MOTION COMPENSATED THREE DIMENSIONAL WAVELET TRANSFORM BASED VIDEO COMPRESSION AND CODING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYDIN BİÇER

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet ERKMEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. A. Aydın ALATAN Co-Supervisor	Prof. Dr. Zafer ÜNVER Supervisor

Examining Committee Members

Prof. Dr. Mete SEVERCAN	(METU, EEE)	_______________
Prof. Dr. Zafer ÜNVER	(METU, EEE)	_______________
Assoc. Prof. Dr. A. Aydın ALATAN	(METU, EEE)	_______________
Assoc. Prof. Dr. Gözde BOZDAĞI AKAR	(METU, EEE)	_______________
Dr. Cengiz ERBAŞ	(ASELSAN)	_______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Aydın BİÇER

Signature :

ABSTRACT

MOTION COMPENSATED THREE DIMENSIONAL WAVELET TRANSFORM BASED VIDEO COMPRESSION AND CODING

BİÇER, Aydın

M.S., Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. Zafer ÜNVER

Co-Supervisor : Assoc. Prof. Dr. Aydın ALATAN

January 2005, 174 pages

In this thesis, a low bit rate video coding system based on three-dimensional (3-D) wavelet coding is studied. In addition to the initial motivation to make use of the motion compensated wavelet based coding schemes, the other techniques that do not utilize the motion compensation in their coding procedures have also been considered on equal footing.

The 3-D wavelet transform (WT) algorithm is based on the "*group of frames*" (GOF) concept. The group of eight frames are decomposed both temporally and spatially to their coarse and detail parts. The decomposition process utilizes both orthogonal and bi-orthogonal wavelet analysis filters. The transform coefficients are coded using an embedded image coding algorithm, called the "*Two-Dimensional Set Partitioning in Hierarchical Trees*" (2-D SPIHT). Due to its nature, the 2-D SPIHT is applied to the individual subband frames.

In the reconstruction phase, the 2-D SPIHT decoding algorithm and the wavelet synthesis filters are employed, respectively. The Peak Signal to Noise Ratios (PSNRs) are used as a quality measure of the reconstructed frames. The investigations reveal that among several factors, the multi-level (de)composition is the dominant one effective both on the signal compression and the PSNR level. The encoded videos compressed to the ratio of 1:9 are reconstructed with the PSNR of about 30 dB in the best cases.

Keywords: 3-D Wavelet Transform, Motion Compensation, SPIHT, Video Coding, Video Compression.

ÖZ

HAREKET DENGELEMELİ ÜÇ BOYUTLU DALGACIK DÖNÜŞÜM TABANLI VİDEO SIKIŞTIRMA VE KODLAMA

BİÇER, Aydın

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. Zafer ÜNVER

Ortak Tez Yöneticisi : Doç. Dr. Aydın ALATAN

Ocak 2005, 174 sayfa

Bu tezde, üç boyutlu dalgacık dönüşüm tabanlı, düşük bit hızlı video kodlanma sistemi üzerinde çalışılmıştır. Hareket dengelemeli dalgacık dönüşüm tabanlı kodlama sistemleri öncelikli hedefler arasında yer alsa da, bu yöntemi kullanmayan hareket dengelemesiz sistemler de eş değerde ele alınmıştır.

Üç boyutlu dalgacık dönüşüm algoritması "*video karelerinin gruplandırılması*" kavramına dayanmaktadır. Sekizli video karelerden oluşan gruplar, zaman ve video kare boyutlarında kaba ve ayrıntılı kısımlarına ayrıştırılmıştır. Ayrıştırma işleminde dikgen ve çift dikgen dalgacık süzgeçlerinden faydalanılmıştır. Dönüşüm katsayıları, "*İki Boyutta Hiyerarşik Grup Ayrıştırması*" olarak adlandırılan gömülü imge kodlama tekniği kullanılarak kodlanmıştır. Kodlama, bu tekniğin doğası gereği grup içindeki her bir video karesi üzerinde ayrı yapılmıştır.

Yeniden oluşturma evresinde, sırasıyla "*İki Boyutta Hiyerarşik Grup Ayrıştırması*" kod çözümleme algoritması ve dalgacık sentez süzgeçleri kullanılmıştır. Yeniden oluşturulan karelerin niteliğinin belirlenmesi için işaret gürültü oranları kullanılmıştır. Çalışmalar, bir çok etmenin arasında, işaret sıkıştırması ve gürültü seviyeleri üzerinde en baskın etmenin çok-seviyeli ayrıştırma/oluşturma yaklaşımı olduğunu göstermiştir. 1:9 oranında sıkıştırılan kodlanmış videoların, en iyi durumlarda yaklaşık 30 dB işaret gürültü seviyesinde yeniden oluşturulduğu gözlenmiştir.

Anahtar Kelimeler: Üç Boyutlu Dalgacık Dönüşümü, Hareket Dengeleme, İki Boyutlu Hiyerarşik Grup Ayrıştırması, Video Kodlama, Video Sıkıştırma

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my supervisor Prof. Dr. Zafer ÜNVER for his guidance, advice, criticism, and insight throughout the research.

I would also like to thank my co-supervisor Assoc. Prof. Dr. Aydın ALATAN for his encouragements and suggestions on the research subject.

I wish to express my sincere appreciation to my manager Dr. Cengiz ERBAŞ in Aselsan Inc. for his valuable criticism and suggestions on this thesis.

I would also like to thank my family for their encouragement.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

1-D	One-Dimensional
2-D	Two-Dimensional
3-D	Three-Dimensional
CD	Compact Disc
CWT	Continuous Wavelet Transform
dB	Decibel
Db-4	Daubechies-4
Db-9/7	Daubechies-9/7
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
EZW	Embedded Zerotree Wavelet
FT	Fourier Transform
GAP	Geometrical Analysis Procedure
GOF	Group of Frames
H	Haar
HDTV	High Definition Television
HP	Highpass
HVS	Human Visual System
IDWT	Inverse Discrete Wavelet Transform
JPEG	Joint Picture Expert Group
L	Level
LP	Lowpass
MC	Motion Compensated
MCP	Motion Compensated Predictive
MCTF	Motion Compensated Temporal Filtering
ME	Motion Estimation
MOC	Magnitude Ordered Coefficients

MPEG	Motion Picture Expert Group
MV	Motion Vector
PSNR	Peak Signal to Noise Ratio
QMF	Quadrature Mirror Filters
SB	Subband
SBC	Subband Coding
SLOC	Source Lines of Code
SNR	Signal to Noise Ratio
SPIHT	Set Partitioning in Hierarchical Trees
STFT	Short Time Fourier Transform
SVC	Scalable Video Coding
t	Temporal
t + 2-D	Spatial-Temporal
VoD	Video on Demand
VQ	Vector Quantization
WT	Wavelet Transform

CHAPTER 1

INTRODUCTION

1.1 Digital Video Applications

Digital video is rapidly replacing the traditional analog video. It has a number of unique properties that make applications that could not be realized using analog video possible. Digital video offers numerous advantages over analog systems in terms of information processing such as *production of higher quality information, ease of copying information without quality loss, higher interactivity, ease of storage and retrieval, ease of distribution*, and *higher security.* Having these advantages, digital video is driving a variety of applications such as *telephone, video e-mail, video conference, video CD players, video streaming over Internet and Intranet, DVD players* and *HDTV*.

The ability to easily store and transmit digital video is by far its most important property. It allows users to add video attachments to e-mail, sometimes called *v-mail* and makes possible video telephony, or video conferencing, which are expected to become the first widespread application of digital video [1].

The low bandwidth requirements of digital video also make it easy to store. Movies stored on Compact Discs (CDs), known as *Video CD* can be played on CD Interactive (CD-I) devices.

Because compressed digital video can be transmitted using less bandwidth than analog television, it is possible to provide many channels where before there were only a few or none. By exploiting similar technologies, cable TV systems can have enough capacity to provide hundreds of channels of digital video.

Video-on-Demand (VoD), currently available only on a trial basis, allows viewers to choose the movie and the time that they watch it. Entertainment companies that convert cinemas into digital are aiming to form interactive television systems. The degree of interaction can vary from application to application, such as choice of a program as in VoD to continuous interaction as in games.

Integrating digital video into interactive applications along with other media, such as sound, animation, photographs and text, known as *multimedia*, is one of the most popular applications of digital video. The low cost production of video sequences by non television professionals and the low bandwidth requirements of digital video allow it to be stored on CDs or hard disks and to be displayed on computer screens. In addition, the ease with which various segments can be accessed allows them to be integrated into highly interactive applications.

High quality movies and sounds stored on *DVD*, known as DVD-Video/Audio, can be played on computer systems. The technology used in DVDs let digital data to be stored from seven times to over 25 times that of a CD.

High Definition Television (HDTV) has been desirable for many years. Although manufacturers and governments have until recently failed to agree on a standard, HDTV is expected to be such a huge success that it offers impressive quality advantages over conventional television. Images will be sharper and screens will be larger with more suitable aspect ratios.

The digital video applications are much more extended than mentioned above. "As the world goes digital more and more, the researchers go on working on digital video with the support of military, telecommunication, TV, security, government, health, publishing, and entertainment markets" [5].

1.2 The Need for Video Compression

The high bit rates that result from the various types of digital video make their transmission through their intended channels very difficult. In addition, the processing power needed to manage raw volumes of data for storage and transmission makes the receiver hardware very expensive. To overcome the problem, engineers and researchers work not only on developing high speed processors and peripherals that are capable of processing huge volumes of data but also on establishing video compression standards. The latter facilitates manipulation and storage of full-motion video as a form of computer data, and its transmission over existing and future computer networks, or over worldwide broadcast channels.

Compression is the conversion of data to a format that requires fewer bits, usually so performed that the data can be stored or transmitted more efficiently. The size of the data in compressed form (C) relative to the original size (O) is known as the *compression ratio* ($R=C/O$). If the inverse of the process, *decompression*, produces an exact replica of the original data then the compression is *lossless* [1]. On the other hand, *lossy compression* has a higher compression ratio but does not allow reproduction of an exact replica of the original image [42].

The success of data compression depends largely on the data itself. Some data types are inherently more compressible than others. Generally, some elements within the data are more common than others and most compression algorithms exploit this property, known as *redundancy*. The greater the redundancy within the data, the more successful the compression of the data is likely to be. Fortunately, digital video contains a great deal of redundancy and thus is very suitable for compression.

A great number of compression techniques have been developed and some lossless techniques can be applied to any type of data. Development of lossy techniques in recent years specifically for image data has contributed a great deal to the realization of digital video applications. The areas of application for digital video compression standards include all areas mentioned in Section 1.1.

A device (software or hardware) that compresses data is often known as an *encoder* or a *coder*, whereas a device that decompresses data is known as a *decoder*. A device that acts as both a *co*der and a *dec*oder is known as a *codec*.

1.3 Video Compression Methods and Standards

Basically, compression is performed when an input video stream is analyzed and information that is indiscernible to the viewer is discarded. Each event is then assigned a code. Commonly occurring events are assigned few bits and rare events have more bits. These steps are commonly called signal analysis, quantization and variable length encoding, respectively. Four common methods for compression are *discrete cosine transform* (DCT), *vector quantization* (VQ), *fractal compression*, and *discrete wavelet transform* (DWT).

The DCT based compression algorithm samples an image at regular intervals, analyzes the frequency components present in the sample, and discards those frequencies which do not affect the image as the human eye perceives it. DCT is the basis of standards such as JPEG, MPEG, H.261, and H.263.

The VQ based compression algorithm looks at an array of data, instead of individual values. It can then generalize what it sees, compressing redundant data, while at the same time retaining the desired object or data stream's original intent. Fractal compression is a form of VQ. The compression is performed by locating self-similar sections of an image, then using a fractal algorithm to generate the sections.

Like DCT, the DWT based compression algorithm mathematically transforms an image into frequency components. As opposed to the other methods (DCT), which work on smaller pieces of the desired data, the process is performed on the entire image. The result is a hierarchical representation of an image, where each layer represents a frequency band.

Standardization of compressed digital video formats has been an ongoing work for about two decades. International organizations have described several standards for digital audio-videos and images based on the methods defined above [2], [7-9]. The following paragraphs give a brief description of these standards:

Moving Picture Experts Group (MPEG), an ISO/IEC working group was established in 1988 to develop standards for digital audio and video formats. There are five MPEG standards being used or in development. They include MPEG-1, 2, 4, 7, and 21. Each compression standard was designed with a specific application and bit rate, although MPEG compression scales well with increased bit rates [39].

H.261 and H.263 are ITU standards. H.261 was designed for two-way communication over ISDN lines (video conferencing) and supports data rates which are multiples of 64Kbit/s and CIF, QCIF resolutions. H.263 is based on H.261 with enhancements that improve video quality over modems. It supports CIF, QCIF SQCIF, 4CIF and 16CIF resolutions.

Besides the ISO/IEC and ITU standards, there are also some other industrial standards, such as DV or DivX. Most of these schemes (MPEG-1, 2, H.261, 263, DV) employ DCT as a compression method. Besides its wide spread application in video compression techniques, the still image compression standards, such as JPEG, also utilize DCT. Researchers have been studying on wavelets for image and video applications to overcome the problems related to DCT [5]. Following the determination of wavelets as the fundamental tool for JPEG-2000 standard, more and more efforts to explore DWT based schemes have begun to grow [2].

1.4 Motivation for DWT Based Schemes

In the 1930s, several researchers working on the representation of functions using *scale-varying basis functions*, so called *wavelets* revealed that Haar basis function, which had appeared as the first mention of wavelets (1909), is superior to the Fourier basis functions to analyze small complicated details in motion [40]. After that time, investigations of wavelets had continued mostly in the field of Mathematics and Physics. In 1985, Mallat [34] discovered the relationships between orthonormal wavelet bases and pyramid algorithms. Influenced by these results, Meyer [41] constructed the first non-trivial wavelets. Unlike the Haar wavelets, the Meyer wavelets are continuously differentiable; however they do not have compact support, which means that they do not vanish outside of a finite interval. A few years later, Daubechies [17] used Mallat’s work to construct a set of refined wavelet orthonormal basis functions, which have become the cornerstone of wavelet applications today.

To fully understand why DWT is being used in very low bit rate image and/or video applications, one has to understand the use of *transform coding* techniques. In transform image coding, image is transformed to a domain significantly different from the image intensity domain, and then transform coefficients are coded. The technique attempts to reduce the correlation that exists among image pixel intensities. When the correlation is reduced, redundant information does not have to be coded repeatedly. In addition, transform coding techniques exploit the observation that for typical images a large amount of energy is concentrated in small fraction of the transform coefficients. This is the *energy compaction property*. Due to this property, it is possible to code only a fraction of the transform coefficients without seriously affecting the image. This allows images to be coded at bit rates below 1 bit/pixel with a relatively small sacrifice in image quality and intelligibility. The utilization of DWT in the image coding can also be extended for the video coding schemes. A more detailed study on the wavelet analysis of video signals is given in Chapter 3.

1.5 Survey of Previous Work

There are extensive investigations on *video compression and coding* concept in the literature. The *hybrid coding*, *subband/wavelet coding* and some comparative studies on different *transform coding* techniques are found. A short summary of them are given in the following paragraphs.

Motion compensated (MC) predictive coding has a *hybrid* structure whereby subband/transform coding in the spatial domain is applied to MC prediction error signal. That is to say, employing prediction with motion compensation along the temporal axis and 2-D DCT (for its nice *de-correlation and energy compaction properties*) coding in the spatial domain is the technique used in most of the current video compression standards (e.g., MPEG-2 [7] and H.263 [8]).

Other works based on *hybrid* coding had been reported on motion compensated sub-band coding (SBC), e.g., [10-11] where the DCT frequency decomposition was replaced by 2-D subband filterbanks. Indeed, SBC had emerged as a superior technique for encoding of 2-D image signals, which could overcome the blocking artifacts inherent in DCT schemes. Zhang [12] introduced a zero-tree wavelet video coder for which the main issue was the scalability. The wavelet transform, being as one of the transform coding techniques, may also be regarded as a special case of SBC, with the transform's bases functions interpreted as the impulse response of a filterbank.

3-D based schemes came no sooner than *hybrid* coding in the literature. Nicoulin *et al*. [13] modified the initial motion compensated 3-D DCT concept, where so-called *unconnected* pixels had been treated separately, by adding a MC prediction error signal on the original frame and then applying the DCT in the temporal domain. Ohm [14] made progress by processing *unconnected* pixels simultaneously with the *connected* ones and proposed 3-D SBC by cascading temporal two-channel subband analysis. He initially performed the temporal analysis with two-tap MC filters using block matching and later extended his temporal filtering method to more general motion vector fields. Taubman and Zakhor [15] introduced a multi-rate video coding system using global motion compensation for camera panning. Woods [16] reported a related work to Ohm's [14] with reduction in MC filtering artifacts and got better results than MPEG-1[9].

The work on "Image Coding Using Wavelet Transform" [17] became popular in the image and later video coding community as a new scheme for image compression. It proposed *wavelet transform* in order to obtain a set of bi-orthogonal sub-classes of images and employed *vector quantization* to form a bit plane. Following this study, a very effective and computationally simple technique for image compression called an *embedded zerotree wavelet* (EZW) coding was introduced [18]. The technique was based on three concepts: (1) wavelet transform or hierarchical subband decomposition, (2) prediction of the absence of significant information across scales by exploiting the self-similarity inherent in images, and (3) hierarchical entropy-coded quantization. Pearlman *et al.* [19] later offered an alternative explanation of the principles of EZW operation with a new and different implementation based on *set partitioning in hierarchical trees* (SPIHT) which provided even better performance. The 3-D version of the SPIHT video coder

[20] proved a comparable performance to H.263, but with much lower complexity when MC is not used.

While good results obtained by wavelet coders (e.g., EZW, SPIHT coder) are partly assignable to the wavelet transform, Xiong *et al*. [21] emphasized that image and video compression algorithm should be addressed from the overall system viewpoint: quantization, entropy coding, and the complex interplay among elements of the coding system are more important than spending all efforts on optimizing the transform. This study illustrated that the main factors in image and video coding are the quantizer and entropy coder rather than the difference between wavelet transform and the DCT in their comparative study of DCT- and wavelet-based coding for both still images and video sequences.

Following the determination of wavelets as the fundamental tool for JPEG-2000 standard [22], more researches are being published on wavelet transform based video schemes today: Currently, MPEG is investigating a possible scalable video coding standard (SVC); a dream within the video coding research community, based on motion compensated temporal filtering (MCTF) with lifting scheme [23]. The lifting scheme provides a flexible framework for building wavelet transforms [24]. The advantages of this scheme are both in terms of complexity ("in-place" calculation) and additional functionalities [25]. More recently, there have been many journals investigating the benefits of the formalism for motion compensated wavelet decomposition [26-31].

1.6 Wavelet Based Video Compression and Coding

In this thesis, a low bit rate video coding system based on three-dimensional (3-D) wavelet coding is studied. Using the different *wavelet filters* and including the *motion estimation* in the transform algorithm are the methods employed in this study to compare variant impacts on the video coding system.

Still image compression using wavelets is accomplished by applying the wavelet transform to de-correlate the image data, quantizing the resulting transform coefficients, and encoding the quantized values. This is usually carried out in a row-by-row then a column-by-column basis when using a separable one-dimensional wavelet transform. A natural extension is to apply wavelet compression to a 3-D image, with the temporal axis being the third dimension. Thus, in 3-D wavelet coding schemes, the frequency decomposition is carried out in addition along the temporal axis of a video signal. Slow movement in the video content will produce mostly low frequency components and the high frequency components will be almost negligible. This *energy compaction property* of wavelet decomposition allows us to ignore some frequency components and still have a good representation of the signal.

The obvious advantage of utilizing a temporal decomposition is that representation of temporal changes can be very efficient in the transform domain when movement is slow. The forward transform is implemented using analysis filters, and, generally, better analysis is achieved when filters are longer and so that correlation among more frames can be discovered. One problem inherent in such a scheme is that if it is required to analyze video over long sequences, one must first buffer a large number of frames. However, in interactive applications, such as video phone, buffering of frames results in a corresponding time delay. This problem is solved by buffering a moderate number of frames and using only Haar wavelet, a filter with a length of two taps, in the temporal direction.

If motion occurs, the correlation along the temporal filter path of the subband/wavelet analysis may be lowered. Therefore, to attain high energy compaction in the case of motion, it would be convenient to employ motion compensated temporal analysis, which means that the temporal wavelet filtering is performed along the motion trajectory. Since the success of the process strictly depends on the accuracy of motion estimation, it is necessary to use an appropriate motion estimation technique with a good motion model. It is also important for this model to ease the implementation of temporal filtering.

Among a number of motion estimation models, block-based motion estimation, called a *block motion model*, is the most popular approach where it is assumed that the image is composed of moving blocks. The block-based motion compensation has been adopted in the international standards for digital video compression, such as H.261 and MPEG 1-2. There are a few block-motion models and a special domain search approach, called the *block-matching* method is introduced here. The basic idea behind block matching is to determine the displacement for a pixel in the present frame by considering a block centered about the pixel, and searching target frame for the location of the best-matching block of the same size. The block size is one of the most important parameters in any block-based motion estimation algorithm. It usually involves a tradeoff between two conflicting requirements: (1) the window must be large enough in order to be able to estimate large displacement vectors. On the other hand, (2) it should be small enough so that the displacement vector remains constant within the window. To achieve a balance between the two, resizing the block and the search window is necessary in each decomposition level.

Differing from the order of spatial and temporal filterings, there are two main approaches for implementing 3-D wavelet transform algorithms: (1) the temporal transform is followed by the spatial transform (t + 2-D), and (2) the spatial subband/wavelet transform occurs first, followed by the temporal transformation of each subband (2-D + t). The order of spatial and temporal filtering results in different coder complexity and gain [32], and the former is most commonly employed in video coding applications including this study. The subband/wavelet decomposition in the spatial domain following the temporal decomposition helps to capture the most spatial redundancies in neighboring pixels in individual temporal subbands. Of course, the efficiency of decomposition is directly related to both scene content and the wavelet kernel used. Since the size of frames is usually large enough to exploit redundancies in neighboring pixels, using longer tap filters on the spatial domain would be more beneficial. In fact, we employed such a filter of Daubechies [17] with 9/7 tap, symmetric, bi-orthogonal filter characteristics in our comparative studies with shorter ones.

The implementation ends up with quantizing the resulting transform coefficients, and encoding the quantized values. In this work, a simple, yet remarkably effective image coding algorithm to encode the individual frames is described. The coding algorithm is based on three key concepts: (1) partial ordering of transformed frame elements by magnitude, with transmission of order by a subset partitioning algorithm that is duplicated at the decoder, (2) ordered bit plane transmission of refinement bits, and (3) exploitation of the self-similarity of the image wavelet transform across different scales. These concepts actually bring out the property that the bits in the bit stream are generated in order of importance, and yielding a fully hierarchical image compression suitable for embedded coding or progressive transmission. Given a frame bit stream, the decoder can cease decoding at any point in the bit stream and still produce exactly the same frame that would have been encoded at the bit rate corresponding to the truncated bit stream.

The crucial parts of the coding process – the way subsets of coefficients are partitioned and how the significance information is conveyed – are fundamentally different from aforementioned works where arithmetic coding of the bit streams was essential to compress the ordering information as conveyed by the results of the significant tests. Here, the effective subset partitioning algorithm allows us to achieve satisfactory performance even without binary un-coded transmission.

A feasibility study on the discrete wavelet transformation is initially performed on Matlab. Then, a C++ program with graphical user interface has been developed on the Microsoft Visual Studio running on Intel® processor. Including the comments, over ten thousand Source Lines of Code (SLOC) has been developed to implement the following modules:

Work Packet	**SLOC**
Matlab Functions	~0250
Haar Wavelet Analysis/Synthesis	~0850
MC Haar Wavelet Analysis/Synthesis	~1500
Daubechies-9/7 Wavelet Analysis/Synthesis	~1350
MC Daubechies-9/7 Wavelet Analysis/Synthesis	~1850
2-D SPIHT Encoding/Decoding	~4100
Test Programming	~1250

TOTAL	**~11150**

The test program consists of the different test cases, such as temporal only, spatial only and spatial-temporal wavelet analysis and synthesis implementations. In these test cases, the coding written in other work packages are mostly reused.

1.7 Organization of the Thesis

This thesis study is organized in five chapters including the introductory chapter. In Chapter 2, fundamentals of image and video compression techniques are given. Among several redundancy types, the spatial redundancy procedure is taken into consideration in detail to remove via the DCT and the DWT.

In Chapter 3, the 3-D wavelet analysis of video signals is studied. The realization of temporal only and spatial-temporal subbands of different video signals is performed using the Haar and Daubechies-9/7 wavelet filters. Several test cases are generated to asses the factors effective on the signal compression.

In Chapter 4, the 2-D SPIHT coding algorithm is employed to encode the coefficients that are created after either a spatial only or a spatial-temporal decomposition. The compression ratios obtained in different test cases are compared to evaluate the factors effective on the signal compression.

Finally, in Chapter 5, a brief summary of this study is given together with the comments. Suggestions are given to improve the system under consideration.

CHAPTER 2

FUNDAMENTALS OF IMAGE & VIDEO COMPRESSION

2.1 Human Visual System (HVS) and Compression

"The aim of an image/video compressor is to decrease the information content in all image/video signals by removing the information which cannot be perceived by HVS and eliminating the redundant information that is repeating itself" [5].

Depending on the methods used, c*ompression* may be a reversible operation. The quality of a reconstructed image/video is related with the perception of the HVS. Therefore, it is important to understand the features of HVS. Among various HVS features, the most commonly used are luminance contrast sensitivity and frequency sensitivity: The human visual system drops off with increasing spatial frequency. A small variation in intensity is more visible in slowly varying regions than in busier ones, and also more visible in luminance compared to a similar variation in chrominance.

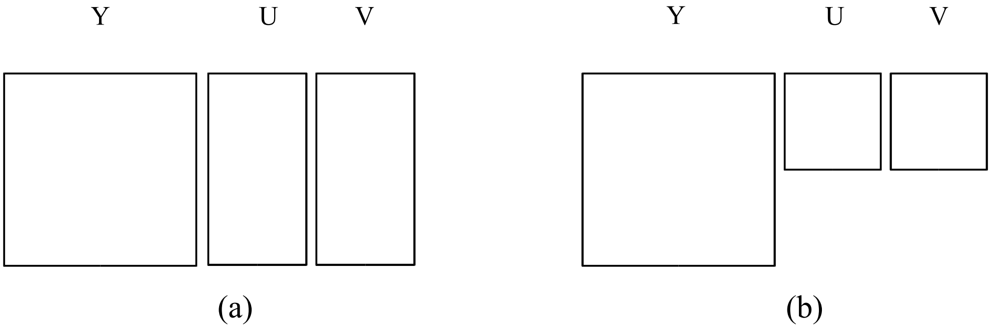
Fortunately, compression of image/video data without significant degradation of the visual quality is usually possible because images/videos contain a high degree of redundancy including the data which is not perceived by the HVS. In brief, the types of redundancy can be listed as follows: (1) spectral redundancy, due to correlation among the color components, (2) spatial redundancy, due to correlation between the neighboring pixels, (3) temporal redundancy, due to correlation among the group of frames, and (4) statistical redundancy, due to correlation among the symbols in bit planes.

In the following sections, a detailed description of such redundancies and the removal procedures are discussed.

2.2 Spectral Redundancy

"In general, the input for image/video compression schemes, i.e., the output of a professional video camera, is a full color video signal in RGB (Red-Green-Blue) color space. The RGB space is spectrally redundant. Since human eye is more sensitive to luminance than to chrominance, it is common to transform the color space into the YUV space as the first step of compression" [5], where Y is linked to the component of luminance, and U and V are linked to the components of chrominance (see Appendix A). The YUV space is better to eliminate the spectral redundancies.

