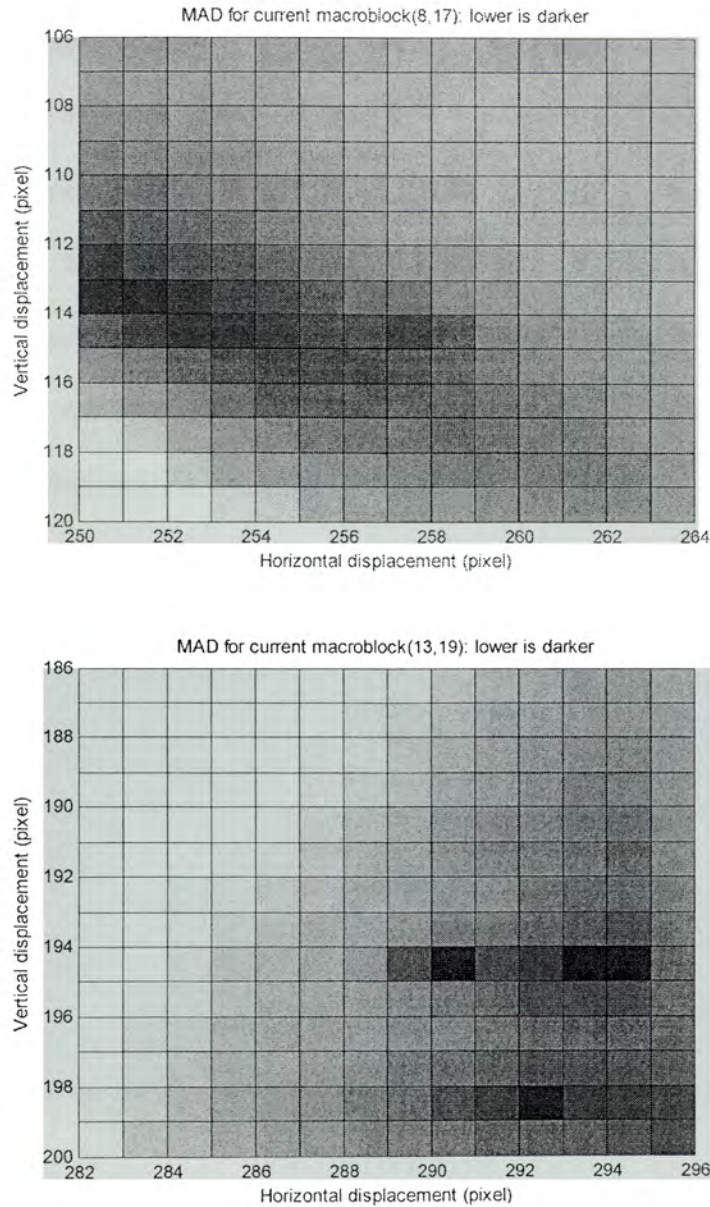


Figure 4.4. Matching function distributions over the search area

In general, most motion vectors are center-biased distributed, especially for low bit rate sequences. Thus, the stationary block for the initial search as a change detector is included, and the search is processed from near to far. If the matching function is low enough under a threshold, the heavy computation can be ceased in such a reasonable way.



To save on the bandwidth, shorter motion vectors are preferred if the matched blocks have the same minimum criterion. Results of motion estimation using full search method is shown in Figure 4.5. The motion vectors for intermode compressed blocks are drawn after the best matching block is found, otherwise a square is marked for intramode compressed block. The angle of the motion vector shows the moving direction and the length shows the moving distance from the best matching block to the current block. The motion vectors for the two blocks in Figure 4.4 can be verified by pointing the darkest position over the whole search area to the center.

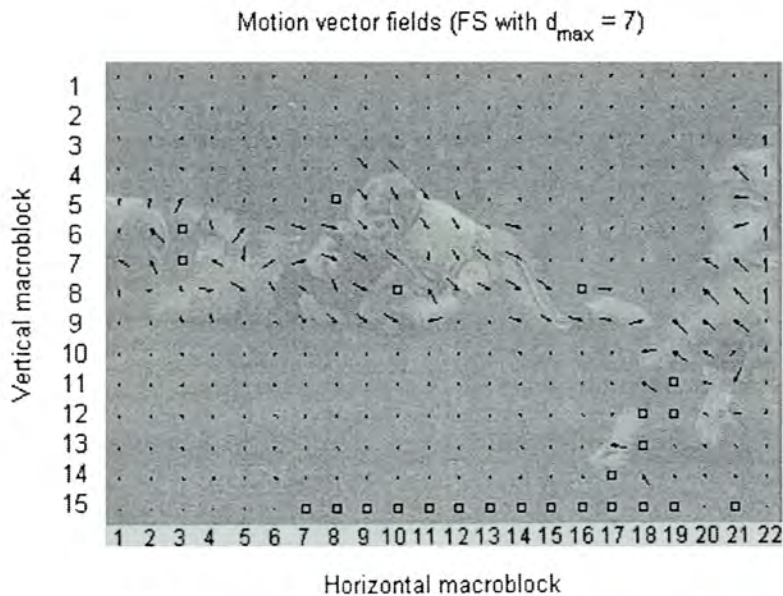


Figure 4.5. Motion vector fields by FS method

The difference between the current block and the predictive best matching block after motion estimation is called a differential error or residual block. If the energy of the differential block is higher than the original current block, motion compensation is not necessary and intramode coding is applied on the original block. But in most cases the

differential block contains much lower entropy than the original block, and therefore intermode coding on the differential block is used. The basic procedure of motion estimation and compensation is shown in Figure 4.6.

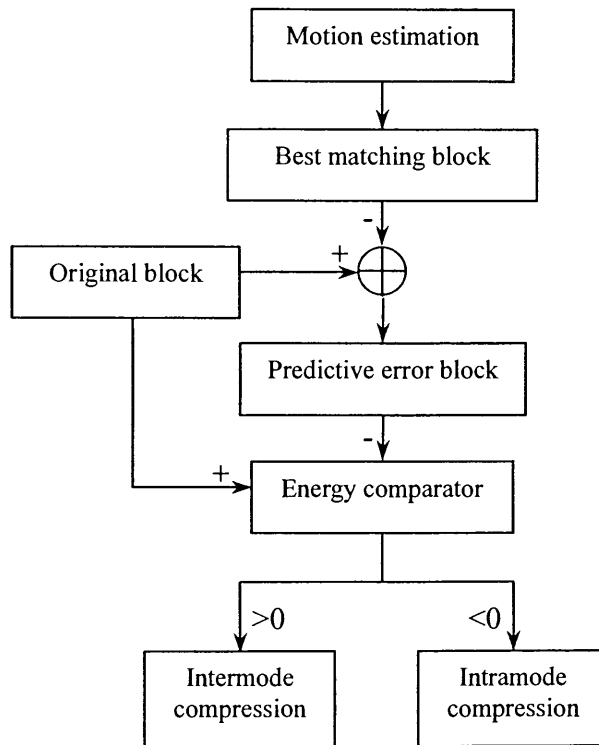


Figure 4.6. Intermode and intramode compression in motion estimation and motion compensation

Motion compensation is performed on all color components using the same motion vectors obtained from motion estimation on luminance blocks. The resulting prediction error frames are obtained separately in  $Y$ ,  $C_r$ , and  $C_b$  channels as illustrated in Figures 4.7, 4.8 and 4.9. It can be demonstrated that most of the energies are contained in the luminance frame which is used for motion estimation, and the energies of the prediction error frames are, as a consequence, much decreased for entropy coding.

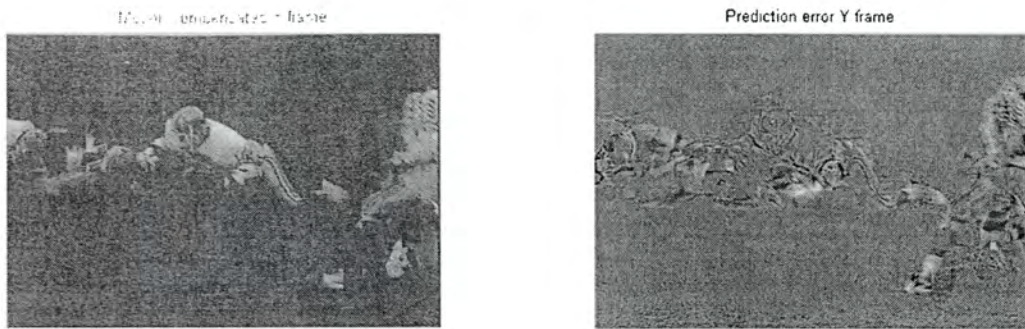


Figure 4.7. Y component motion compensation after FS motion estimation



Figure 4.8.  $C_r$  component motion compensation after FS motion estimation

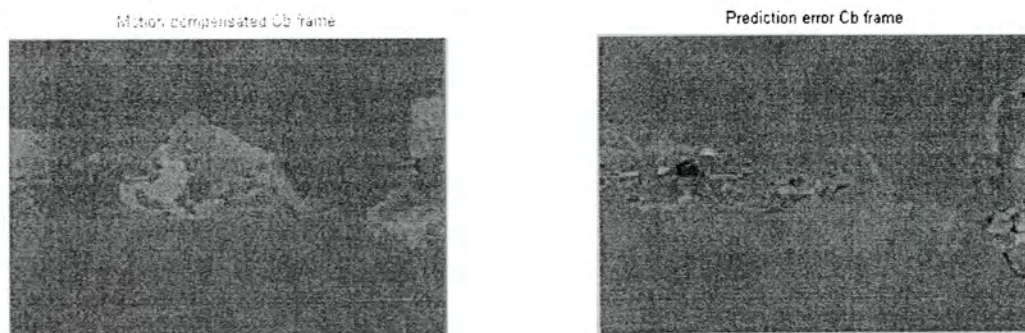


Figure 4.9.  $C_b$  component motion compensation after FS motion estimation

The computational load of MAD matching function is proportional to the block size  $N_1 \times N_2$ ,

$$L_B \propto N_1 N_2 \quad (4.3)$$

Because the search area is  $(2M_1 + 1) \times (2M_2 + 1)$ , and in most cases, the displacements in both directions are defined the same,

$$d = M_1 = M_2 \quad (4.4)$$

the computational load using full search method for each macroblock is determined as:

$$L_{FS} = L_B (2d + 1)^2 \quad (4.5)$$

## 4.2 Fast search methods

Although motion estimation is an efficient approach to video compression, it comprises most of the encoder complexity. Full search algorithm is the optimal solution for spatial motion estimation but is time consuming, thus fast motion estimation techniques are needed. The basic idea to achieve fast search is to perform a subsampling on the search area, assuming that the matching function is monotonically converging towards the best matching point, which is the center of the best matching block. Step search and block-based gradient descent search are good motion estimation methods for large and small motion video compressions respectively. A meaningful best matching block can be obtained inside the search area although it does not always correlate well with the actual motion. A number of articles have been published to realize fast search in aspects of adaptive search window [48, 49], motion tracing [50, 51], optimized matching criteria

[52, 53, 54], differential method [19], and motion estimation and motion compensation strategies [55, 56, 57, 58].

#### 4.2.1 Step search method

For center-biased step search (SS), the center position is initially defined in the reference frame at the same position of the current block in the current frame. In the 1<sup>st</sup> search step, the criteria function is run on the center point as well as other 8 corner points with  $\frac{d}{2}$  distance away, if the search window is bounded at  $\pm d$  pixel displacement. The step size is halved in the following steps around the best matching point from the last search step. The same refinement procedure is continued until the step size is 1. In reference to Figure 4.10, the maximum search displacement is 7 pixels and the current block shown in darker background is centered at location 7E. Reference blocks centered at 7E and its 8 side and corner checking blocks 4 pixels away are given by 3A, 3E, 3I, 7A, 7I, 11A, 11E and 11I, with large cross marks being the initial checking blocks for the 1<sup>st</sup> step search. The block with minimum MAD is the new best matching block, centered at 3I. All blocks centered by square marks around this new best block with 2 pixels away are the candidates for the 2<sup>nd</sup> step search. Assuming that the best block is found at 5K, the final search is reached by only 1 pixel displacement for the blocks marked with small cross centers. The block centered at 6L is the best in this last step, so it is the resulting best matching block. The overall motion vector is the addition of all the motion vectors from all steps, as shown by the arrow in Figure 4.10.



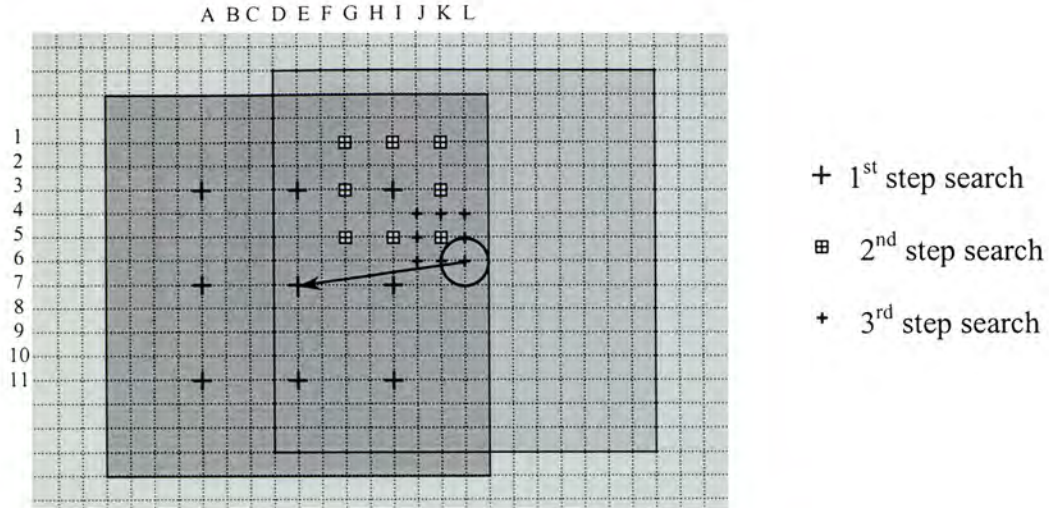


Figure 4.10. Step search algorithm

If the matching criteria have the same results for some checking points, the search procedure has to follow for each of them as if they are the new centers until better checking points are found. For example, if the blocks centered at 3E and 3I obtain the same matching function minima in the 1<sup>st</sup> search step, the side and corner checking positions around both of them are the candidates in the 2<sup>nd</sup> search step, which are at  $XY(X = \{1,5\}, Y = \{C, E, G, I, K\})$  and  $X = \{3\}, Y = \{C, G, K\}$ , where 1G, 3G and 5G are repeated for these two refinements. Thus, the recalculation problem needs to be avoided by memorizing the computed positions inside the search area. The results of motion vector field after motion estimation using step search method are demonstrated in Figure 4.11.

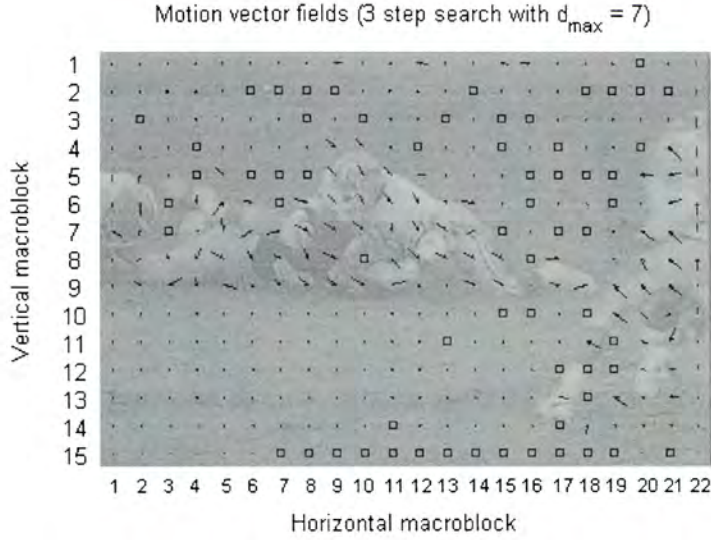


Figure 4.11. Motion estimation and compensation from step search method

For an  $N$ -step search, the total displacement is given by

$$d = \sum_{i=1}^N 2^{i-1} = 2^N - 1 \quad (4.6)$$

The computational load of the criterion function using step search method for each macroblock is

$$L_{SS} = (8N + 1)L_B = L_B \{8 * \lfloor \log_2(d + 1) \rfloor + 1\} \quad (4.7)$$

assuming that one best matching block is found in each search step. The  $\lfloor x \rfloor$  operator rounds  $x$  to the largest integer that is less than  $x$ . The computational load of step search method is much less significant as compared to the full search method as contrasted in Table 4.1.

Table 4.1. Computational load comparison using full search and step search method

Displacement	Computational load (in unit of $L_B$ )			
	$d = 1$	$d = 4$	$d = 7$	$d = 15$
$L_{FS}$	9	81	225	961
$L_{SS}$	9	17	25	33

#### 4.2.2 Block-based gradient descent search method

In block-based gradient descent search (BBGDS) method, the search procedure starts from the stationary center and its neighboring unchecked points. If the best matching point is a corner, 4 checking points would be calculated and 5 new ones would be included in the next round. On the other hand, if the best matching point is in the middle of the edge, 6 points would be calculated and 3 points would be considered new points for the next round. Since the step size is already as minimum as 1 pixel, the search procedure continues until the best matching point is the center or the search window boundary is reached. Figure 4.12 provides an example of this search method with the center of the current block shown in darker background is at location 3F. Together with its neighbor points around as 2E, 2F, 2G, 3E, 3G, 4E, 4F and 4G, 9 checking blocks are considered in the 1<sup>st</sup> round search. If the best checking point is at 3G, the checking blocks using large square marks centered at 2H, 3H and 4H are compared with 3G for the 2<sup>nd</sup> round. Assuming that 4H is the new best checking point, the blocks centered with small cross marks are run for the 3<sup>rd</sup> round. If 5H is the best and it's also better than its neighbor unchecked points with small square marks, 5H is then considered as the center position of the final best matching block.

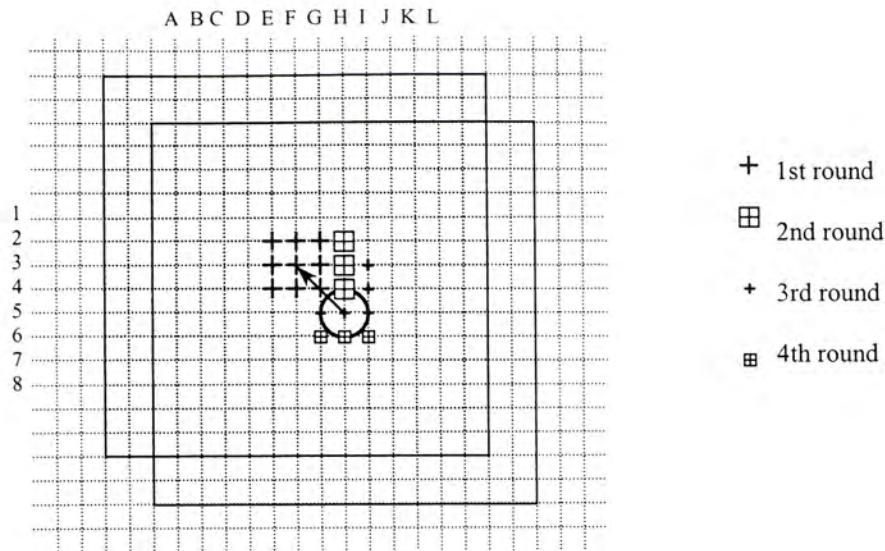


Figure 4.12. Block-based gradient descent search algorithm

Since the new checking points could be overlapped between rounds, the computed checking points need to be saved to avoid recalculation. Figure 4.13 shows the results of motion estimation using BBGDS method by drawing motion vectors for intermode blocks and squares for intramode blocks.

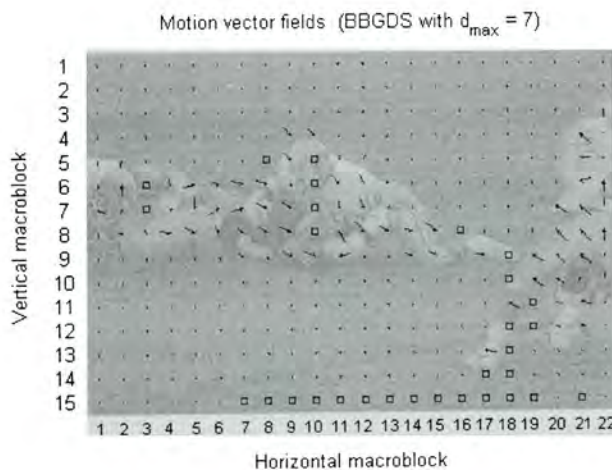


Figure 4.13. Motion estimation and compensation from BBGDS method

The computational load of BBGDS method within  $d$ -pixel displacement area after  $N$  search rounds is:

$$L_{BBGDS} = L_B(9 + \sum_{i=2}^N L_i) \quad (4.8)$$

with

$$-d \leq \sum_{i=1}^N d_i \leq d \quad (4.9)$$

$d_i$  is the displacement and  $L_i$  is the computational load in the  $i^{th}$  search round.  $L_i = 5$  if the best matching point from the last round is at the corner, and  $L_i = 3$  if it is in the middle, given that only one best matching block is found in each round. Thus, the computational load depends on the matching function distribution from the center throughout the searching area. The convergence of the matching function is fast for small motions and slow for large motions until the local minimum is reached.

### 4.3 Multiresolution motion estimation

During the motion estimation process, matching criteria function will decrease and will converge to a local minimum inside the search window. However, the matching function usually is not monotonically distributed over the whole search region but possible for the region around the global optimum, so the best matching block can be found by the refinement procedure.

Step search makes large search steps at the beginning which is not good for small motions while BBGDS is always taking only 1 pixel step size. The motion vector field results indicate that the best matching blocks are missed or are incorrect by using SS and

BBGDS in some locations. A few best matching blocks can't be detected from BBGDS method around the players in large movements and from SS method in the almost stationary grass area. From the MAD plot of macroblock (8,17) in Figure 4.4, a small displaced local minimum stops BBGDS calculation. This error can be verified by the different motion vectors between FS in Figure 4.5 and BBGDS in Figure 4.13 at this block. An error of large motion using SS method occurs at block (13,19) from Figure 4.4. This can also be seen by comparing the motion vectors of FS and SS method, from Figure 4.5 and 4.11 at this block. Since the motion speed and the frame interpolation distances may vary, fixed step size search methods are prone to falling into local minima. To increase the search efficiency and accuracy, the initial search points need to be located reasonably, not too large for small motions and not too small for large motions, to avoid being trapped into undesirable convergent region. Motion estimation can be implemented globally but not fully, using a multiresolution description of the video frames to obtain initial and refined search. Because motion activities are still highly correlated at different resolutions, frame pyramid is build up and the search starts from coarse to fine scales. Due to the fact that the motion speed is doubled by the halved resolution between successive levels, multiresolution method is hence regarded as a fast search method.

#### **4.4 Video codec simulation and compression results**

The basic structure of video encoder is illustrated in Figure 4.14 in which motion estimation and motion compensation are the main components. The predictive frame is generated via motion compensation according to the motion vectors obtained from motion estimation. After subtracting the predictive frame from the input current frame,

frequency transformation is applied on the residual frame due to the fact that the correlations between error pixels after MEMC are still very strong [59]. Since the differential frame contains a lot of high frequency components, a scalar quantization matrix is utilized, instead of the frequency-dependent quantization matrix used in image compressions.

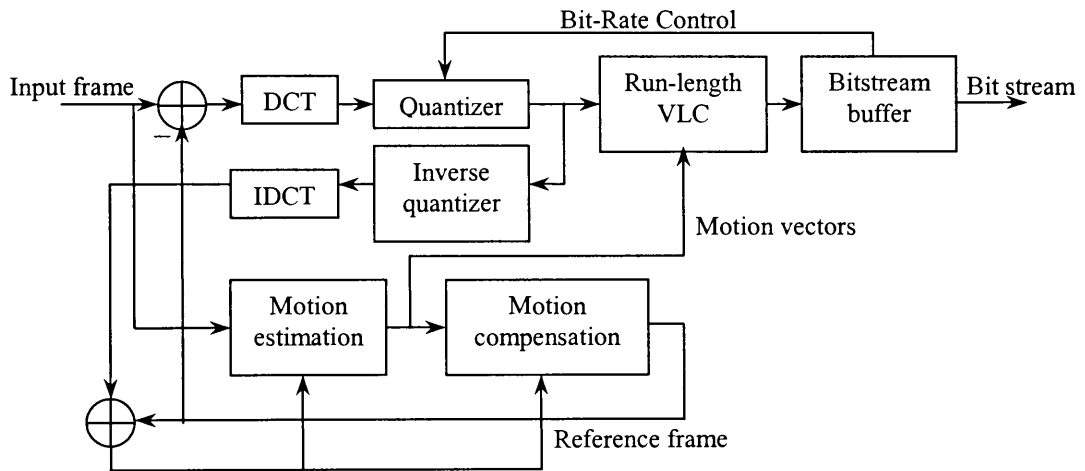


Figure 4.14. Video encoder kernel

In order to prevent error build-up at the decoder, the prediction at both the encoder and decoder must be identical [47]. A local decoder is set up after the lossy quantizer to simulate the reconstructed reference frame. The bit rate of the buffer and the transmission can be regulated via the adaptive quantization factor on the differential frame with the trade-off between the video quality. Only the motion vectors and the low entropy compensated error signals are sent to the other end, therefore, the channel capacity required for video transmission is reduced.

Without motion estimation, the decoder shown in Figure 4.15 has much less complexity than the encoder. The predictive frame is produced by the best matching blocks from the



reference frame and the decoded motion vectors. The addition of the predictive frame and the differential error frame after dequantization and inverse DCT reconstructs the current frame.

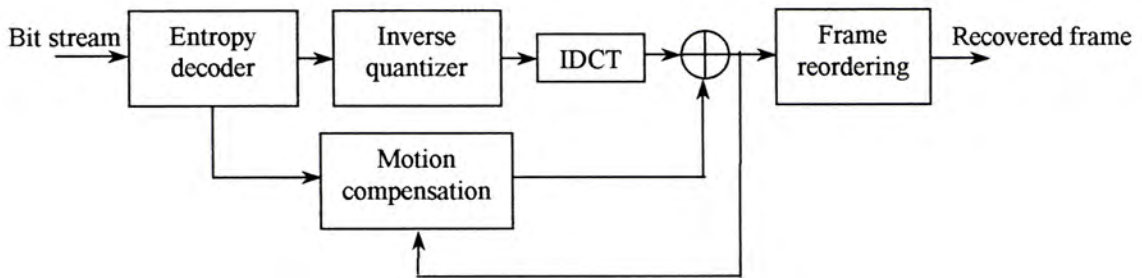


Figure 4.15. Video decoder kernel

The video codecs are implemented by software to compare the compression performances using FS, SS and BBGDS motion estimation methods. A translational-motioned sequence such as the “foreman” sequence is shown in Figure 4.16. The current frame compressions of “football” by 60 times at a bit rate of about 0.4bpp using these methods are shown in Figure 4.17 and for “foreman” compressed by 120 times at bit rate of 0.2 bpp is shown in Figure 4.18.



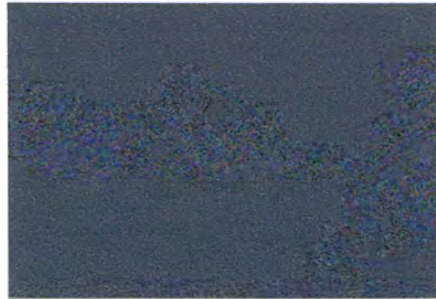
Figure 4.16. Reference-current frame pair from test video sequence “foreman”



Reconstruction from FS MEMC  
Bitrate = 0.398(bpp) PSNR = 29.754(dB)



Reconstruction error



Reconstruction from SS MEMC  
Bitrate = 0.409(bpp) PSNR = 29.101(dB)



Reconstruction error



Reconstruction from BBGDS MEMC  
Bitrate = 0.394(bpp) PSNR = 29.666(dB)

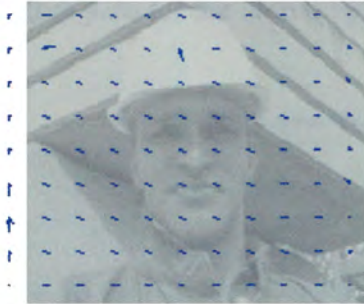


Reconstruction error



Figure 4.17. Video compression and reconstruction for sequence “football”

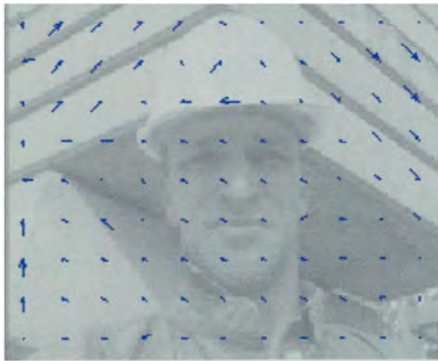
Motion vector fields (FS with  $d_{\max} = 7$ )



Reconstruction from FS MEMC  
Bitrate = 0.204(bpp) PSNR = 33.662(dB)



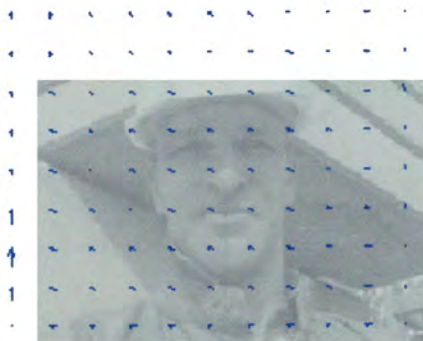
Motion vector fields (3 step search with  $d_{\max} = 7$ )



Reconstruction from SS MEMC  
Bitrate = 0.210(bpp) PSNR = 31.897(dB)



Motion vector fields (BBGDS with  $d_{\max} = 7$ )



Reconstruction from BBGDS MEMC  
Bitrate = 0.203(bpp) PSNR = 33.744(dB)



Figure 4.18. Video compression and reconstruction for sequence “foreman”

The compression performances at various bit rates vs. PSNR for low bit rate compressions from 0.2~1.0 bpp are illustrated in Figure 4.19 for the “football” sequence and in Figure 4.20 for the “foreman” sequence.

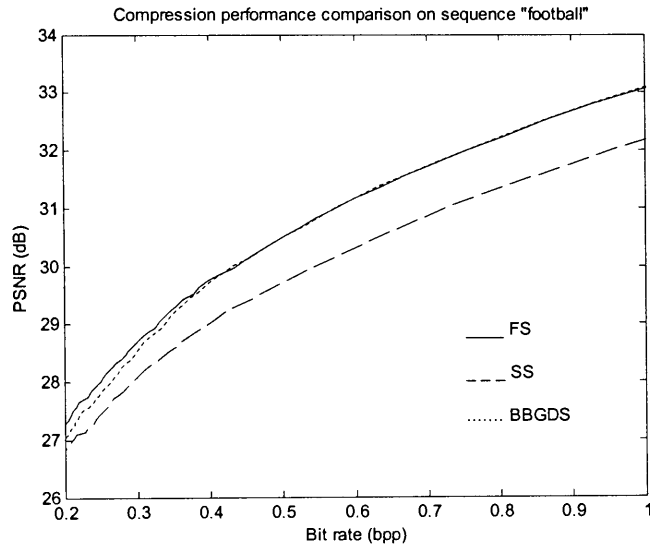


Figure 4.19. Bit rate vs. PSNR for sequence “football”

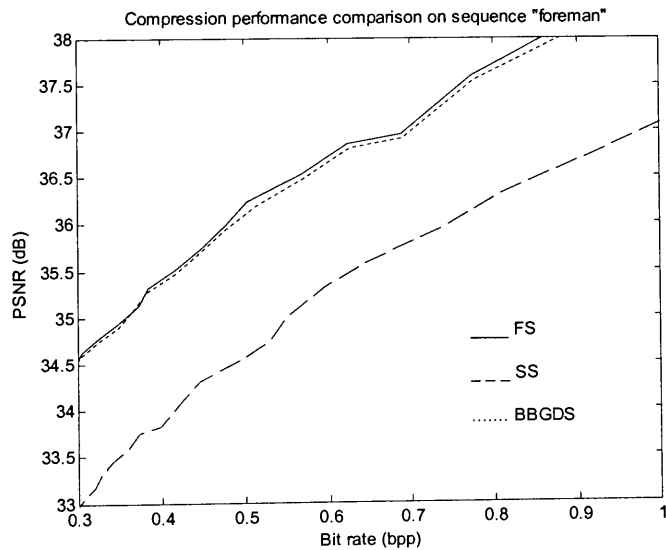


Figure 4.20. Bit rate vs. PSNR for sequence “foreman”

The blocking artifacts appear at low bit rate compressions of 0.13bpp in Figure 4.21 due to DCT still image compression, as well as the degradation of transform coding on the differential signals after motion compensation.

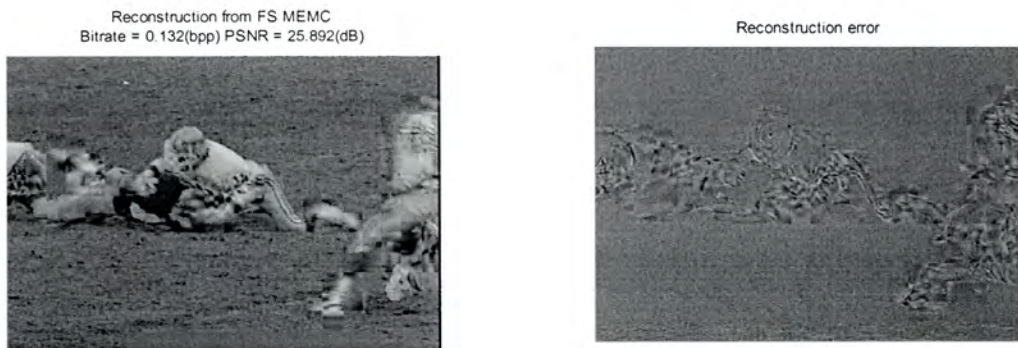


Figure 4.21. Low bit rate video compression

## CHAPTER 5

### Wavelet-Based Video Compression

#### 5.1 Wavelet-based video codec

The idea of multiresolution motion estimation and compensation lead to the investigation of wavelet-based video compression methods. By the fact that wavelet transformed coefficients still contain spatial domain information, motion structure within the subbands at the same scale and between different scales are highly correlated. The motion fields are more robust to detect both small and large motion activities from the refinement mechanism of hierarchical wavelet search. It is expected that the wavelet domain video compression can achieve good performance, especially at low bit rates [58], by the experiment results of image compressions using wavelet transform. Furthermore, the differential error frame after motion estimation and compensation has lower entropy than the original video frame, which results in more efficient compressions.

Since the high frequencies are contained at the sharp motion compensated block edges and the small spatial correlations of the differential error blocks decrease the coding efficiency, global wavelet transform is taken before motion estimation, which is different from the conventional video codecs. It was demonstrated that DWT applied on the original video frames outperforms the DWT operating on the differential frame after motion compensation in terms of PSNR and the subjective evaluations [60, 61]. Further compression is enabled by applying quantization schemes [62] to the transformed data and then by allocating bandwidth to the quantized coefficients statistically. Vector

quantization using a codebook is frequently used in image compression [63, 64]. Although the quantization matrix can be optimized based on psychovisual and frequency characteristics [65, 66, 67], the improvements by these processes on the decorrelated differential data inside the wavelet-transformed domain are not obvious [68]. Wavelet-based video compression has been studied in some papers by means of search strategies [61, 69, 70, 71, 72, 73, 74], complex-valued wavelet transform [75], overcomplete discrete wavelet transform [76] and wavelet packets [77].

The basic structure of wavelet-based video encoder is shown in Figure 5.1. Both the reference frame and the current frame are first color space transformed (CST) from RGB into  $YC_bC_b$  format and then shifted to their average luminance and chrominances before wavelet transformation is applied. Multiresolution motion estimation and compensation are performed on the decomposed subframe pyramids, based on the strong correlation between LL subbands after wavelet filtering. Due to the similar motion activities in the approximation and detail subbands, motion fields can also be estimated in a fast and simplified way by using subband refinement approaches. In addition, appropriate motion compensation may obtain efficient coding when the entropy is getting lower in the high frequency subbands. Uniform quantization is performed on the wavelet compensated frame, which still keeps the energy distribution routine within different frequency subbands. Each coefficient is divided by a quantization factor and rounded to a nearest integer, regardless of intermode or intramode compression. The bit stream containing side information of the Huffman tables, frame format, filter coefficients, transformation parameters, quantization factor, and the compressed data are sent to the receiver.

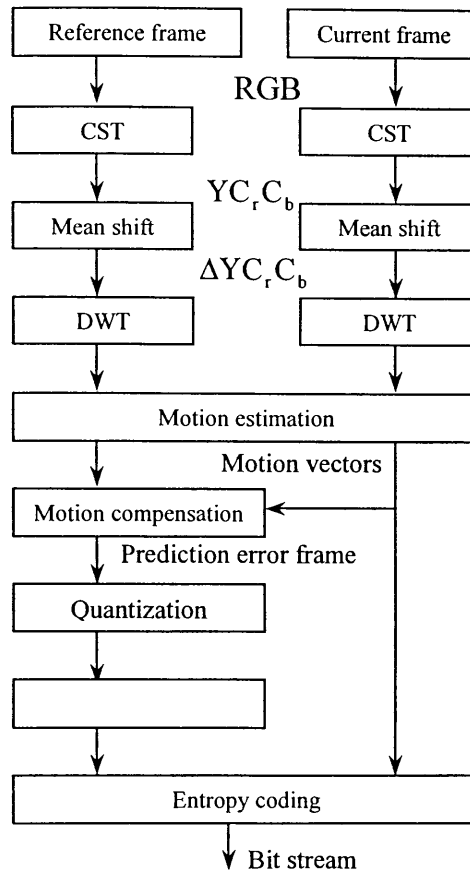


Figure 5.1. Wavelet video encoder

Accordingly, the lossless decoder extracts the Huffman table to decompress the sequential bit stream, as illustrated in Figure 5.2. Intermode and intramode compressions are performed from the decoded frame data and the motion vectors for each current block level by level. The dequantized coefficients with quantization and rounding errors approximate the wavelet transformed frame. Since no inverse transformation is included in the encoder, features such as scalability and progressive transmission can be realized. After all the necessary compensated subframes are reconstructed, inverse wavelet transform is taken until the full frame resolution is reached. Color space conversion from  $YC_rC_b$  back into RGB format is performed at last to achieve the reconstructed frame.



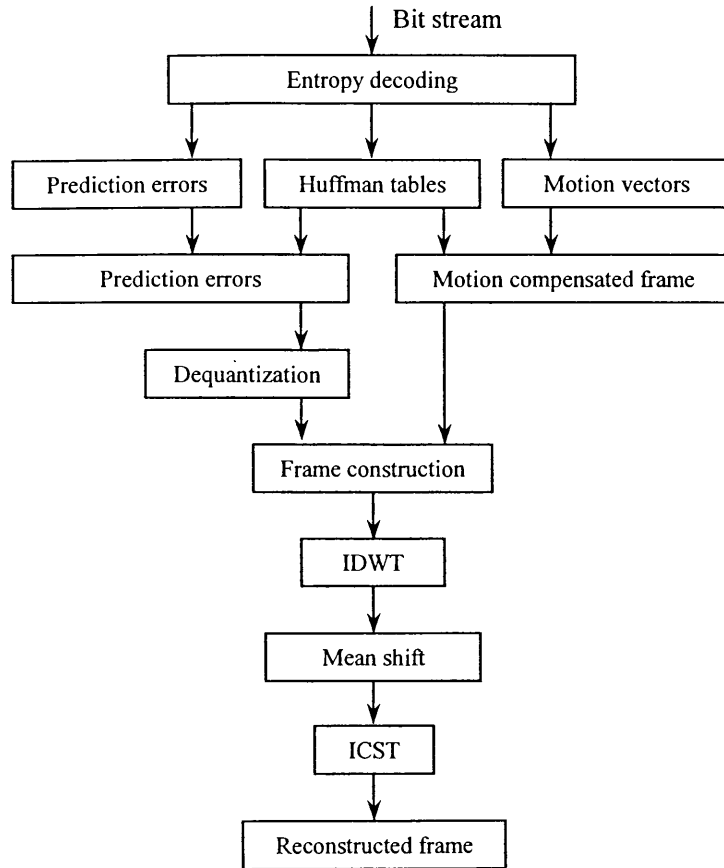


Figure 5.2. Wavelet video decoder

## 5.2 Wavelet domain multiresolution motion estimation and compensation

Wavelet domain motion estimation is processed by searching the best matching block between the current subframe and the corresponding reference subframe, from coarse to fine resolutions hierarchically. At the lowest resolution level, full search algorithm is applied in the LL subframe as an initial estimate. Due to the correlated motion structures between adjacent frequency subbands, motion estimation in the detail HL, LH and HH subbands are refined by the motion vectors obtained from the LL subband at the same level, and the motion fields in higher resolution LL subframe are refined by those from



the lower resolution LL subframes. Defining  $MV_i$  as a motion vector in the LL subframe at the  $i^{th}$  decomposition level, motion fields in subbands  $HL_i$ ,  $LH_i$  and  $HH_i$  are then refined around the position appointed by  $MV_i$  as:

$$MV_{n_i} = MV_i + \Delta MV_{n_i} \quad (5.1)$$

where  $n = \{HL, LH, HH\}$  constitutes the detail subbands. When more frequency components are added, more accurate motion fields can be achieved in the higher resolution subframe  $LL_{i-1}$ , based on the rescaled motion vectors from the last subframe  $LL_i$ . Because of the half-resolution decomposition, the number of the processing blocks in all subframes is identical accomplishing one-to-one correspondence and the rescaling of the motion vectors between two levels is performed by doubling as:

$$MV_{i-1} = 2MV_i \quad (5.2)$$

where  $MV_{i-1}$  is the motion vector from the higher resolution level. The refinement is made around the matched block defined by the rescaled motion vector  $2MV_i$  as:

$$MV_{i-1} = 2MV_i + \Delta MV_{i-1} \quad (5.3)$$

This assumes that a 2-level wavelet decomposition is applied on the reference frame and the current frame, as shown in Figure 5.3. Motion estimation starts from the lowest resolution subframe  $LL_2$ , and  $MV_2$  is the resulting motion vector for a certain block when its best matching block is found. If  $MV_2$  is not available, full search is reinitiated for the motion estimation in the detail  $HL_2$ ,  $LH_2$  and  $HH_2$  subbands respectively. The rescaling in the 1<sup>st</sup> level is:

$$MV_1 = 2MV_2 \quad (5.4)$$

and the motion vector of the  $LL_1$  subframe is refined as:

$$MV_1 = 2MV_2 + \Delta MV_1 \quad (5.5)$$

which is the initial motion vector for all the detail subbands in the first level. By referring to equation (5.1) for  $i = 1$ , the motion vectors in the  $HL_1$ ,  $LH_1$  and  $HH_1$  subbands are refined from  $MV_1$  of the  $LL_1$  subframe:

$$MV_{n_1} = MV_1 + \Delta MV_{n_1} \quad (5.6)$$

as represented by  $MV_{HL_1}$ ,  $MV_{LH_1}$  and  $MV_{HH_1}$ .

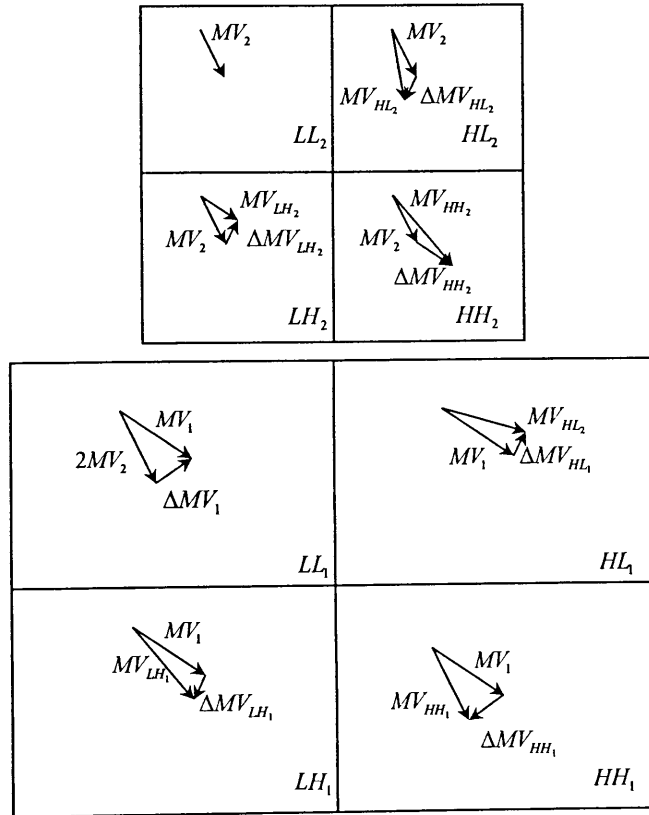


Figure 5.3. Multiresolution wavelet motion estimation

Thus, an  $N$ -level multiresolution motion estimation can be concluded as:

$$LL_i \rightarrow LL_{i-1} \quad (i = 2 \sim N)$$

$$LL_i \rightarrow HL_i, LH_i, HH_i \quad (i = 1 \sim N)$$

where the right side subband is taking refinement reference from the left side. Table 5.1 summarizes the motion vector refinement relationships between the 2-level decomposed subbands, assuming that the best matching block is found in the reference subband, otherwise full search algorithm is used instead.