"In the original digitized video signal, there are red (R), green (G) and blue (B) components for each pixel. Each of these components is usually represented with 8 bits. In YUV color space, there are one luminance (Y) and two chrominance (U and V) components for each pixel. Since U and V spaces are redundant, these signals can be downsampled by a factor of 2. In 4:2:2 YUV systems, the down sampling is applied only in horizontal direction, giving a 3:2 compression ratio. Whereas in 4:1:1 YUV conversion, U and V signals are downsampled in both horizontal and vertical directions and a 2:1 compression ratio is achieved" [5]. The YUV color space representation is shown in Figure 2.1.



(a) (b)

Figure 2.1 YUV Systems: (a) 4:2:2; (b) 4:1:1.

2.3 Spatial Redundancy

Besides the spectral redundancy, image/video signals have spatial information that cannot be perceived by human eye. Specifically, HVS is more sensitive to lower frequencies in the spatial domain. Therefore, higher frequencies in the signal can be compressed better without distorting the visual quality. This can be achieved by first transforming the signal into another domain where a better energy compaction is possible [5]. The transformation makes the signal be represented by numbers, called transform coefficients, in the transform domain. These coefficients, in general, give idea about the frequency content of the signal when the transformation techniques which analyze spectral components in the signal are used. Therefore, the suppression of higher frequencies actually means the suppression of transform coefficients which refer to those spectral components.

The success of the spatial redundancy elimination depends on distinguishing the coefficients that refer to different spectral components precisely. Fortunately, the way that transformation techniques are applied provides us with the information of the frequency localization. Furthermore, the magnitudes of coefficients give an idea about the relative weight of the frequency component in that location. Having this information, efficient spatial redundancy elimination is possible with a proper quantization of the coefficients.

Two main techniques are used to remove spatial redundancy: (1) Discrete Cosine Transform (DCT) and (2) Discrete Wavelet Transform (DWT). The DCT has been widely used in most compression schemes so far. The DWT is a more recent technique and its applications are still growing. Although they have advantages over one another, both techniques aim to localize the signal energy at different frequencies. Today, there are ongoing researches for the replacement of the DCT in image/video codecs with the DWT to take unique advantages of the wavelet transform, such as time localization of the spectral subbands.

2.3.1 The Discrete Cosine Transform (DCT)

The DCT helps separate images into spectral subbands of differing importance with respect to the image's visual quality.

In DCT-based image coding schemes, such as JPEG, images are divided into 8×8 blocks and processed separately. The DCT input is the 8 by 8 array of integers, which contains each pixel's gray scale level from 0 to 255. These blocks are formed to ease the implementation, since it does not need special requests of memory. It also reduces the number of operations to transform the entire image. The transformation is performed via DCT matrix of the same size as the block. Figure 2.2 illustrates a typical discrete cosine transformation of an 8 by 8 block from the "*Lena*" image and the transformation matrix used in the process. The output array of the DCT coefficients contains floating numbers due to multiplication of the input block from both sides.

For most images, much of the signal energy lies at low frequencies. The transform coefficients referring to low frequencies appear in the upper left corner of the DCT block. The higher the magnitude of coefficients is, the higher the energy of the corresponding spectral components and vice versa. Therefore, most of the signal energy is carried by those coefficients in the upper left corner of the DCT block. The lower right values represent higher frequencies, and are often small – small enough to be neglected with little visible distortion.

The resultant DCT coefficients at each block are quantized using 8×8 matrices, which contain the quantization step size for each coefficient. The quantization matrix is obtained by multiplying a base matrix by a quantization parameter. This quantization parameter is typically used to tune the image/video coding: A larger quantization parameter results in lower quality as well as smaller size (in bits) of the encoded image or video frame. The quantized DCT coefficients are finally variable-length coded for a more compact representation. The decoder performs the opposite operations.

8×8 Input Block

76	74	77	105	136	122	73	37
85	112	136	137	103	49	24	35
122	104	94	102	92	48	21	32
105	81	74	92	88	47	35	65
86	84	97	118	98	42	28	62
89	83	87	101	92	50	32	52
66	79	85	79	58	34	42	76
66	79	81	71	59	44	43	59

8×8 DCT Transform Matrix

0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536
0.4904	0.4157	0.2778	0.0975	-0.0975	-0.2778	-0.4157	-0.4904
0.4619	0.1913	-0.1913	-0.4619	-0.4619	-0.1913	0.1913	0.4619
0.4157	-0.0975	-0.4904	-0.2778	0.2778	0.4904	0.0975	-0.4157
0.3536	-0.3536	-0.3536	0.3536	0.3536	-0.3536	-0.3536	0.3536
0.2778	-0.4904	0.0975	0.4157	-0.4157	-0.0975	0.4904	-0.2778
0.1913	-0.4619	0.4619	-0.1913	-0.1913	0.4619	-0.4619	0.1913
0.0975	-0.2778	0.4157	-0.4904	0.4904	-0.4157	0.2778	-0.0975

8×8 DCT Output Matrix

600.625	138.223	-79.971	-31.917	60.375	-20.015	-0.023	0.128
60.023	24.150	-55.694	39.704	-0.098	0.411	-0.503	-0.039
-0.067	-35.330	-16.095	11.550	-40.047	0.212	-0.088	-0.161
21.237	-35.750	-16.095	14.618	0.173	-0.131	-0.592	-0.015
0.375	-44.337	-0.720	28.023	0.125	-0.004	-0.107	0.176
-11.757	-35.790	28.236	31.882	-0.034	-0.214	0.313	0.001
0.163	0.311	-0.088	43.892	-0.324	-0.224	0.345	0.352
0.035	0.473	-0.207	0.337	-0.147	0.009	-0.452	-0.554

Figure 2.2 8×8 Block DCT

The crucial point in the quantization is that the DC coefficient, the lowest indexed coefficient in both horizontal and vertical dimensions (600.625 in Figure 2.2) is the most important coefficient in the block and should be coded with minimum loss. The other coefficients, called AC coefficients, on the other hand, are scanned in the order of their frequencies as shown in Figure 2.3. The aim in such a scheme is (1) to map 8×8 block DCT coefficients to a 1×64 vector, and (2) to group low frequency coefficients in top of the vector. Since HVS is insensitive to details, higher frequency coefficients can be compressed using a quantization with larger decision levels.

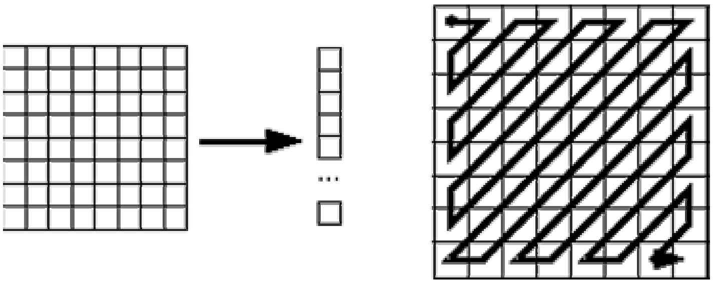


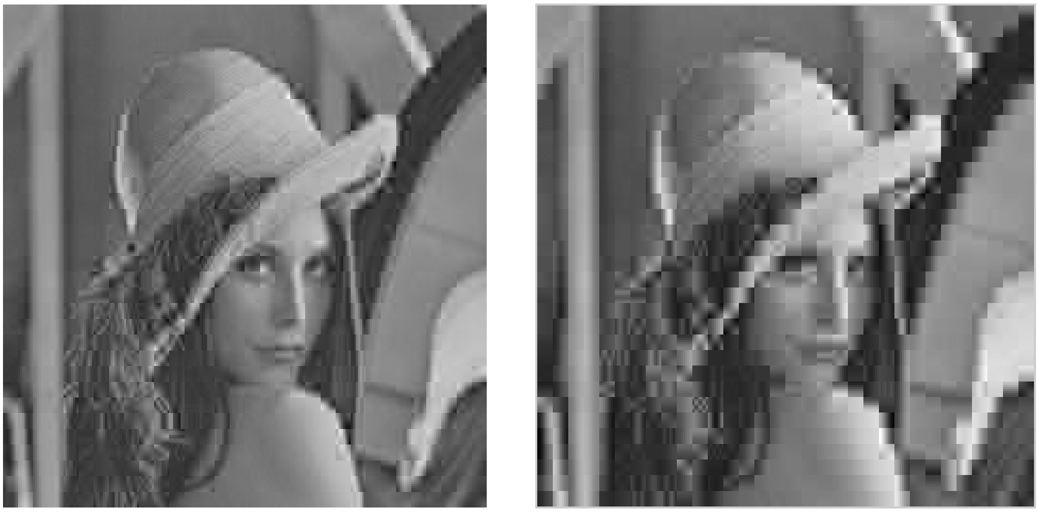
Figure 2.3 Ordering of the DCT Coefficients by a Zigzag Scan

In practice, the quantization level in the DCT based codecs is adjusted based on the target bit rate [7-9]. While the target bit rate gets smaller, higher frequencies are more suppressed. Since each block is coded and quantized independently, the block boundaries can be detected at low bit rates. This is one of the main disadvantages of the DCT. To get rid of the blocking artifact, researchers proposed modified techniques, such as the *lapped orthogonal transform* [33].

In order to better realize the DCT-based image compression schemes, we tested the "*Lena*" image (256 × 256) by Matlab. The original image is reconstructed using less number of transform coefficients. Two test cases are considered: (1) reconstruction using 2 DCT coefficients in each DCT block and (2) reconstruction using 4 DCT coefficients in each DCT block. The coefficients used in the reconstruction are determined according to their order of importance. That is, most of the coefficients in the coefficients' vector are ignored, and only 6.25% and 3.125% of the coefficients remained are used, respectively. The Peak Signal to Noise Ratios (PSNRs) computed according to Equation 2.1 are used as a quality measure of the reconstructed images. Figure 2.4 shows the reconstructed images and their PSNRs.

$$PSNR = 20 \times \log_{10}\left(\frac{255}{\sqrt{\frac{\sum_{r,c}\left(I_{r,c} - \tilde{I}_{r,c}\right)^2}{N \times M}}} \right) \tag{2.1}$$

where $I_{r,c}$ and $\tilde{I}_{r,c}$ denote the pixel values at position (*r-row, c-column*) of the original and reconstructed images, respectively; assuming the images have size of *N* rows (height) and *M* columns (width) in pixels.



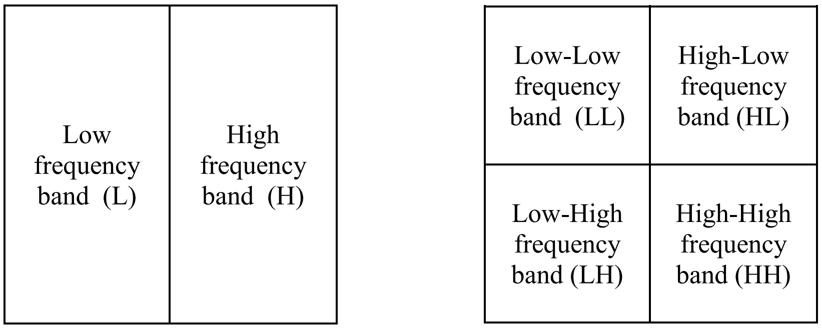
(a) (b)

Figure 2.4 Reconstruction of the "*Lena*" Image from its DCT Coefficients: (a) 4 coefficients (6.25%), 30.62 dB; (b) 2 coefficients (3.125%), 26.97 dB.

In Figure 2.4, there are occurrences of blocking artifacts. The irritation to the human eye in the reconstruction (a) is not as much as the reconstruction in (b). The blocking artifacts are the nature of the DCT coding especially at very low bit rates as in Figure 2.4-b.

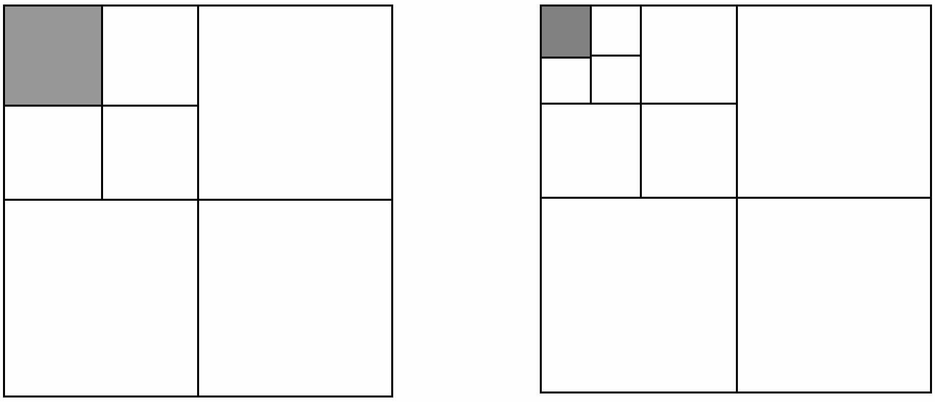
2.3.2 The Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transforms used on images is perfomed by applying the 1-D DWT (see Appendix B) on two directions: first on rows and then on columns. Sub-sampling operates in two directions reduces the image pixel count to a quarter in the transformed image as shown in Figure 2.5-b. This scheme is different than the block-wise DCT decompostion since the DWT operates on the entire image. Further wavelet transform of the low frequency portion (LL subband) can be taken by increasing the depth (level) of the transformation as illustrated in Figure 2.6-a and 2.6-b. The original image can still be reconstructed exactly but the inverse transform will have to be taken 3 times.


Low frequency band (L)
High frequency band (H)
Low-Low frequency band (LL)
High-Low frequency band (HL)
Low-High frequency band (LH)
High-High frequency band (HH)

(a) (b)

Figure 2.5 1-level 2-D DWT on Images:
(a) DWT along rows; (b) DWT along columns.



(a) (b)

Figure 2.6 Multi-level 2-D DWT on Images:
(a) 2-level DWT; (b) 3-level DWT.

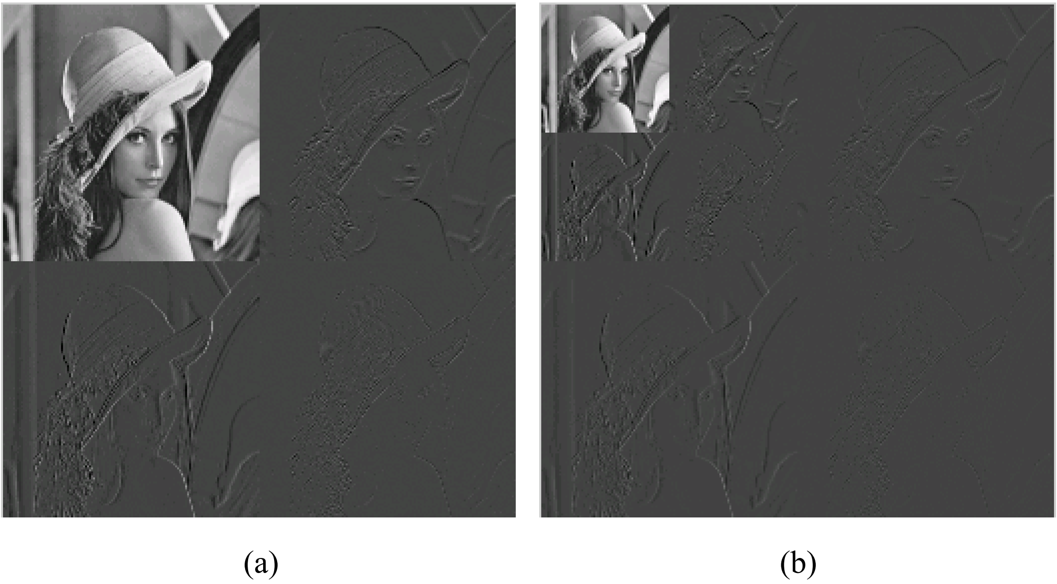


Figure 2.7 DWT of the "*Lena*" Image:
(a) 1-level decomposition; (b) 2-level decomposition.

Figure 2.7-a shows a one level decompostion of the "*Lena*" image. The two-dimensional wavelet transform decomposes the "*Lena*" image into four frequency bands, each one quarter the size of the original. The LL subband is a coarse scale approximation of original image, and the rest of frequency bands (HL, LH, HH) are detail signals along the horizontal, vertical and diagonal orientations, respectively. The lowest resolution band (LL) corresponding to low frequency portion contains the coefficients by which most of the signal energy is captured. All the other subbands contain the detail information corresponding to high frequencies. The high frequency bands have small magnitudes and easy to compress. Since the human eye is not sensitive to high frequency (detail) information, the high frequency quarters of the transformed image could be discarded even without thresholding. Images reconstructed without the high frequency information will be very much like the original.

The transform can be applied recursively on the LL sub-image (on the top left corner) to obtain decomposition at coarser (lower) scales as shown in Figure

2.7-b. This yields a hierarchical decomposition of the image. The number of decomposition levels depends on the original image size. It may last until one coefficient remains to represent the entire image.

A properly designed quantizer and entropy encoder are required along with the wavelet transformation to accomplish the compression. Being an advanced wavelet coding algorithm, *Zerotree Coding* is used in most wavelet-based image coding schemes. The image is represented by a tree-structured data constructed as illustrated in Figure 2.8. The root node, which is the parent of three nodes, represents the scaling function coefficient in the lowest frequency band. All the nodes inside the tree correspond to wavelet coefficients at a scale determined by their height in the tree. Each of these coefficients has four children which correspond to the wavelets at the next finer scale having the same location in space. At the bottom of the data structure lie the leaf nodes, which have no children. If the node value is insignificant according to a pre-specified threshold, all its children are ignored. These ignored coefficients are decoded as zeros at the decoder. If the node value is significant, each of its four children is selected as root of a new tree. The process continuos until all trees are evaluated.

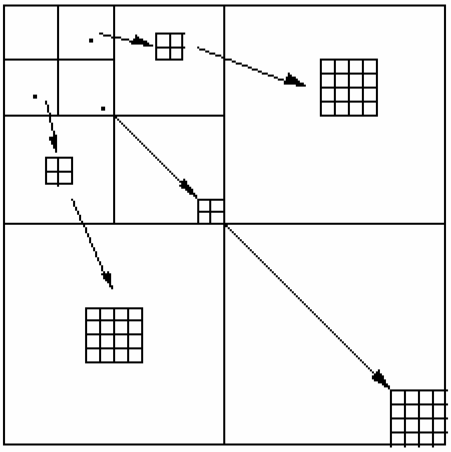


Figure 2.8 Zerotree Structures

In order to better realize the DWT in image compression schemes, we tested the "*Lena*" image (256 × 256) by Matlab. The Haar and Daubechies-4 (Db-4) analysis filter coefficients used in the decompositions are given below:

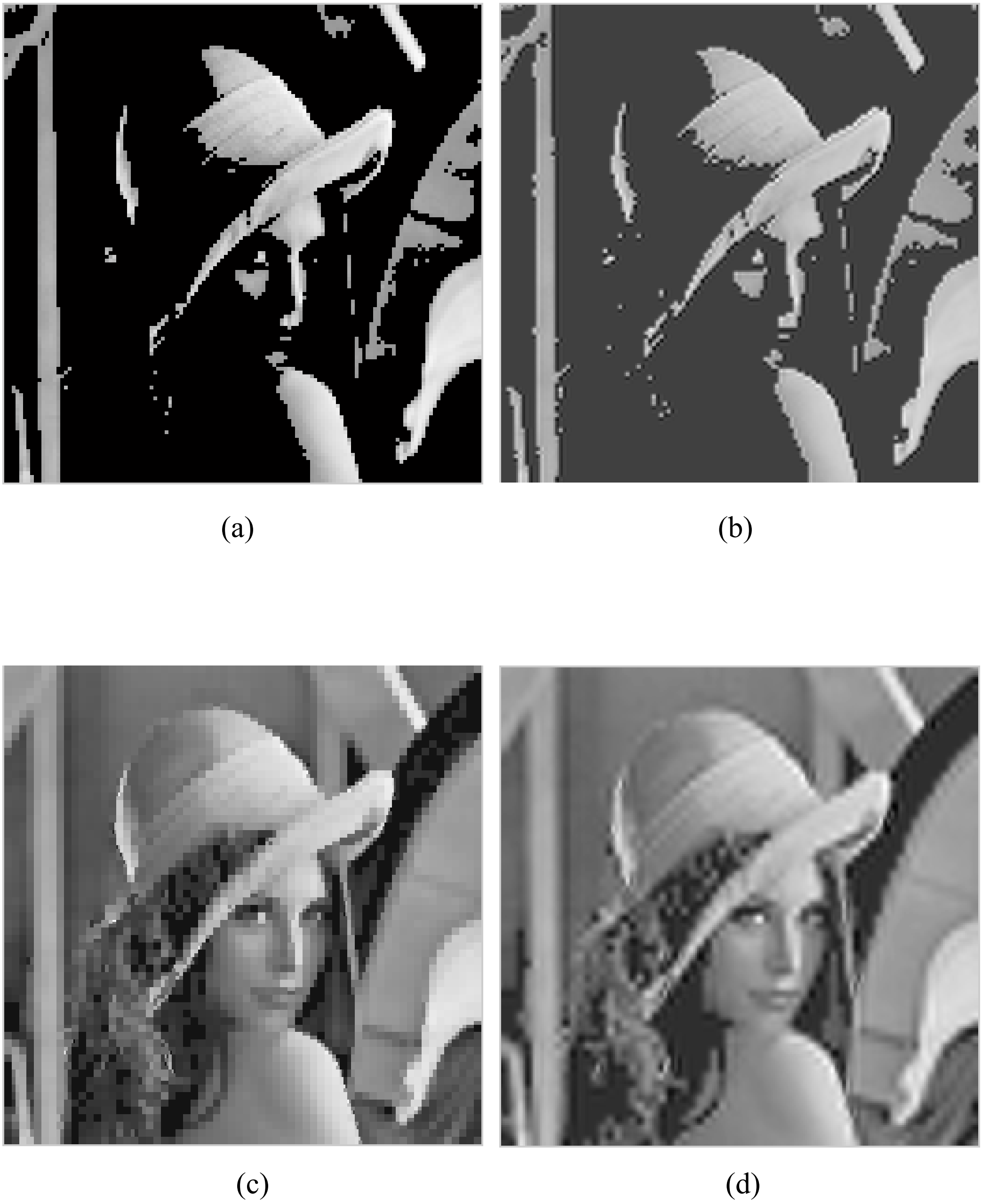
h_{haar} = [1/√2 1/√2],
g_{haar} = [-1/√2 1/√2].

$h_{db\text{-}4}$ = [-0.0106 0.0329 0.0308 -0.1870 -0.0280 0.6309 0.7148 0.2304],
$g_{db\text{-}4}$ = [-0.2304 0.7148 -0.6309 -0.0280 0.1870 0.0308 -0.0329 -0.0106].

While reconstructing the original image, only 4096 and 2048 transform coefficients are used for two different test cases. The effect of the multi-level decomposition on the quality of the image is observed. Figure 2.9 and Figure 2.10 show the reconstructed images. The PSNR for each reconstruction is also given in decibel (dB).

It is observed that the Haar and the Daubechies-4 filters have different affects on the quality of the reconstructed images. Having longer taps, Daubechies-4 filter is expected to better de-correlate information content. This, in turn, makes it moderately stronger candidate in image/video compression algorithms. We say "*moderately*", since the strength comes with complexity in processing the extended number of samples in the transform domain. In fact, the number of wavelet coefficients at each level (from level-1 to level-4) is counted as 68644, 70528, 71542, and 72034, respectively. Correspondingly, the ratio of the number of coefficients used in the reconstruction process (4096) is 6, 5.8, 5.7, and 5.6 percentages of all samples at respective levels. However, the percentage is constant (6.25%) for the Haar synthesis because the Haar decomposition does not change the total number of coefficients (65536) at all levels. This proves the expectation that entropy of the coefficients is closer to zero and that better de-correlation is achieved in the case of Daubechies-4.

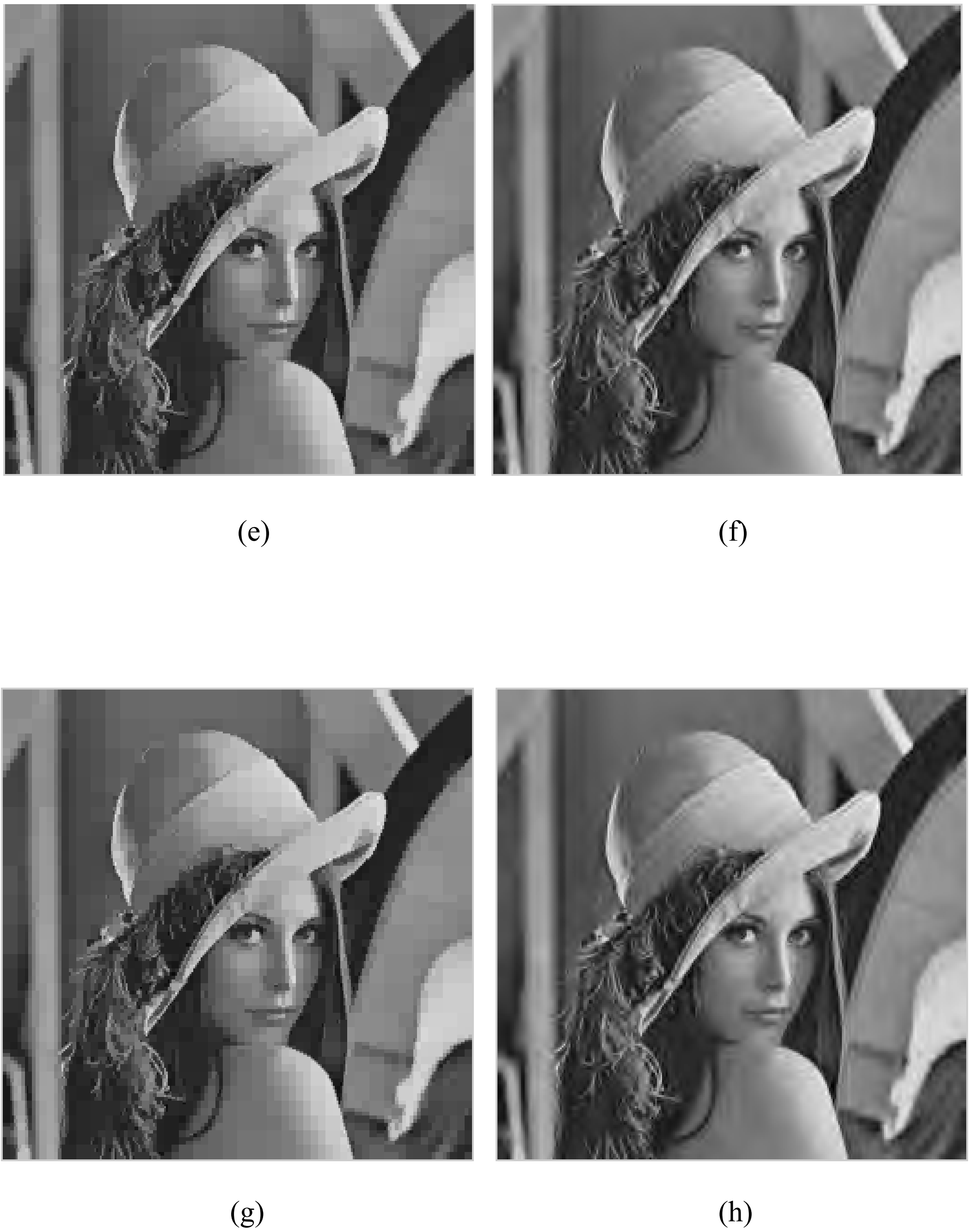
If PSNRs are taken as a measure of the quality for the '*Lena*' image, the best performance is achieved when it is reconstructed from the 4th level. The wavelet transformation makes the signal energy concentrated mostly in LL subbands.