Table 5.1. Motion vector refinements between the subbands in wavelet domain motion estimation

Subband	$LL_2$	$HL_2$	$LH_2$	$HH_2$	$LL_1$	$HL_1$	$LH_1$	$HH_1$
Reference subband	N/A	$LL_2$	$LL_2$	$LL_2$	$LL_2$	$LL_1$	$LL_1$	$LL_1$

The refinement process is continued successively until motion estimation and compensation are realized in all the subbands of the current frame. Each subband obtains a motion vector array to represent the motions of all current blocks. The difference between the best matching block and the current block is encoded, called intermode compression, as seen from the output after the subtractor in Figure 5.4. If the energy of the differential block is larger than the current block, the current block is encoded instead without motion compensation to achieve lower entropy coding, which is called the intramode compression. The motion vectors and the compression modes are sent together for the motion compensation in the decoder.

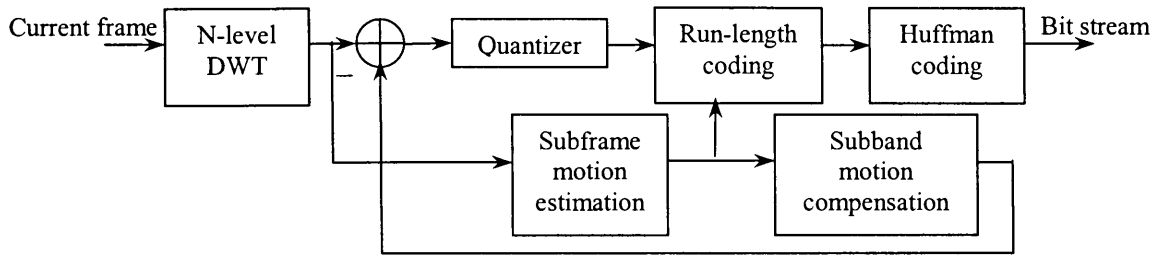


Figure 5.4. Wavelet-based video encoder kernel

Motion compensation is performed on the  $LL$  subband in the lowest resolution level and the detail subbands, while higher level  $LL$  subbands as the initial refinement reference for the whole resolution level are not included. The basic structure of wavelet-based video decoder is shown in Figure 5.5. Motion compensations are realized by adding the differential data to the matched blocks defined by the motion fields of each subband. The motion compensated frame serves as the input to the inverse wavelet transformer to get the reconstructed frame.

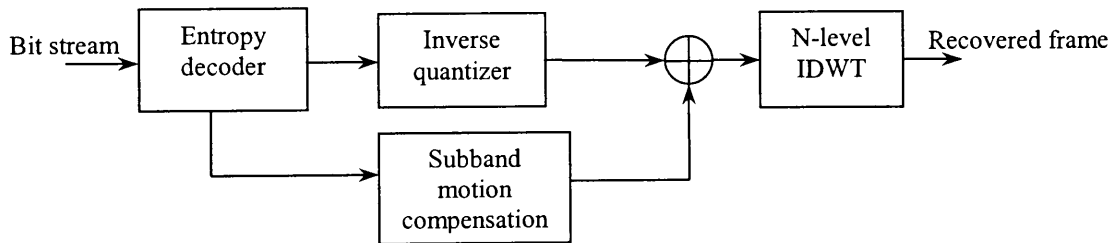


Figure 5.5. Wavelet-based video decoder kernel

Although the block size relative to the coverage of the full frame remains the same, the physical block size to compute the matching criterion function is halved. Assume the maximum displacement in the lowest resolution level is  $d_0$ -pixel using full search

method after  $N$ -level wavelet decomposition, and the refinements in other subframes and levels are  $d_r$ -pixel, the equivalent search displacement  $d_{ws}$  in frame resolution is:

$$d_{ws} = 2^N d_0 + \sum_{i=1}^N 2^i d_r = 2^N d_0 + (2^{N+1} - 2)d_r \quad (5.7)$$

Because the block size is  $\frac{N_1}{2^i} \times \frac{N_2}{2^i}$  in level  $i$ , the computational load of a current block

in all levels for a full frame size of  $N_1 \times N_2$  using wavelet search is:

$$\begin{aligned} L_{ws} &= \frac{N_1}{2^N} \cdot \frac{N_2}{2^N} [(2d_0 + 1)^2 + 3(2d_r + 1)^2] + \sum_{i=1}^{N-1} \frac{N_1}{2^i} \cdot \frac{N_2}{2^i} \cdot 4(2d_r + 1)^2 \\ &= L_B \left\{ \left(\frac{1}{4}\right)^N [(2d_0 + 1)^2 + 3(2d_r + 1)^2] + \frac{4}{3} \left[1 - \left(\frac{1}{4}\right)^{N-1}\right] (2d_r + 1)^2 \right\} \end{aligned} \quad (5.8)$$

assuming that refinements are performed in each of the subsequent levels and subbands.

Usually, the refinement of motion vector is 1 pixel ( $d_r = 1$ ) owing to the full search method used in the lowest resolution level and the 2:1 decimation process.

### 5.3 Entropy coding

The progressively transmitted data composed of the frame and motion information are entropy encoded to reduce coding redundancies. Embedded zerotree wavelet (EZW) coding, which is peculiarly used for wavelet image compression [51, 78], can also be applied in video compression [79]. EZW is making use of the correlations between wavelet coefficients in different frequency subbands, thus the coding on the decorrelated differential signals is not that efficient. Although arithmetic coding [59] can attain better entropy by assigning non-integer bits to the coded symbols, Huffman coding is employed for real implementation from the processing complexity point of view. The data of the bit

stream follows the order of: from motions to coefficients, from luminance to chrominances, from approximation to detail subbands, and from low to high resolution levels.

The fundamental principle of entropy coding is to represent the frequently occurring symbols by short codewords and infrequent ones by long codewords, which is the so-called variable-length coding. The lossless strategy is that no codeword is the prefix of another, as defined by the basecode lookup table. Unlike the static Huffman entropy coding using a fixed table, dynamic Huffman encoding reads the frame data twice. The first pass generates the frame dependent basecode table from a binary tree build by the symbol histograms and the second pass uses this table to encode the data. Although the entropy table is dynamically assigned in each frame, this extra bandwidth is quite small comparing to the overall amount of data in a frame, and the coding efficiency is greatly enhanced to deal with the uncertain distribution of the wavelet differential frame.

Due to mean shifting and quantization, a large portion of the coefficients are zeroed out, and run lengths of zeros in a subband are counted. Thus, the entropy coding is computed on the basis of the statistic distributions of the current frame data including run lengths, motion vectors, and the non-zero coefficients. The encoder treats the categories of all these values as symbols and assigns unique codewords to classify them. In the first pass of the encoding, the occurrence of successive zero-valued coefficients is counted as 1 and its run length is counted as a non-zero coefficient. The symbol histograms are categorized by powers of 2, starting from 0 to a final category, which contains the maximum value of all the data, as:

$$[0][-1,1][-3,-2,2,3][-7 \sim -4,4 \sim 7].....[-2^K + 1 \sim -2^{K-1}, 2^{K-1} \sim 2^K - 1]$$

The categories are designed in this way so that a value in category  $i$  needs  $i$  more bits to be represented after the basecode. Once the last non-zero coefficient is reached in a subband, an end of block (EOB) symbol is counted into the last histogram category. Since some intermediate categories may contain no coefficients, category and its corresponding basecode are transmitted as a pair to the decoder without empty categories included.

### 5.3.1 Motion vector coding

If the best matching blocks are found in both the current and its reference subband, the differential motion vector, which include the horizontal and vertical refinements are coded; if the best matching block is found in the current level but not in its reference subband, real motion vectors are coded. In both cases, compression mode bit of 1 is assigned in front of the motion vector, and the quantized differential data is to represent intermode compression. Zero-valued motion vectors are expressed by only the basecode of category zero. Non-zero motion vectors are coded in the same way as a non-zero coefficient. If the best matching block is not found, compression mode bit of 0 is assigned followed by the intramode compressed data of this block.

The differential motion vector coding for the 2-level decomposed subframes and subbands are demonstrated in Table 5.2. The motion vectors of  $LL$  subbands are necessary to act as the refinement references of the  $LL$  subbands in higher levels and the detail subbands at the same level. Except for the  $LL$  subband of the lowest level that is

center-based, the motion vector refinements of other  $LL$  subbands are referred to the doubling of the motion vectors from the lower  $LL$  subbands. If the reference motion vector does not exist, real motion vectors are coded since the center-biased full search is applied.

Table 5.2. Differential motion vector coding in wavelet domain  
motion estimation and compensation

Subband	$LL_2$	$HL_2$	$LH_2$	$HH_2$	$LL_1$	$HL_1$	$LH_1$	$HH_1$
Reference motion vector	0	$LL_2$	$LL_2$	$LL_2$	$2LL_2$	$LL_1$	$LL_1$	$LL_1$

### 5.3.2 Coefficient coding

After the coefficients and motion vectors are all counted, a basecode lookup table is generated. The runs of zeros are substituted by the basecode of zero, the basecode of the run length category and the binary value of the run length. A positive coefficient is coded by its category basecode and the binary representation of its value. A negative coefficient is expressed by the category basecode and the binary complement of its absolute value. Each color subband is ended up by an end of block (EOB) symbol to replace the long runs of zeros after the last non-zero coefficient. If all the coefficients in a subband are zero, only an EOB is coded.

The wavelet domain video compression is performed basically as follows. Both the reference and the current frame from sequence “football” are 2-level decomposed as shown in Figure 5.6. Multiresolution motion estimation is performed on these subframe pairs using 2-pixel full search in the lowest resolution level and 1-pixel refinements. The obtained motion vector fields as well as the intramode compressed blocks are illustrated



in Figure 5.7. The prediction error frame histograms of  $Y$ ,  $C_r$ , and  $C_b$  components are shown in Figure 5.8. Most of the coefficients after motion compensation are distributed around zero. Uniform quantization is applied on all color components, which generates centralized histograms for entropy coding. The current frame is compressed by about 50 times at bit rate of 0.5bpp with reconstruction quality PSNR of 29.2dB as shown in Figure 5.9.

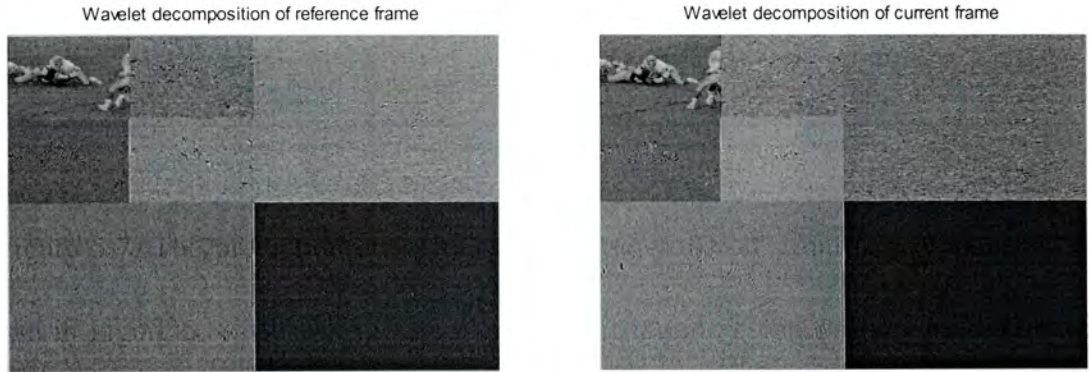


Figure 5.6. Wavelet decomposition of reference frame and current frame

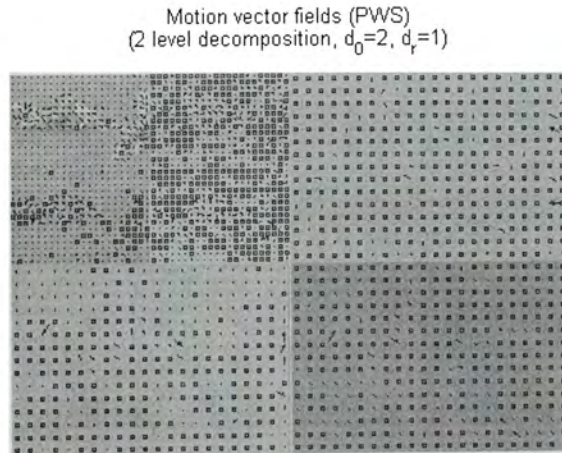


Figure 5.7. Motion vector fields of wavelet domain motion estimation and compensation for "football"

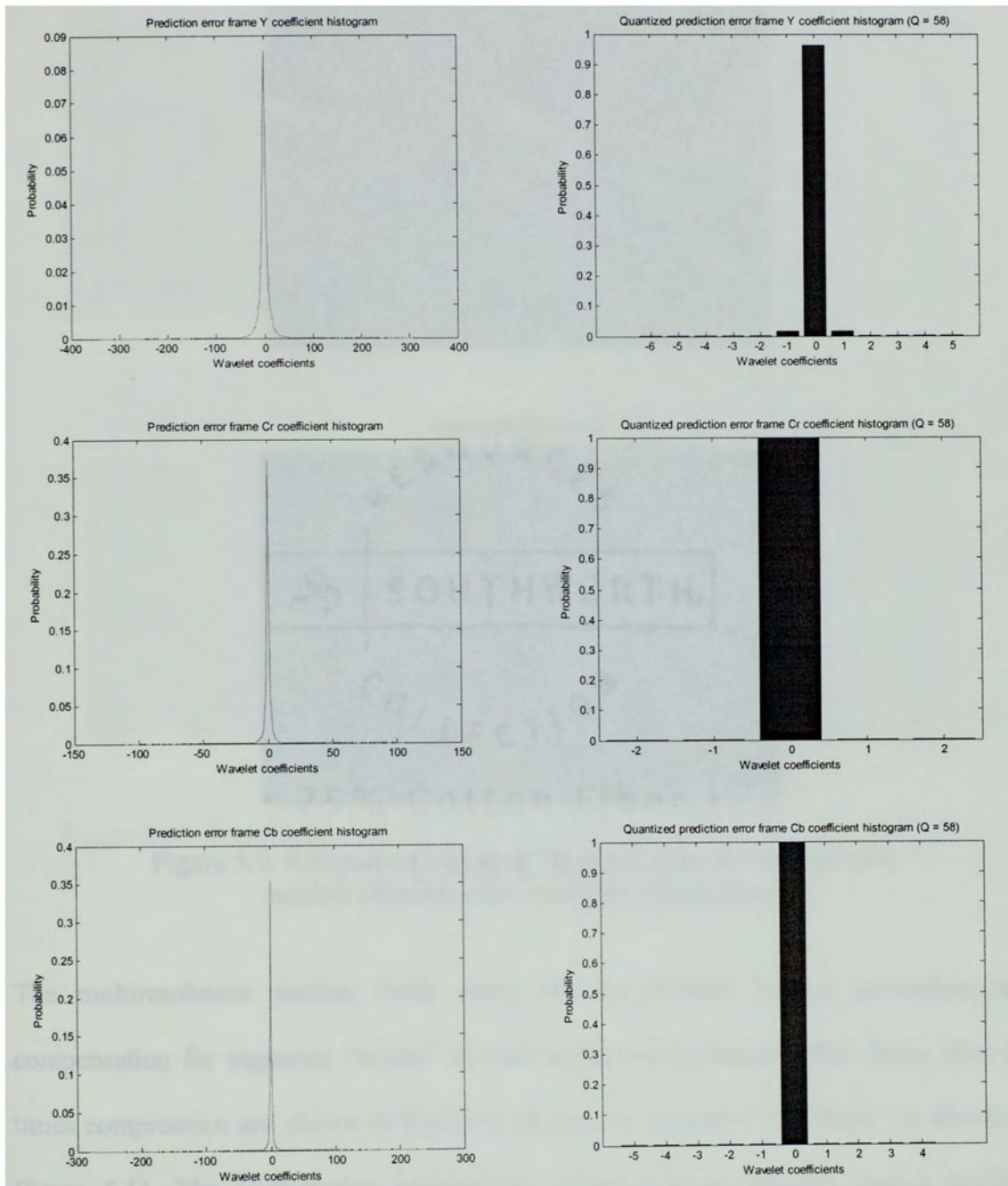


Figure 5.8.  $Y, C_r, C_b$  prediction error frame histograms before and after quantization

Reconstruction from 2 level PWS MEMC  
Bitrate = 0.516(bpp) PSNR = 29.226(dB)



Reconstruction error



Figure 5.9. Reconstruction frame "football" from wavelet domain motion estimation and compensation method

The multiresolution motion fields using wavelet domain motion estimation and compensation for sequence "trover" as well as the reconstructed video frame after 50 times compression are shown in Figure 5.10, and for sequence "carphone" is shown in Figure 5.11. The compression performance shows that the wavelet domain MEMC method is close to or better than the spatial domain FS method in a range of compressions for frame "trover" as shown in Figure 5.12 and for "carphone" as shown in Figure 5.13.



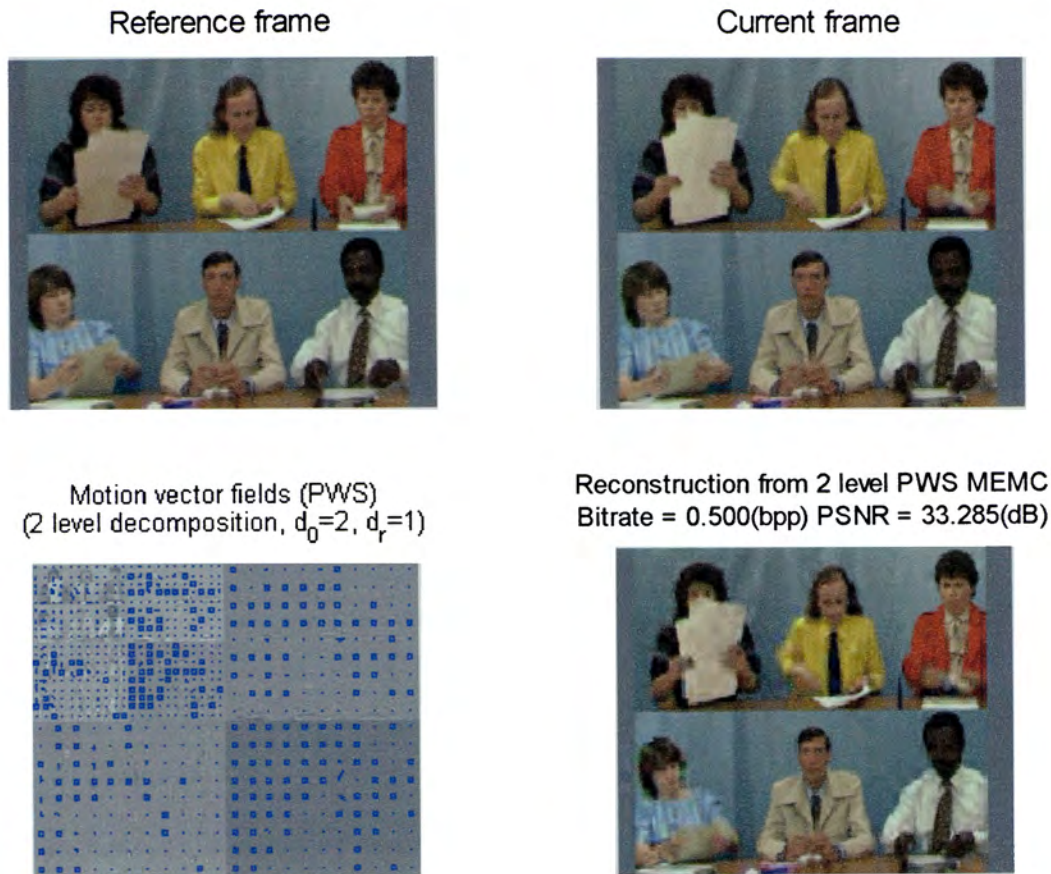


Figure 5.10. Wavelet domain compression on frame "trover" using multiresolution motion estimation and compensation

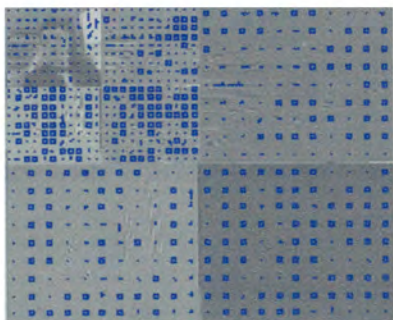
Reference frame



Current frame



Motion vector fields (PWS)  
(2 level decomposition,  $d_0=2$ ,  $d_1=1$ )



Reconstruction from 2 level PWS MEMC  
Bitrate = 0.510(bpp) PSNR = 35.135(dB)



Figure 5.11. Wavelet domain compression on frame "carphone" using multiresolution motion estimation and compensation

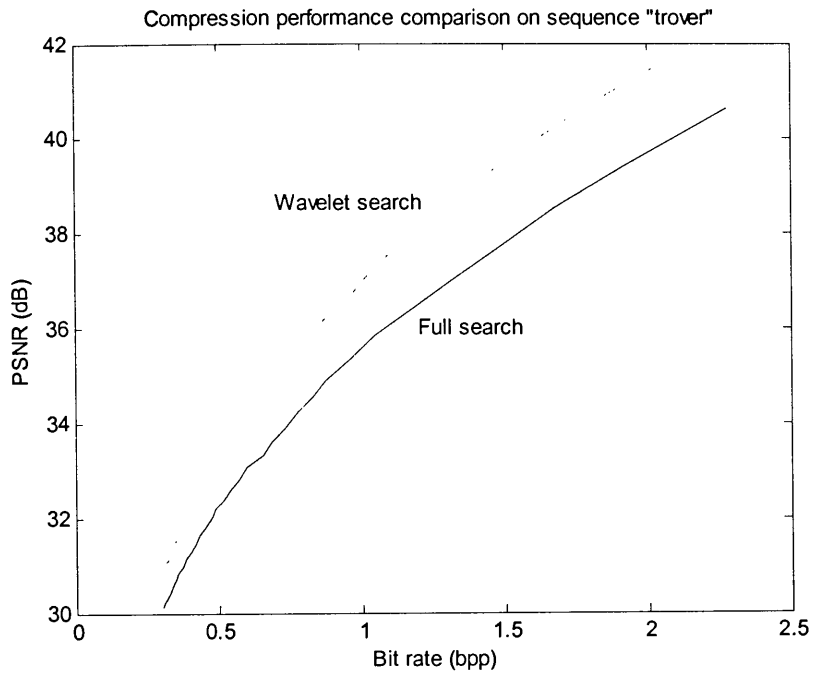


Figure 5.12. Wavelet domain search and spatial domain FS MEMC on frame "trover"

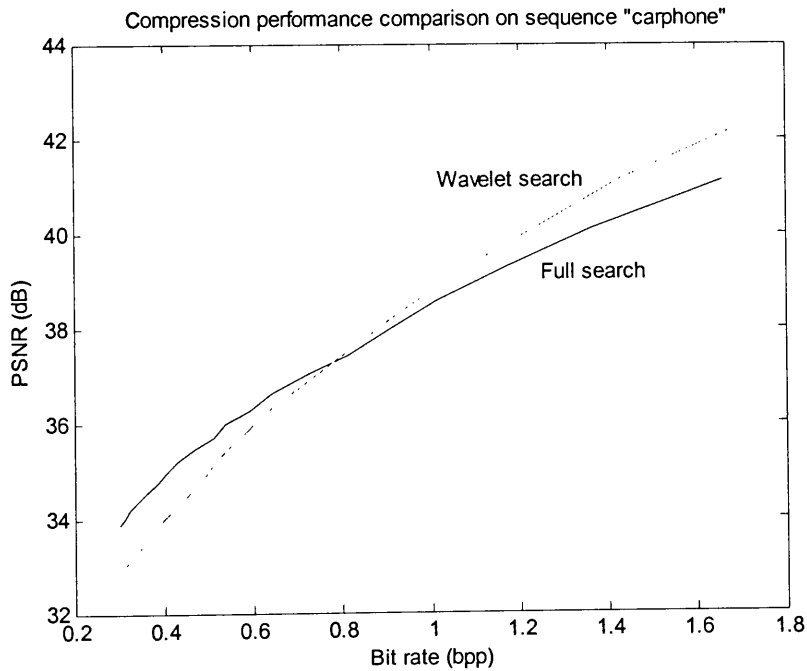


Figure 5.13. Wavelet domain search and spatial domain FS MEMC on frame "carphone"

## 5.4 Simplified wavelet domain motion estimation and compensation methods

The refinement mechanism of wavelet domain motion estimation method is based on the motion similarities between the low frequency and high frequency subbands. As illustrated in Figure 5.14, the correlations between the LL subframes of different levels are actually low frequency signals with more details; while the correlations between the approximation subband and the detail subbands are actually adjacent bandpass signals. Therefore, the motion estimation procedure can be simplified by these correlations to save the computational load and motion information bandwidth.

Firstly, the motion vectors are refined in each level by the correlations between the LL subframes, as in the level-refined motion estimation method. Secondly, the motion vectors are refined in each subband based on the close motion structures in the passband neighbors, as in the subband-refined motion estimation method.

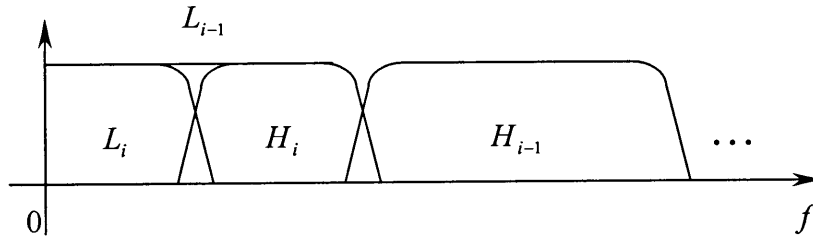


Figure 5.14. Frequency coverage of the lowpass and highpass subbands between levels

### 5.4.1 Level-refined motion estimation

In the level-refined (LR) motion estimation method, the motion vectors of LL subframe in a higher resolution level are refined by those from the lower resolution LL subframe after doubling, and represent the motion vectors for the three detail subbands at the same

level. Thus, one motion vector array is shared by all the subbands in a level without extra motion estimation computation and associated motion information. For an  $N$ -level decomposed motion estimation, the motion structure between two levels is refined as:

$$LL_i \rightarrow LL_{i-1} \quad (i = 2 \sim N) \quad (5.9)$$

where the right side subband is taking refinement reference from the left side. A 2-level motion field refinements using level-refined motion estimation method is shown in Figure 5.15, where the refined motion vector

$$MV_1 = 2MV_2 + \Delta MV_1 \quad (5.10)$$

is also used in  $HL_1$ ,  $LH_1$  and  $HH_1$  subbands for motion compensation.

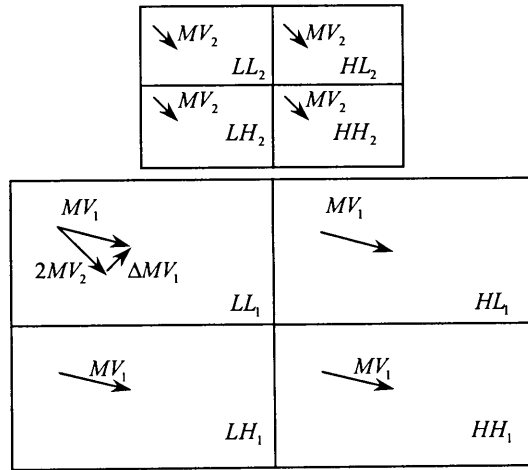


Figure 5.15. Motion field refinements in level-refined motion estimation method

The detail subbands use the same motion structure as the LL subbands without further processing and coding. The relationships of the motion field refinements and the differential motion vector coding with respect to their reference subbands are illustrated in Table 5.3.



Table 5.3. Motion vector refinements and coding  
in level-refined motion estimation method

Subband	$LL_2$	$HL_2$	$LH_2$	$HH_2$	$LL_1$	$HL_1$	$LH_1$	$HH_1$
Reference subband	N/A	$LL_2$	$LL_2$	$LL_2$	$LL_2$	$LL_1$	$LL_1$	$LL_1$
Reference motion vector	0	N/A	N/A	N/A	$2LL_2$	N/A	N/A	N/A

Since no subband refinement is performed, motion compensation in the detail subbands is applied by the motion vectors obtained from the  $LL$  subframe. The maximum displacement in the lowest resolution level is assumed  $d_0$ -pixel using full search method after  $N$ -level wavelet decomposition, and the refinements in other levels are  $d_r$ -pixel, the equivalent search displacement  $d_{WS\_LR}$  in frame resolution is:

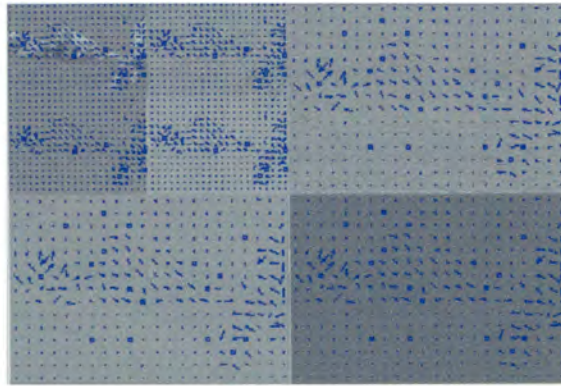
$$d_{WS\_LR} = 2^N d_0 + \sum_{i=1}^{N-1} 2^i d_r = 2^N d_0 + (2^N - 2)d_r, \quad (5.11)$$

and the computational load of a current block in all levels of the  $N_1 \times N_2$  full frame using level-refined motion estimation method is:

$$\begin{aligned} L_{WS\_LR} &= \frac{N_1}{2^N} \cdot \frac{N_2}{2^N} (2d_0 + 1)^2 + \sum_{i=1}^{N-1} \frac{N_1}{2^i} \cdot \frac{N_2}{2^i} (2d_r + 1)^2 \\ &= L_B \left\{ \left(\frac{1}{4}\right)^N (2d_0 + 1)^2 + \frac{1}{3} \left[1 - \left(\frac{1}{4}\right)^{N-1}\right] (2d_r + 1)^2 \right\} \end{aligned} \quad (5.12)$$

assuming that the motion vectors are obtained in all the lower resolution levels. An experiment on sequence “football” using 2-pixel full search in the 2<sup>nd</sup> level  $LL$  subframe and 1-pixel refinement in the 1<sup>st</sup> level  $LL$  subframe is shown in Figure 5.16. The motion fields of the subbands at the same level are identical. The resulting signal to noise ratio of the reconstructed frame is 27.3dB after about 50 times compression.

Motion vector fields (level-refined)  
(2 level decomposition,  $d_0=2$ ,  $d_1=1$ )



Reconstruction from 2 level level-refined MEMC  
Bitrate = 0.508(bpp) PSNR = 27.292(dB)



Figure 5.16. Wavelet domain compression using level-refined motion estimation and compensation method

#### 5.4.2 Subband-refined motion estimation

In subband-refined (SR) motion estimation method, motion estimation is developed by the correlated motion structures between the successive frequency subbands with refinement mechanism in both directions. The search starts from the LL subframe of the lowest resolution level. The obtained motion vector array is used as the initials for HL, LH and HH subbands at the same level:

$$MV_{n_N} = MV_N + \Delta MV_{n_N} \quad (5.13)$$

where  $n = \{HL, LH, HH\}$  subbands and  $MV_{n_N}$  is the motion vector of subband  $n$  at the lowest level;  $MV_N$  is the motion vector of the initial LL subframe. The motion vectors of higher levels are refined by those from the same orientational subband in the lower levels after doubling, as:

$$MV_{n_{i-1}} = 2MV_{n_i} + \Delta MV_{n_{i-1}} \quad (5.14)$$

where  $i = 2 \sim N$ .  $MV_{n_i}$  is the motion vector of subband  $n$  at the  $i^{th}$  decomposition level.

Thus, for an  $N$ -level decomposed motion estimation, the motion vector refinements between the subbands are derived as:

$$\begin{aligned} LL_N &\rightarrow HL_N, LH_N, HH_N \\ HL_i &\rightarrow HL_{i-1} \\ LH_i &\rightarrow LH_{i-1} \\ HH_i &\rightarrow HH_{i-1} \end{aligned} \quad (5.15)$$

where the right side subband is taking refinement reference from the left side. A 2-level motion estimation by subband-refined method is illustrated in Figure 5.17.

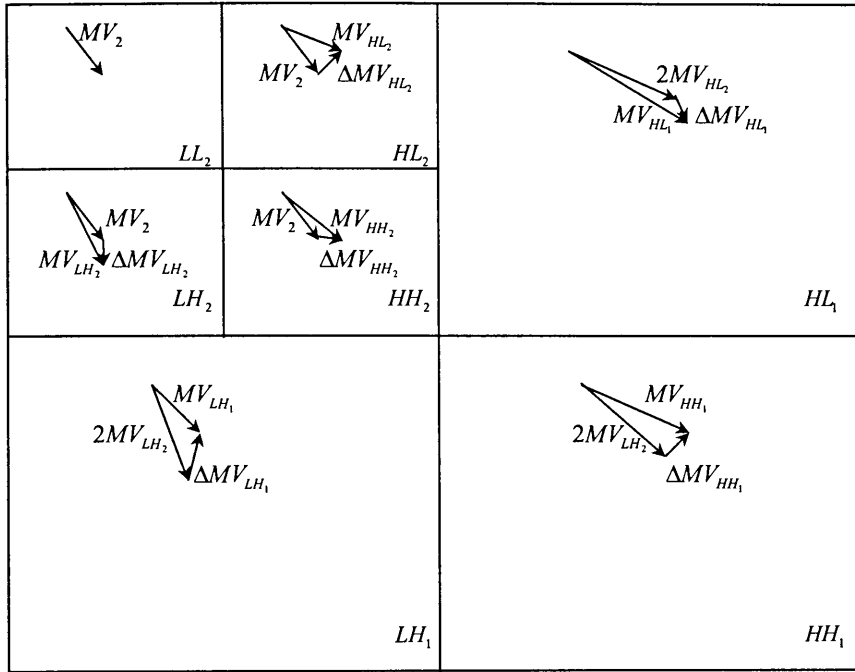


Figure 5.17. Wavelet domain subband-refined motion estimation

Table 5.4 shows the relationships between the subbands for motion vector refinements and the differential motion vector coding.

Table 5.4. Motion vector refinements and coding in subband-refined motion estimation and compensation method

Subband	$LL_2$	$HL_2$	$LH_2$	$HH_2$	$LL_1$	$HL_1$	$LH_1$	$HH_1$
Reference subband	N/A	$LL_2$	$LL_2$	$LL_2$	N/A	$HL_2$	$LH_2$	$HH_2$
Reference motion vector	0	$LL_2$	$LL_2$	$LL_2$	N/A	$2HL_2$	$2LH_2$	$2HH_2$

Assume the maximum displacement in the lowest resolution level is  $d_0$ -pixel using full search method after  $N$ -level wavelet decomposition, and the refinements in other subbands are  $d_r$ -pixel, the equivalent search displacement  $d_{ws\_sr}$  in frame resolution is:

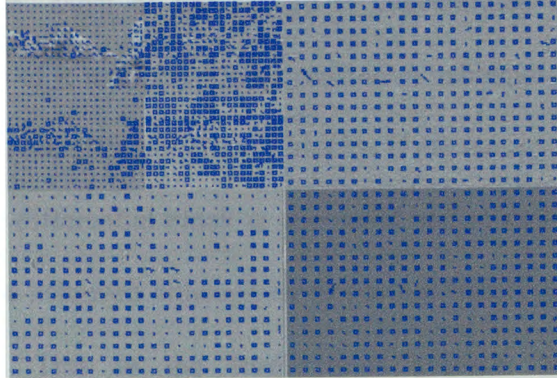
$$d_{ws\_sr} = 2^N d_0 + \sum_{i=1}^{N-1} 2^i d_r = 2^N d_0 + (2^N - 2) d_r \quad (5.16)$$

and the computational load of a current block in all the levels of a full frame  $N_1 \times N_2$  using subband-refined motion estimation method is:

$$\begin{aligned} L_{ws\_sr} &= \frac{N_1}{2^N} \cdot \frac{N_2}{2^N} (2d_0 + 1)^2 + 3 \times \sum_{i=1}^N \frac{N_1}{2^i} \cdot \frac{N_2}{2^i} (2d_r + 1)^2 \\ &= L_B \left\{ \left(\frac{1}{4}\right)^N (2d_0 + 1)^2 + \left[1 - \left(\frac{1}{4}\right)^N\right] (2d_r + 1)^2 \right\} \end{aligned} \quad (5.17)$$

assuming that the motion vectors are available in each of the reference subbands. Figure 5.18 shows the motion fields using 2-pixel full search in the lowest LL subframe and 1-pixel refinements in the detail subbands. The reconstructed frame is also shown for about 50 times compression at bit rate about 0.5bpp with PSNR of 29.34dB.

Motion vector fields (subband-refined)  
(2 level decomposition,  $d_0=2$ ,  $d_r=1$ )



Reconstruction from 2 level subband-refined PWS  
Bitrate = 0.516(bpp) PSNR = 29.339(dB)



Figure 5.18. Wavelet domain compression using subband-refined motion estimation and compensation method

### 5.5 Level-refined motion estimation and subband compensation method

In level-refined wavelet motion estimation method, motion compensation in the detailed HL, LH and HH subbands are using the same positioned compensation blocks in order to save the motion information bandwidth and computational load of the motion field refinements. As illustrated in subband-refined method, the increased intrablocks in high

resolution levels after motion compensation indicates the energy distributions in the decomposed subbands. Most of the energy is concentrated in the lower resolution subbands, while little of it is in the higher resolution subbands, which is possibly even less than that found in the differential error blocks. If the energy comparison between the current block and the differential block is computed before motion compensation, lower entropy coding can be achieved although more motion bandwidth is used to specify the compression modes. This is the level-refined motion estimation and subband compensation (LRSC) algorithm. If the energy of the differential block is larger than the current block, the compression mode bit is set as 1 and intermode compression is coded with the corresponding motion vectors. Otherwise, the compression mode bit is set as 0 to represent intramode compression on the current block. If the motion vector in a LL subframe is not available, the assumption is that all the detail subbands at the same level are using intramode compression.

Actually, the intramode compression of the current block in the detail subbands does not change the motion field reference for the next level refinements, but it will in the subband-refined method and cost the motion estimation computational load by reinitializing the full search around the stationary center. Therefore, based on the level-refined method that only one motion vector array is shared by all the subbands in this level, LRSC method contains an extra compression mode bit array to distinguish intramode or intermode compression in each detail subband. Indeed, the bandwidth of compression mode array by only one bit for each current block in each level is quite small comparing to a block of data. The LRSC method illustrates the efficiency of

wavelet domain multiresolution motion estimation and compensation instead of the conventional approaches.

In LRSC method, the motion vector refinements in each level and the differential motion vector coding remain the same as the level-refined method in Table 5.3. The computational loads are also identical in terms of motion estimation. The additional computation from LRSC method is the comparison of the energies contained in the current block and the differential block before motion compensation. The computational loads  $L$  including the current blocks in all decomposition levels by the introduced wavelet-domain motion estimation methods with respect to the search displacement  $d$  are listed in Table 5.5 and plotted in Figure 5.19, where a 2-level decomposition with refinement  $d_r = 1$  pixel is used.

Table 5.5. Computational load comparison of a current block motion estimation using wavelet domain methods

Displacement	Computational load (in unit of $L_B$ )			
	$d_0 = 1$	$d_0 = 2$	$d_0 = 3$	$d_0 = 4$
$d_{WS}$	10	14	18	22
$L_{WS}$	11.25	12.25	13.75	15.75
$d_{WS\_LR} (d_{WS\_LRSC})$	6	10	14	18
$L_{WS\_LR} (L_{WS\_LRSC})$	2.81	3.81	5.31	7.31
$d_{WS\_SR}$	6	10	14	18
$L_{WS\_SR}$	9	10	11.5	13.5



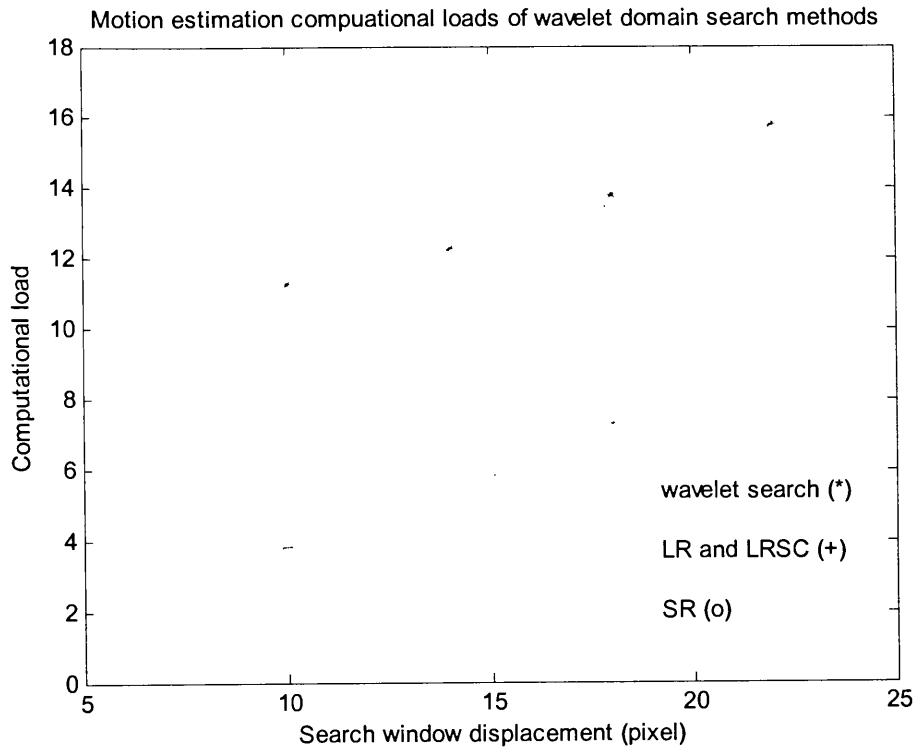


Figure 5.19. Computational loads of motion estimation in wavelet domain search methods

Figure 5.20 shows the motion fields and compression modes of a 2-level wavelet domain motion estimation and compensation using LRSC method for frame “football” and the resulting reconstructed frame with PSNR about 29.3dB at bit rate of 0.5bpp.

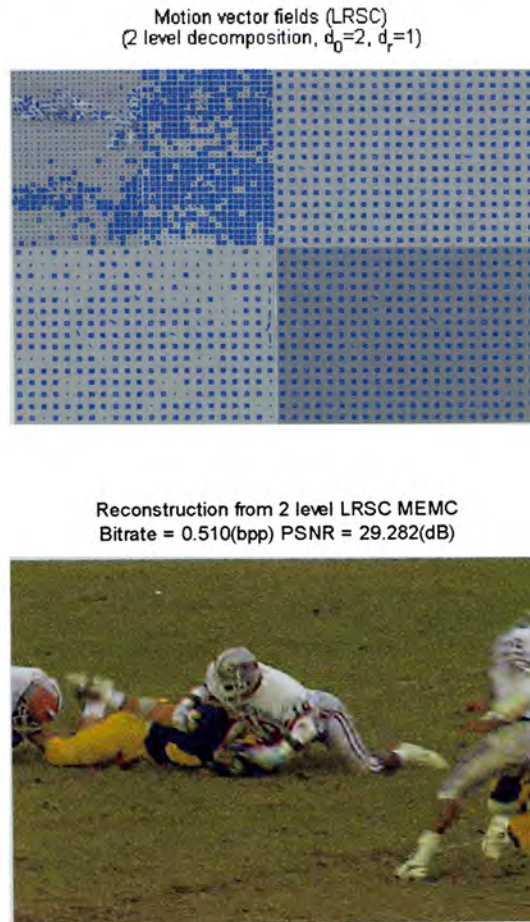


Figure 5.20. Wavelet domain compression using level-refined motion estimation and subband compensation method

The compression performances of wavelet search, level-refined, subband-refined and LRSC methods at low bit rates of about 0.1~2bpp are compared in Figure 5.21 for sequence “football” and in Figure 5.22 for sequence “carphone”. The quality of compression is inverse to the computational load of motion estimation. The results demonstrate that the compression performance of LRSC method is close to or better than other wavelet domain search methods, while the motion estimation process contains only about  $\frac{1}{3}$  of the computational complexity.