(a) (b)

(c) (d)

Figure 2.9 Reconstruction of the “*Lena*” Image from 4096 Coefficients:
(a) 1-level Haar, 12.39 dB; (b) 1-level Db-4,12.37 dB;
(c) 2-level Haar, 28.11 dB; (d) 2-level Db-4, 27.19 dB;
(e) 3-level Haar, 32.81 dB; (f) 3-level Db-4, 34.07 dB;
(g) 4-level Haar, 33.16 dB; (h) 4-level Db-4, 34.29 dB.



(e) (f)

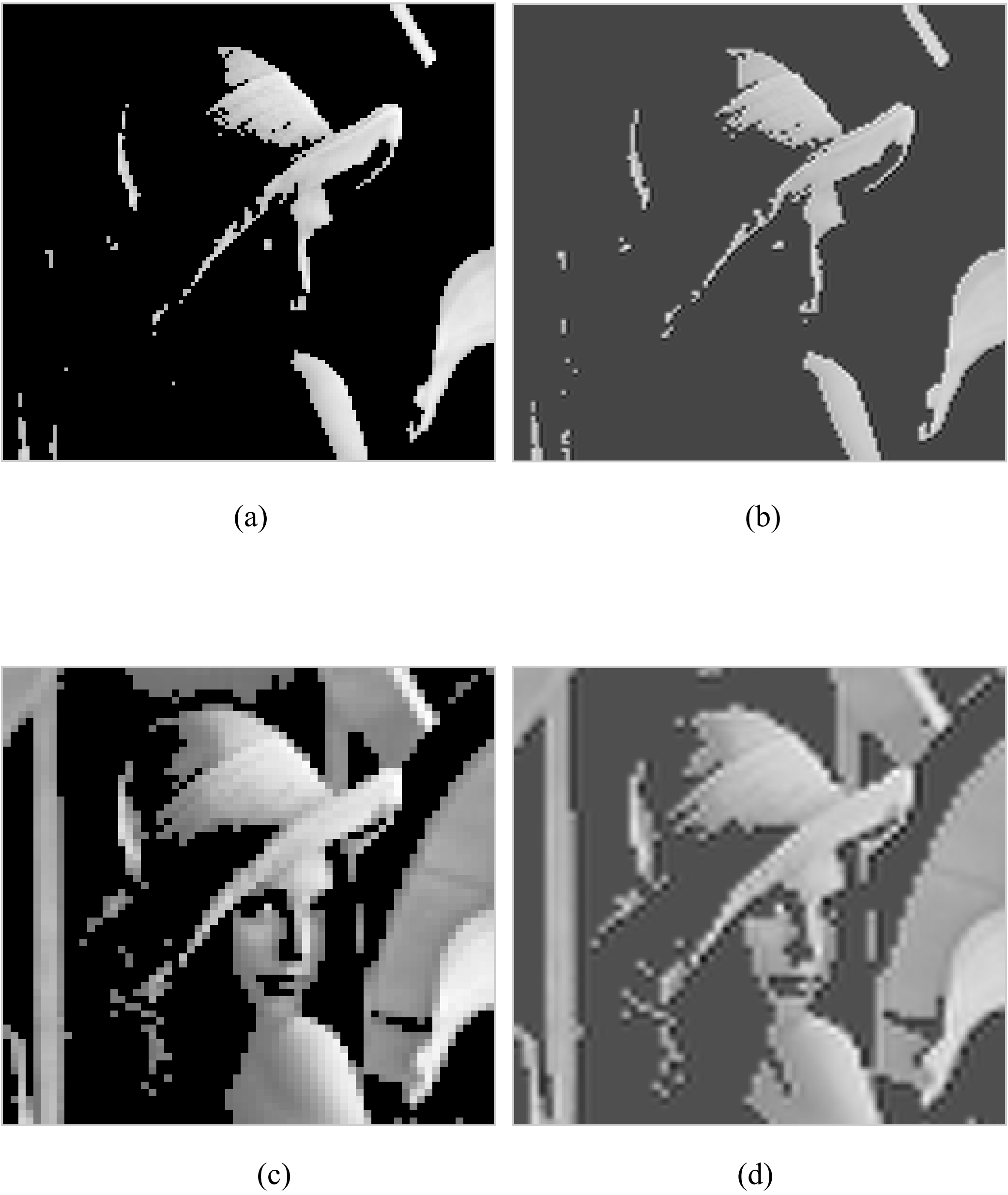
(g) (h)

Figure 2.9 Reconstruction of the “*Lena*” Image from 4096 Coefficients (cont’d)

The number of samples (coefficients) in LL subbands of the "*Lena*" image decomposed by the Haar filters are 16384 at level-1, 4096 at level-2, 1024 at level-3, and 256 at level-4. It is even more in the case of Daubechies-4 (17161, 4761, 1444, and 484). Having these numbers, the majority of the signal energy is shared by approximately one forth of all coefficients. The 4096 coefficients used in the reconstruction are not sufficient to comprehend the coefficients in the LL band of the first level. Therefore, reconstruction from the first level is not satisfactory. The coefficients are only sufficient when reconstructing from the second level, where less number of coefficients (4096 and 4761 in the case of Haar and Db-4, respectively) represents the entire signal. The energy compaction is better achieved. This helps us taking all significant coefficients when determining the 4096 coefficients in the reconstruction process. The decomposition to further levels decorrelates the samples even more. At each level, the LL subband represents coarser information than the subband of the previous levels. Thus, the energy of the signal cumulates in narrower bands as we go down the decomposition tree. It becomes easy to allocate all the coefficients of the coarsest (LL) subband plus more coefficients from the finer subbands (LH, HL, HH). This turns out to be better reconstruction of the original image.

The second case of the Matlab test is concerned with reconstructing the original image via 2048 coefficients. The same procedure is followed as in the previous case. The impacts of the Haar and Daubechies-4 wavelet filters on the reconstruction process are investigated. Moreover, the effects of the depth of the decomposition tree on the quality of the reconstructed image are also considered. Together with the previous test case, it is also aimed to complete a comparative study with DCT-based spatial redundancy elimination introduced in Section 2.2.1. The reconstructed images and their PSNRs are given in Figure 2.10.

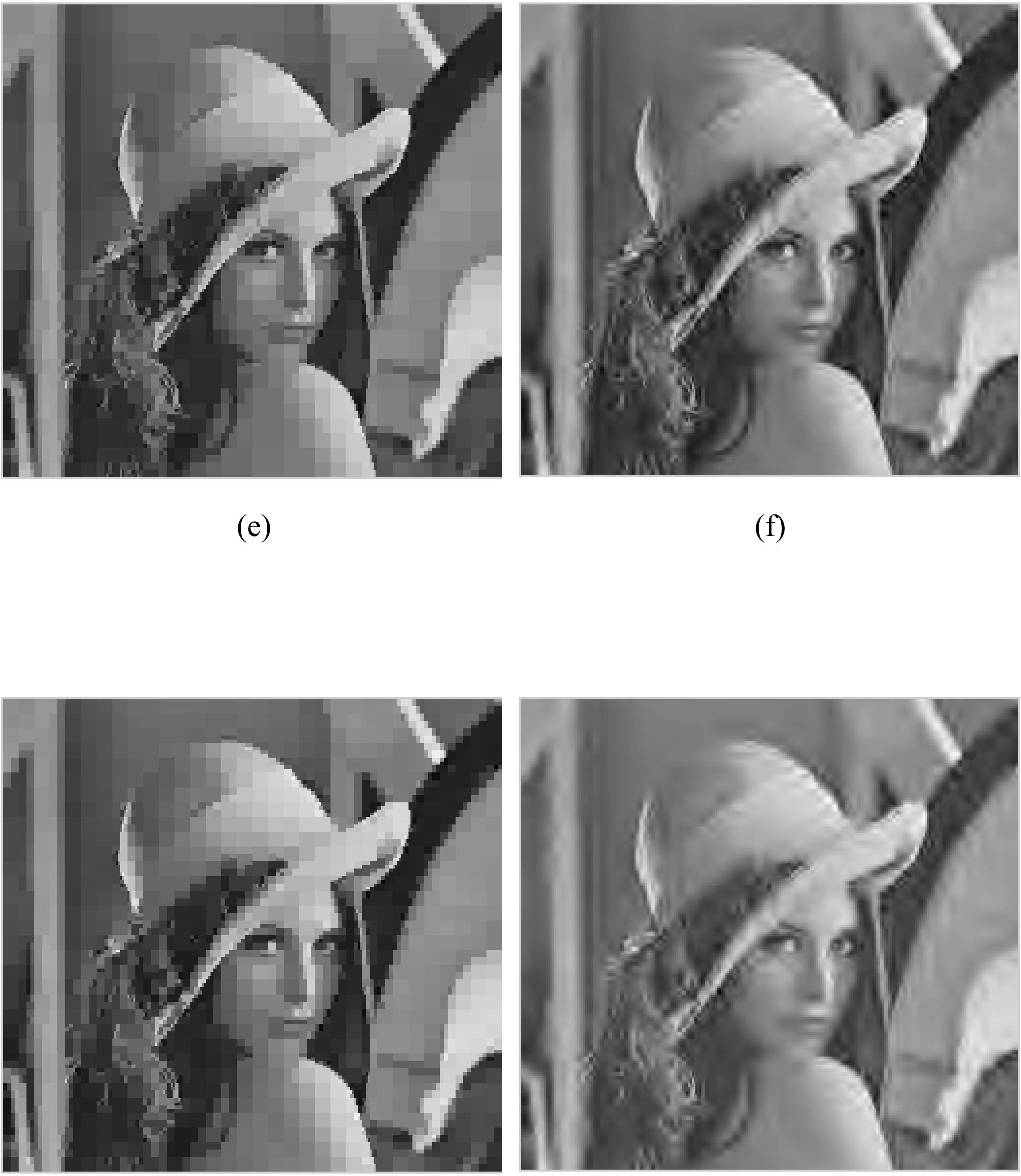
The results of the second test are consistent with that of the first one. The PSNRs and correspondingly the quality of reconstructed images became worse in the second test case. It is the number of coefficients that caused this reduction in the quality. In the Haar case, the reconstruction was performed using only 3.125% of all coefficients at all levels. The ratios in the Daubechies-4 case are 2.98, 2.90,



(a) (b)

(c) (d)

Figure 2.10 Reconstruction of the "*Lena*" Image from 2048 Coefficients:
(a) 1-Level Haar, 10.42 dB; (b) 1-Level Db4, 10.43 dB;
(c) 2-Level Haar, 16.72 dB; (d) 2-Level Db4, 15.98 dB;
(e) 3-Level Haar, 28.85 dB; (f) 3-Level Db4, 29.45 dB;
(g) 4-Level Haar, 29.64 dB; (h) 4-Level Db4, 30.49 dB



(e) (f)

(g) (h)

Figure 2.10 Reconstruction of the "*Lena*" Image from 2048 Coefficients (cont'd)

2.86, and 2.84 at levels 1, 2, 3, and 4, respectively. The more samples we suppress, the more compression ratio we get. Since twice these ratios were the case in the previous test, the compression ratio here is twice the previous one.

2.3.3 The Comparison of DCT and DWT

In wavelet transform, similarities between the wavelets and the original signal are checked whereas in DCT the check is between a fixed basis function cosine and the original signal. This comes with advantages for the DWT based schemes. The basis functions can be changed to suit the range of images. For example, while a basis function could be used for smooth images, another basis function can be selected for images that contain a lot of sharp edges.

Naturally, the noticeable and annoying *blocking artifacts* exist in DCT based image coders where images are partitioned into blocks of 8×8 pixels. In the DWT test, we did not observe the *blocking artifacts* in the reconstructed images. Moreover, PSNR values are as good as that of the DCT case. If the synthesis starts from deeper decomposition levels, it is even possible to obtain better reconstructions. A few words to mention about the success of the DCT based image compression schemes may be its frequency localization property. While the PSNR values are about 30 dB in the DCT test, the same performance is achieved when the synthesis starts from the second level in the DWT case.

From Section 2.2 up to here, the spatial redundancy removal techniques in image/video coding are given. We concentrated on commonly used DCT and the recent DWT coding. Each is effectively used in JPEG and JPEG2000, the standardized still image coding techniques, respectively [2]. MPEG-4, the latest video compression standard, includes wavelet transform among its procedures for the still texture compression [38]. On the other hand, MPEG-1, MPEG-2, H261, and H263 employ DCT [6].

2.4 Temporal Redundancy

The temporal redundancy elimination is a subject related to video compression algorithms. “The video signals are sampled at a rate of 25 or 30 frames per second. This makes the consecutive frames of video sequence to be almost similar. Especially, the background information will be the same if the camera movements and illumination changes are insignificant. This causes a temporal redundancy among the frames.

In order to remove the temporal redundancy, the first and perhaps the easiest approach is to subtract the previous frame from the current frame to reduce the information to be transmitted by assuming no camera motion. Another algorithm is to divide the image into blocks and try to estimate the current frame from previous frame with the motion information of each block. By sending this motion information, camera movement and motion of objects can be taken into account while reducing the temporal redundancies. This operation is called *motion compensation*. After estimating the motion, current frame is generated using the motion information and the previous frame. This method decreases the bit rate since instead of the whole image only the error between the blocks is sent to the decoder.

There are several approaches for estimating the motion in video. The most widely used system is to use fixed size rectangular blocks with a logarithmic or exhaustive search within a search window. Another approach is to use hierarchical motion estimation: Several resolutions of image are obtained by filtering and decimation operations. Motion estimation is applied at the lowest resolution and refined either by fixed or variable block size at each upper level.

Another approach is to utilize three-dimensional (3-D) transforms instead of motion estimation and the 2-D DCT on the spatial domain [5]". Transform coding techniques such as 3-D DCT or 3-D DWT are applied both in time and spatial domains. Today, more research is performed on 3-D based video coders. Researchers are especially working on fast and efficient 3-D transform techniques [3-4], [23], [25-31].

2.5 Statistical Redundancy

In addition to the redundancy types mentioned in the previous sections, there are also statistical redundancies in the compressed signal. The statistical properties of the video signals make happen such redundancies. In order to eliminate these redundancies, some statistical lossless coding schemes such as Run-length, Huffman, or arithmetic coding techniques are used [5].

CHAPTER 3

THREE-DIMENSIONAL WAVELET ANALYSIS OF VIDEO

3.1 Introduction

In Chapter 2, the two-dimensional (2-D) wavelet transform (WT) and its application on images were introduced. It was shown that the 2-D WT is performed by two separate 1-D transforms along the rows and the columns of the image data. The procedure creates four frequency bands called subbands (LL, HL, LH, and HH), each is (approximately) one-quarter the size of the original image. The transform is applied recursively to the LL sub-image to obtain decomposition at lower scales, yielding a hierarchical decomposition of the original image. The number of (decomposition) levels to apply depends on the image size. The LL band is a coarse approximation to the original image corresponding to low frequencies, and all the other bands corresponding to high frequencies are detail information in the horizontal, vertical and diagonal orientations. The energy of the high frequency bands is low, so they can be ignored when compressing the image.

The three dimensional (3-D) WT is a natural extension of the 2-D WT over a three dimensional signal. Video signals are formed by sequence of images (frames) and therefore, treated as 3-D signals with temporal axis being the third dimension. The 3-D wavelet decomposition is computed by applying three separate 1-D transforms along the coordinate axes of the video data: (1) rows and (2) columns of the image or frame, and (3) time. Being a more natural way of implementing the 3-D wavelet decomposition, (t + 2-D) approach is preferred in this study. It treats video as volumetric data. The temporal transform before the spatial transform is especially useful if the motion compensated (MC) decomposition is considered. The input video signal has the highest resolution before the motion estimation (ME), which makes the estimation more accurate. Moreover, embedded 3-D coding of video, which is discussed in Chapter 4, is possible with (t + 2-D) scheme.

The representation of temporal changes is more useful in the transform domain. After the temporal decomposition, the transform coefficients give information about the mutual connection (correlation) of frames. The changes among frames occur when there is movement of objects or scene changes. In case of slow motion, the frames can be regarded as still pictures and are highly correlated. To de-correlate the signal, temporal wavelet decomposition (without motion compensation) can be used. If there are objects with fast translational motion, new frequency components are introduced. The significant change of temporal frequency causes much of the signal energy to accumulate in the temporal direction. If the energy is so wide spread, it becomes difficult to benefit the "*energy compaction*" property of the wavelet transform. The "*energy compaction*" means the energy concentration in the lowest temporal frequency band. The higher the number of moving pixels means the higher the amount of energy in the high frequency components. In order to compensate the distribution of energy over high frequency components, a "*motion compensated temporal decomposition*" (MCTD) technique is employed.

In Chapter 2, we mentioned about the characteristics of Human Visual System and the type of redundancies in multi-dimensional signals. In video signals, spatial-temporal redundancy exists due to existence of high frequency components. It is crucial to localize these frequency components to achieve compression in video signals. The following sections explain how 3-D wavelet decomposition eliminates redundancy and becomes a useful tool for video compression.

3.2 Temporal Wavelet Decomposition

The temporal decomposition for 3-D WT algorithm is based on the "*group of frames*" (GOF) concept. Figure 3.1 shows a GOF structure with 8 frames. The temporal decomposition is the 1-D wavelet transformation of an input signal along the temporal axis. The input signal consists of pixel values that are collected from the same spatial position in each frame in the GOF. The spatial position at the "*k*"th frame is determined by its row, *r*, and column, *c*, indices and denoted by $(r,\ c)_k$. The 1-D input signal consists of 8 samples (pixel values) collected from respective positions: $(r,\ c)_1$ to $(r,\ c)_8$. Assuming each frame has size of *N* rows (height) and *M* columns (width) in pixels; *N* × *M* number of input signals are formed, and *N* × *M* number of 1-D WT are performed to accomplish the temporal decomposition.

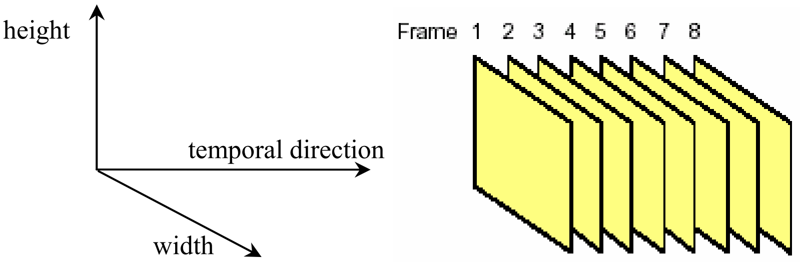


Figure 3.1 GOF Structure

The decomposition is a recursive procedure. A one-level decomposition of GOF structure results in one-half the GOF (½ GOF) of temporal high frequency bands and ½ GOF of temporal low frequency bands, provided that two-tap analysis filters are used in the decomposition. Such decomposition of the GOF structure is illustrated in Figure 3.2. The sequence is decomposed into temporal frequency bands by recursive decomposition of the low temporal subbands. The solid lines indicate the achieved temporal low frequency frames, and the dashed lines indicate the achieved temporal high frequency frames in the same decomposition. Better frequency localization is achieved at deeper levels. The filter type used in the decomposition has effects on the success of 1-D WT application. Although it is commonly accepted that better analysis is achieved when longer filters are used to de-correlate longer sequence of frames, inherent difficulties are faced in such schemes. Forming a GOF with a large number of frames causes buffering problems and processing delays. In this thesis, eight frames are allocated to be decomposed by two-tap Haar wavelet filters in order to solve the problem.

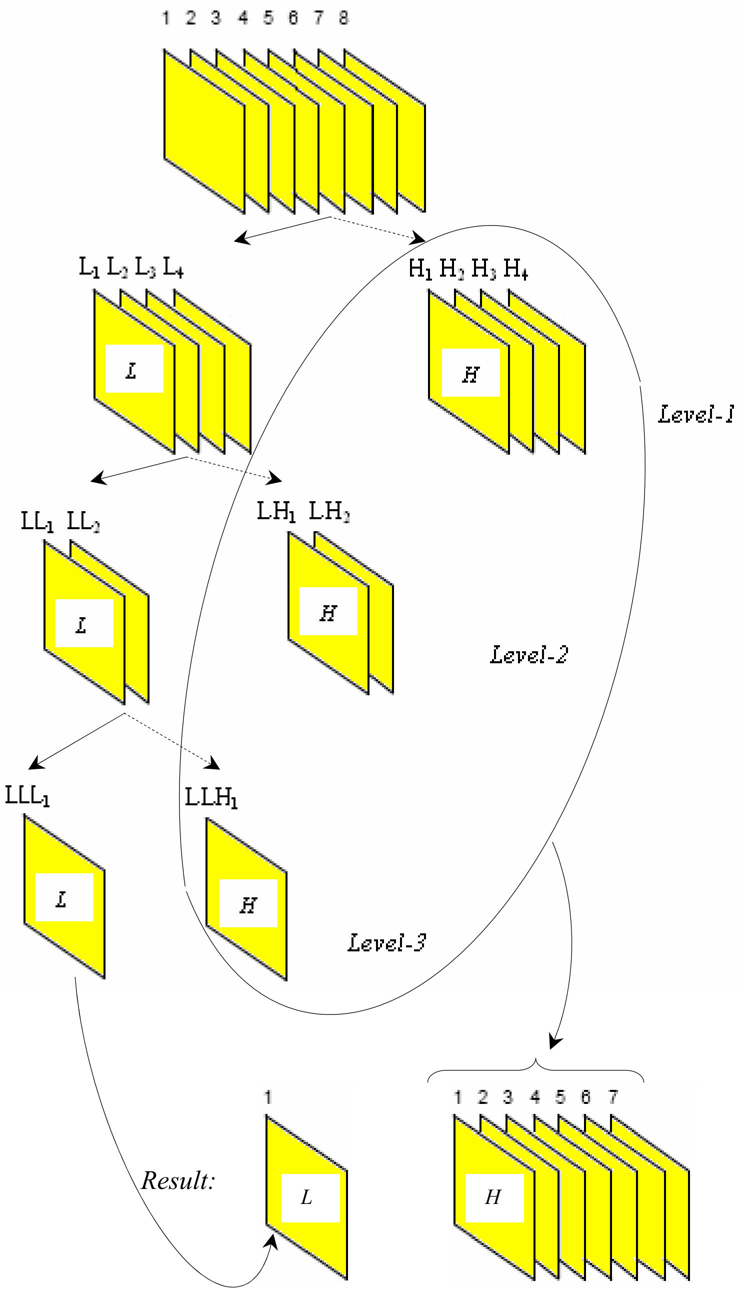


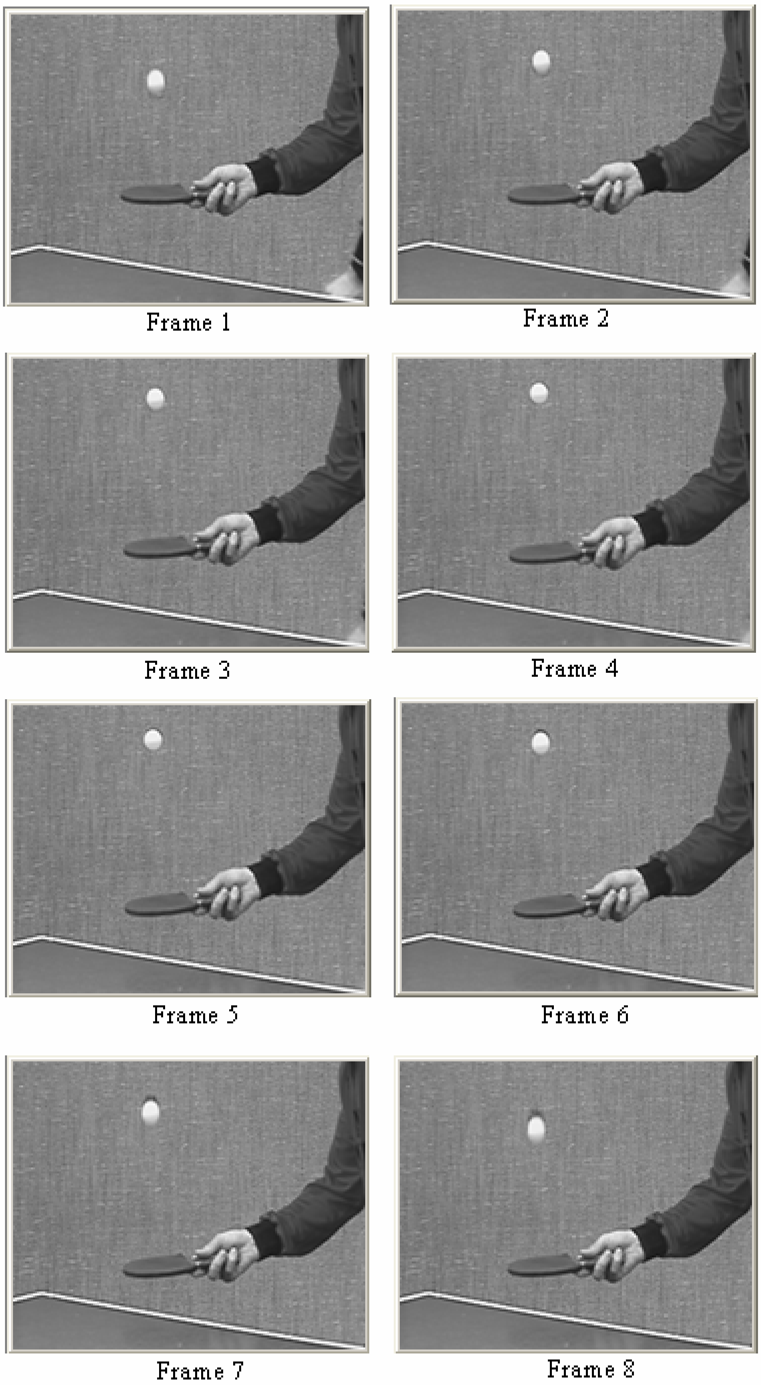
Figure 3.2 3-Level Temporal Wavelet Decomposition of GOF

The realization of temporal subbands is achieved by testing the "*Table Tennis*" video. The video has QCIF resolution (144×176 pixels in vertical and horizontal directions, respectively). The pixels are represented in YUV color space. The luminance (Y) component in frames is shown in Figure 3.3. The pixels have unsigned 8-bit integer gray scale value (ranging from 0 to 255). The chrominance components (U & V) are ignored in all processes and only the luminance component is used for decompositions. The 3-Level decomposition is performed using Haar analysis filters. The low pass (h_{haar}) and high pass (g_{haar}) filter coefficients are given below:

$h_{haar} = \sqrt{2} \times [0.5 \quad 0.5]$,
$g_{haar} = \sqrt{2} \times [-0.5 \quad 0.5]$.

The resulting temporal subbands are shown in Figures 3.4, 3.5, and 3.6. Figure 3.4 shows two subbands, low (L1-L4) and high (H1-H4), generated at the first level. Each low subband frame is (approximately) the average of two original frames. Therefore, motion of the ball and the arms seem blurred. In Figure 3.5, with two more temporal low (LL1-LL2) and high subbands (LH1-LH2) better frequency localization is obtained. Since average of the averaged motion is calculated at the second level, the blur in the low subband frames increases. The motion in the original video is entirely represented by a single subband (LLL1) frame at the third level in Figure 3.6. The resultant subband frames after the three-level decomposition are as follows: 1 LLL and 1 LLH at level-3, 2 LH at level-2, and 4 H at level-1. Other than the LLL frame, the remaining 7 frames show the details in the motion. They can be regarded as the difference information in successive frames. The coefficients in these bands are relatively small compared to that of the low bands. In every successive level, the coefficients in the low subbands become larger by a factor of √2. To display the low subband frames, the coefficients are scaled by factors of 1/√2, 1/2, and 1/2√2, respectively for each level. This procedure can be regarded as normalizing the coefficients to 255.

The energy of the signal is proportional to the magnitude of coefficients in the transform domain. In order to analyze the energy distribution, we took into



Frame 1

Frame 2

Frame 3

Frame 4

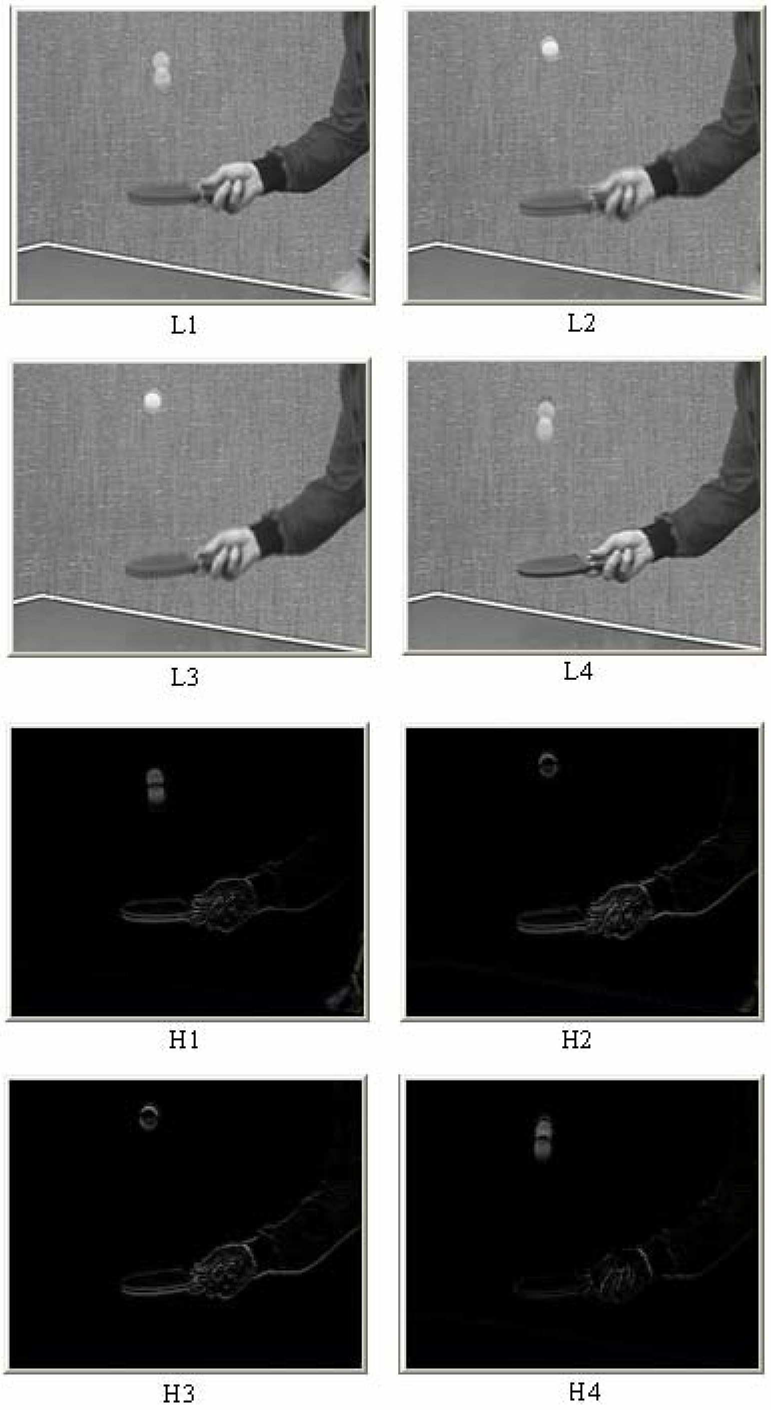
Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.3 Original “*Table Tennis*” Video Signal



L1

L2

L3

L4

H1

H2

H3

H4

Figure 3.4 Temporal Only Decomposition (L-1)

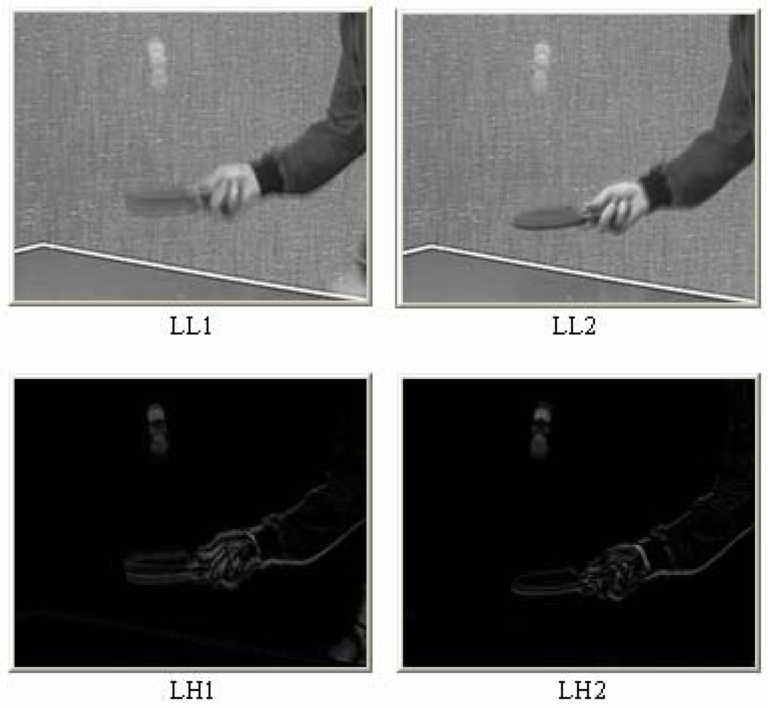


Figure 3.5 Temporal Only Decomposition (L-2)

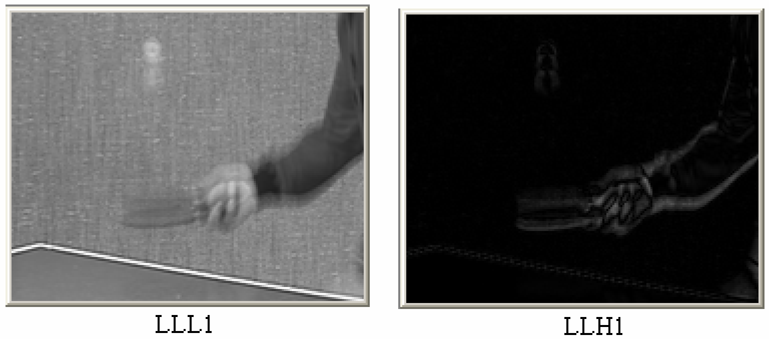


Figure 3.6 Temporal Only Decomposition (L-3)

account the distribution of coefficients at each level. We utilize the entropy of coefficients to comment as one step ahead of histogram drawings. The entropy is calculated according to the following equation:

$$H = -\sum_{i=1}^{2} p_i \log_2 p_i \qquad (3.1)$$

where p_i denotes the probability of a coefficient to be one of two types. According to Equation 3.1, the system is considered to yield 2 types of messages about coefficients: A coefficient is either significant or insignificant depending on its magnitude. Suppose that p_1 denotes the probability of the occurrence of insignificant coefficients and p_2 denotes the probability of the occurrence of significant coefficients. If $p_1 = 1$ and $p_2 = 0$, the entropy H is 0, that is it has the minimum possible value. In this case, the message tells the information all about the insignificant coefficients. At the other extreme, if $p_1 = p_2 = ½$, the entropy is 1 that is it has the maximum possible value. In this case, two message possibilities are equally likely. Receiving the message adds new information [37].

A threshold value is set to determine the type of a coefficient. Our observations on many video signals revealed that coefficients at high subbands are close to zero in magnitude. On the other hand, the maximum coefficient is found to be below √2 × 255 at level-1, 2 × 255 at level-2, and 2√2 × 255 at level-3. Therefore, we set a threshold value of 25.0 based on our experiences. It constitutes 6.9%, 3.4%, and 2.9% of the maximum available coefficient in respective levels. These ratios are illustrated via diagrams in Figure 3.7.

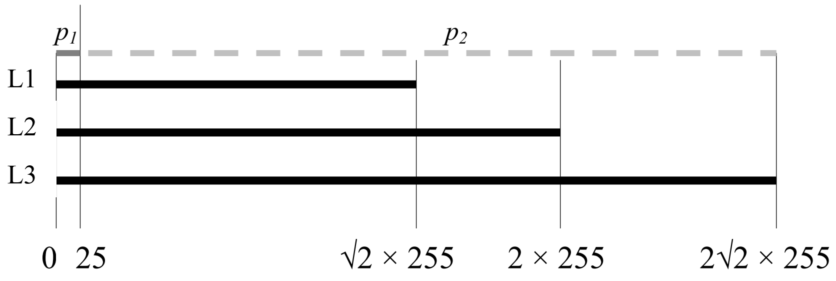


Figure 3.7 Setting Threshold for Coefficients (t only)

In Figure 3.7 the coefficients below 25.0 in magnitude are treated as insignificant. Their probability of occurrence is denoted by p_1. The rest of the coefficients are significant and their probability is denoted by p_2. It is aimed to maximize the accumulation of coefficients around zero so that most of the signal energy is shared by significant coefficients, the number of which is less. Equation 3.1 tells us information about the accumulation around zero: The higher the accumulation is the lower the entropy.

Entropy calculations for "*Table Tennis*" video signal are summarized in Table 3.1. The entropy *H* is found to be 0.999, almost unity at level-1. It means that receiving coefficients in the interval [0, 25.0) is equally likely with receiving coefficients in the interval [25.0, √2 × 255]. This worst case tells us that half of the samples in time domain are shifted to a region with less energy in transform domain, and most of the signal energy is concentrated on the other half. This can be proven by the increases in magnitudes of coefficients in the low subbands. At level-2, the entropy is 0.846. The decrease from 0.999 is actually an expected result for the second level. The number of high subbands increases and most of the significant coefficients are compacted only in LL1 and LL2 bands. The best accumulation of coefficients near zero, or energy concentration in the lowest frequency band (LLL1) is achieved at level-3, where *H* is 0.635. This generates a slowly moving average video signal with a lower resolution.

Table 3.1 Entropy of Coefficients (t only)

level-1	level-2	level-3
0.999	0.846	0.635

3.3 Temporal Wavelet Decomposition with Motion Compensation

In Section 3.2, we observed the effect of decomposition levels on the distribution of signal energy. Table 3.1 says that as we increase the decomposition level, more coefficients with less energy appear in temporal high subbands, and most of the signal energy accumulates in temporal low subbands. This section investigates whether the "*motion compensated temporal decomposition*" (MCTD) can contribute such effect of the multi level decomposition introduced in the previous section.

The basic difference of MC technique is the determination of input signal. The input signal consists of pixel values collected at positions $(r, c)_k$, where r, c, and k are indices of row, column and frame, respectively in the GOF structure. The index values can not be determined straightaway but a motion estimation methodology has to be used to calculate them. Once they are calculated, it is possible to trace an object motion as illustrated in Figure 3.8. The MC temporal analysis requires the 1-D WT to be performed along this trajectory. The number of computations of 1-D WT is less than $N \times M$ because there can be as many motion trajectories as the number of pixels at most.

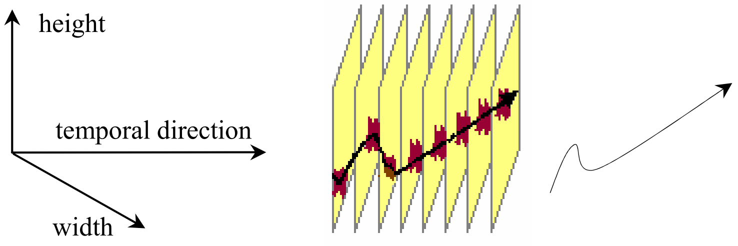


Figure 3.8 Motion Trajectory

3.3.1 Motion Estimation (ME)

The motion estimation methodology is based on the *block motion model* [6]. The model assumes that frames are composed of moving blocks. The moving blocks with reference to the previous frame are estimated in the current frame. The motion estimation utilizes displacement information of the moving blocks. To do so, a specific *block-matching* method is used. The *block-matching* searches for the location of the best-matching block of a fixed size in the current frame based on a distance criterion. The search is restricted to a smaller *search area* centered at the position of the target block in the referenced frame as shown in Figure 3.9. This limits the *maximum displacement* on how far objects move between frames. The maximum displacement (*dx, dy*) is specified as the maximum number of pixels in the horizontal and vertical directions that a candidate (matching) block is from the position of the target block. For example, if a pixel with reference at position (r, c) is found at position (r', c') in the current frame, the displacement vector is calculated by the equation $(dx, dy) = (c'-c, r'-r)$.

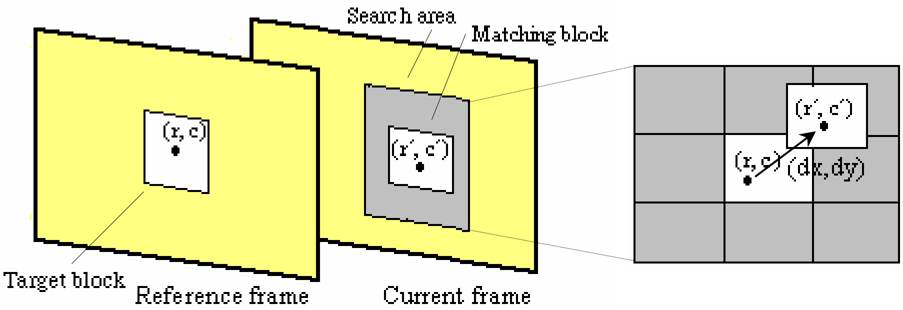


Figure 3.9 Block-Matching Motion Estimation

3.3.2 Motion Compensated (MC) Filtering with Haar Filter

The "*motion compensated temporal filtering*" (MCTF) combined with block matching starts with grouping frames in GOF as shown in Figure 3.10. The *block-matching* motion estimation is performed between the first (referenced) and the second (current) frames of each pair (P1, P2, P3, P4). Once a unique motion vector (MV) for a block and correspondingly for each pixel in the block is determined, the temporal filtering is performed on the motion trajectory.

The motion estimation suffers from the existence and uniqueness problems. Because of the occlusions and overlapping blocks, finding a unique MV for a pixel may not be possible. Occlusion refers to the *covering* / *uncovering* of a surface due to 3-D rotation and/or translation of an object which occupies only part of the field of view [6]. In Figure 3.11, the covered and uncovered background concepts are

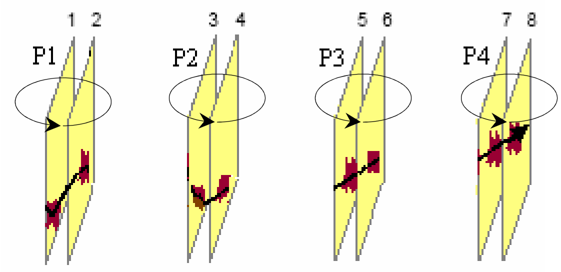


Figure 3.10 Frame Pairs in GOF

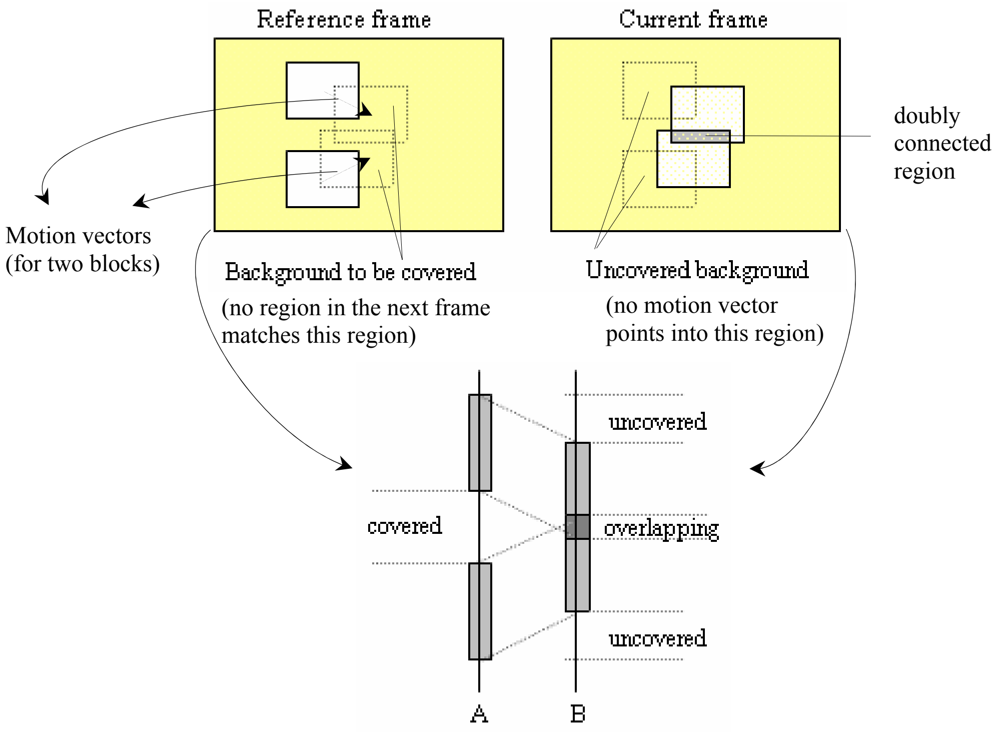


Figure 3.11 Covered/Uncovered Background Problems

illustrated for the two-dimensional translational block motion model. *A* and *B* denote two successive original image frames. The blocks in solid lines are translated in the direction of motion vector from referenced (*A*) to current (*B*) frames. The dotted blocks in *A* indicate the background to be *covered* in *B*. It is not possible to find a correspondence for pixels inside these blocks in *B*. The dotted blocks in B indicate the background *uncovered* by motion of the block. There is no correspondence for these pixels in *A*. Because of the correspondence problem, only a subset of the referenced and current frames is used for the motion compensation operation.

If a one-to-one correspondence is present, pixels are uniquely "*connected*"; hereafter we call them "*connected*" pixels. In all other cases, the pixels are either "*not connected*" or "*double connected*". The "*connected*" pixels enter to (temporal) filtering operation without any pre-processing. If we only encode and transmit MC filtering outputs, we have the problem that not-filtered ("*not connected*" & "*double connected*") pixels are left out, which has to be solved.

The "*connected*", "*double connected*", and "*not connected*" pixels need special treatments. Ohm [14] proposed such a treatment, called "*geometrical analysis procedure*", (GAP), to accomplish the MCTD. The GAP is a four-step procedure and illustrated in Figure 3.12. Only the motion in the vertical direction is shown, but the extension to the horizontal dimension is straightforward. In Figure 3.12, three adjacent blocks are regarded, with typical motion values of 1, 2, and 1 pixels, respectively. Each step identifies the status of pixels and can be summarized as follows: (1) All pixels with reference exclusively inside the same block are found; (2) All pixels with reference in the adjacent block lying in the direction of motion are found; (3) All pixels with "*double connection*" are found; (4) All remaining pixels are classified as "*not connected*". The first two steps of the GAP lead to the coordinate positions, at which the motion-compensated filtering is applied. The third and fourth steps identify the remaining *unconnected* pixels. Specifically, the pixels with reference in *A* are labeled as "*not connected*" and others in *B*, as "*double connected*".

All pixels with one of three statuses are processed separately as illustrated in Figure 3.13, where MC decomposition of a two-sample signal is given. Each sample is positioned either at *(r, c)* in *A* or at *(r′, c′)* in *B*. The *L* and *H* images/frames represent the resulting lowpass and highpass components, respectively such that the resultant wavelet coefficients are positioned either at *(r, c)* in image *H* or at *(r′, c′)* in image *L*.

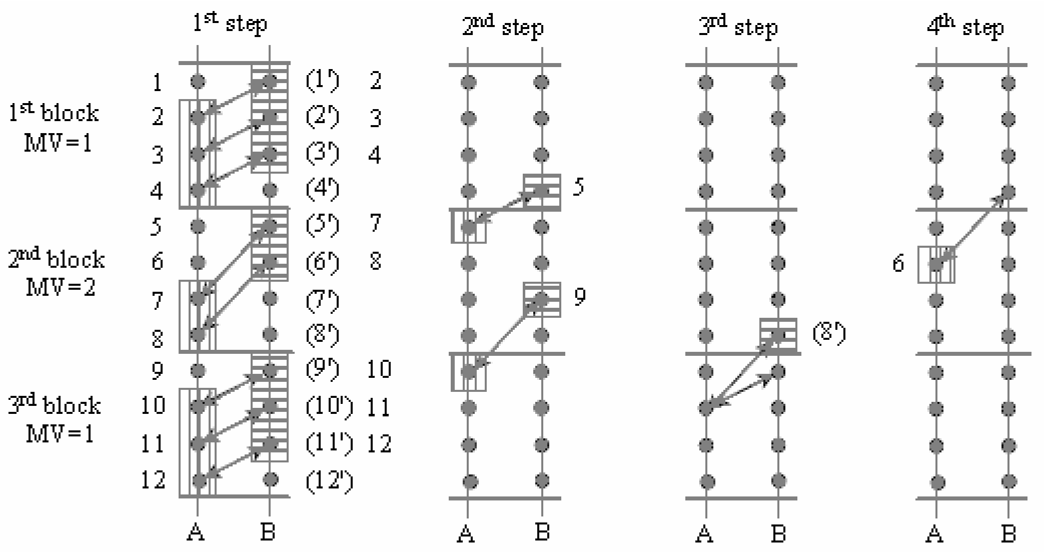
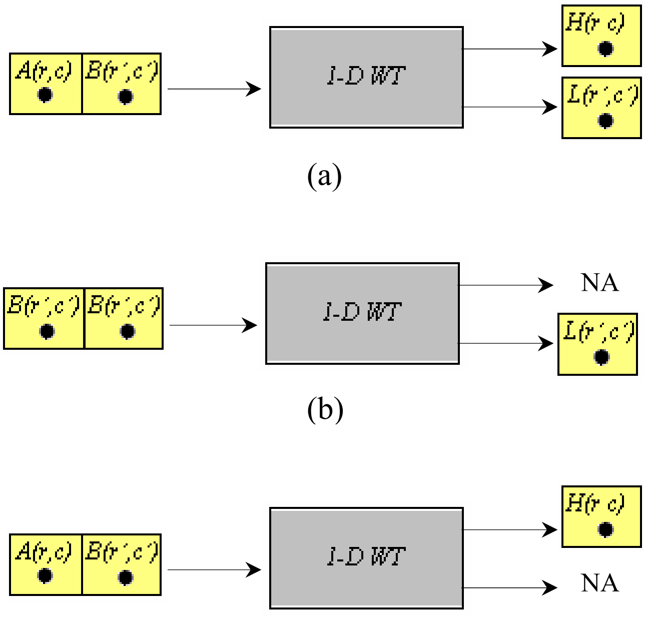


Figure 3.12 Geometrical Analysis Procedure (GAP)



(a)

(b)

(c)

Figure 3.13 MCTD of a 2-pixel Signal with Haar Filter:
(a) Uniquely connected, (1st and 2nd step of GAP);
(b) Double connected, (3rd step of GAP);
(c) Not connected, (4th step of GAP)

In Figure 3.13-a, the positions *(r, c)* and *(r′, c′)* are connected by a unique motion vector. The wavelet transformation on the motion vector yields one approximate and one detail coefficients. The former is placed at *(r′, c′)* in image *L*, and the latter is placed at *(r, c)* in image *H*. In Figure 3.13-b, an *unconnected* (or *double connected*) pixel at *(r′, c′)* in frame B is processed. The pixel is replicated to form a two-tap temporal signal. The resulting approximate coefficient is placed at the same position in image *L*. In Figure 3.13-c, an empty location in image *H* is filled. The samples of the input signal are related to each other by an artificial MV estimated for the block in which the pixel at *(r, c)* exists.

3.3.3 Multi-Level MC Decomposition

Multi-level MC WT is illustrated in Figure 3.14. The arrow "*Next decomposition*" shows the recursive operation on low subbands. The difference from Figure 3.2 is the insertion of GAP to achieve MC filtering. To calculate the coefficients at level-2, motion estimation is performed between the coefficients in *L* images (the low subband frames: L1-L4). LL1 and LL2 subband frames are processed according to the GAP to perform the third level MC decomposition.

The realization of the temporal subbands is achieved by testing the "*Table Tennis*" video as in Section 3.2. Three-level MC Haar decomposition is obtained. The subbands generated at level-1, level-2, and level-3 are displayed in Figures 3.15, 3.16, and 3.17, respectively. Square blocks with dimensions of 8 by 8 pixels are used in the block matching algorithm. Three times the block dimensions (24×24) are used as dimensions for the search window.

A visual improvement in temporal low subband frames can be perceived in Figures 3.15, 3.16, and 3.17 since the motion blur is eliminated. The difference information implied by high subband frames is lack of motion, especially around the ball, the hand, and the arm. The change in the entropy of MC coefficients also reflects this improvement. The MC entropies in Table 3.2 are less than those in Table 3.1 at all levels. Because of the insufficient three digit accuracy, the difference at level-1 is not perceivable.