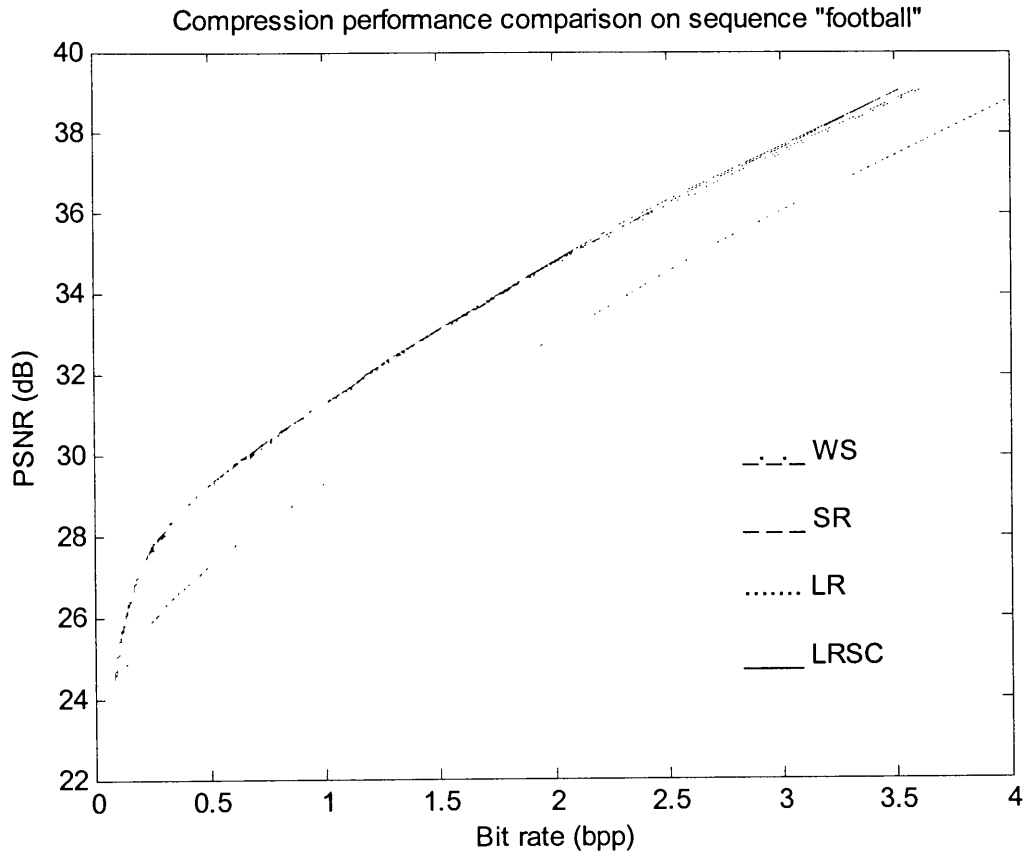


Figure 5.21. Wavelet domain MEMC comparisons on frame "football"

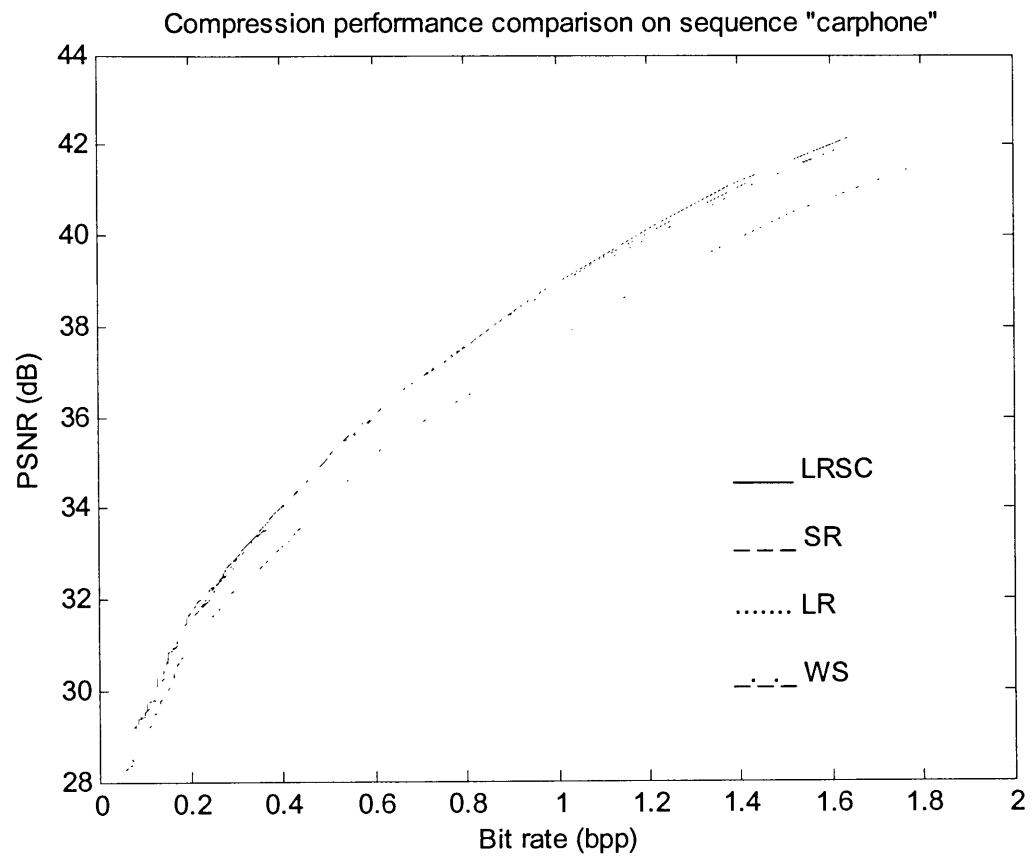


Figure 5.22. Compression comparison of sequence "carphone" by wavelet domain MEMC methods

## CHAPTER 6

### Symmetric-Extended Wavelet Transform

The underlying task in wavelet transform is the convolution between the decimated signal and the wavelet filters. Since image signals are not continuous at the boundaries, problems of coefficient expansion and boundary distortion are faced in the implementation of the filtering on the finite length signal. Circular convolution instead of linear convolution can eliminate coefficient expansion but introduce boundary artifacts, especially when more levels of decomposition are involved. Symmetric-extended wavelet transform (SWT) is thus introduced in this chapter to address the boundary artifacts with improved performances in wavelet domain image and video compressions.

#### 6.1 Symmetric-extended signal

Assuming that a signal to be processed has finite length  $L$ . A whole point symmetry extension  $W_s$  of the signal means that all samples except for the two end points are repeated as a period before a signal is said to be circularly extended. Each period thus has length  $2L - 2$ .  $W_s$  extension is also called (1,1) mode extension. Figure 6.1 shows the original signal samples which are in black dots, and its  $W_s$  extension by performing the symmetric extension without the first and last samples as the gray dots and then periodically extended as the white dots.

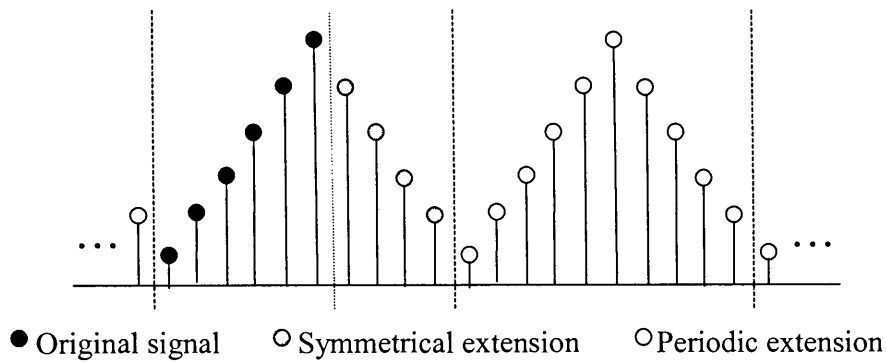


Figure 6.1. W extension of the finite-length signal

A half point symmetry extension  $H_s$  of the signal is that the whole length of the original signal are repeated before it is periodically extended. Each period now has length  $2L$ .  $H_s$  extension is also called (2,2) mode extension. As shown in Figure 6.2, the original signal as the black dots are all symmetrically extended into one period as the gray dots and circularly extended as the white dots.

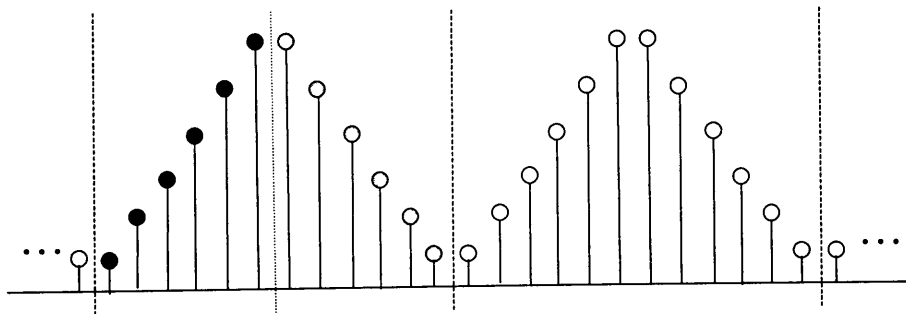


Figure 6.2. H extension of the finite-length signal

If the signal is symmetrically extended except for the first sample point as a period and then circularly extended, it's called a (1,2) mode extension of the signal. On the other hand, if the signal is symmetrically extended except for the last sample point as a period

before the circular extension, it's called a (2,1) mode extension of the signal. Both (1,2) and (2,1) mode extensions have  $2L - 1$  samples in one period as shown in Figure 6.3 and 6.4, respectively.

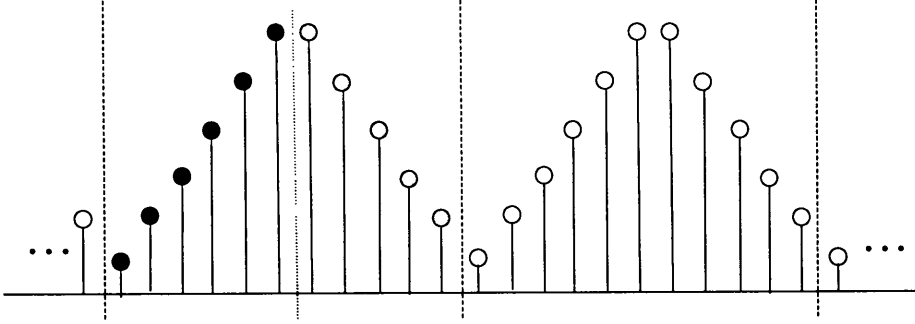


Figure 6.3. (1,2) mode extension of the finite-length signal

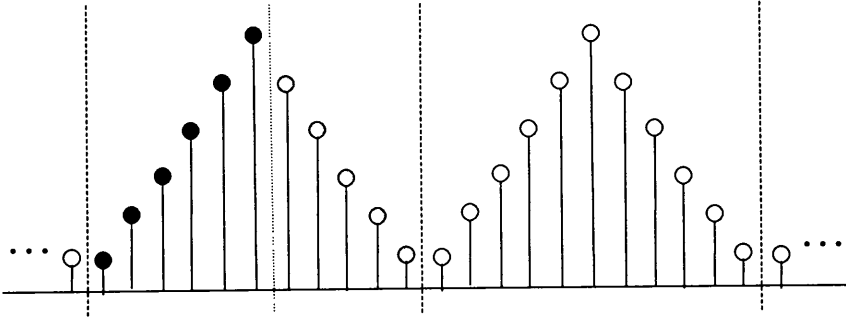


Figure 6.4. (2,1) mode extension of the finite-length signal

## 6.2 Symmetric-extended wavelet transform

Linear convolution of a signal in length  $L_s$  by a filter in length  $L_f$  generates the filtered signal in length  $L_s + L_f - 1$ , which is expanded with respect to the original signal. To realize multilevel decomposition by keeping half length in the decomposed signals, discrete wavelet transform using circular convolution is applied. Although the coefficient

expansion problem is solved, boundary distortion problem still exists because of the jumping points on both ends of the signal from the periodic padding in circular convolution. Thus, the solution to these problems requires that the decomposed signal remains in half length even after circular convolution with the expanded signal to deal with the boundary artifacts.

To smooth the boundary discontinuity, the signal is first symmetrically extended before the periodic repeating to perform circular convolution. It's known that a symmetric signal has linear phase, and the resulting signal is also symmetric when convolved with linear phase filters [23]. Thus, the combination of the symmetric extension of the signal and the biorthogonal symmetric wavelet filter is an effective approach to reducing the boundary distortion after half resolution decomposition. The main concern is that the signal from wavelet transform might not be symmetric after decimation, so the condition of perfect reconstruction in half resolution cannot be realized. Therefore, the extension and decimation need to be desired in such a way that the approximation and detail signals after decimation are still symmetrically extended, so that the first half truncation as the decomposed signal is possible.

A digital filter is defined as a  $W_f$  filter if its impulse response is a finite odd-length symmetric sequence. Furthermore, a digital filter is defined as an  $H_f$  filter if its impulse response is a finite even-length symmetric sequence. Observe from Figure 6.5 that half decimation from  $W_s$  signal is also a symmetric-extended signal in (1,2) extension mode as the black samples or in (2,1) extension mode as the white samples.



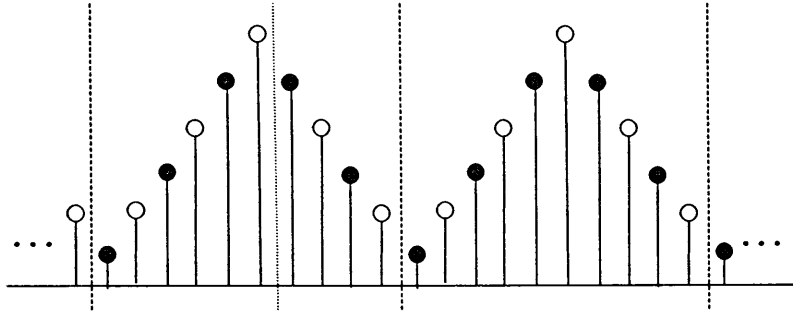


Figure 6.5. Half decimation on  $W$  symmetric-extended signal

But the half downsampling from  $H_s$  signal does not have symmetric properties as illustrated in Figure 6.6 for both the white and black decimated samples.

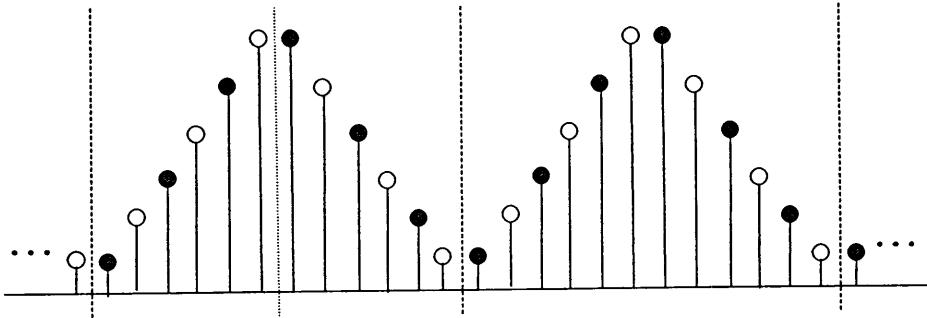


Figure 6.6. Half decimation on  $H$  symmetric-extended signal

Since the original signal is almost doubled in length by the symmetric extension before circular convolution with the wavelet filters, the decimated signals need to be symmetric, and half length decomposition should be made possible for perfect reconstruction. Thus, the wavelet filtered signal needs to be a  $W_s$  signal so that the half decimations are still symmetric. To obtain a  $W_s$  signal after circular convolution with the biorthogonal filter, the symmetry mode of the extended signal should be matched with that of the applied

filter. That is to say, for wavelet transformation using a  $W_f$  filter, the signal is  $W_s$  extended and made periodic; and for transformation using an  $H_f$  filter, the signal is  $H_s$  extended and made periodic. In such a way, the decimation on the convolved  $W_s$  signal will be in (1,2) or (2,1) extension mode, then only the first half samples need to be saved in the decomposition for perfect reconstruction.

The symmetric-extended wavelet transform using the casual biorthogonal 9/7 filter is shown in Figure 6.7. For this odd-length symmetric  $W_f$  filter, the finite-length signal is  $W_s$  extended into length  $2L - 2$  as the input to the wavelet transform including filtering and downsampling. The convolution is aligned on the center of the filter to the first sample point in a period of the signal. The wavelet transform decomposes the  $W_s$  signal into the approximation signal in (1,2) extension mode from the lowpass branch and the detail signal in (2,1) extension mode from the highpass branch. Truncating these symmetric signals by keeping the first half will achieve the half resolution outputs with removed discontinuity artifacts.

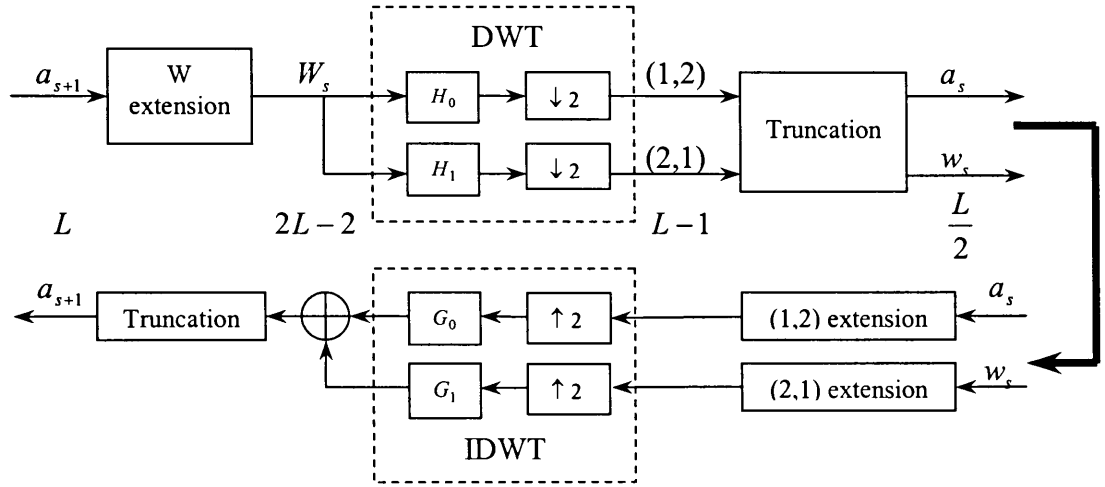


Figure 6.7. One dimensional SWT and inverse SWT using biorthogonal 9/7 filter

The inverse symmetric-extended wavelet transform is performed by making the corresponding extension before applying the inverse wavelet transform and truncated by the original length as the reconstructed signal. It has better performance dealing with the boundary artifacts than the traditional periodic-extended convolution, trading-off with more computation since the signal is almost doubled after extension. But this process is necessary, especially for compression and hierarchical motion estimation when the subframe size gets smaller and convolving with the wavelet filter.

### 6.3 Symmetric-extended wavelet transform on image compression

From the computational investigation, the lowpass signal from symmetric-extended wavelet transform by a  $W_f$  filter is (1,2) extended and the highpass signal is (2,1) extended, and the symmetric properties of the two-dimensional image signal by the symmetric-extended wavelet transform in horizontal and vertical directions are as

indicated in Table 6.2. This is required for each half-resolution subband so that the synthesis part of the filter bank can make the corresponding extensions before inverse wavelet transform for perfect reconstruction.

Table 6.1. Image symmetric extension modes using biorthogonal 9/7 filter

Symmetric mode	LL	HL	LH	HH
Horizontal	(1,2)	(1,2)	(2,1)	(2,1)
Vertical	(1,2)	(2,1)	(1,2)	(2,1)

Compression performances using JPEG (Joint Photographic Experts Group), the conventional periodic-extended wavelet transform (PWT) and the symmetric-extended wavelet transform (SWT) are compared on image “azalea” in Figure 6.8 at bit rate of 0.5bpp after about 50 times compression. JPEG is implemented in Visual C++ using FDCT algorithm [80, 81] to compress the windows device independent bitmap (DIB) image. In order to get a better subjective performance of the decompressed image, the average luminance and chrominances are refined to be the same as in the original. The blocking artifact appears in JPEG compression but not in wavelet compression as a global transform, which brings with it better reconstruction quality.

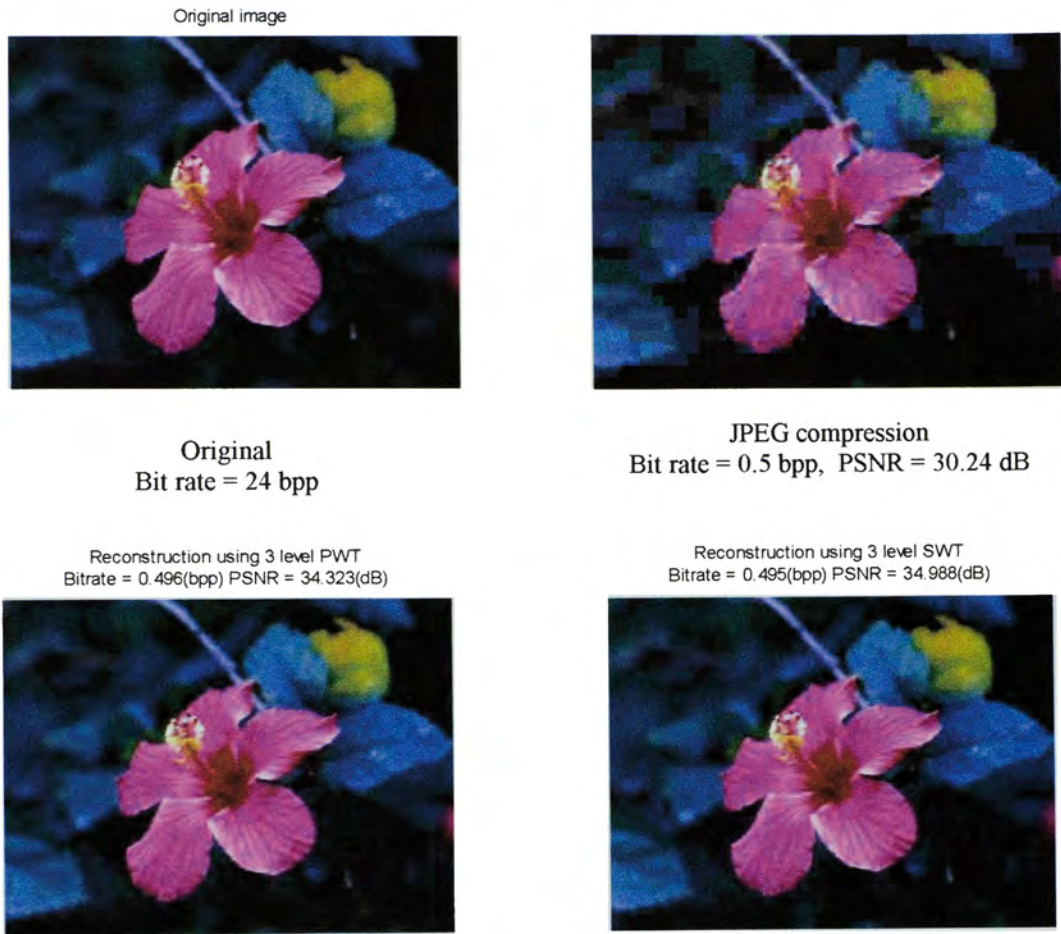


Figure 6.8. Image compression comparisons using JPEG, PWT and SWT at bit rate of 0.5bpp

Low bit rate color image compressions by JPEG, PWT and SWT are compared in Figure 6.9 for image “azalea” and in Figure 6.10 for image “parrots” in terms of the reconstruction image qualities, in which SWT is demonstrated to have the best result.