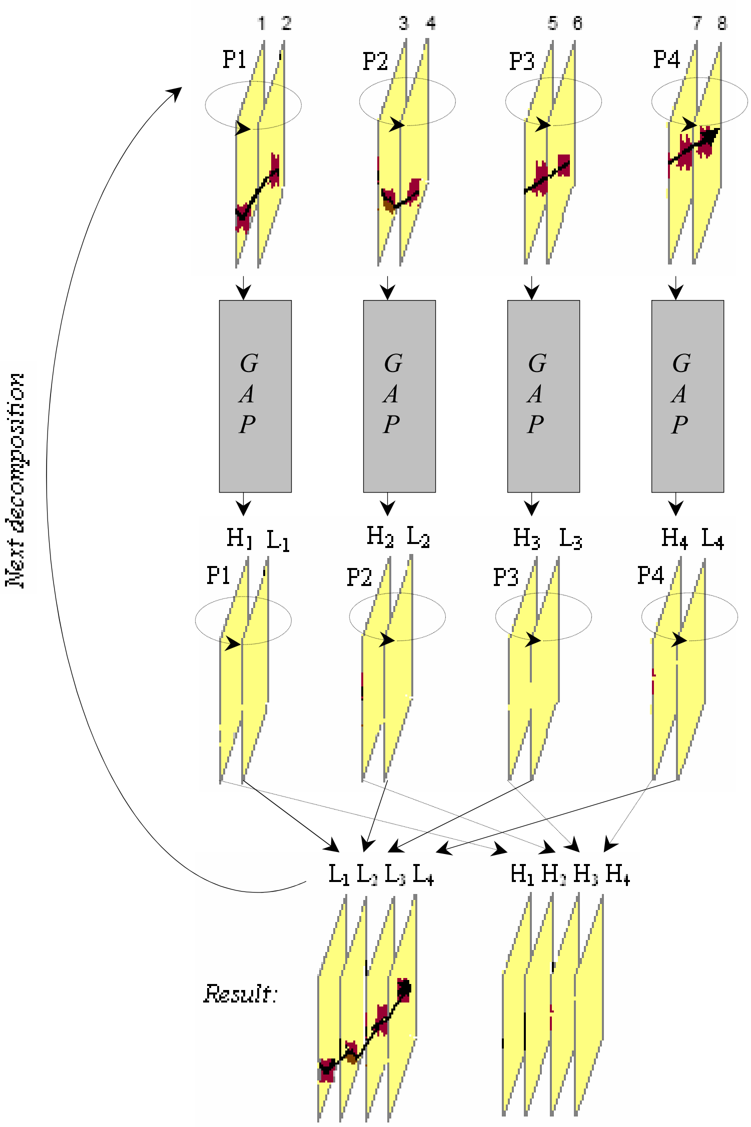
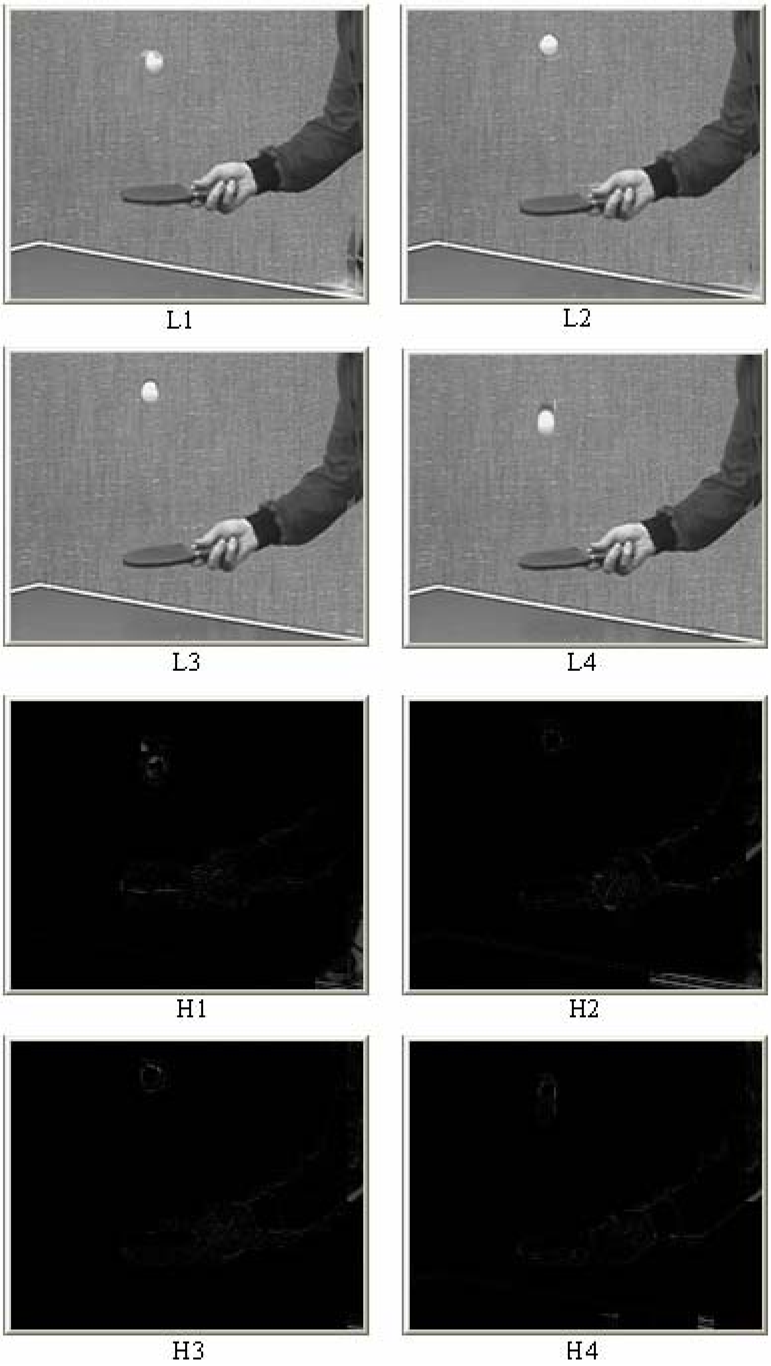


Figure 3.14 MCTD of the GOF with 2-tap QMF



L1

L2

L3

L4

H1

H2

H3

H4

Figure 3.15 MC Temporal Only Decomposition (L-1)

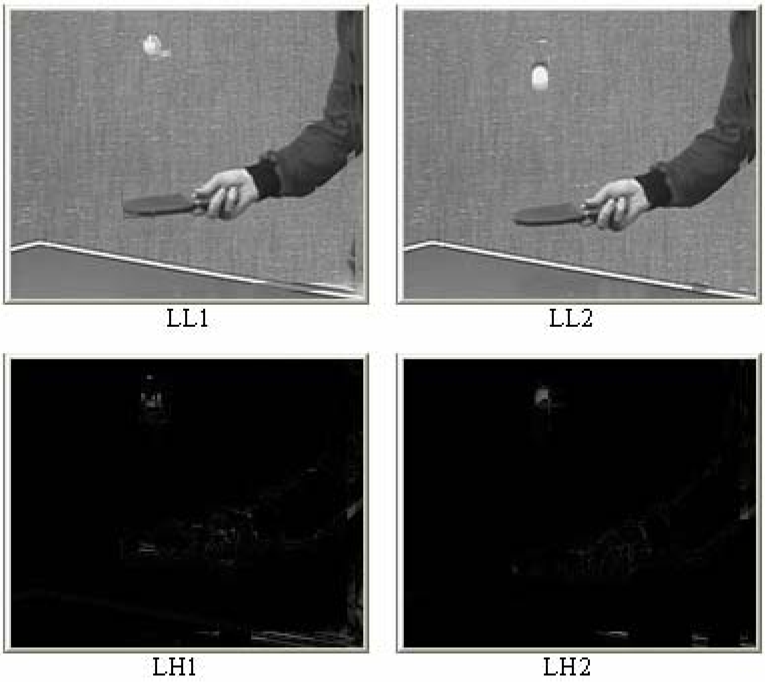


Figure 3.16 MC Temporal Only Decomposition (t only, L-2)

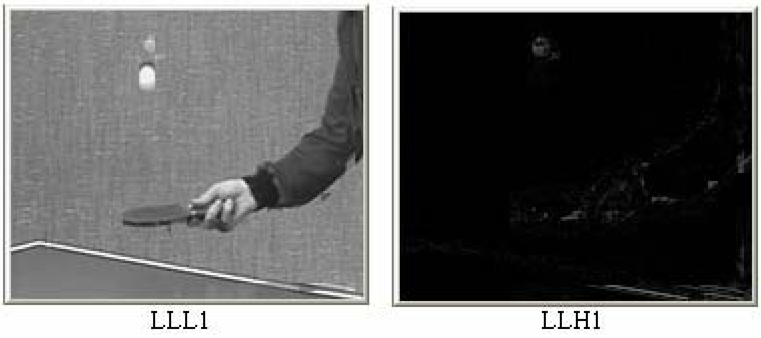


Figure 3.17 MC Temporal Only Decomposition (t only, L-3)

Table 3.2 Entropy of MC Coefficients (t only)

level-1	level-2	level-3
0.999	0.825	0.584

The insertion of GAP into Figure 3.14 increases the algorithmic complexity. The complexity of block matching is O(n^4), and the complexity of wavelet decomposition is O(n^2). The overall complexity of the system is determined by the motion estimation. Correspondingly, the increase in the number of operations (multiplications and additions) causes processing delays. An open source C++ function from the Intel® Computer Vision Library [36] is used to calculate the MVs for each block. The MVs are given in integer accurate pixels. The time estimation for a level decomposition results in more than 5 seconds in "Debug" mode over a Pentium® 4 processor with a speed of 1.6 GHz. It is hard to estimate real times because the Operating System controls multiple tasks. An appropriate way to estimate processing times is to select the least time delay as reference and to estimate all relative times in seconds. Table 3.3 gives the relative times needed to complete decompositions for three levels.

Table 3.3 Relative Decomposition Times (t only)

decomposition	level-1	level-2	level-3
No MC	1	1.45	1.73
MC	1.09	1.64	1.91

Besides processing delays, buffering problems occur since newly generated MVs require being stored at each level. The number of MVs calculated in the first two steps of the GAP is given in Table 3.4. They constitute the pixels illustrated in Figure 3.13-a. Initially, there are 8×144×176 = 202752 pixels (samples) in the video signal. Since the ME is performed between a pair of frames, 202752 / 2 = 101376 pixels are connected at most in the best case. According to Table 3.4, 96.6% of the pixels have one-to-one correspondence from referenced to current frames. Only the 3.4% are unconnected, and the 3rd and 4th steps of GAP are applied to them.

At level-1, motion is estimated between four subband frames (L1-L4). The ratio of the connected pixels is 94.6%. It is not as successful as the estimation in the initial step. This is caused by the additional motion gap between the successive subband frames at level-2. The additional motion gap causes inconvenience to find out the motion within the search window. At level-2, the number of motion hits constitutes only 90.9% of the entire frame. The result is worse due to the same reason as for the previous level.

Table 3.4 Number of Connected Pixels (t only)

Frame 1-8	L1-L4	LL1-LL2
97895	47930	23030

The percentage of *unconnected* pixels is summarized in Table 3.5 as alternative to Table 3.4. The previous studies [14] have comments on the success of the MCTD based on *unconnected* pixels, such as ME is useful if 3-5% of the samples (pixels or coefficients) are represented by *unconnected* ones.

Table 3.5 Average Percentage of Unconnected Pixels (t only)

Frame 1-8	L1-L4	LL1-LL2
3.4%	5.4%	9.1%

3.4 Spatial - Temporal Wavelet Decomposition

The spatial-temporal decomposition of video signal implements the (t + 2-D) scheme. The temporal decomposition, "t", is followed by the spatial decomposition, "2-D". The former is explained in detail in Section 3.2. The latter is identical to the 2-D WT of a frame. Its analytical explanation is also given in Chapter 2 while examining the "*Lena*" test image. Only the stand-alone applications of "t" or "2-D" decompositions are illustrated in the referred sections. Their integration to the (t + 2-D) scheme extends the dimension of decomposition to three. The GOF structure is regarded as volumetric data with dimensions *t-time*, *y-*

height, and *x-width* as shown in Figure 3.18, where a 3-level hierarchical 3-D decomposition of the volume is illustrated.

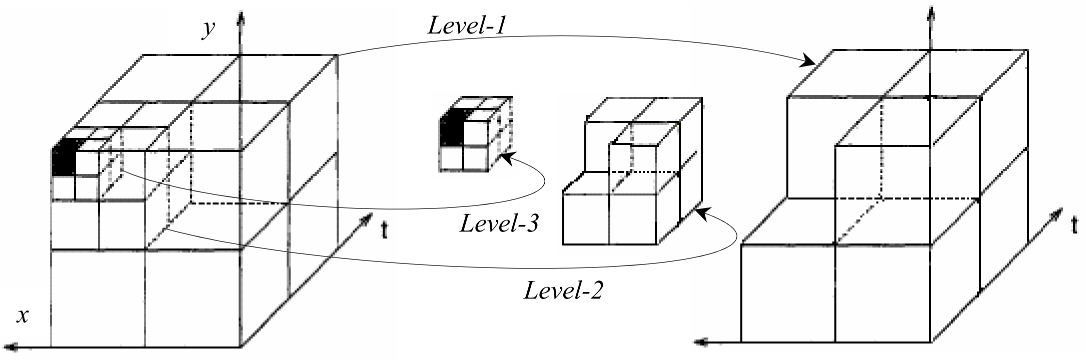


Figure 3.18 3-Level Spatial-Temporal Wavelet Decomposition of GOF

The decomposition starts with 1-level temporal decomposition of the GOF as shown in Figure 3.2. Then, the low (L1, L2, L3, and L4) and high (H1, H2, H3, and H4) subbands are spatially decomposed. This entire process is described as one-level spatial-temporal decomposition and creates 8 different subbands: (1) LLL, (2) LLH, (3) LHL, (4) LHH, (5) HLL, (6) HLH, (7) HHL, and (8) HHH. The number of each subband frame is four. They constitute sub-cubical volumes shown by the arrow "*Level-1*" in Figure 3.18. The sizes of subband frames are (approximately) one-eighth the volume of the GOF structure. They exhibit different characteristics of the original video signal: The LLL frames represent the slowly moving *average signal* and the remaining seven frames are the *detail signals*. The details are directionally sensitive: for example, the HLL frames emphasize the quickly changing average signal, the LLH frames emphasize slowly changing horizontal image features, the LHL frames emphasize slowly changing vertical image features, and the HHH frames emphasize quickly changing diagonal features. Further decomposition of the LLL subbands produces an additional 8 different subbands at the second level too. They constitute sub-cubical volumes shown by the arrow "*Level-2*" in Figure 3.18. More high frequency subbands are obtained with the recursive operation. The depth of the decomposition depends on both the initial number of frames and their spatial sizes.

Differing from the previous observations in temporal and spatial only decompositions, Figure 3.18 shows 8 subbands after one level spatial-temporal decomposition. It provides a better frequency localization. Since more details corresponding to high frequency components are identified, we can better resolve the energy distribution of the video signal. The coefficients in the high subbands have relatively small magnitudes. More coefficients are expected to accumulate near zero.

The analysis of the coefficients is performed by testing the "*Table Tennis*" video as in the previous sections. Both normal and MC temporal decomposition techniques are used. In all test cases, temporal decompositions are performed using Haar filters. Either Haar or Daubechies-9/7 analysis filters are employed for spatial decompositions. As opposed to Haar filters, Daubechies-9/7 filters are bi-orthogonal and the coefficients are irrational excluding the √2 factor. The low pass ($h_{db\text{-}9/7}$) and high pass ($g_{db\text{-}9/7}$) filter coefficients given below are truncated decimal expansions of irrational numbers:

$h_{db\text{-}9/7} = \sqrt{2} \times [0.026\ -0.016\ -0.078\ 0.266\ 0.602\ 0.266\ -0.078\ -0.016\ 0.026]$,
$g_{db\text{-}9/7} = \sqrt{2} \times [-0.045\ 0.028\ 0.295\ -0.557\ 0.295\ 0.028\ -0.045]$.

The test cases can be summarized as follows: (1) 3-D Haar decomposition, (2) MC 3-D Haar decomposition, (3) 3-D Haar & Daubechies-9/7 decomposition, and (4) MC 3-D Haar & Daubechies-9/7 decomposition. The experiments help us observe the differences in the distribution of coefficients. The entropy results are compared with the results of temporal only decompositions. The effects of using different wavelet filters are analyzed. The costs of using longer tap filters and the MC technique are estimated.

Eight frames let us perform up to three levels temporal decomposition, and correspondingly, up to three levels (t + 2-D) decomposition. There is no restriction with the spatial domain since frames' sizes are large enough (144 × 176). Because the WT of a 1-D signal requires filtering (convolution) with the wavelet filter, the size of the resultant signal in the transform domain changes. The convolution extends size of the input signal. Assuming the input signal has length n and the filter has length L, the length of the output signal becomes $n + L - 1$ after the filtering. The decimation operation following the filtering drops every other sample (coefficient), yielding a signal with approximately one half the length of the convolved signal. The convolution with Haar filter and dropping every other sample creates temporal low and high band signals with size of half of the input video signal. The number of subband frames is 4 at level-1, 2 at level-2, and 1 at level-3, as shown in Figure 3.2. Spatial dimensions are (72 × 88), (36 × 44), and (18 × 22) at levels 1, 2 and 3, respectively. Therefore, in test cases (1) and (2), exactly one eighth the volume size of the previous low band is created at all levels. It is not so for case (3) and (4), in which Daubechies-9/7 analysis filters are used. Since $h_{db-9/7}$ and $g_{db-9/7}$ have different lengths, low and high subband frames result in different sizes. In Table 3.6, we only summarized the size of the slowly moving average signal (LLL subband frames) and quickly changing detail signal (HHH subband frames) at all 3-levels. The width of the LLH or HLH subband frames is equal to the width of the LLL frames, and the height of the LLH or HLH subband frames is equal to the height of the HHH frames. The notation, $a \times (b \times c)$ in Table 3.6 denotes the number of subband frames for a, the width of subbands for b, and the height of subbands for c.

Table 3.6 Size of Subband Frames (t + 2-D, test case: 3 & 4)

subbands	level-1	level-2	level-3
LLL	4 × (76 × 92)	2 × (42 × 50)	1 × (25 × 29)
HHH	4 × (75 × 91)	2 × (41 × 49)	1 × (24 × 28)

The spatial dimension of the GOF structure and the volumetric data after the (t + 2-D) decomposition is given by its *height* and *width*. The number of pixels in a frame of the original video signal is 144×176 = 25344. The first level decomposition produces a spatial dimension of 151×183, holding 27633 coefficients. That means a 9.0% increase in samples in the transform domain with respect to pixels of the original signal in the time domain. Next decomposition is performed by the coefficients of LLL subband frames (76 × 92) and produces 8217 coefficients at level-2. The ratio to the input signal is 17.5%. Finally, at level-3, the increase in the number of samples (coefficients) with respect to input signal (coefficients) is calculated as ((49 × 57) - (42 × 50)) / (42 × 50) = 33%. The deeper the decomposition level is, the higher the increase in ratio of samples in the transform domain to the input signal. The cubical volume in Figure 3.18 turns to a rectangular prism at every level but larger than the input cubic by the given amount of ratios. The reason for the increase is an increase in the ratio of Daubechies-9/7 filters length to the signal spatial-dimension in successive levels. The filter length in deeper levels gets more significance with respect to signal dimension. The result becomes tremendous when the dimension decreases so as to be comparable with the filter length. Using Daubechies-9/7 filters in the (t + 2-D) scheme, instead of Haar filters, would result in 8 temporal low subband frames and 7 temporal high subband frames at level-1. Getting so many coefficients is not desirable because it creates buffering problems and increases the processing times. We need to keep each coefficient produced in each level to reconstruct the original video signal.

In test cases (2) and (4), the block size and the search window are initialized to 8×8 and 24×24 pixels, respectively. Since the spatial dimension lessens at the deeper levels, we readjusted the block and search window dimensions. Iteratively, the dimensions are halved. The respective dimensions are 4×4 and 12×12 before decomposition to first level and 2×2 and 6×6 before decomposition to third level. The motion estimation is performed on the spatial-temporal low subband (LLL) frames. At the beginning there are 8 frames belonging to the original video. At level-1, there are 4 LLL frames, and at level-2, there are 2 LLL frames. We faced with restriction that at level-1 and level-2, we had to normalize coefficients in LLL subbands to the interval [0-255] and then to merge them to the nearest integer. The limitation is because of the Intel® OpenCV Library. The same C++ function used in MC temporal only decomposition is employed for the block matching algorithm. The function requires input frames with 8 bit pixel values. Such representation of LLL subband frames is obtained by normalizing the coefficients to 255. The scaling factor to normalize is $[(1/\sqrt{2})^3]^p$, where p is l ("*desired level*") – 1. The "*desired level*" is the level arrived after the MC decomposition.

The realization of temporal subbands is achieved by testing the "*Table Tennis*" video as in the previous sections. The maximum coefficient at (desired) level l is less than $[(\sqrt{2})^3]^l \times 255$. To estimate entropies, the threshold value is set to 25.0 based on our previous experiences. It constitutes 3.4%, 1.2%, and 0.4% of the maximum coefficients at respective levels. The ratios are illustrated in Figure 3.19. The subband frames generated in the first test case is shown in Figures 3.20, 3.21, and 3.22 for levels 1, 2, and 3, respectively. Similarly, the subband frames generated in cases (2), (3), and (4) are shown in Figures 3.23 - 3.31.

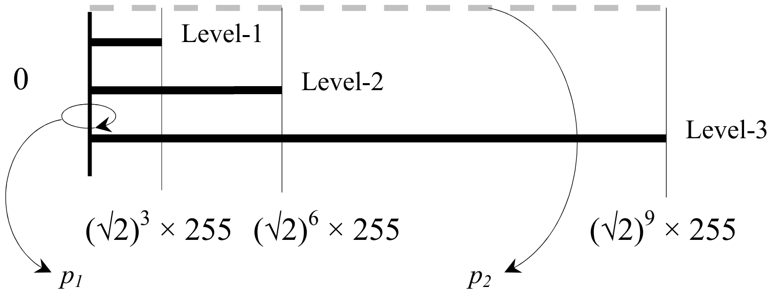

Level-1
0
Level-2
Level-3
$(\sqrt{2})^3 \times 255$
$(\sqrt{2})^6 \times 255$
$(\sqrt{2})^9 \times 255$
p_1
p_2

Figure 3.19 Setting Threshold for Coefficients (t + 2-D)

Figure 3.19 uses the same notation given in Figure 3.7. The probability of the occurrence of (insignificant) coefficients below the threshold is denoted by p_1, and the probability of the remaining coefficients are denoted by p_2. The scale of the range bars is not the same as in Figure 3.7; that is why the interval [0, 25) not distinguishable around zero. The same scale can be obtained by magnifying the range bars 10 times. The 3-D decomposition produces coefficients with higher magnitudes to accumulate in low subbands when compared to the coefficients after the temporal only decomposition. The energy concentration at low subbands is expected to be more. Consequently, more coefficients accumulate around zero.

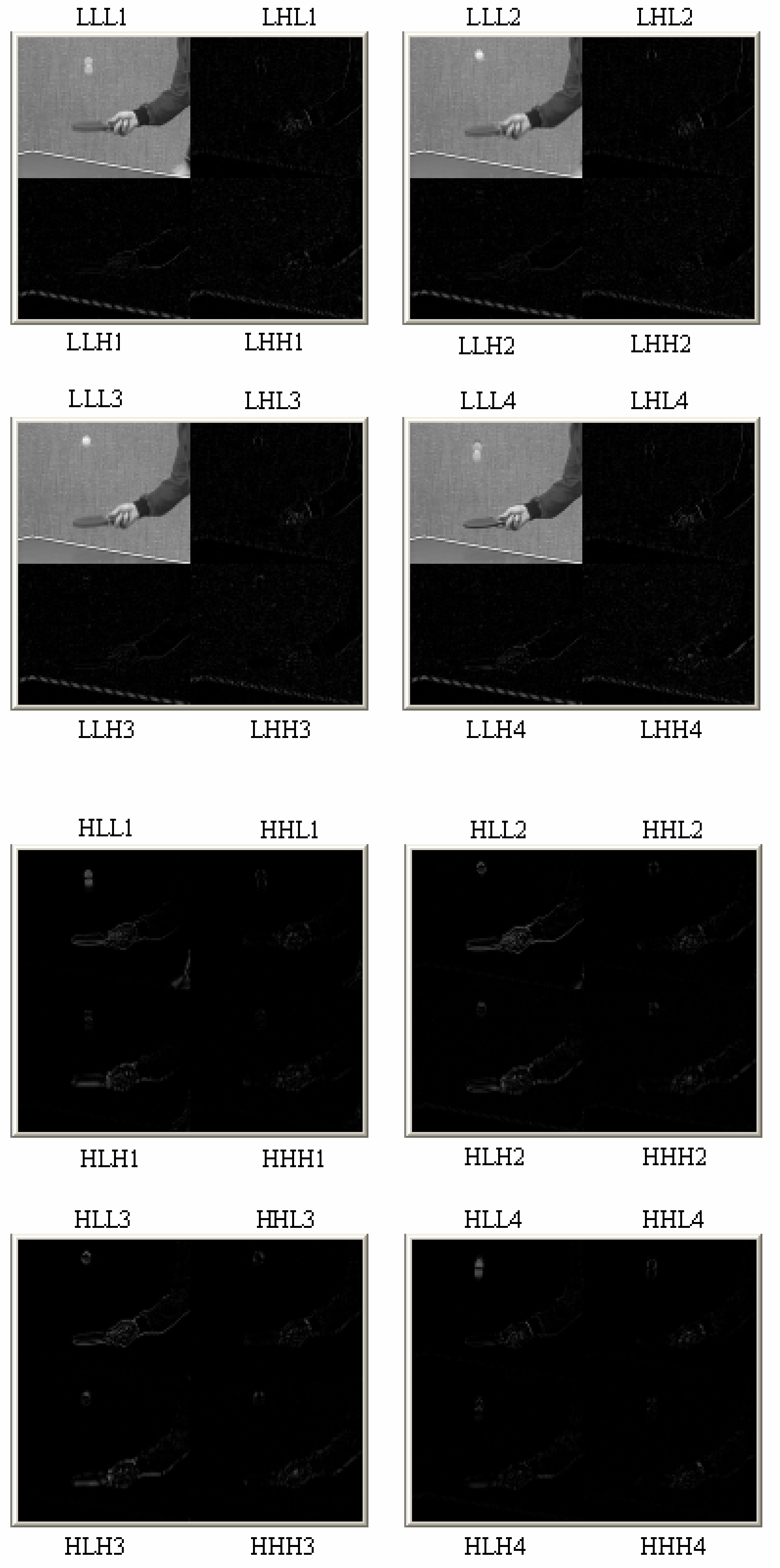


Figure 3.20 Spatial-Temporal Haar Decomposition (L-1)

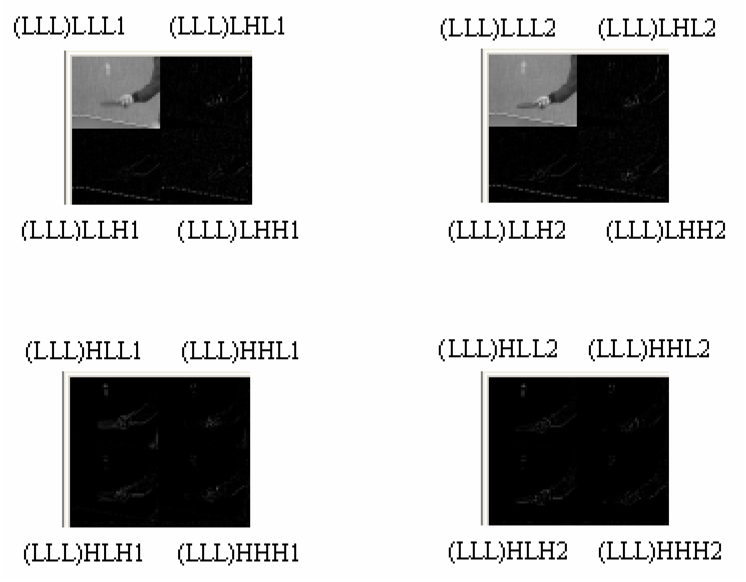


Figure 3.21 Spatial-Temporal Haar Decomposition (L-2)

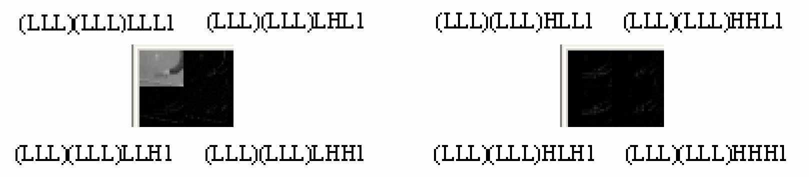


Figure 3.22 Spatial-Temporal Haar Decomposition (L-3)

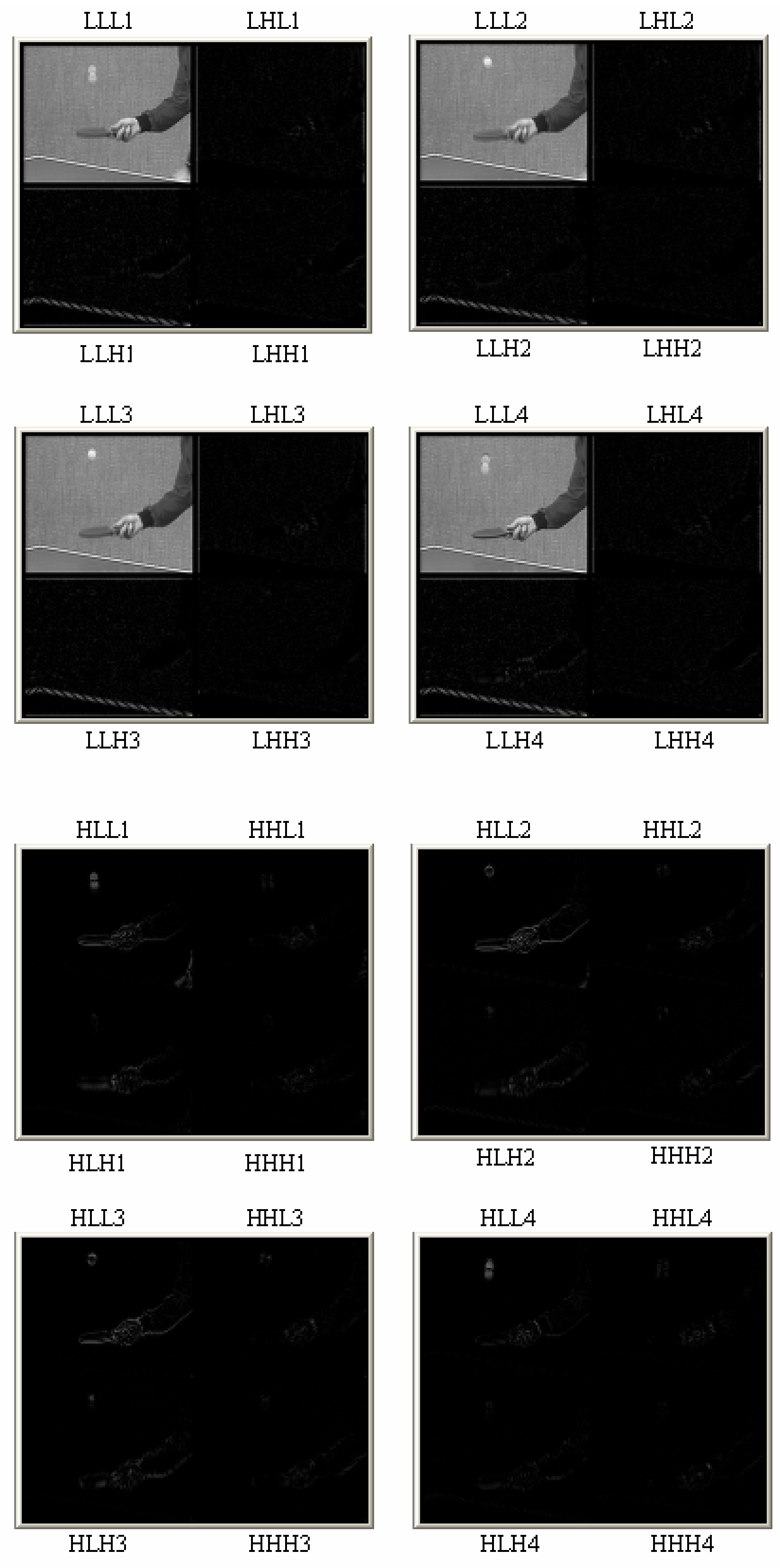


Figure 3.23 Spatial-Temporal Haar & Db-9/7 Decomposition (L-1)

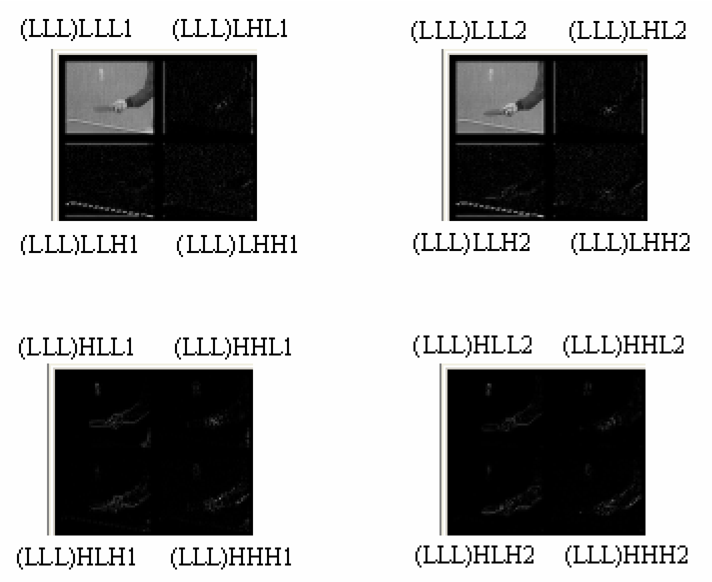


Figure 3.24 Spatial-Temporal Haar & Db-9/7 Decomposition (L-2)

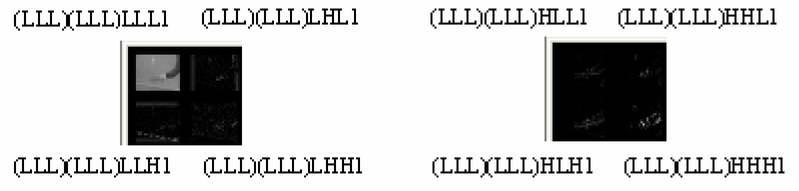


Figure 3.25 Spatial-Temporal Haar & Db-9/7 Decomposition (L-3)

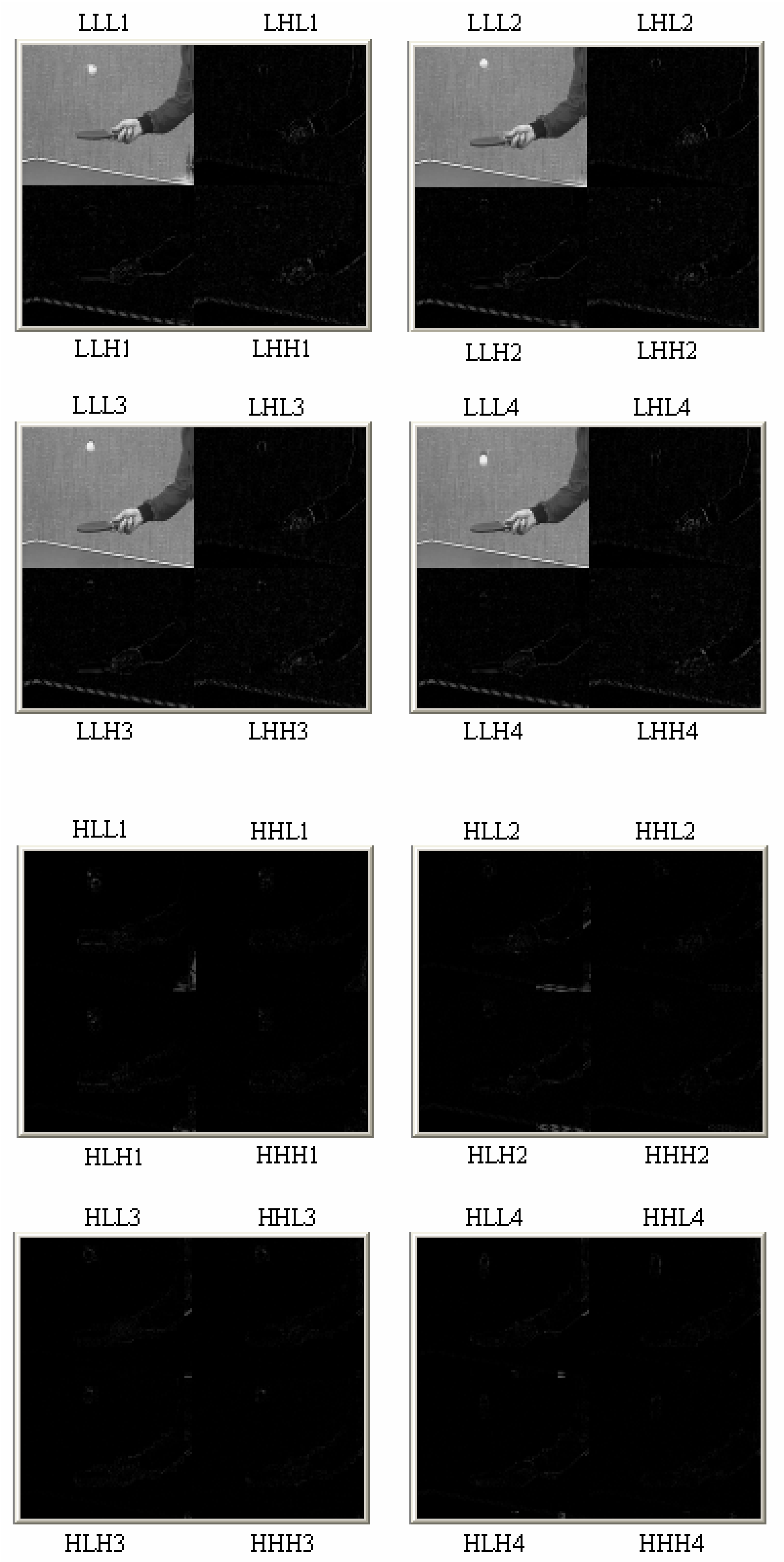


Figure 3.26 MC Spatial-Temporal Haar Decomposition (L-1)

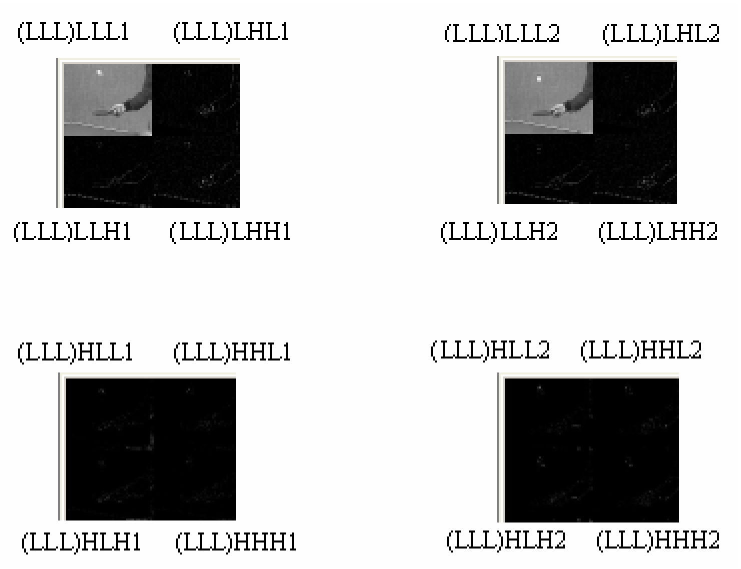


Figure 3.27 MC Spatial-Temporal Haar Decomposition (L-2)

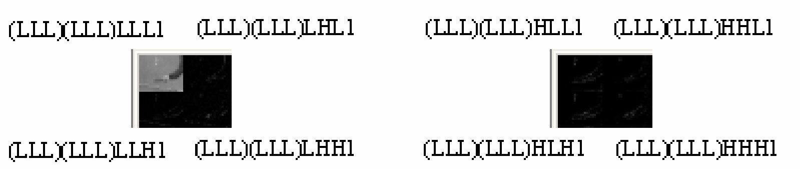


Figure 3.28 MC Spatial-Temporal Haar Decomposition (L-3)

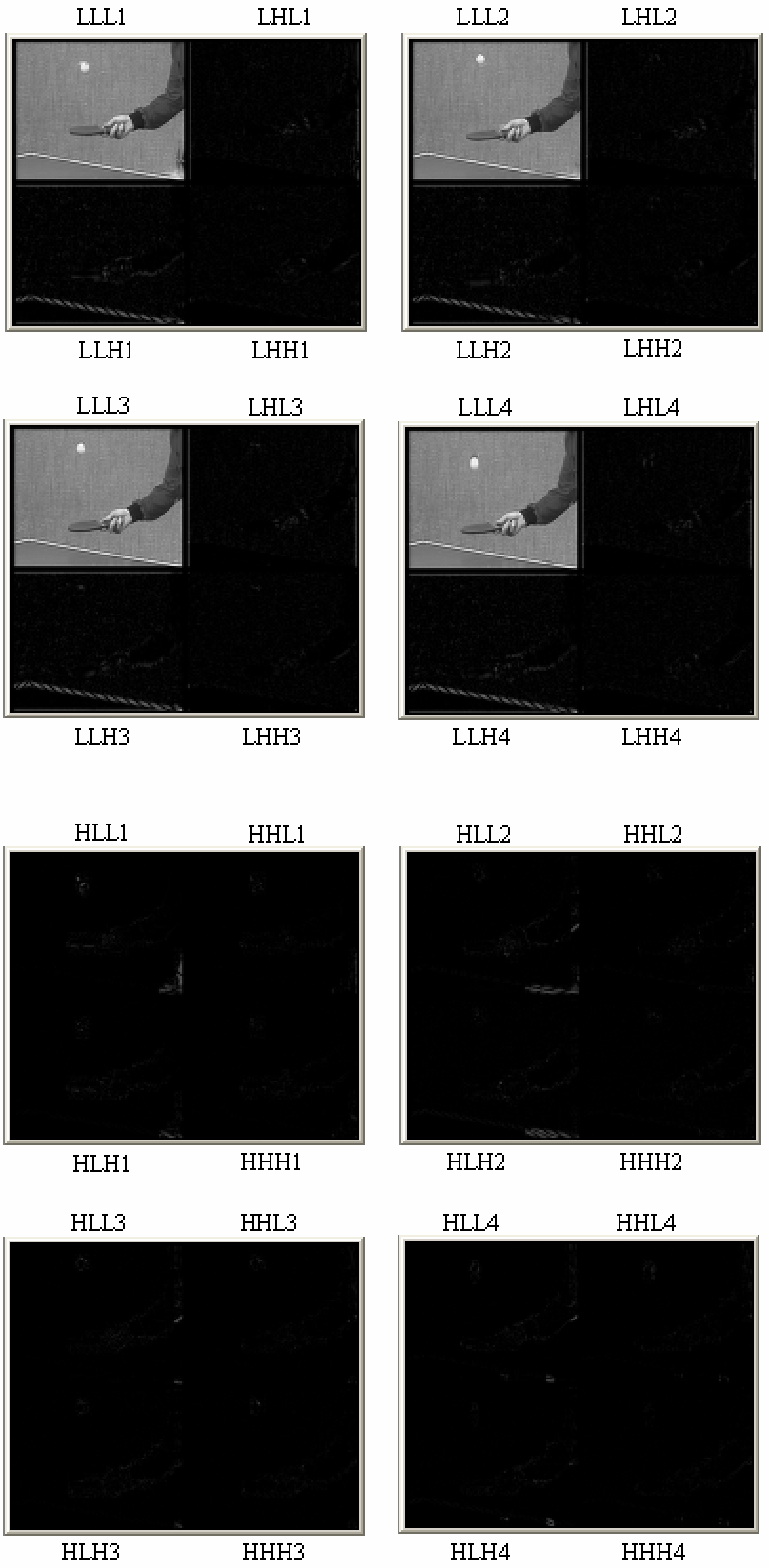


Figure 3.29 MC Spatial-Temporal Haar & Db-9/7 Decomposition (L-1)

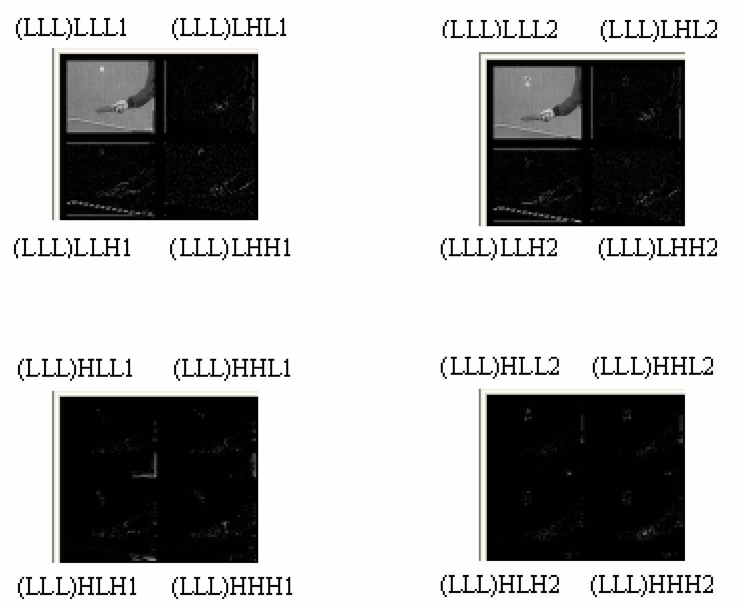


Figure 3.30 MC Spatial-Temporal Haar & Db-9/7 Decomposition (L-2)

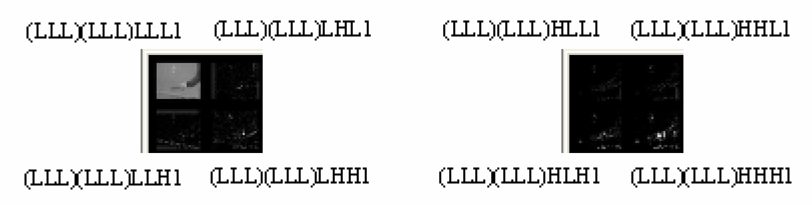


Figure 3.31 MC Spatial-Temporal Haar & Db-9/7 Decomposition (L-3)

As a measure of distribution of coefficients, entropies are calculated for all test cases. The results are given in Table 3.7. In all cases and at all levels, entropies are less than those given in Table 3.1 and 3.2. The abrupt drop is due to the spatial (2-D) decomposition in (t + 2-D) scheme. The contribution of one level spatial-temporal decomposition to distribution of coefficients can only be achieved by three successive MC temporal only decomposition. Three-level decomposition represents the entire video signal as a single frame with averaged motion in temporal only decomposition. Regarding the low subband frames in Figures 3.20, 3.23, 3.26, and 3.29, it is possible to accomplish the continuity of video play between the successive frames after a spatial-temporal decomposition. The cost of the spatial-temporal decomposition is a lower scale video.

Table 3.7 Entropy of Coefficients (t + 2-D)

case	decomposition	level-1	level-2	level-3
(1)	3-D H	0.605	0.294	0.254
(2)	MC 3-D H	0.595	0.271	0.230
(3)	3-D H+Db9/7	0.583	0.286	0.247
(4)	MC 3-D H+Db9/7	0.579	0.284	0.243

Table 3.7 gives a lot more information about the individual test cases. Considering (1) and (3), Daubechies-9/7 filters are more effective in signal de-correlation than Haar filters in general. Entropies at all levels in (3) are less than those in (1). It is apparent especially at the first levels in all cases and weakens at deeper levels. Contrary to the general comment, exceptions occur at the 2nd and 3rd levels of (2) and (4), where Haar filters provide lower entropies. It is probable that weights of coefficients are unevenly spread within the interval [0, 25) in (2). It questions if the selected threshold is low enough to localize insignificant coefficients well. Therefore, we reestimated entropies for (2) and (4) using different threshold values below the current threshold. Results are summarized in Table 3.8 and Table 3.9.

Table 3.8 New Entropy of Coefficients (test case: 2)

threshold	level-1	level-2	level-3
25.0	0.595	0.271	0.230
12.0	0.706	0.499	0.476
8.0	0.805	0.662	0.647

Table 3.9 New Entropy of Coefficients (test case: 4)

threshold	level-1	level-2	level-3
25.0	0.579	0.284	0.243
12.0	0.685	0.490	0.469
8.0	0.781	0.636	0.622

According to the Tables 3.8 and 3.9, entropies in (4) are less than those in (2) at all levels when the threshold is set to either 12.0 or 8.0. The results give clues about the distribution of coefficients specifically in [0, 25). In test case (2), more coefficients exist near 25.0 and the majority of coefficients accumulate in intervals [12, 25) and [8, 25). On the other hand, the Daubechies-9/7 filters contribute more coefficients to accumulate closer to zero in (4).

The contribution by MC technique is observable in the case of current threshold. The realization is achieved by comparing (1) with (2), and (3) with (4) in Table 3.7. Daubechies-9/7 filters are advantageous especially when used over signals with larger size relative to the filters' length as previously mentioned above. In general, all decompositions employing MC are superior in signal de-correlation to those that do not utilize it. Since the motion blur is eliminated, a visual improvement in spatial-temporal low subband frames can also be perceived by naked eye in Figures 3.26 and 3.29 when compared to Figures 3.20 and 3.23, respectively. The difference information implied by high subband frames is lack of motion, especially around the ball, the hand, and the arm. At deeper levels, the perception is less because the subband frames' size, in other words, the spatial resolution decreases.

The insertion of the GAP into Figure 3.18 increases the algorithmic complexity. Previously, we declared the complexity of block matching as O(n^4), and the complexity of temporal wavelet decomposition as O(n^2). The 3-D decomposition is sequentially performed on three dimensions. Since the 3-D decomposition is performed by applying the 1-D WT three times, the algorithmic complexity remains unchanged. The overall complexity of the system is determined by the motion estimation. Only the number of operations (multiplications and additions) increases three times. The operational increase causes processing delays. The time estimation for a level 3-D decomposition results in more than 5 seconds in the "Debug" mode over a Pentium® 4 processor with a speed of 1.6 GHz. Table 3.10 gives the relative times needed to complete decompositions for three levels in all cases.

Table 3.10 Relative Decomposition Times (t + 2-D)

case	decomposition	level-1	level-2	level-3
(1)	3-D H	1.27	1.43	1.45
(2)	3-D H+Db9/7	1.36	1.53	1.55
(3)	MC 3-D H	1.41	1.58	1.60
(4)	MC 3-D H+Db9/7	1.45	1.63	1.66

Besides processing delays, buffering problems occur since newly generated MVs require storage. The number of MVs calculated in the first two steps of the GAP is given in Table 3.11. Initially, 96.6% of the pixels have one-to-one correspondence before decomposition to the first level. The 3rd and 4th steps of GAP are applied for only 3.4% unconnected ones. At level-1, the motion is estimated among four subband frames (LLL1-LLL4). The size of the subband frames is different in (2) and (4), for which the sizes are explicitly given in Table 3.6. The motion estimation at level-1 revealed 11902 pixels out of 12672 (2 × 72 × 88) are connected in (3) and 13244 pixels out of 13984 (2 × 76 × 92) are connected in (4). The numbers are less than those in Table 3.4 because of the low subband frames in lower scale. The ratio of the connected pixels is 93.9% in (3) and 94.7% in (4). The additional motion gap causes the estimation not to be as successful as in the initial step. At level-2, the number of motion hits constitutes only 94.5% and 97.1% for (3) and (4), respectively. The increase in number of connected pixels is mostly contributed by the motion hits within the block. They constitute the pixels referred in the 1st step of GAP. Since the search window is very small and the motion is averaged, most of the hits occur in the same blocks with referenced. The majority of generated MVs have zero values. Therefore, the information content does not have much significance.

Table 3.11 Number of Connected Pixels (t + 2-D)

case	decomposition	Frame 1-8	level-1	level-2
(2)	MC 3-D H	97895	11902	1498
(4)	MC 3-D H+Db9/7	97895	13244	2040

Table 3.12 Average Percentage of Unconnected Pixels (t + 2-D)

case	decomposition	Frame 1-8	level-1	level-2
(2)	MC 3-D H	3.4%	6.1%	5.4%
(4)	MC 3-D H+Db9/7	3.4%	5.2%	2.8%

Another point to pay attention is the difference between the cases (3) and (4) in Tables 3.11 and 3.12. The spatial decomposition by Daubechies-9/7 filters increases the number of motion hits due to two reasons: (1) The motion estimation is performed over an increased spatial dimension; (2) More MVs with zero value is generated for the coefficients around the boundary of subband images. Since the filtering around the boundaries is performed by zero padding, the resultant coefficients have smaller magnitudes relative to coefficients representing the image in its true dimensions. These insignificant coefficients is observed as black regions in low subband frames at all levels as shown in Figures 3.23 - 3.25 and 3.29 - 3.31.

3.5 Spatial - Temporal Wavelet Synthesis

In Section 3.4, we analyzed wavelet coefficients of 3-D decomposed video signals. Coefficients in low subbands have greater magnitudes and correspondingly higher energies than the coefficients in high subbands. They are called either significant or insignificant depending on their magnitudes. The determination of significance of a coefficient is not a well-defined procedure and depends on the signal statistics. Both temporal and spatial frequency components affect the magnitudes of coefficients. When compressing video, the coefficients having relatively small magnitudes are ignored. Since the information is lost, this type of compression is called *lossy compression*. The size of the data in compressed form (C) relative to the original size (O) is known as the *compression ratio* (R=C/O). Size of the compressed data is given in terms of bits. Coefficients must be quantized and coded to accomplish the bit representation. In order to obtain some feelings about wavelet-based video compression, we just omitted the bit-wise representation in this section, and directly reconstructed the original signal by less number of coefficients. It is very similar to the experiment in Chapter 2 when reconstructing the "*Lena*" image from minority of its 2-D WT coefficients.

The reconstruction of video signal from its coefficients is called "*spatial-temporal synthesis*" operation or "*inverse*" 3-D WT. In other words, the synthesis of video signal implements the (t + 2-D) scheme in reverse order. That is, the spatial ("2-D") composition is followed by the temporal composition, "t". The former and the latter are exactly inverses of the 2-D and 1-D WTs, respectively. They are mentioned in Chapter 2. Figure 3.32 illustrates a 3-level hierarchical 3-D composition of the volume data. The composition starts with one level spatial composition of subbands at level-3. Then, resultant low and high subbands are temporally composed. This entire process is described as one-level spatial-temporal synthesis, and creates the LLL subband at level-2. It can be regarded as merging of sub-cubicals at level-3 to a larger volumetric data that fits exactly to empty location of the cubic at level-2 in Figure 3.32. The synthesis is an iterative procedure. The eight subbands, including LLL at level-2 are spatial-temporal composed, yielding the LLL subband at level-1. Finally, the composition is performed once more for 8 subbands to obtain the original video.

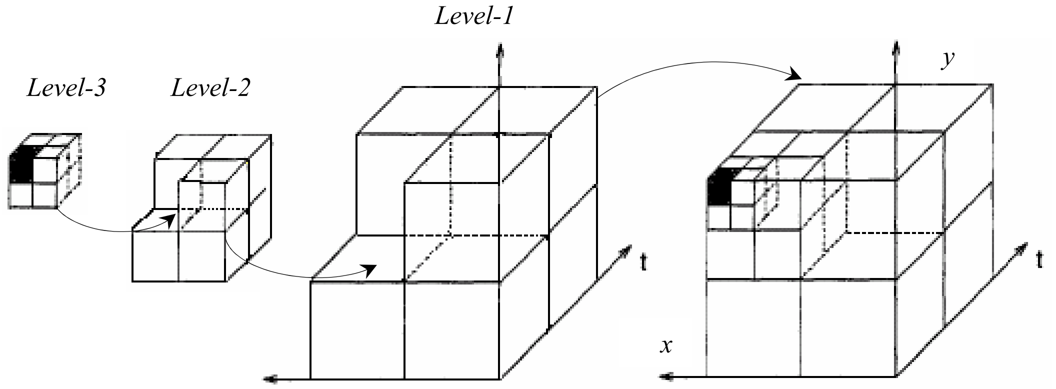


Figure 3.32 3-Level Spatial-Temporal Wavelet Synthesis of GOF

The spatial-temporal synthesis is performed on the coefficients of "*Table Tennis*" video for each test case in Section 3.4. In all cases, temporal compositions are performed using Haar synthesis filters. Either Haar or Daubechies-9/7 synthesis filters are employed for spatial compositions. The synthesis low pass (*h′*) and high pass (*g′*) filter coefficients for both filter types are given below:

$$h'_{haar} = \sqrt{2} \times [0.5 \quad 0.5] = [1/\sqrt{2} \quad 1/\sqrt{2}],$$
$$g'_{haar} = \sqrt{2} \times [0.5 \quad -0.5] = [1/\sqrt{2} \quad 1/\sqrt{2}].$$

$$h'_{db\text{-}9/7} = \sqrt{2} \times [-0.045\ -0.028\ 0.295\ 0.557\ 0.295\ -0.028\ -0.045],$$
$$g'_{db\text{-}9/7} = \sqrt{2} \times [-0.026\ -0.016\ 0.078\ 0.266\ -0.602\ 0.266\ 0.078\ -0.016\ -0.026].$$

The up-sampling part in the inverse transformation is performed by inserting a zero to every other sample in both channels. The convolution part is performed by zero padding around the subband frames. The synthesis with Daubechies-9/7 filters needs more operations. After the decomposition with Daubechies-9/7 filters, each row and column of subband frames has additional wavelet coefficients from the beginning and from the end. They usually have small magnitudes and appear as black regions around the subband frames. The convolution increases the number of these coefficients more when composing subbands, and thus enlarges the black regions around the reconstructed LLL subband frame. Since coefficients in black regions of reconstructed frames have zero magnitudes, they are discarded before placing the subband into the empty location in the previous level as shown in Figure 3.32. The number of discarded coefficients in each row and column of the subband is constant, 14. In other words, 7 coefficients both from the beginning and from the end of either a row or a column are discarded. The mathematical explanation of the extension of low and high subband frames is given by (3.2) and (3.3), respectively, in both analysis (decomposition) and synthesis (composition) phases below.

$$\mathit{int}\,[((\ldots + 9 - 1) + 1) / 2] \times 2 + 7 - 1 = \ldots + 14 \tag{3.2}$$

$$\mathit{int}\,[((\ldots + 7 - 1) + 1) / 2] \times 2 + 9 - 1 = \ldots + 14 \tag{3.3}$$

The total number of coefficients in test cases (1) and (2) is 8 × (144 × 176) = 202752; and it is constant at all levels. The number of coefficients in (3) and (4) depends on the decomposition level. According to the number of subband frames in Figures 3.23 - 3.25 and 3.29 - 3.31 and the subband frames' sizes in Table 3.6, the total number of coefficients at each level are calculated as follows:

L-1: $$8 \times (151 \times 183) = 221064. \tag{3.4}$$

L-2: $$\begin{aligned} &4 \times (151 \times 183) + \\ &4 \times (91 \times 76 + 92 \times 75 + 91 \times 75) + \\ &4 \times (83 \times 99) = 225964. \end{aligned} \tag{3.5}$$

L-3: $$\begin{aligned} &4 \times (151 \times 183) + \\ &4 \times (91 \times 76 + 92 \times 75 + 91 \times 75) + \\ &2 \times (83 \times 99) + \\ &2 \times (49 \times 42 + 50 \times 41 + 41 \times 49) + \\ &2 \times (57 \times 49) = 227350. \end{aligned} \tag{3.6}$$

The "*Table Tennis*" video is reconstructed using 6336 coefficients. They constitute 3.125% of total number of coefficients at all levels for cases (1) and (2). It is analogous to Matlab tests in Chapter 2, where the "*Lena*" image is reconstructed using its 2048 (3.125%) wavelet coefficients. The ratio differs for (3) and (4) depending on the decomposition level as explained by (3.4) – (3.6) above.