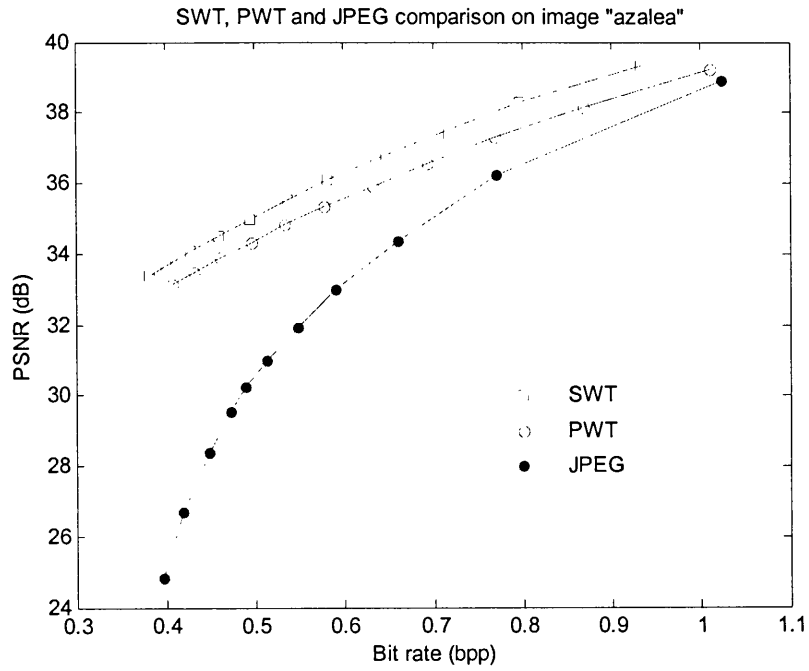


Figure 6.9. JPEG, PWT and SWT performances by bit rate vs. PSNR for "azalea"

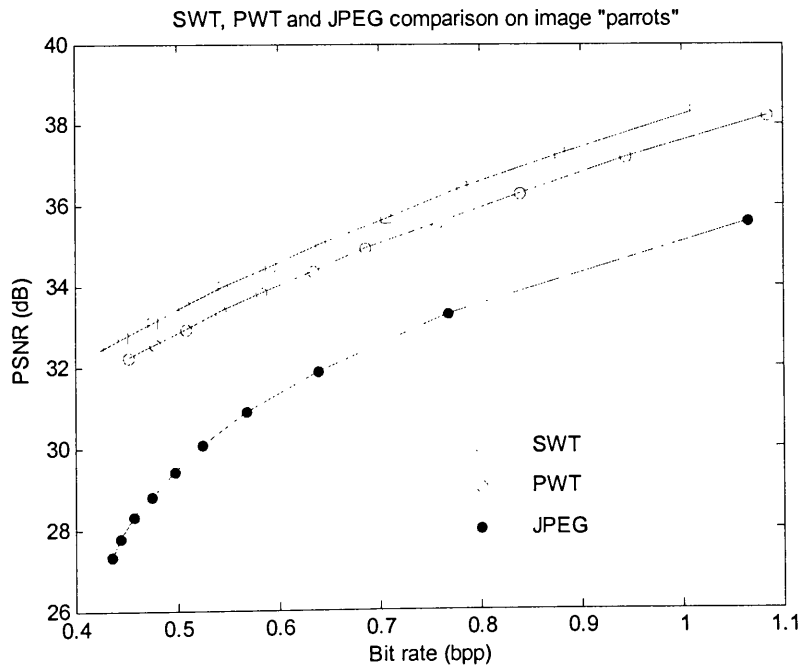


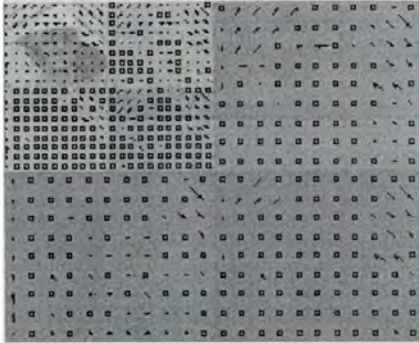
Figure 6.10. JPEG, PWT and SWT performances by bit rate vs. PSNR for "parrots"

#### 6.4 Symmetric-extended wavelet transform on video compression

In multiresolution motion estimation and compensation, when more levels of decomposition are applied, the circularly extended boundary samples convolved with the wavelet filters will generate deeper border distortions along the subframes. Symmetric-extended wavelet transform applied on video compressions are desired to improve motion estimation and compensation performances on the boundary blocks in the decomposed subframes. The improvement is more effective for the video frames that contain miscellaneous background because high frequency artifacts will appear on the boundary by the conventional wavelet transformation using discontinuous extension without the smoothing operations. Based on the LRSC motion estimation and compensation method introduced in Chapter 5, the subframes are decomposed using symmetric-extended wavelet transform to obtain better matching blocks so that the compensated frame will have lower entropy to increase the compression efficiency.

An experiment on video sequence “foreman” using symmetric-extended wavelet transform is shown in Figure 6.10 with the motion fields after multiresolution motion estimation and compensation as well as the reconstructed frame at compressed bit rate of 0.5bpp. Based on the level-refined motion estimation and subband compensation method, a 2-level wavelet decomposition is applied on the reference frame and the current frame. The full search with maximum displacement  $d_0 = 2$  pixel is used in the lowest LL subframe and the refinements are  $d_r = 1$  pixel in the higher resolution levels.

Motion vector fields (LRSC SWS)  
(2 level decomposition,  $d_0=2$ ,  $d_r=1$ )



Reconstruction from 2 level LRSC MEMC  
Bitrate = 0.512(bpp) PSNR = 32.294(dB)



Figure 6.11. Video compression on sequence "foreman" using SWT method

Video compressions using SWT-LRSC motion estimation and compensation method comparing to the PWT method on "foreman" at low bit rate range about 0.1~2bpp is illustrated in Figure 6.12, where the improvement of PSNR is above 0.5dB using symmetric-extended wavelet method.



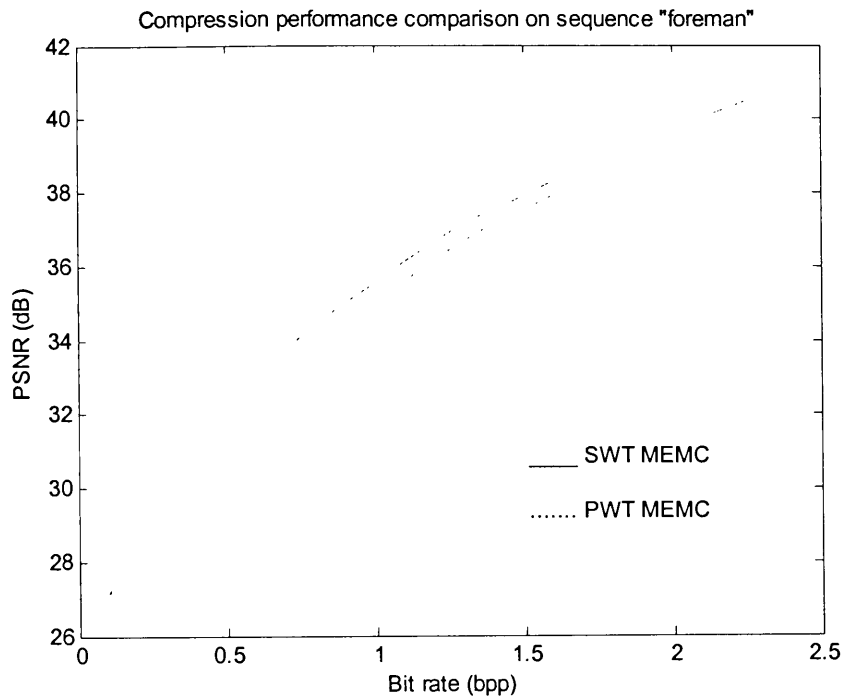


Figure 6.12. Wavelet-based video compression on sequence "foreman" using PWT and SWT motion estimation and compensation

## CHAPTER 7

### Wavelet Domain Sub-decimation Decomposition Method

#### 7.1 Sub-decimation decomposition method

Due to the decimation operation in wavelet filtering, discrete wavelet transform is shift variant. Assume the input signal to the wavelet transformer is  $x(n)$ , as shown in Figure 7.1, the output signal  $y(n)$  after one-level decomposition is determined as:

$$y(n) = \sum_k x(k)h(2n - k) \quad (7.1)$$

If the input  $x(n)$  is shifted by  $p$  samples,

$$x'(k) = x(k - p) \quad (p \in Z) \quad (7.2)$$

the output can be identically shifted only under the condition that  $p$  is an even number,

$$p = 2m \quad (m \in Z) \quad (7.3)$$

then  $y(n)$  is shifted by half of the input,

$$\begin{aligned} y'(n) &= \sum_k x'(k)h(2n - k) \\ &= y(n - m) \end{aligned} \quad (7.4)$$

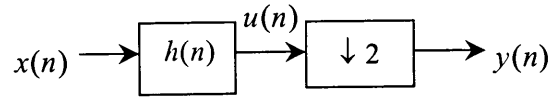


Figure 7.1. A branch of the discrete wavelet transform

Based on the condition of Equation (7.3), shift invariance requires that the shift from the input to be  $p = q \cdot 2^N$  ( $q \in Z$ ) after  $N$ -level decomposition. Thus, a dyadic wavelet

transform is shift invariant only if the shift in the time or spatial domain is a multiple of the sampling period.

Obviously from Equation (7.1), shift variance is produced due to the half decimation on the filter-convolved signal. By choosing odd or even decimation samples, the output is not alike by one level decomposition, as illustrated in Figure 7.2, and could be much more different after several levels of decomposition. Therefore, both odd and even decimations should be counted in the decomposed signal to preserve the necessary information instead of using one predefined decimation pattern only.

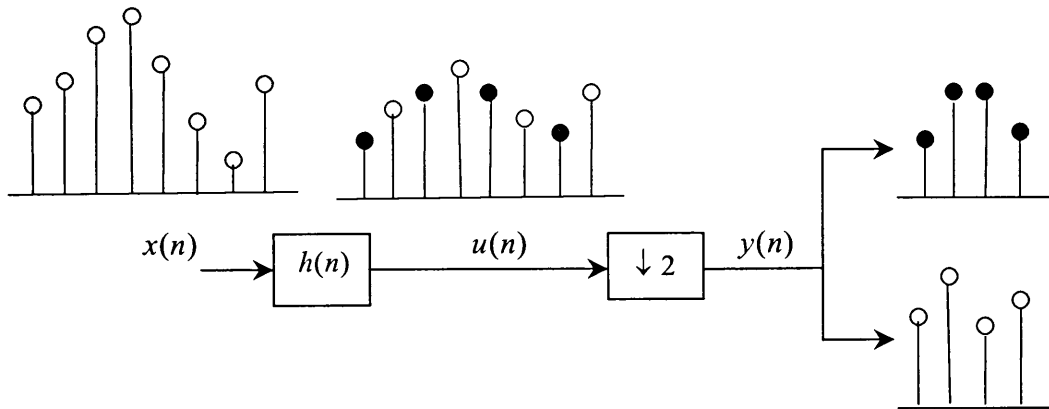


Figure 7.2. Decimation patterns on one-dimensional wavelet decomposition

In a two-channel wavelet filter bank as in Figure 7.3, after convolving with the analysis filters  $H_0$  and  $H_1$  followed by half downsampling, the lowpass filter branch generates the half-resolution approximation signal  $a$ , and the highpass branch generates the half-resolution detail signal  $w$ . To be further specified by odd and even decimation modes,

the sub-decimation decomposition keeps 2 approximations and 2 details in each level by defining a decimation parity  $P$  using one bit, as:

$$P = \begin{cases} 0 & (\text{odd} - \text{decimation}) \\ 1 & (\text{even} - \text{decimation}) \end{cases} \quad (7.5)$$

In such a way, the decomposed signals can be labeled using a layered binary representation after multiple levels of decomposition:

$$D_{L+1} = D_L + 2^L P_{L+1} \quad (0 \leq D_{L+1} < 2^{L+1}) \quad (7.6)$$

where  $D_L$  and  $D_{L+1}$  are the decimation patterns in level  $L$  and the lower higher resolution level  $L+1$ . The properties of the decimation pattern are:

- 1) The odd-decimation keeps the same decimation pattern from the last level;
- 2) The even-decimation moves the decimation pattern to  $D_L + 2^L$ ;
- 3) Totally  $2^{L+1}$  sub-decimation decomposed signals are generated in level  $L+1$ ;
- 4) Since one decimation pattern represents one decomposed subband signal,  $2^{L+1}$  approximation and  $\sum_{i=1}^{L+1} 2^i$  detail subbands are generated after  $L+1$  levels of sub-decimation decomposition.

Set the initial value  $D_0$  be 0 before decomposition, the decimation patterns of a 2-level decomposition is illustrated in Table 7.1, where  $P_1$  and  $P_2$  are the decimation parities in the 1<sup>st</sup> and 2<sup>nd</sup> decomposition level. Figure 7.3 shows the flow chart of the decomposed subband signals  $a_{L,D}$  and  $w_{L,D}$ , defined by decomposition level  $L$  and decimation pattern  $D$ .

Table 7.1. Two-level sub-decimation decomposition

1 <sup>st</sup> level decimation mode	Odd		Even	
$P_1$	0		1	
$D_1$	0		1	
2 <sup>nd</sup> level decimation mode	Odd	Even	Odd	Even
$P_2$	0	1	0	1
$D_2$	0	2	1	3

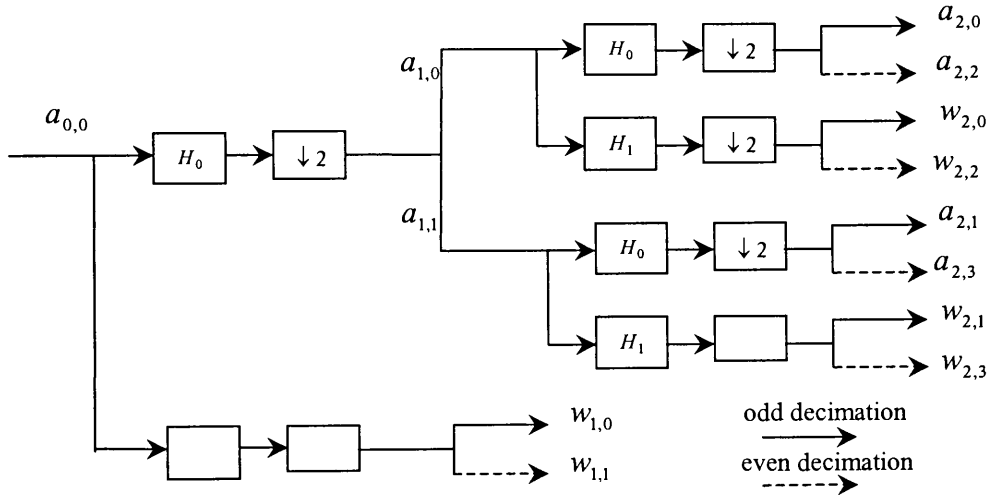


Figure 7.3. Sub-decimation decomposition in wavelet filter bank

Shift invariance is more important for nonstationary image signals due to the fact that high frequency detail information could be removed after subsampling. For two-dimensional signals, the general wavelet decomposition produces 1 approximation and 3 orientational detail subbands, while sub-decimation decomposition generates another 3 decimation subbands by applying one-dimensional SDD on both the horizontal and vertical directions respectively [46]. They are one-pixel horizontal, vertical and diagonal shift decimated from the predefined decomposed subband, as illustrated in Figure 7.4.

Therefore, one sub-decimation decomposition produces 4 approximations and 12 detail subbands, totaling  $4^N$  approximation subbands and  $3 * \sum_{i=1}^N 4^i$  detail subbands after  $N$ -level decomposition.

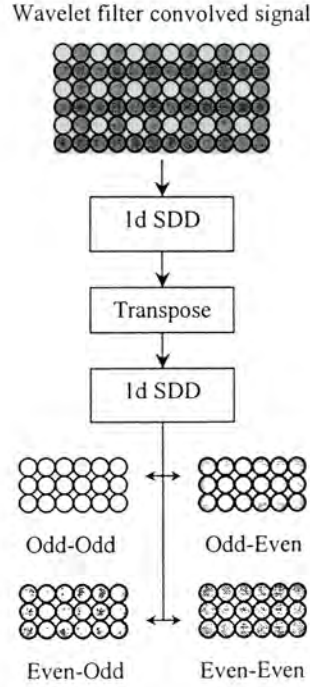


Figure 7.4. Two-dimensional sub-decimation decomposition method

## 7.2 Sub-decimation decomposition motion estimation and compensation

Wavelet-domain shift invariant motion estimation has been investigated in some literatures. One approach is to perform motion estimation on the lowpass-filtered frame before decimation to make up for the translation inconsistency [71]. Another approach is developed by the way that the reference frame is shifted by one pixel along the horizontal, vertical and diagonal directions in spatial domain before decomposition [82,

83]. In order to realize shift invariance without shifting on the discontinuous image signals, sub-decimation decomposition (SDD) motion estimation and compensation method is proposed.

If an object in the current frame is moved from a block in the reference frame by a non-multiple step of the sampling period, the decomposed coefficients of these two blocks are elusive for motion estimation in the approximation subframes even though the lowpass filtering is performed, as well as for motion compensation in the detail subbands by the rapidly changing signals. Since  $2^N$  approximation subbands are generated after  $N$ -level decomposition, even-shifted blocks in horizontal or vertical direction can only be found by  $\frac{1}{2^N}$  of the decimation patterns after  $N$ -level decomposition; and the odd shifts are undetectable without the matching discrimination. Thus, the shift variance feature is essential to the performance of wavelet domain motion estimation and compensation.

Instead of identical decompositions applied on the reference-current frame pair, the reference frame is decomposed by the sub-decimation decomposition method, while the current frame is decomposed by a predefined decimation mode. The motion vectors are obtained by motion estimation between the decomposed subframes at the same level, and motion compensations are performed in the lowest approximation subband as well as the detail subbands based on these motion vectors. A reasonable search mechanism to find the best matching subblocks with hierarchical refinements is developed, so that all the translations in the original frame can be counted from the subframes in the lowest resolution level. The procedure is mainly described into 3 steps as below.

### 1) Generation of the sub-decimation decomposed subframes on the reference frame

All the approximation subframes are decomposed level after level using sub-decimation decompositions recursively. Figure 7.5 illustrates one level decomposition from a high level approximation subband into four approximation subbands as well as their corresponding detail subbands (HL, LH, HH) by odd-odd, odd-even, even-odd and even-even decimation modes. The decimation mode is defined in vertical-horizontal directions, for example, odd-even decimation keeps odd-numbered rows and even-numbered columns after the two-dimensional filter convolution. Therefore, each sub-decimation decomposed subband can be specified by the parameters of bandwidth, decomposition level and decimation patterns.

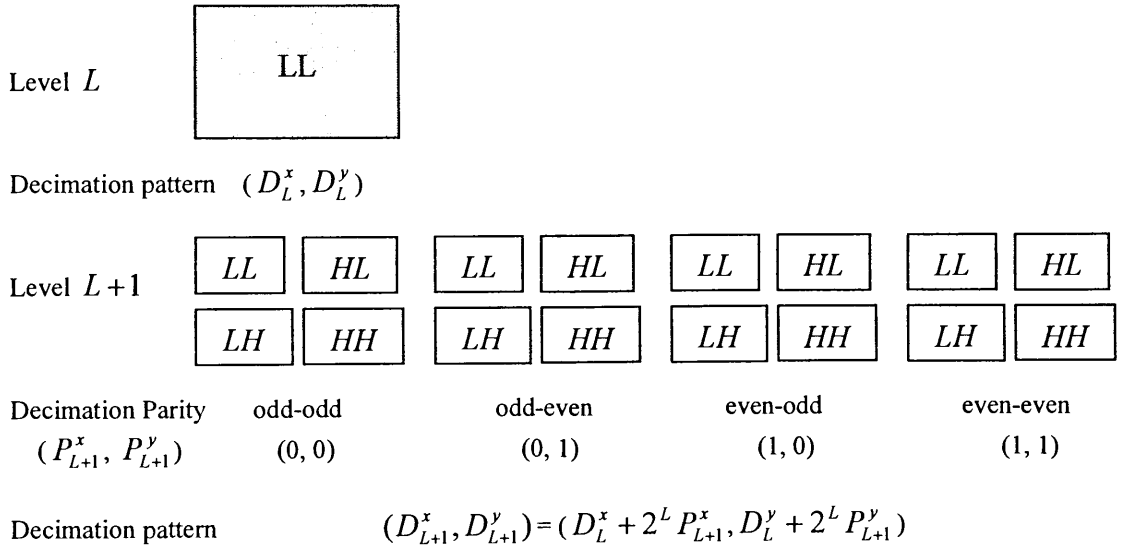


Figure 7.5. Sub-decimation decomposition method

### 2) Development of the shift invariant relationships between the subbands

The pixel location  $d^y$  in the full size frame can be represented by the pixel location  $S_L^y$  and the decimation pattern  $D_L^y$  for a sub-decimation decomposed subframe at the  $L^{\text{th}}$



level in both the horizontal ( $v = x$ ) and vertical ( $v = y$ ) directions, as:

$$d^v = 2^L S_L^v + D_L^v \quad (7.7)$$

where

$$S_L^v = \left\lfloor \frac{d^v}{2^L} \right\rfloor \quad (7.8)$$

and

$$D_L^v = d^v \% 2^L \quad (0 \leq D_L^v < 2^L) \quad (7.9)$$

% is the modulation operation. Since the parameter pair  $(S_L^v, D_L^v)$  has a one to one correspondence to  $d^v$ ,

$$d^v \leftrightarrow (S_L^v, D_L^v) \quad (7.10)$$

the location of a block in the original frame is defined by its left-top corner coordinate  $(d^x, d^y)$  and the location of a subblock in a subframe is defined by its left-top coordinate  $(S_L^x, S_L^y)$  with decimation pattern  $D_L$ , so that pixel motion in full frame can be determined. On the other hand, a subblock  $[S_L^x \sim S_L^x + \frac{B_x}{2^L} - 1, S_L^y \sim S_L^y + \frac{B_y}{2^L} - 1]$  in the approximation subframe at the  $L^{\text{th}}$  decomposed level is corresponding to a block with pixels  $[2^L S_L^x - 2^L + 1 \sim 2^L S_L^x - 2^L + B_x, 2^L S_L^y - 2^L + 1 \sim 2^L S_L^y - 2^L + B_y]$  in a full-size frame by a default decimation pattern 0.  $B_x$  and  $B_y$  are the current block size of the original frame, and the four values are the corner coordinates in horizontal and vertical directions.

### 3) Shift invariant motion estimation and compensation

Define  $w_x$  and  $w_y$  as the maximum search window displacements in horizontal and vertical directions in the full size frame. Motion estimation is started from the lowest resolution level  $N$  using the spatial domain full search method. Since decimation pattern 0 is used in the current frame decomposition, for a subblock located at  $(S_N^x, S_N^y)$  as shown in Figure 7.6, the location of the corresponding current block in the full-size current frame is  $(2^N S_N^x - 2^N + 1, 2^N S_N^y - 2^N + 1)$ . The location  $(d^x, d^y)$  of the candidate searching blocks in the full size reference frame is in the range of:

$$2^N S_N^x - 2^N + 1 - w_x \leq d_x \leq 2^N S_N^x - 2^N + 1 + w_x \quad (7.11)$$

$$2^N S_N^y - 2^N + 1 - w_y \leq d_y \leq 2^N S_N^y - 2^N + 1 + w_y \quad (7.12)$$

and the boundary conditions of the search windows are:

$$1 \leq d^x \leq N_x - \frac{B_x}{2} + 1 \quad (7.13)$$

$$1 \leq d^y \leq N_y - \frac{B_y}{2} + 1 \quad (7.14)$$

where  $N_x$  and  $N_y$  are the full frame size.