The percentages are 2.86 at level-1, 2.80 at level-2, and 2.78 at level-3. In order to obtain these percentages, zero is assigned to all coefficients with magnitudes less than or equal to certain threshold values in Table 3.13.

Table 3.13 Zeroing Threshold Values for the "*Table Tennis*" Video (CN*: 6336)

case	composition	level-1	level-2	level-3
(1)	3-D H	384.66	45.60	32.17
(2)	MC 3-D H	384.31	41.25	28.99
(3)	3-D H+Db9/7	384.62	65.20	37.34
(4)	MC 3-D H+Db9/7	386.61	65.33	38.58

* Number of coefficients used in the reconstruction

Table 3.14 Average PSNR of the Reconstructed "*Table Tennis*" Video (CN: 6336)

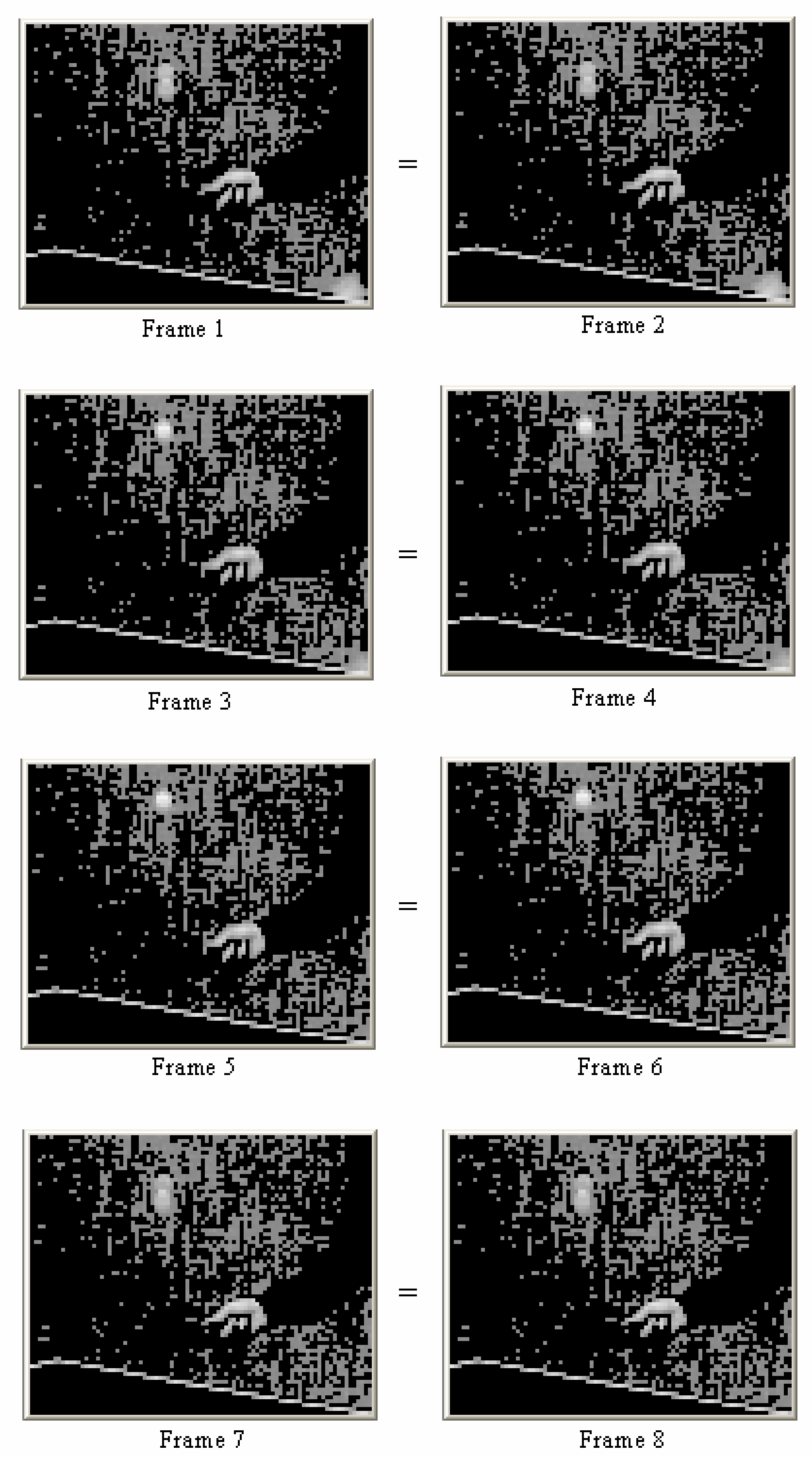
case	composition	level-1	level-2	level-3
(1)	3-D H	7.68	30.88	32.11
(2)	MC 3-D H	7.69	31.48	32.65
(3)	3-D H+Db9/7	8.11	29.95	31.69
(4)	MC 3-D H+Db9/7	8.13	29.94	31.58

The threshold values in Table 3.13 give us information about the effects of composition levels on synthesizes. As we go further down the composition, the zeroing threshold values decrease. It is all consistent with entropy values in Table 3.7 such that more coefficients with smaller magnitudes accumulate (in high subbands) at deeper levels. The thresholds are extremely high at level-1; and it is highly probable that zeroing exceeds high subbands and comprehends the coefficients in low subbands. It is not desired to zero coefficients with higher magnitudes in low subbands as it causes majority of information to be lost. Figures 3.33, 3.34, 3.37, and 3.38 show reconstructed frames which suffer from this information loss in all cases. The average PSNRs of the reconstructed frames are given in Table 3.14. They are extremely low at level-1, which is consistent with high threshold values at the same level in Table 3.13.

The threshold values drops abruptly at level-2 and level-3. A great majority of the zeroized coefficients are in high subbands. The information loss is significantly compensated by further decompositions. The corresponding PSNR values increase abruptly at the same levels in Table 3.14. The reconstructed frames from level-2 are shown in Figures 3.35, 3.36, 3.39, and 3.40. If PSNR values are used as a measure of the quality of reconstructed frames, it can be said that a tremendous improvement in image (frame) quality is realized compared to Figures 3.33, 3.34, 3.37, and 3.38. Reconstruction of three-level decomposed frames is not displayed here because the improvement in resolution is hardly distinguishable by naked eye. It can be quantitatively expressed by the difference in PSNR values. Table 3.14 indicates the improvement in average frame quality to be up to 1.7 dB from level-2 to level-3.

In the worst cases (synthesizes at level-1), reconstructed frames are not unique. Every frame is exactly the same as its successive frame in Figures 3.33 and 3.37, which is indicated by the equality sign between the successive frames in the figures. As opposed to the test cases (1) and (3), the continuity in video play is not lost in cases (2) and (4), where motion is saved by MC filtering. The uniqueness of frames is implied by the inequality signs in Figures 3.34 and 3.38. Achievement of continuous video play is perhaps the most significant advantage of MC technique since the profit in PSNRs is not very much. According to Table 3.14, the highest improvement is 0.6 db, which is obtained at level-2 in MC Haar synthesis.

Besides the contribution of MC technique, experiments give us information about the effects of different synthesis filters on the reconstructed video. Both zeroing thresholds and PSNR values for test cases (3) and (4) in Tables 3.13 and 3.14 are worse than those in (1) and (2). Although our expectations for MC technique are satisfied with PSNRs in the Haar case, we could not achieve the same performance for Daubechies-9/7 ones. If we benefit Daubechies-9/7 filters well in the experiments may be questionable; and results motivate us for retesting the reconstruction to better answer the best performance case.



Frame 1

Frame 2

Frame 3

Frame 4

Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.33 Reconstructed “*Table Tennis*” Video (test case: 1, L-1, CN: 6336)

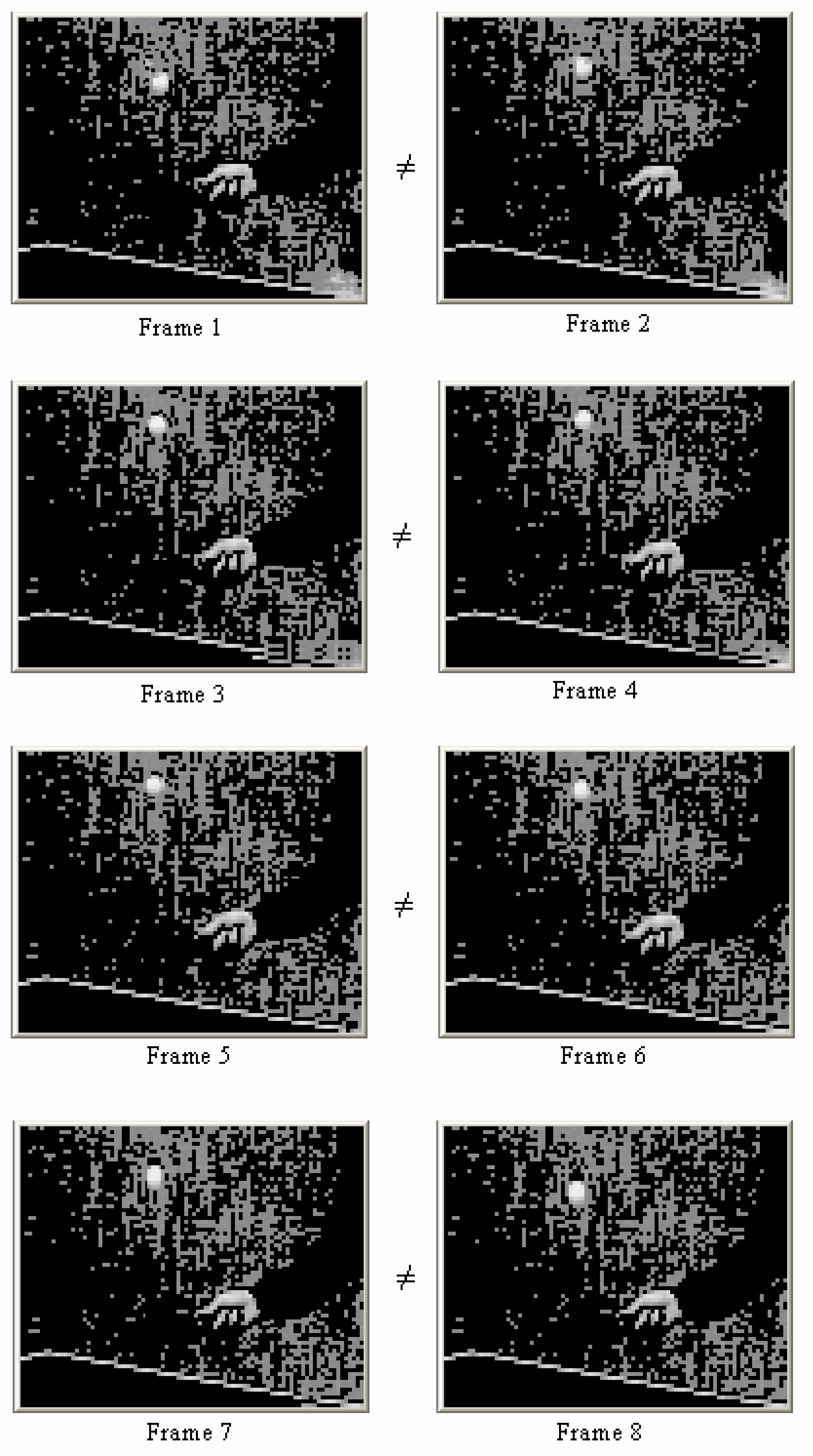
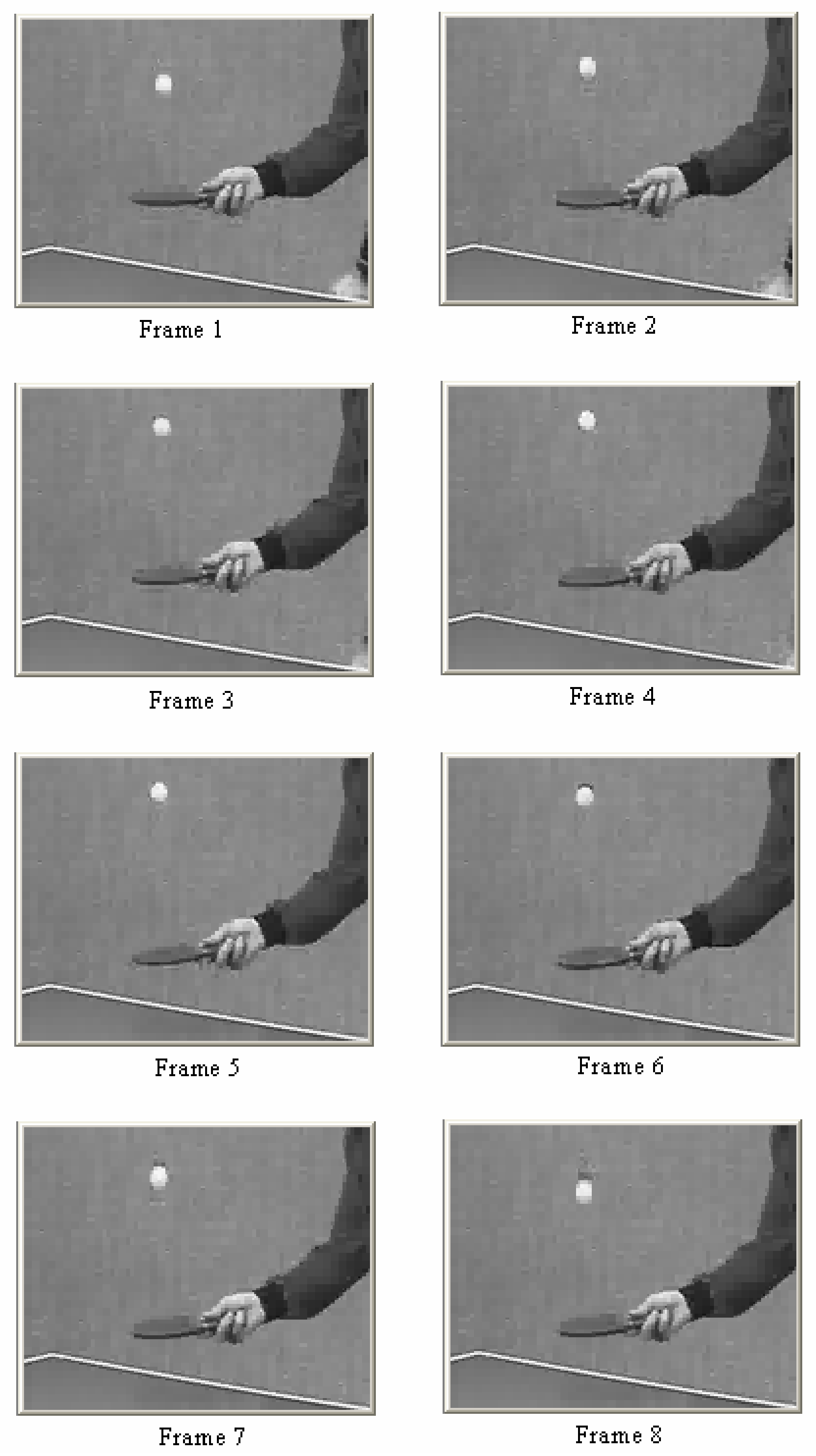


Figure 3.34 Reconstructed “*Table Tennis*” Video (test case: 2, L-1, CN: 6336)



Frame 1

Frame 2

Frame 3

Frame 4

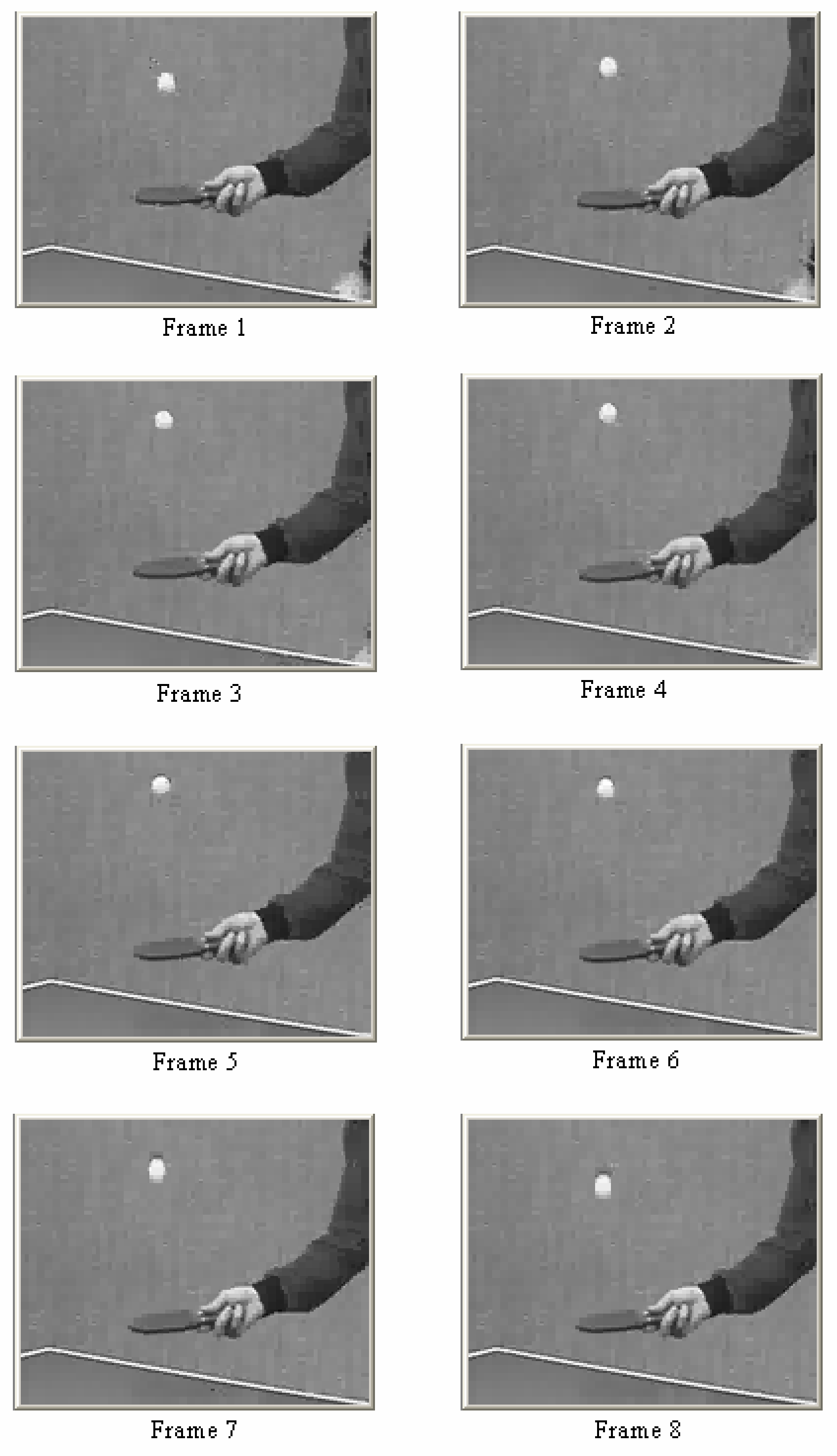
Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.35 Reconstructed "*Table Tennis*" Video (test case: 1, L-2, CN: 6336)



Frame 1

Frame 2

Frame 3

Frame 4

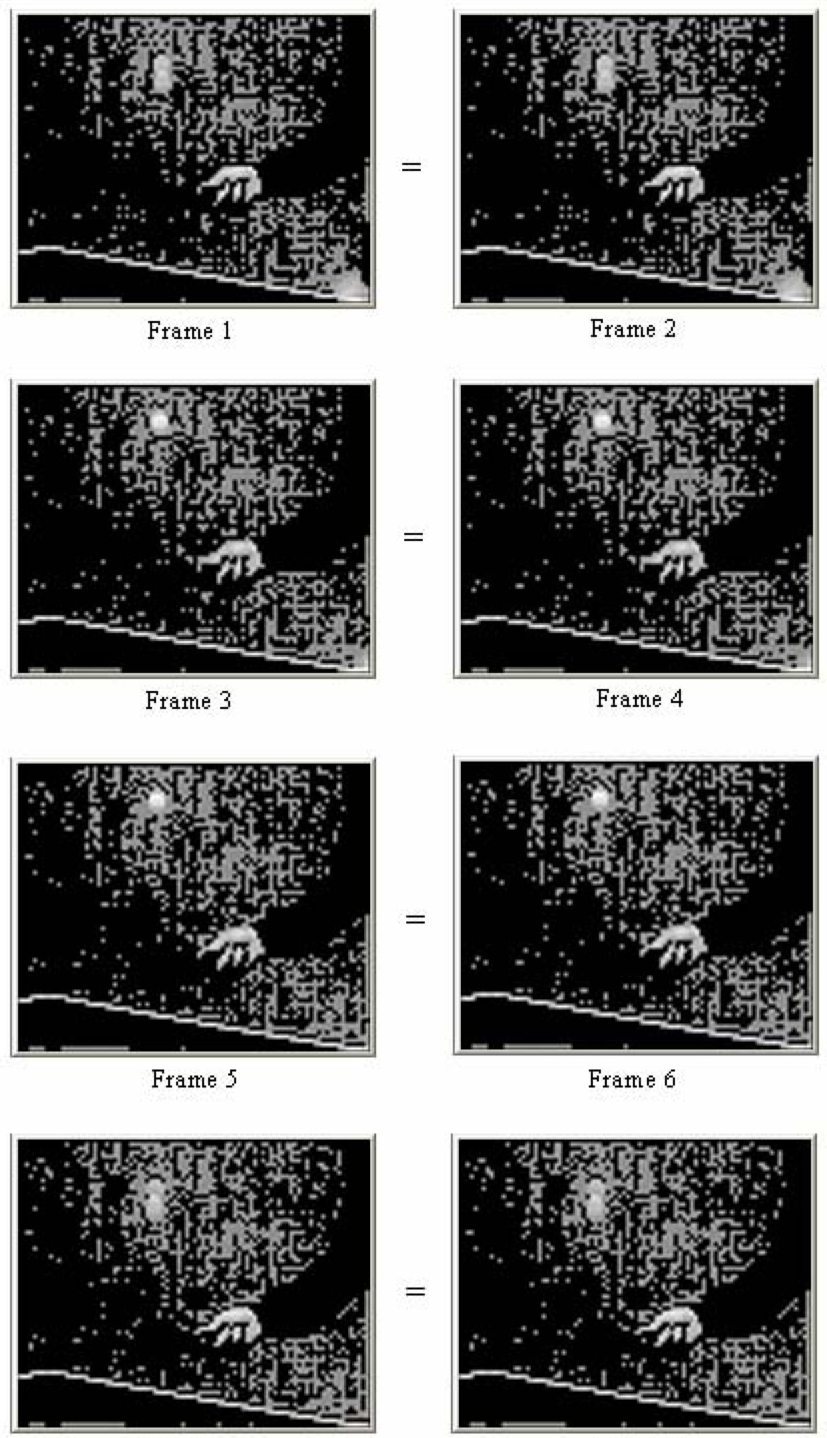
Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.36 Reconstructed “*Table Tennis*” Video (test case: 2, L-2, CN: 6336)



Frame 1

Frame 2

Frame 3

Frame 4

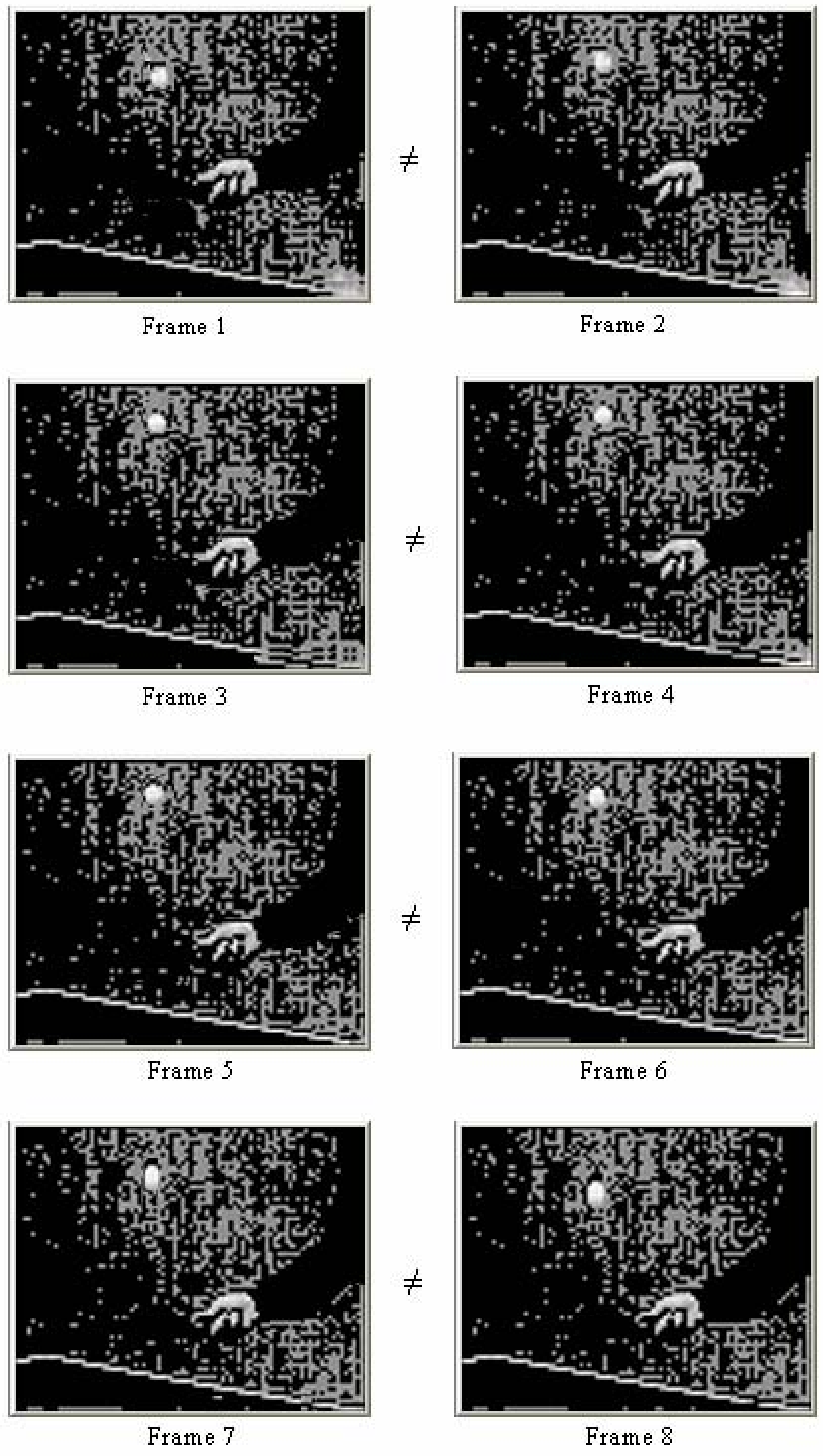
Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.37 Reconstructed “*Table Tennis*” Video (test case: 3, L-1, CN: 6336)