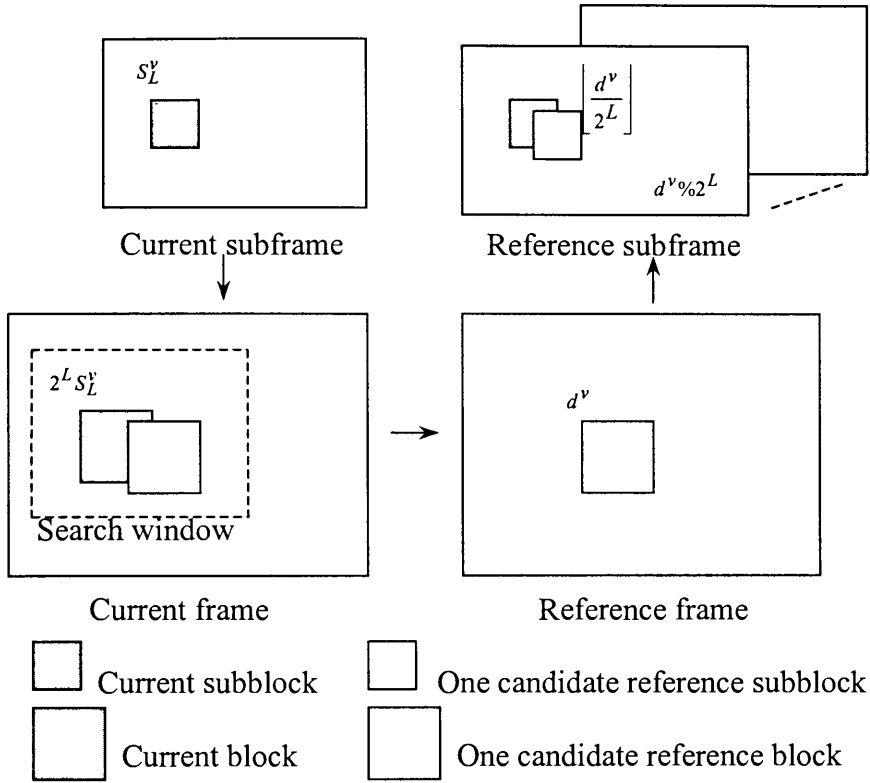


Figure 7.6. Motion estimation between the reference-current frame and subframe pairs

For each candidate block  $(d^x, d^y)$  in the reference frame, the corresponding searching subblock in a decomposed subframe is specified by the left-top corner coordinate  $\left(\left\lfloor \frac{d^x}{2^L} \right\rfloor, \left\lfloor \frac{d^y}{2^L} \right\rfloor\right)$  and decimation pattern  $(d^x \% 2^L, d^y \% 2^L)$  by Equations (7.8) and (7.9).

The best matching subblock is searched within all these reference subblock candidates by Equations (7.11~7.14). The motion vectors in a higher resolution level are refined in the approximation subframe by those from the adjacent lower resolution approximation subframe after doubling, and represent the motion vectors of the three corresponding detail subbands with the same decimation pattern. Due to the 2:1 decimation, the

refinements of the motion vectors are performed within  $\pm 1$  pixel around the location of the best matching block found from the last level in the full frame size. If the best matching block is not found, full search method is applied again, as in the lowest resolution level. Motion compensation in the detail subbands are directly using the motion vectors obtained from the approximation subband at the same level to save the computational load. The refinement process is continued successively until the required dimension and scalability of the video frame is reached.

Assume the maximum displacement in the lowest resolution level is  $d_0$ -pixel using full search method, after  $N$ -level wavelet decomposition, and the refinements in other levels are  $d_r$ -pixel, the equivalent search displacement  $d$  in full frame resolution is:

$$\begin{aligned} d_{SDD} &= 2^N d_0 + \sum_{i=1}^{N-1} 2^i d_r \\ &= 2^N d_0 + (2^N - 2) d_r \end{aligned} \quad (7.15)$$

The computational load for a current block in all levels of a  $N_1 \times N_2$  frame using sub-decimation decomposed motion estimation method is:

$$\begin{aligned} L_{SDD} &= 4^N (2d_0 + 1)^2 \cdot \frac{N_1}{2^N} \cdot \frac{N_2}{2^N} + \sum_{i=1}^{N-1} (2d_r + 1)^2 \cdot \frac{N_1}{2^i} \cdot \frac{N_2}{2^i} \\ &= L_B \left\{ (2d_0 + 1)^2 + \frac{1}{3} \left[ 1 - \left( \frac{1}{4} \right)^{N-1} \right] (2d_r + 1)^2 \right\} \end{aligned} \quad (7.16)$$

with that the motion vectors obtained in all of the lower resolution levels. Table 7.2 compares the computational loads of using spatial domain full search method and wavelet domain sub-decimation decomposition method by 2-level decomposition and 1-pixel refinement in each level.

Table 7.2. Computational load comparison between full search method and sub-decimation decomposition method ( $d_r = 1, N = 2$ )

Displacement	Computational load (in unit of $L_B$ )		
	$d = 6$	$d = 10$	$d = 14$
$L_{FS}$	169	441	841
$L_{SDD}$	11.25	27.25	51.25

Differential motion vectors in each level as well as binary decimation patterns are combined to represent motion information. If a best matching block is found in the  $k^{th}$  resolution level,  $k$  bits are used to represent the decimation pattern for the current subblock in one direction. For a  $16 \times 16$  macroblock in the original frame,

$2 \sum_{i=1}^N i = N(N+1)$  more bits are used after the  $N$ -level decomposed sub-decimation

MEMC, which results in  $\frac{N(N+1)}{256} bpp$  of the increased bit rate. When  $N = 2$ , 0.02bpp is

needed to represent the decimation information. However, the coding efficiency is highly dependent on the energy distribution of the prediction error subframe, and the energy of the prediction subframe is strongly related to the precision of the motion vectors obtained from motion estimation. Better PSNR quality can be achieved by sub-decimation decomposition because it is applied on the reference frame only, the data transmitted are still the differential wavelet coefficients for the current frame. Therefore, the bandwidth is highly decreased by the lower-entropy compensated frame data comparing to the small amount of motion information, so the overall compression performance from the sub-decimation decomposition method is improved.

The motion estimation and compensation using sub-decimation decomposition method on video frames “football” is shown in Figure 7.7. The motion fields are displayed by the block shifts as the pixel movements in the subframes. The reconstructed frame is about 30dB at 0.4bpp after 60 times compression. The comparisons between the wavelet domain search method (PWS) and the sub-decimation decomposition method (SDD\_PWS) at low bit rates are illustrated for “football” in Figure 7.8 and for “foreman” in Figure 7.9.

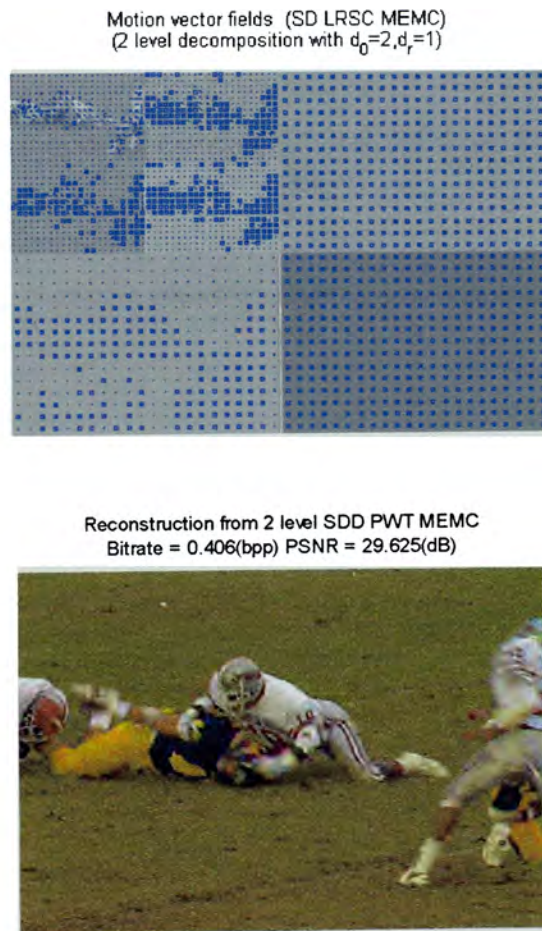


Figure 7.7. Video compression using sub-decimated decomposition method

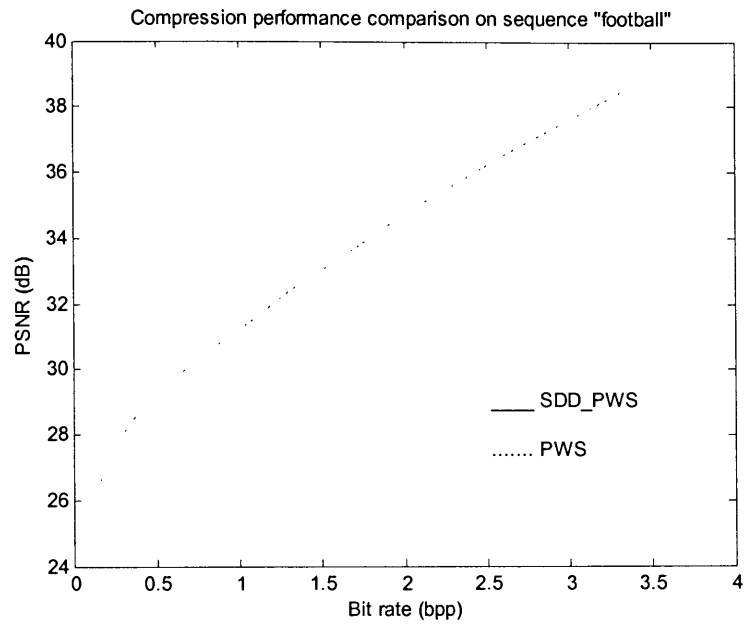


Figure 7.8. Compression comparison between PWS and SDD\_PWS methods for "football"

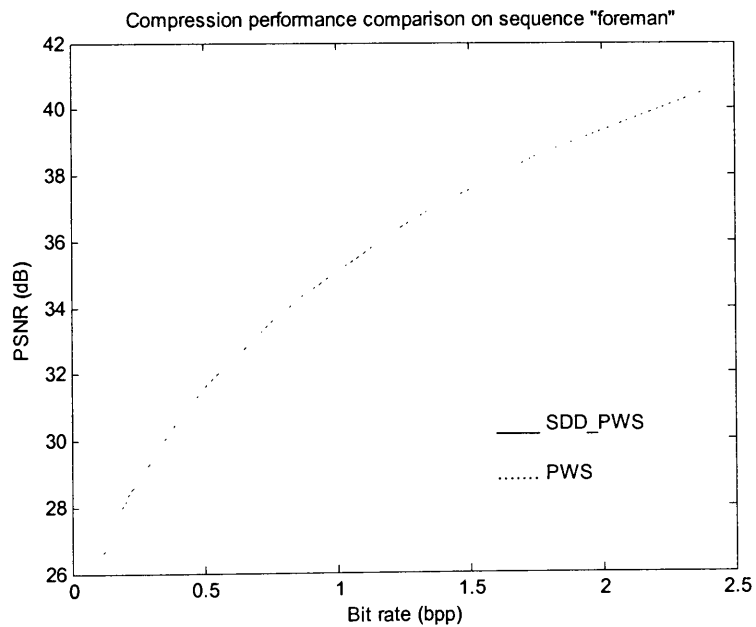


Figure 7.9. Compression comparison between PWS and SDD\_PWS methods for "foreman"

### 7.3 Sub-decimation decomposition motion estimation and compensation by symmetric-extended wavelet transform

From the results provided in chapter 6, the lowpass decimated output by the symmetric-extended wavelet transform using a  $W_f$  filter is in (1,2) extension mode, and the highpass output is in (2,1) extension mode. This is the even decimation result, while the odd decimation has the opposite extension modes in sub-decimation decomposition method. Table 7.3 shows the extension modes of the one-dimensional symmetric-extended signal convolved with the biorthogonal 9/7 wavelet filters after half decimation.

Table 7.3. One-dimensional signal extension modes after SWT by biorthogonal 9/7 filter

Decimation	Odd	Even
$h_0$ (LPF)	(2,1)	(1,2)
$h_1$ (HPF)	(1,2)	(2,1)

The sub-decimation decomposition using symmetric-extended wavelet transform by the odd length symmetric wavelet filter is illustrated in Figure 7.10, with the input being a finite-length signal in length  $L$  at scale  $s + 1$ . The W extension is performed on the input signal into length  $2L - 2$  to match the symmetric extension mode of the  $W_f$  filter before wavelet transformation, as shown in Figure 7.11. The first sample point of the signal is aligned to the symmetric center of the filter to get the center-biased convolved signal. Although the extension modes of the transformed coefficients in the two filter bank branches and the two decimations are different, truncations of the first half as the decomposed signal can keep all the necessary information. The terms  $a_{odd}$  and  $w_{odd}$  are



the approximation and detail output signals by odd decimation, while  $a_{even}$  and  $w_{even}$  are the outputs by even decimation.

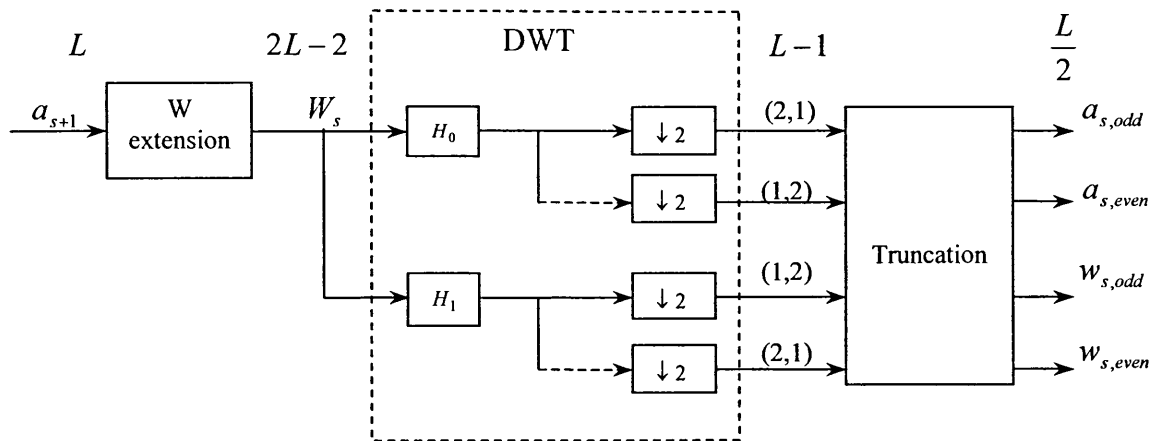


Figure 7.10. One-dimensional sub-decimation decomposition using symmetric-extended wavelet transform by biorthogonal 9/7 filters

The extension modes of two-dimensional signal in odd and even decimations can be obtained by applying the one-dimensional operations in both the horizontal and vertical directions. Table 7.4 illustrates the extension modes of the decomposed signal in each orientational subband after the two-dimensional sub-decimations. Each decimation pattern, as the odd-odd, odd-even, even-odd and even-even can perfectly reconstruct the original signal although the coefficients could be quite different in the same subbands. These extension modes are necessary for reconstruction by making the corresponding symmetric extensions before applying the inverse wavelet transformation.

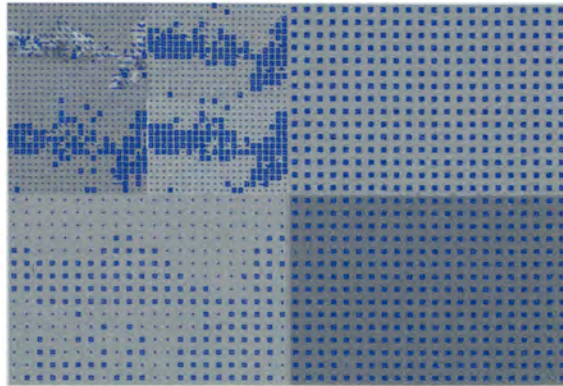
Table 7.4. Two-dimensional signal extension modes after SWT by biorthogonal 9/7 filter

Decimation	Odd-Odd		Odd-Even		Even-Odd		Even-Even	
Orientation	Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical
LL	(2,1)	(2,1)	(1,2)	(2,1)	(2,1)	(1,2)	(1,2)	(1,2)
HL	(2,1)	(1,2)	(1,2)	(1,2)	(2,1)	(2,1)	(1,2)	(2,1)
LH	(1,2)	(2,1)	(2,1)	(2,1)	(1,2)	(1,2)	(2,1)	(1,2)
HH	(1,2)	(1,2)	(2,1)	(1,2)	(1,2)	(2,1)	(2,1)	(2,1)

The sub-decimation decomposed video compression by symmetric-extended wavelet transform is based on the level-refined motion estimation and subband compensation method from chapter 5. Both the reference frame and the current frame are decomposed by symmetric-extended wavelet transform, so that the boundary artifacts are not extended into the subframes for motion estimation. In addition, the reference subframe is decomposed by the four decimation patterns, while the current frame still keeps the original decimation. After all the decomposed subframes by symmetric-extended wavelet transform are generated, the motion estimation and compensation follow the procedure of the sub-decimation decomposition method as in section 7.2.

The video compression on frame “football” using sub-decimation decomposition method by symmetric-extended wavelet transform with biorthogonal 9/7 filters is shown in Figure 7.11.

Motion vector fields (SDD LRSC SWS)  
(2 level decomposition with  $d_0=2, d_r=1$ )



Reconstruction from 2 level SDD SWS  
Bitrate = 0.515(bpp) PSNR = 30.343(dB)



Figure 7.11. Video compression using sub-decimation decomposition method by symmetric-extended wavelet transform with biorthogonal 9/7 filters on frame "football"

A range of compression performances using sub-decimation decomposed SWT search method (SDD\_SWS) are compared with SDD\_PWS method as shown in Figures 7.12 and 7.13 for frame "foreman" and for frame "carphone", respectively.

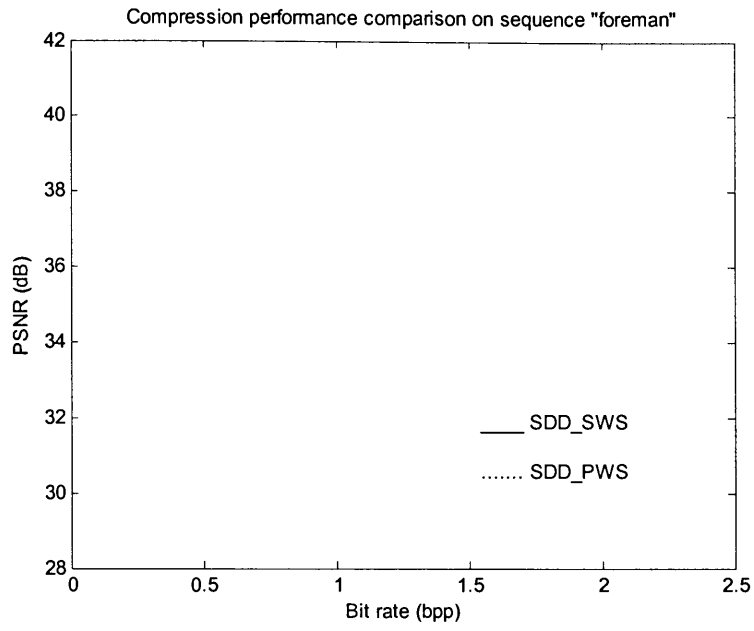


Figure 7.12. Compression comparison of SDD\_SWS and SDD\_PWS method for frame "foreman"

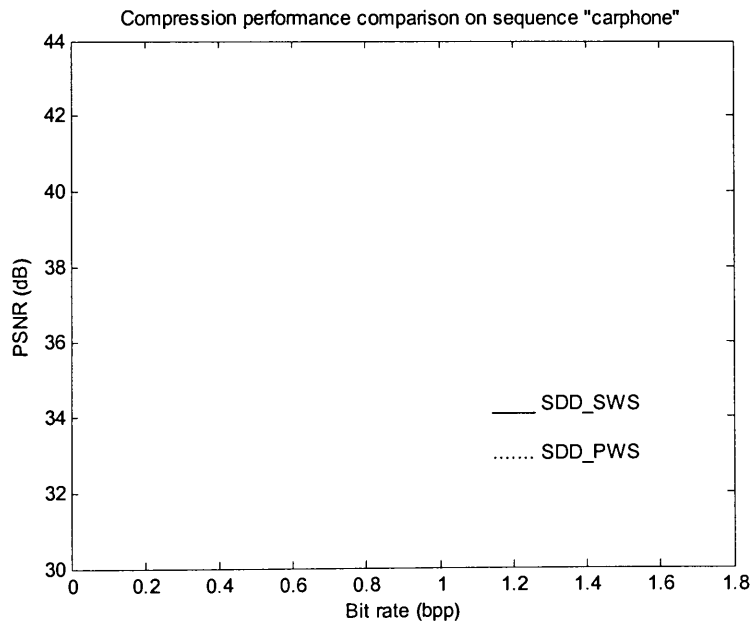


Figure 7.13. Compression comparison of SDD\_SWS and SDD\_PWS method for frame "carphone"

To illustrate the compression performances of the motion estimation and compensation methods, the reconstructed frames are put together to compare the performances under the same bit rates for sequence “football” and “carphone” in Figure 7.14 and 7.15 respectively.

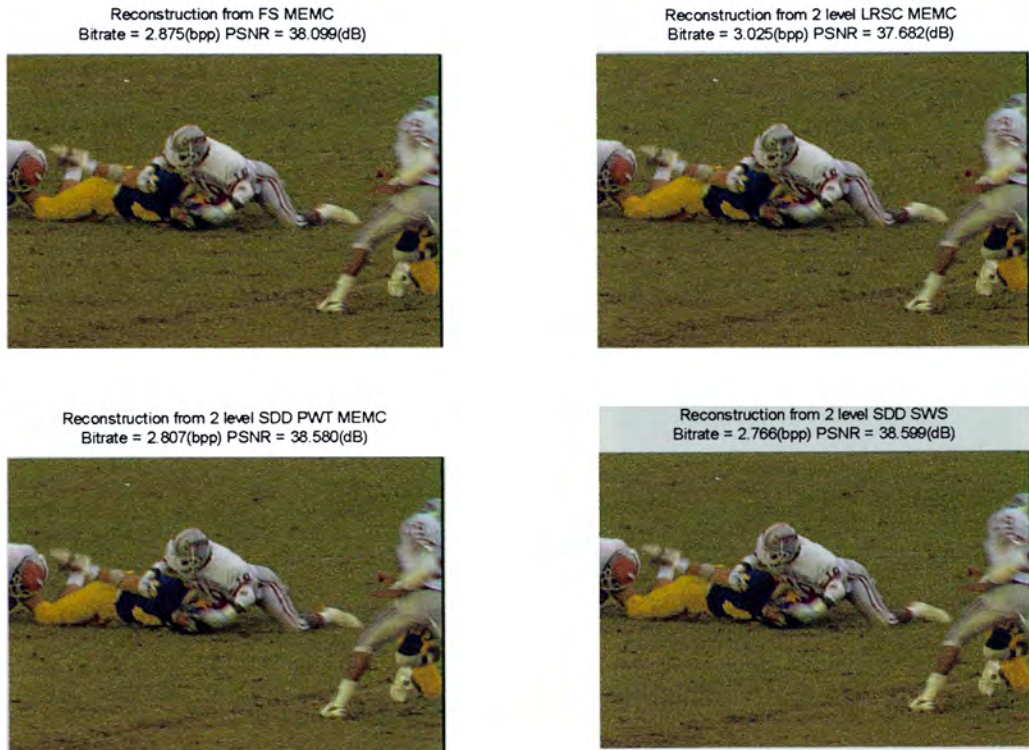


Figure 7.14. Compression comparison for frame "football" at  $CR = 8$

Reconstruction from FS MEMC  
Bitrate = 1.015(bpp) PSNR = 38.570(dB)



Reconstruction from 2 level LRSC MEMC  
Bitrate = 1.074(bpp) PSNR = 39.398(dB)



Reconstruction from 2 level SDD PWT MEMC  
Bitrate = 1.023(bpp) PSNR = 39.524(dB)



Reconstruction from 2 level SDD SWS  
Bitrate = 1.005(bpp) PSNR = 39.590(dB)



Figure 7.15. Compression comparison for frame "carphone" at  $CR = 24$

## CHAPTER 8

### Conclusions

The efficiency gain obtained by interframe compression demonstrates the functionality of motion estimation and compensation based on the strong temporal correlations between video frames. Fast search algorithms of step search (SS) and block-based gradient descent search (BBGDS) contain much less computational loads than the full search in motion estimation, which are compared by the formulated evaluations. The displayed criterion function distributions and motion vector fields match the essential principles of the search strategies. However, the compression qualities of the fast search methods depend on the motion speed of the video sequence, thus the performances are decreased as indicated by the increased number of intramode coded blocks.

An entropy coding system to evaluate the compression performance is designed in the encoder and decoder to realize lossless recovery of the frame data and motion information. All data including the differential errors, run lengths and motion vectors are counted into the coefficient histogram distributions with a frame-dependent Huffman table, so that a small header bandwidth is in exchange of great coding efficiency. The better the prediction from motion estimation, the lower are the residual signal and the bit rate, and the higher is the compression efficiency. Since only the motion vectors and the low entropy differential error frame are transmitted, the data bandwidth is saved by the compressed bit rate.



The global wavelet transform overcomes the blocking artifact, which is the major drawback of DCT, thus it brings much better subjective quality on video compression, especially at low bit rates. The multiresolution nature of wavelet transform satisfies hierarchical motion estimation with scalability by motion compensation in each level using the refined motion vectors. Since the motion speed is doubled by the half image size between successive levels using wavelet transform, the multiresolution method can be regarded as a fast search method while the compression performances are close to or better than the full search algorithm by the illustrated examples.

Based on the similar motion structures between adjacent frequency subbands, the motion vector fields by level-refined and subband-refined mechanisms simplify the wavelet domain motion estimation and compensation procedure, as compared by the computational load formulations. Although the level-refined motion estimation and compensation method has low computational load, the tradeoff is the compression performance due to the sharing of the same motion structure for all the subbands at the same level.

The energy distributions and the increased intramode compressed blocks in the decomposed subbands indicate that the energy contained in the high frequency subbands is even lower than the differential error blocks after motion estimation. Thus, an efficient approach called level-refined motion estimation and subband compensation (LRSC) method is introduced by applying selective motion compensation in the detail subbands using the estimated motion vectors from the approximation subband. It realizes the possible intrablocks in the subbands for lower entropy coding while keeping the



computational loads of motion estimation as low as the level-refined method, achieving as a consequence both temporal compression quality and computational simplicity. The wavelet domain video compressions in this research endeavor are all based on this new motion estimation and compensation method.

Circular convolution in wavelet transform is used to realize signal filtering without coefficient expansion but the performance of the filtered signal depends on the padding of the finite-length signal. The operations of extension and decimation between the symmetric extension of the finite length signal and the biorthogonal symmetric wavelet filters realizes the symmetric-extended wavelet transform (SWT). The first half truncation as the decomposed signal removes the discontinuous boundary artifacts and achieves half resolution decomposition although the signal is expanded before circular convolution. Under the considerations of the energy conservation and symmetric characteristics, biorthogonal 9/7 filter bank is applied in the wavelet transform system. To match the symmetric feature of the biorthogonal 9/7 filter, the decomposed signals are extracted by the predefined extension modes as (1,2) in the lowpass filter branch and (2,1) in the highpass branch in SWT, and extended by the corresponding extension modes in inverse SWT.

The symmetric-extended wavelet transform applied in video compression dealt with the boundary artifacts by circular convolutions in the reference and current frame signals, so that more accurate subframe decompositions are obtained for motion estimation. The scalable images obtained are in the same size as from the periodic-extended wavelet

transform without the discontinuous boundary artifacts, as proven by the experiments of reference-current frame pair compressions.

The sub-decimation decomposition method introduced in this dissertation improves the wavelet domain motion estimation and compensation, especially for piecewise translational movements in the original frame. The odd and even decimation patterns considered in the decomposed signals make the elusive shift invariance property possible, so that motion consistency between decomposed subframes is attained. The specification of motion shift and decimation pattern in the subframes defines any shift in the original frame by one of the sub-decimation decomposed subframes. Therefore, the matching blocks found are more accurate after motion estimation, and the compensated differential error frame contains lower entropy with only a small amount of motion information, so as to gain the compression efficiency.

## References

- [1] Leonardo Chiariglione, "MPEG and Multimedia Communications", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 1, Feb. 1997.
- [2] Peter Kuhn, *Algorithms, Complexity, Analysis and VLSI Architectures for MPEG-4 Motion Estimation*, Kluwer Academic Publishers, 1999.
- [3] Thomas Sikora, "The MPEG-4 Video Standard Verification Model", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 1, Feb. 1997.
- [4] Susan Gauch, Wei Li and John Gauch, "The Vision Digital Video Library", *Information Processing & Management*, Vol. 33, No. 4, pp. 413-426, 1997.
- [5] Zixiang Xiong, Kannan Ramchandran, Michael T. Orchard and Ya-Qin Zhang, "A Comparative Study of DCT- and Wavelet-Based Image Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 5, Aug. 1999.
- [6] Diego Santa-Cruz and Touradj Ebrahimi, "A Study of JPEG 2000 Still Image Coding Versus Other Standards", *Proc. of the X European Signal Processing Conference*, Vol. 2, pp. 673-676.
- [7] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Addison-wesley, 1992.
- [8] Athanassios Skodras, Charilaos Christopoulos and Touradj Ebrahimi, "The JPEG 2000 Still Image Compression Standard", *IEEE Signal Processing Magazine*, pp. 36-58, Sep. 2001.
- [9] Charilaos Christopoulos and Athanassios Skodras and Touradj Ebrahimi, "The JPEG2000 Still Image Coding System: an Overview", *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 4, pp. 1103-1127, Nov. 2000.
- [10] Diego Santa-Cruz and Touradj Ehrahimi, "An Analytical study of JPEG 2000 functionalities", *2000 International Conference on Image Processing (ICIP-2000)*, Sep. 10-13, 2000.
- [11] A. N. Skodras, C. A. Christopoulos and T. Ebrahimi, "JPEG2000: The Upcoming Still Image Compression Standard", *Proceedings of the 11<sup>th</sup> Portuguese Conference on Pattern Recognition*, Porto, Portugal, May 11-12, 2000.

- [12] Thomas Meier, King N. Ngan and Gregory Crebbin, "Reduction of Blocking Artifacts in Image and Video Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 3, Apr. 1999.
- [13] Hsueh-Ming Hang, Yung-Ming Chou and Sheu-Chih Cheng, "Motion Estimation for Video Coding Standards", *Journal of VLSI Signal Processing* 17 (1997), pp. 113-136.
- [14] Marek Domanski, Adam Luczak and Slawomir Mackowiak, "Scalable MPEG Video Coding with Improved B-Frame Prediction", *IEEE International Symposium on Circuits and Systems*, May 28-31, 2000, Geneva, Switzerland.
- [15] Vivek K. Goyal, "Theoretical Foundations of Transform Coding", *IEEE Signal Processing Magazine*, pp. 9-21, Sep. 2001.
- [16] Henrique S. Malvar and David H. Staelin, "The LOT: Transform Coding Without Blocking Effects", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 37, No. 4, Apr. 1989.
- [17] Yuen-Wen Lee, Faouzi Kossentini, Rabab Ward and Mark Smith, "Towards MPEG4: An Improved H.263-Based Video Coder", *Signal Processing: Image Communication* 10 (1997), pp. 143-158.
- [18] Yeong-An Jeong and Cha-Keon Cheong, "A DCT-Based Embedded Image Coder Using Wavelet Structure of DCT for Very Low Bit Rate Video Codec", *IEEE Transactions on Consumer Electronics*, Vol. 44, No. 3, Aug 1998.
- [19] T. Morris and D. Britch, "Intra-frame Wavelet Video Coding", *International Workshop on Video Processing and Multimedia Communications*, Jun. 28-30, 2000.
- [20] Jaideva C. Goswami and Andrew K. Chan, *Fundamentals of Wavelets: Theory, Algorithms and Applications*, Wiley-Interscience, 1999.
- [21] Shie Qian and Dapang Chen, "Joint Time-Frequency Analysis", *IEEE Signal Processing Magazine*, pp. 53-80, Mar. 1999.
- [22] Christopher Torrence and Gilbert P. Compo, "A Practical Guide to Wavelet Analysis", *Bulletin of the American Meteorological Society*, Vol. 79, No. 1, Jan. 1998.
- [23] Gilbert Strang and Truong Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

- [24] Amara Graps, "An Introduction to Wavelets", *IEEE Computational Science and Engineering*, Vol. 2, No. 2, 1995.
- [25] Gerald Kaiser, *A Friendly Guide to Wavelets*, Birkhauser Boston, 1994.
- [26] John Sadowsky, "Investigation of Signal Characteristic Using the Continuous Wavelet Transform", *John Hopkins APL Technical Digest*, Vol. 17, No. 3, 1996.
- [27] Christopher E. Heil and David F. Walnut, "Continuous and Discrete Wavelet Transform", *SIAM Review*, Vol. 31, No. 4, pp. 628-666, Dec. 1989.
- [28] Ole Christensen, "Frames, Riesz Bases and Discrete Gabor/Wavelet Expansions", *Bulletin of the American Mathematical Society*, Vol. 38, No. 3, pp. 273-291, Mar. 2001.
- [29] Raghuveer M. Rao and Ajit S. Bopardikar, *Wavelet Transforms: Introduction to Theory and Applications*, Addison-wesley, 1998.
- [30] Stephane Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.
- [31] Stephane G. Mallat, "Multiresolution Approximation and Wavelet Orthonormal Bases of  $L^2(\mathbb{R})$ ", *Transactions of the American Mathematical Society*, Vol. 315, Issue 1, pp. 69-87, Sep. 1989.
- [32] Martin Vetterli and Didier Le Gall, "Perfect Reconstruction FIR Filter Banks: Some Properties and Factorizations", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 37, No. 7, Jul. 1989.
- [33] Martin Vetterli, "Wavelets, Approximation and Compression", *IEEE Signal Processing Magazine*, pp. 59-73, Sep. 2001.
- [34] Martin Vetterli and Cormac Herley, "Wavelet and Filter Banks: Theory and Design", *IEEE Transactions on Signal Processing*, Vol. 40, No. 8, Sep. 1992.
- [35] T. K. Sarkar, C. Su, M. Salazar-Palma and L. Garcia-Castillo and Rafael R. Boix, "A Tutorial on Wavelets from an Electrical Engineering Perspective, Part 1: Discrete Wavelet Techniques", *IEEE Antenna and Propagation Magazine*, Vol. 40, No. 5, Oct. 1998.
- [36] Olivier Rioul and Martin Vetterli, "Wavelets and Signal Processing", *IEEE Signal Processing Magazine*, Oct. 1991.
- [37] Ingrid Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, No 61, 1992.

- [38] Y. T. Chan, *Wavelet Basics*, Kluwer Academic Publishers, 1994.
- [39] Truong Q. Nguyen and P. P. Vaidyanathan, "Two-Channel Perfect-Reconstruction FIR QMF Structures Which Yield Linear-Phase Analysis and Synthesis Filters", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 37, No. 5, May. 1989.
- [40] John D. Villasenor, Benjamin Belzer and Judy Liao, "Wavelet Filter Evaluation for Image Compression", *IEEE Transactions on Image Processing*, Vol. 4, No. 8, Aug. 1995.
- [41] Michael D. Adams and Faouzi Kossentini, "Reversible Integer-to-Integer Wavelet Transforms for Image Compression: Performance Evaluation and Analysis", *IEEE Transactions on Image Processing*, Vol. 9, No. 6, Jun. 2000.
- [42] Bryan E. Usevitch, "A tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000", *IEEE Signal Processing Magazine*, Sep. 2001.
- [43] Christopher M. Brislawn and Jonathan N. Bradley, "The FBI Compression Standard for Digitized Fingerprint Images", *Proc. SPIE*, Vol. 2847, pp. 344-355, Aug. 1996.
- [44] Christopher M. Brislawn, "Fingerprints Go Digital", *Notices of the AMS*, Vol. 42, No. 11, pp. 1278-1283.
- [45] Gilbert Strang, "Wavelet Transform Versus Fourier Transforms", *Bulletin of the American Mathematical Society*, Vol. 28, No. 2, pp. 288-305, Apr. 1993.
- [46] Stephane G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, Jul. 1989.
- [47] A. Murat Tekalp, *Digital Video Processing*, Prentice Hall, 1995.
- [48] Hwang-Seok Oh and Heung-Kyu Lee, "Block-Matching Algorithm Based on an Adaptive Reduction of the Search Area for Motion Estimation", *Real-Time Imaging* 6, pp. 407-414, 2000.
- [49] Yun-Hee Choi and Tae-Sun Choi, "Fast Motion Estimation Techniques with Adaptive Variable Search Range", *IEICE Trans. Fundamentals*, Vol. E82-A, No. 6, Jun. 1999.
- [50] Marco Mattavelli and Giorgio Zoia, "Vector-Tracing Algorithms for Motion Estimation in Large Search Windows", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 8, Dec. 2000.

- [51] Jianhua Lu and Ming L. Liou, "A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 2, Apr. 1997.
- [52] Jar-Ferr Yang, Shu-Sheng Hao and Wei-Yuan Lu, "Simplified Block Matching Criteria for Motion Estimation", *IEICE Trans. Inf. & Syst*, Vol. E83-D, No. 4, Apr. 2000.
- [53] Michael Brunig and Wolfgang Niehsen, "Fast Full-Search Block Matching", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 2, Feb. 2001.
- [54] J. Feng, K. T. Lo, H. Mehrpour and A. E. Karbowiak, "Adaptive Block Matching Algorithm for Video Compression", *IEE Proc.-Vis. Image Signal Processing*, Vol. 145, No. 3, Jun. 1998.
- [55] M. K. Mandal, E. Chan, X. Wang and S. Panchanathan, "Multiresolution Motion Estimation Techniques for Video Compression", *Optical Engineering*, Vol. 35, No. 1, pp. 128-136, Jan. 1996.
- [56] Renxiang Li, Bing Zeng and Ming L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 4, Aug. 1994.
- [57] Luis Miguel Lopes Teixeira and Artur P. Alves, "Block Matching Algorithms in MPEG Video Coding", *Proceedings of ICPR '96*, pp. 934-938.
- [58] Detlev Marpe and Hans L. Cycon, "Very Low Bit-Rate Video Coding Using Wavelet-Based Techniques", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 1, Feb. 1999.
- [59] Mohammed Ghanbari, *Video Coding: an Introduction to Standard Codecs*, IEE Telecommunications Series, No. 42, 1999.
- [60] Ya-Qin Zhang and Sohail Zafar, "Motion-Compensated Wavelet Transform Coding for Color Video Compression", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 2, No. 3, Sep. 1992.
- [61] Sohail Zafar, Ya-Qin Zhang and Bihan Jabbari, "Multiscale Video Representation Using Multiresolution Motion Compensation and Wavelet Decomposition", *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 1, Jan. 1993.

- [62] Jonathan M. Bradley and Christopher M. Brislawn, "The Wavelet/Scalar Quantization Compression Standard for Digital Fingerprint Images", *Proc. IEEE ISCAS '94*, London.
- [63] Marc Antonini, Michel Barlaud, Pierre Mathieu and Ingrid Daubechies, "Image Coding Using Wavelet Transform", *IEEE Transactions on Image Processing*, Vol. 1, No. 2, Apr. 1992.
- [64] Marc Antonini, Michel Barlaud, Pierre Mathieu and Ingrid Daubechies, "Image Coding Using Wavelet Transform", *IEEE Transactions on Image Processing*, Vol. 1, No. 2, Apr. 1992.
- [65] Stephen J. Solari, *Digital Video and Audio Compression*, McGraw-Hill, 1997.
- [66] Andrew B. Watson, Gloria Y. Yang, Joshua A. Solomon and John Villasenor, "Visual Thresholds for Wavelet Quantization Error", *SPIE Proceedings*, Vol. 2657, Paper # 44, Human Vision and Electronic Imaging, 1996.
- [67] Zixiang Xiong, Kannan Ramchandran and Michael T. Orchard, "Space-Frequency Quantization for Wavelet Image Coding", *IEEE Transactions on Image Processing*, Vol. 6, No. 5, May 1997.
- [68] Chung-Ming Kuo and Chaur-Heh Hsieh, "Multiresolution Video Coding Based on Kalman Filtering Motion Estimation", *Journal of Visual Communication and Image Representation*, 1998.
- [69] Pankaj N. Topiwala, *Wavelet Image and Video Compression*, Kluwer Academic Publishers, 1998.
- [70] Mi-Sook Chung, Suk Kang, Jae-Ho Choi and Hoon-Sung Kwak, "New Multiresolution Motion Estimation for Wavelet Transform Video Coding", *Proceedings of IEEE Asia Pacific Conference on Circuits and System '96*, Nov. 18~21, 1996, Seoul, Korea, pp. 10-13.
- [71] Suhuai Luo, "On the Improvement of Motion Compensation in Wavelet Transform Video Coding", *1997 IEEE TENCON – Speech and Image Technologies for Computing and Telecommunications*, pp. 735-738.
- [72] El-Sayed Youssef, Ehab Sabry and Mohamed Khedr, "Hybrid Adaptive Motion Search with Wavelet/Wavelet Packet Based Motion Residual Video Coding", *Seventeenth National Radio Science Conference*, Feb. 22-24, 2000, Minufiya University, Egypt, pp. C2 1-11.



- [73] G. Van der Auwera, A. Munteanu, G. Lafruit and J. Cornelis, "Video Coding Based on Motion Estimation in the Wavelet Detail Images", *ICASSP '98*, pp. 2801-2804.
- [74] Seongman Kim and Seunghyeon Rhee, "Interframe Coding Using Two-Stage Variable Block-Size Multiresolution Motion Estimation and Wavelet Decomposition", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 4, Aug. 1998.
- [75] Julian Magarey and Nick Kingsbury, "Motion Estimation Using Complex Wavelets", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, pp. 2371-2374, 1996.
- [76] Radu Zaciuc, Claudiu Lamba, Constantin Burlacu and Gabriel Nicula, "Motion Estimation and Motion Compensation Using an Overcomplete Discrete Wavelet Transform", *ICIP96*, pp. 973-976.
- [77] Michael B. Martin and Amy E. Bell, "New Image Compression Techniques Using Multiwavelets and Multiwavelet packets", *IEEE Transactions on Image Processing*, Vol. 10, No. 4, Apr. 2001.
- [78] Jerome M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, Dec. 1993.
- [79] G. R. Rajugopal, S. Mahmoud and R. H. M. Hafez, "Multi-Resolution Motion Estimation at Low Bit-Rates", *1996 IEEE TENCON – Digital Signal Processing Applications*, pp. 863-866.
- [80] Hsieh S. Hou, "A Fast Recursive Algorithm for Computing the Discrete Cosine Transform", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-35, No. 10, Oct. 1987.
- [81] Madihally J. Narasimha and Allen M. Peterson, "On the Computation of the Discrete Cosine Transform", *IEEE Transactions on Communications*, Vol. COM-26, No. 6, pp. 934-936, Jun. 1978.
- [82] Hyun-Wook Park and Hyung-Sun Kim, "Motion Estimation Using Low-Band-Shift Method for Wavelet-Based Moving-Picture Coding", *IEEE Transactions on Image Processing*, Vol. 9, No. 4, Apr. 2000.
- [83] Hyung Sun Kim and Hyun Wook Park, "Wavelet-Based Moving-Picture Coding Using Shift-Invariant Motion Estimation in Wavelet Domain", *Signal Processing: Image Communication* 16 (2001), pp. 669-679.

## Vita

### Weiting Cai

1993~1997	South China University of Technology Electrical Engineering Bachelor
1998~1999	Florida International University Electrical Engineering Master
2000~2003	Florida International University Electrical Engineering PhD student

### Publications

- [1] Weiting Cai and Malek Adjouadi, "Video Compression Analysis Based on Fast Motion Estimation Algorithms", *International Conference on Computer, Communications and Control Technologies: CCCT '03*, Orlando, Florida, USA, Jul. 31-Aug. 2, 2003.
- [2] Weiting Cai and Malek Adjouadi, "Wavelet-Based Video Compression Using Multiresolution Motion Estimation and Compensation", *The 9<sup>th</sup> International Conference on Distributed Multimedia Systems*, Miami, Florida, USA, Sep. 24-26, 2003.
- [3] Malek Adjouadi and Weiting Cai, "Boundary Distortion Minimization on Scalable Image Compression by Symmetric-Extended Wavelet Transformation", Submitted.
- [4] Weiting Cai and Malek Adjouadi, "An Efficient Approach of Fast Motion Estimation and Compensation in Wavelet Domain Video Compression", Submitted.
- [5] Weiting Cai and Malek Adjouadi, "Wavelet-Domain Shift Invariant Motion Estimation and Compensation", Submitted.
- [6] Weiting Cai and Malek Adjouadi, "Boundary Artifact Minimization on Best Matching Blocks in Wavelet-Based Video Compression", Submitted.
- [7] Weiting Cai and Malek Adjouadi, "Minimization of Boundary Artifacts on Scalable Image Compression Using Symmetric-Extended Wavelet Transform", Submitted.

- [8] Weiting Cai and Malek Adjouadi, "Design and Implementation of Wavelet-Domain Video Compression Using Multiresolution Motion Estimation and Compensation", Submitted.
- [9] Weiting Cai and Malek Adjouadi, "Analysis and Evaluation of Fast Motion Estimation Algorithms in Video Compression", Submitted.
- [10] Weiting Cai and Malek Adjouadi, "Optimized Motion Estimation and Compensation in Wavelet-Based Video Compression", Submitted.
- [11] Weiting Cai and Malek Adjouadi, "A Shift Invariant Motion Estimation and Compensation Method in Wavelet Domain Video Compression", Submitted.