Frame 1

Frame 2

Frame 3

Frame 4

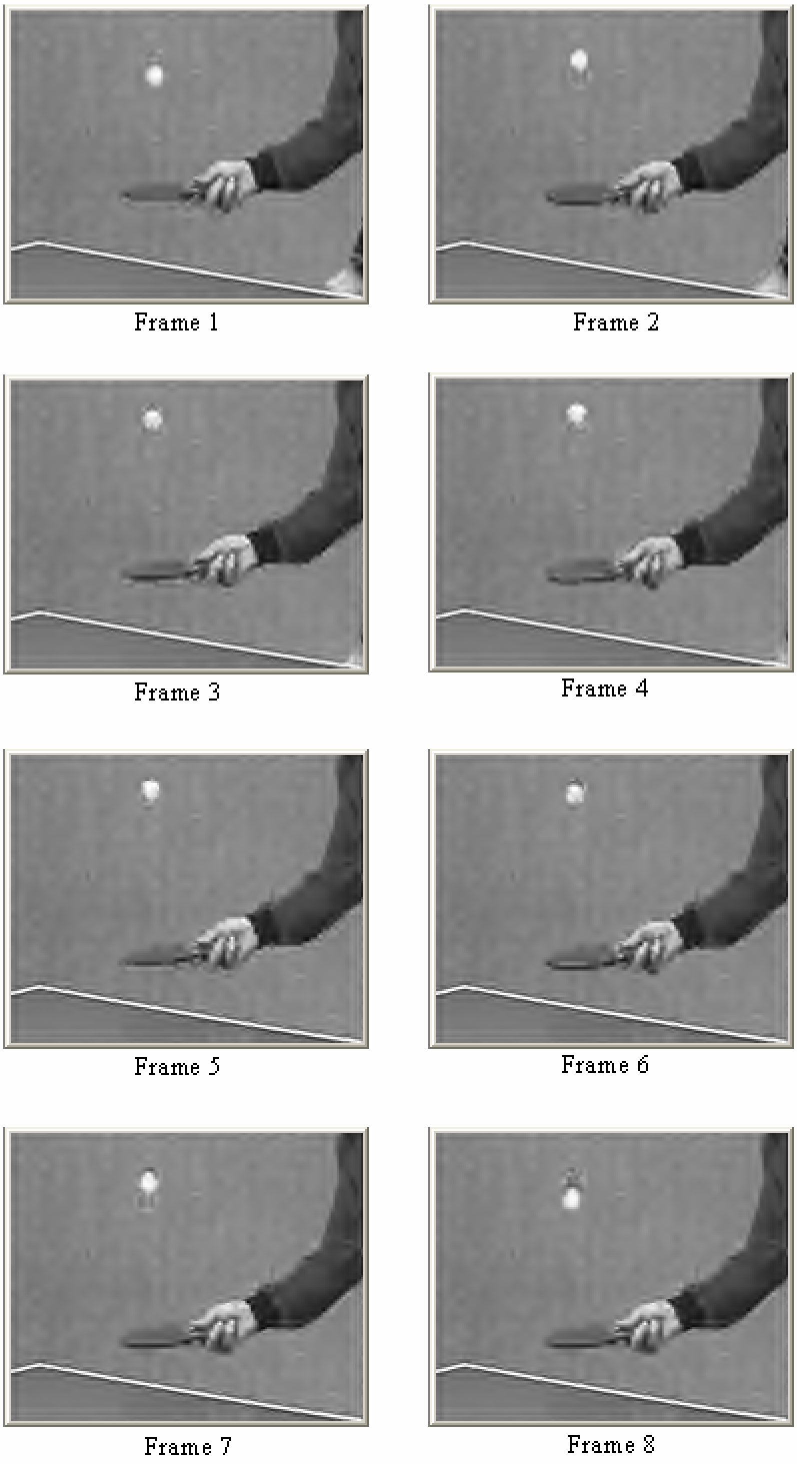
Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.38 Reconstructed “*Table Tennis*” Video (test case: 4, L-1, CN: 6336)



Frame 1

Frame 2

Frame 3

Frame 4

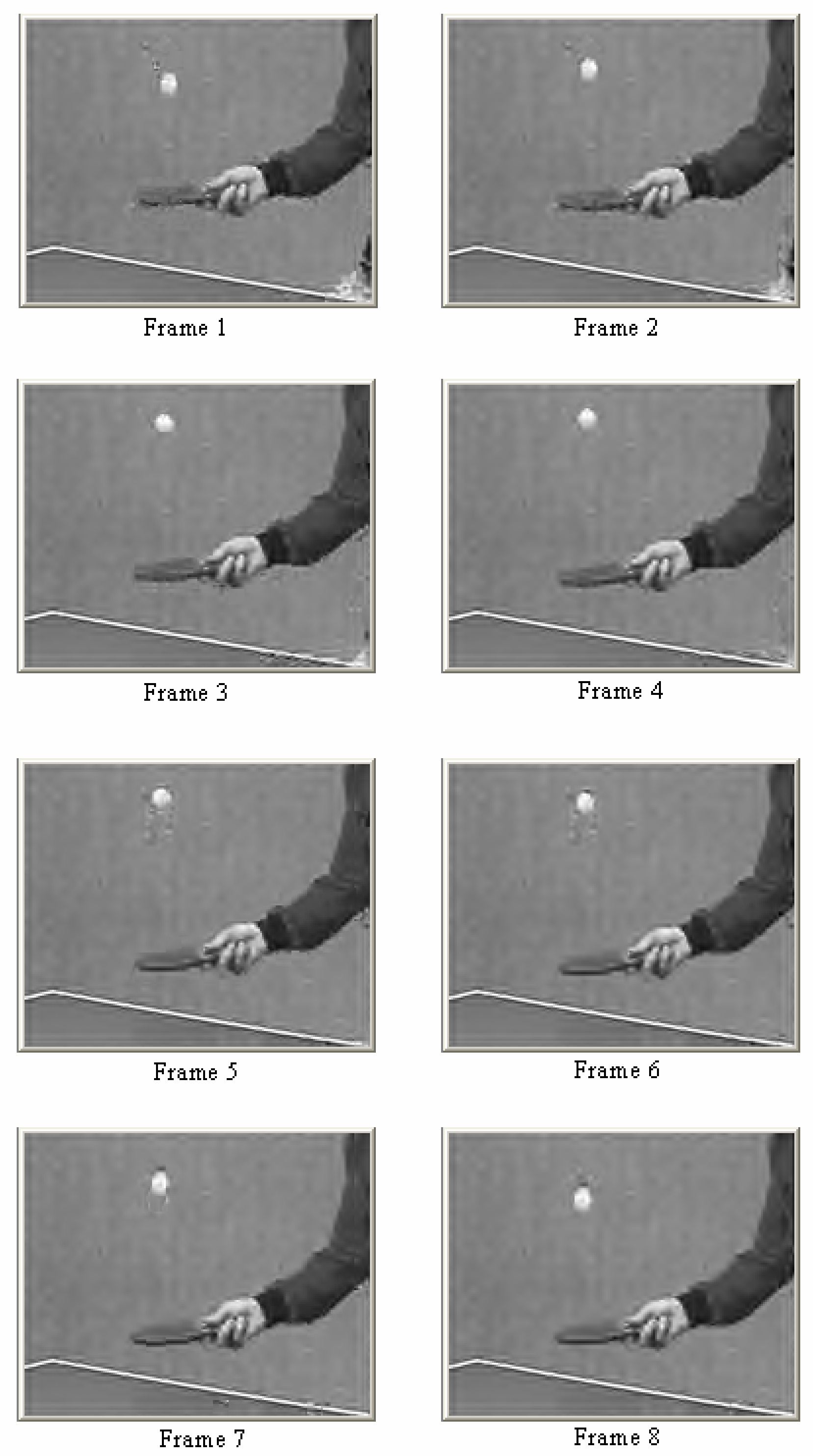
Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.39 Reconstructed “*Table Tennis*” Video (test case: 3, L-2, CN: 6336)



Frame 1

Frame 2

Frame 3

Frame 4

Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.40 Reconstructed “*Table Tennis*” Video (test case: 4, L-2, CN: 6336)

There are two concerns to utilize the Daubechies-9/7 filters: (1) Dimensions of video frames shall be as large as possible compared to filters' length; (2) Different frequency components in the spatial domain shall exist to bring the advantage of Daubechies-9/7 filters in front. The former is already mentioned in Section 3.3 while analyzing the wavelet coefficients. The latter is mentioned in Chapter 2 with a different filter (Db-4) from the Daubechies family but not proven empirically. Therefore, we focus more on the latter here.

The "*Table Tennis*" video is a simple one in a sense that the spatial frequency components are not variants. The background scene and neighboring pixels do not alter much. A better candidate is the "*Conference*" video shown in Figure 3.41. The retesting is performed using "*Conference*" video and the resultant zeroing threshold and average PSNR values are displayed in Table 3.15 and 3.16, respectively.

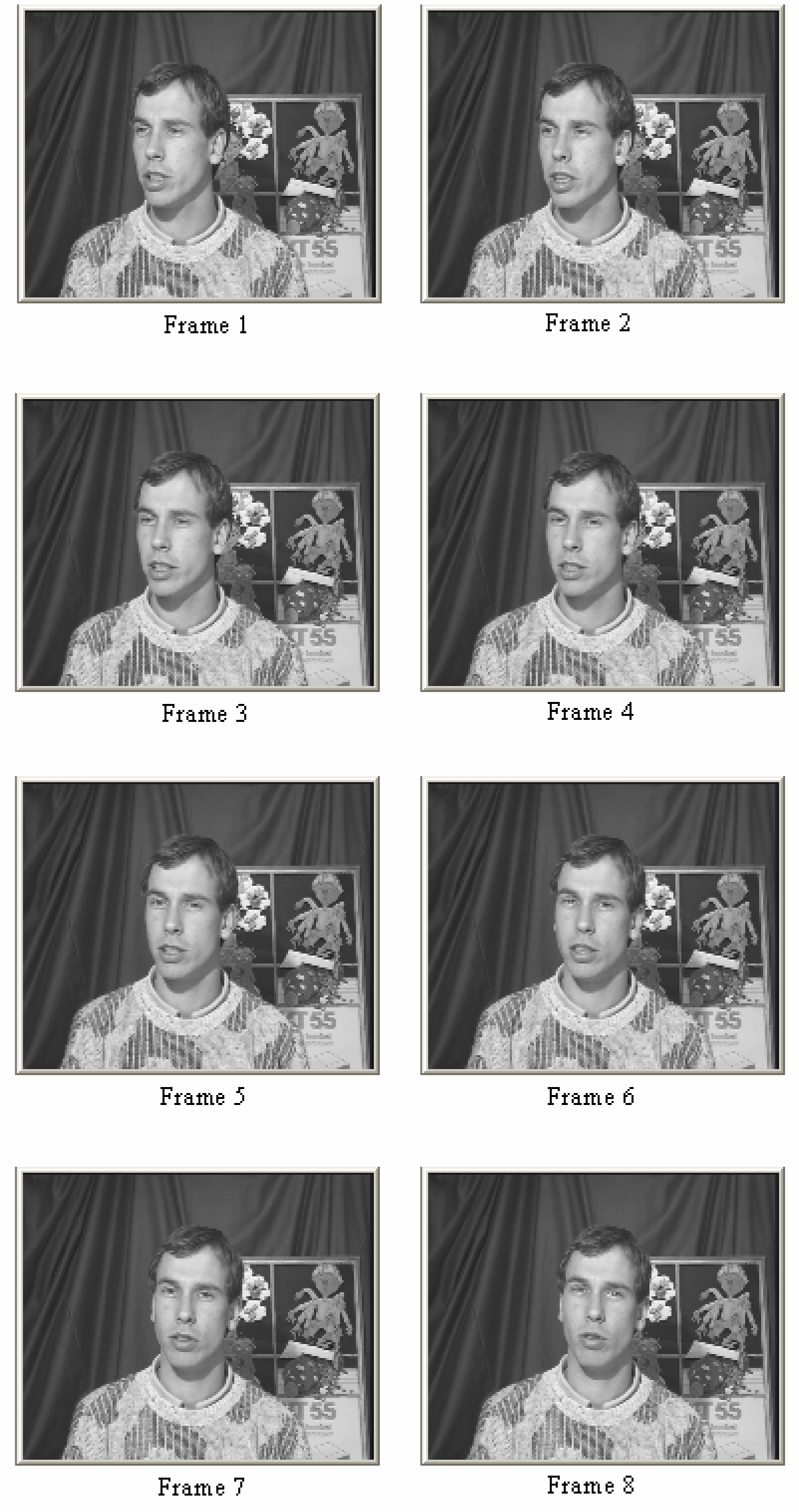
Table 3.15 Zeroing Threshold Values for the "*Conference*" Video (CN: 6336)

case	composition	level-1	level-2	level-3
(1)	3-D H	430.98	76.25	61.51
(2)	MC 3-D H	433.10	75.66	60.10
(3)	3-D H+Db9/7	431.11	72.95	57.43
(4)	MC 3-D H+Db9/7	431.06	75.21	58.43

Table 3.16 Average PSNR of the Reconstructed "*Conference*" Video (CN: 6336)

case	composition	level-1	level-2	level-3
(1)	3-D H	10.67	25.32	26.33
(2)	MC 3-D H	10.71	25.93	27.04
(3)	3-D H+Db9/7	11.04	25.85	26.89
(4)	MC 3-D H+Db9/7	11.10	26.04	27.12

In general, the threshold values in Table 3.15 are greater than those in Table 3.13, and the PSNR values in Table 3.16 are less than those in Table 3.14. The decrease in PSNR values is consistent with the increase in threshold values



Frame 1

Frame 2

Frame 3

Frame 4

Frame 5

Frame 6

Frame 7

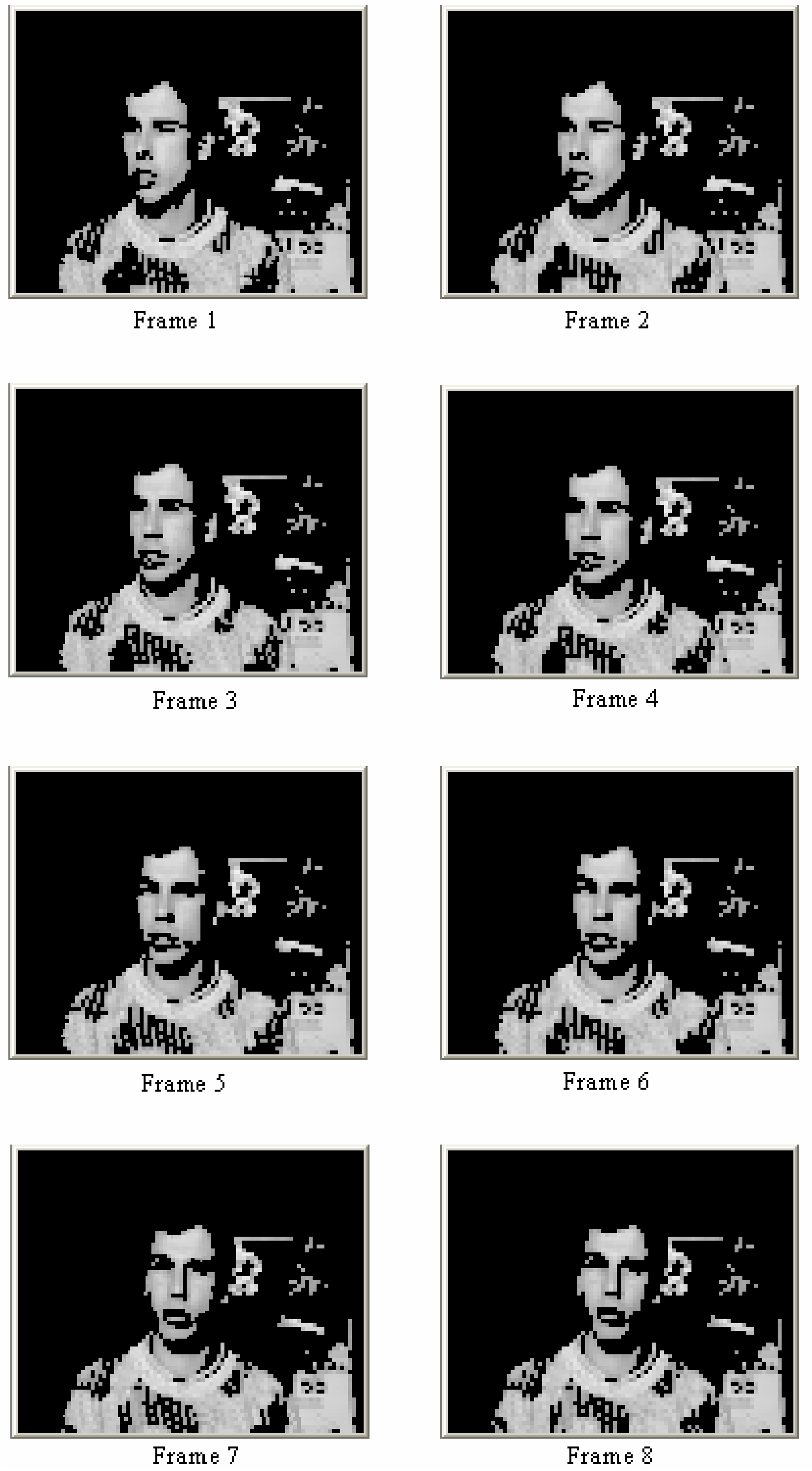
Frame 8

Figure 3.41 Original “*Conference*” Video Signal

since the information (energy of coefficients) loss is more for the "*Conference*" video. Similar to the comments made for the "*Table Tennis*" video, the threshold values decrease as the composition level goes further down and that MC technique provides continuous video play. A different observation for the reconstructed frames is related to Daubechies-9/7 filters. The PSNR values in (3) and (4) are better than those of their counterparts in (1) and (2). The best PSNR improvement via Daubechies-9/7 filters is up to 0.5 dB between (1) and (3). Results are an indication of better de-correlation performance of Daubechies-9/7 filters on video signals with complicated (spatial) frequency components.

Another point related to the Daubechies filters is their smoothing properties. The frames reconstructed using Daubechies-9/7 filters are smoother than those reconstructed via Haar filters. The difference can be perceived by naked eye by comparing Figures 3.44 and 3.45. It is even perceivable in worst case reconstruction at level-1. As opposed to Daubechies-9/7 filters, Haar filters cause rectangular regions to appear where spatial frequency changes occur in reconstructed frames.

All cases using MC technique are superior to those not utilizing it. However, the improvement is not significant in general. The best case is achieved at level-3 in (4) and reconstructed frames are displayed in Figure 3.45. The best contribution to PSNRs via MC is obtained for (2), where MC Haar synthesis achieves 0.7 dB improvement compared to (1).



Frame 1

Frame 2

Frame 3

Frame 4

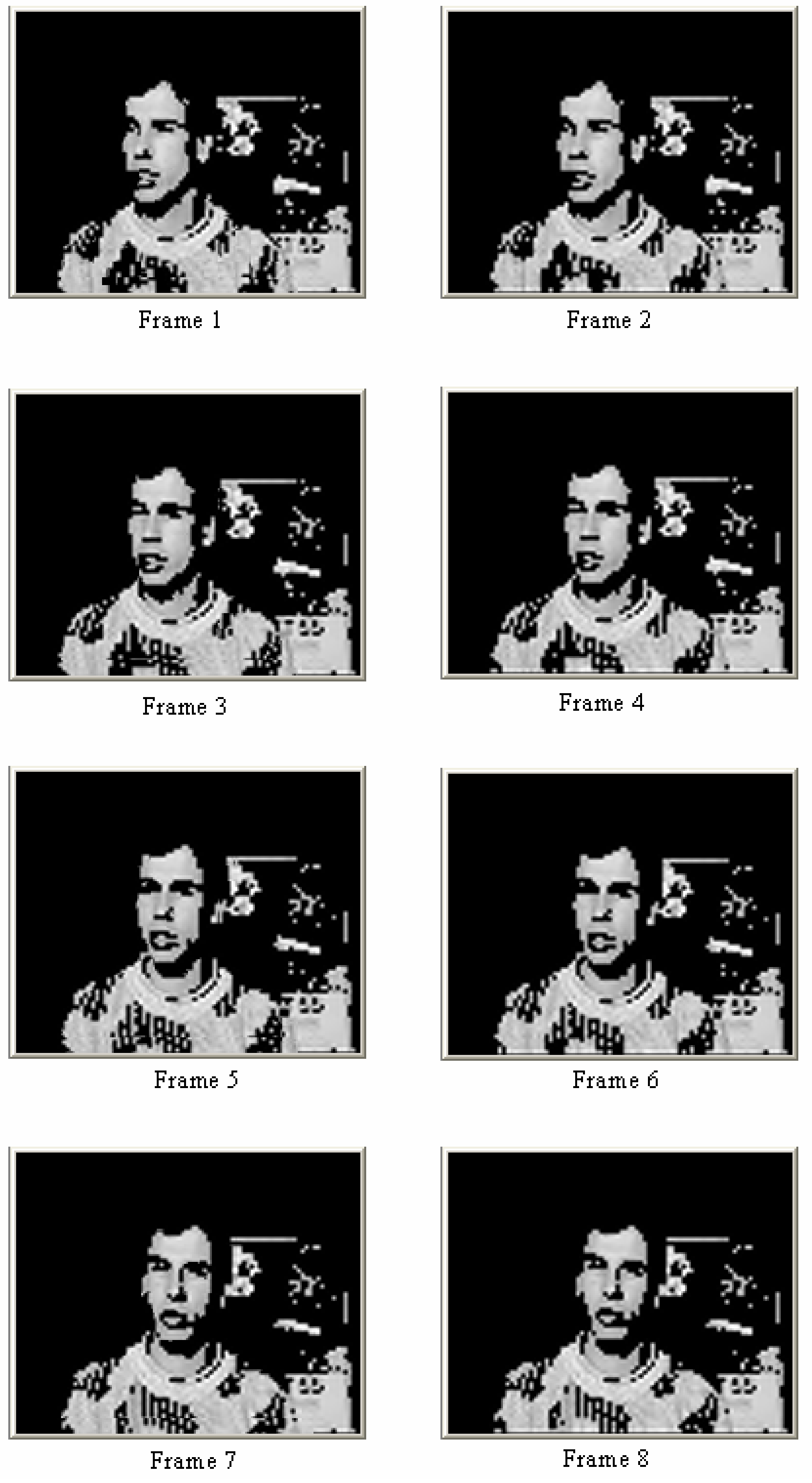
Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.42 Reconstructed "*Conference*" Video (test case: 2, L-1, CN: 6336)



Frame 1

Frame 2

Frame 3

Frame 4

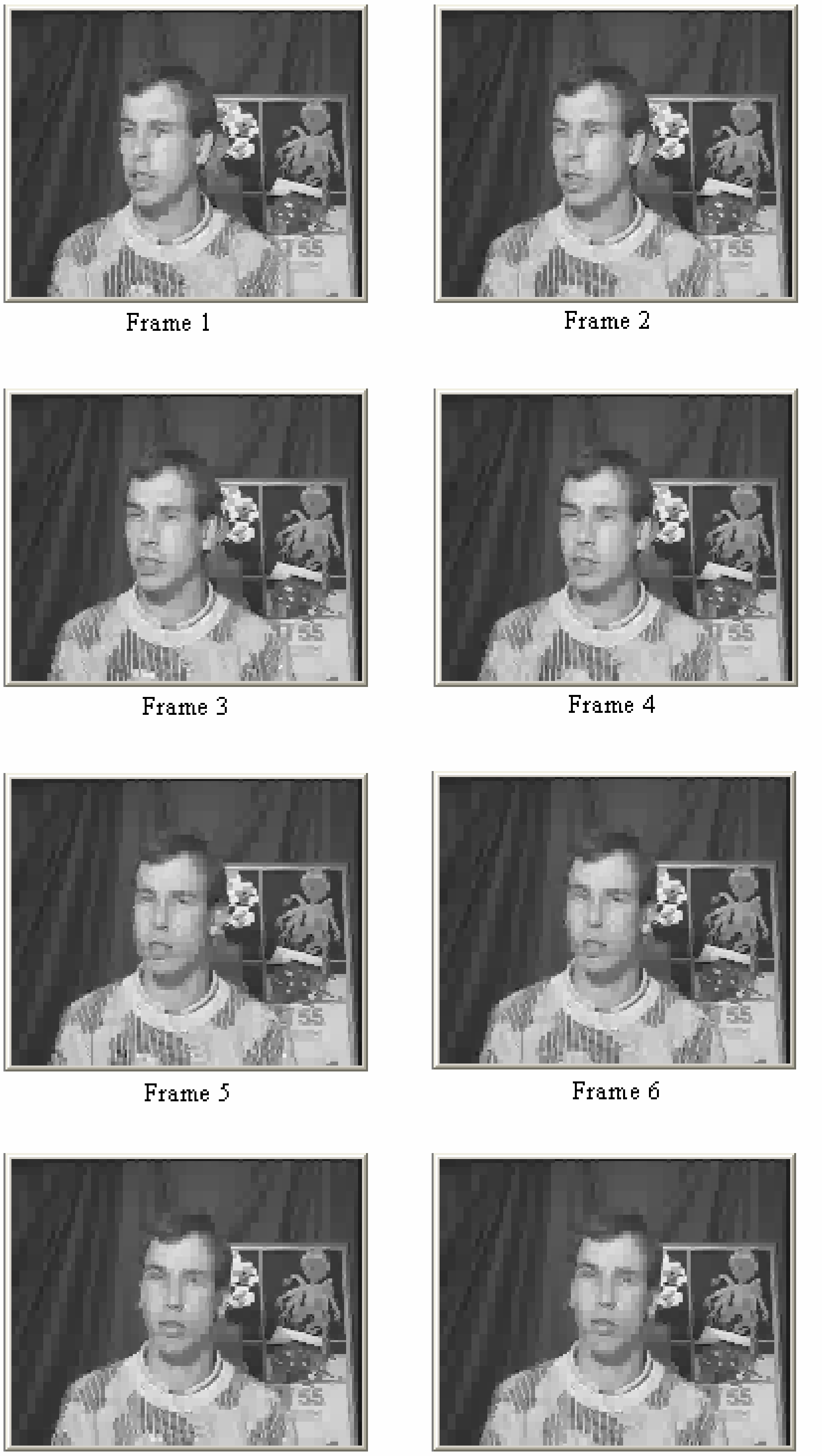
Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.43 Reconstructed "*Conference*" Video (test case: 4, L-1, CN: 6336)



Frame 1

Frame 2

Frame 3

Frame 4

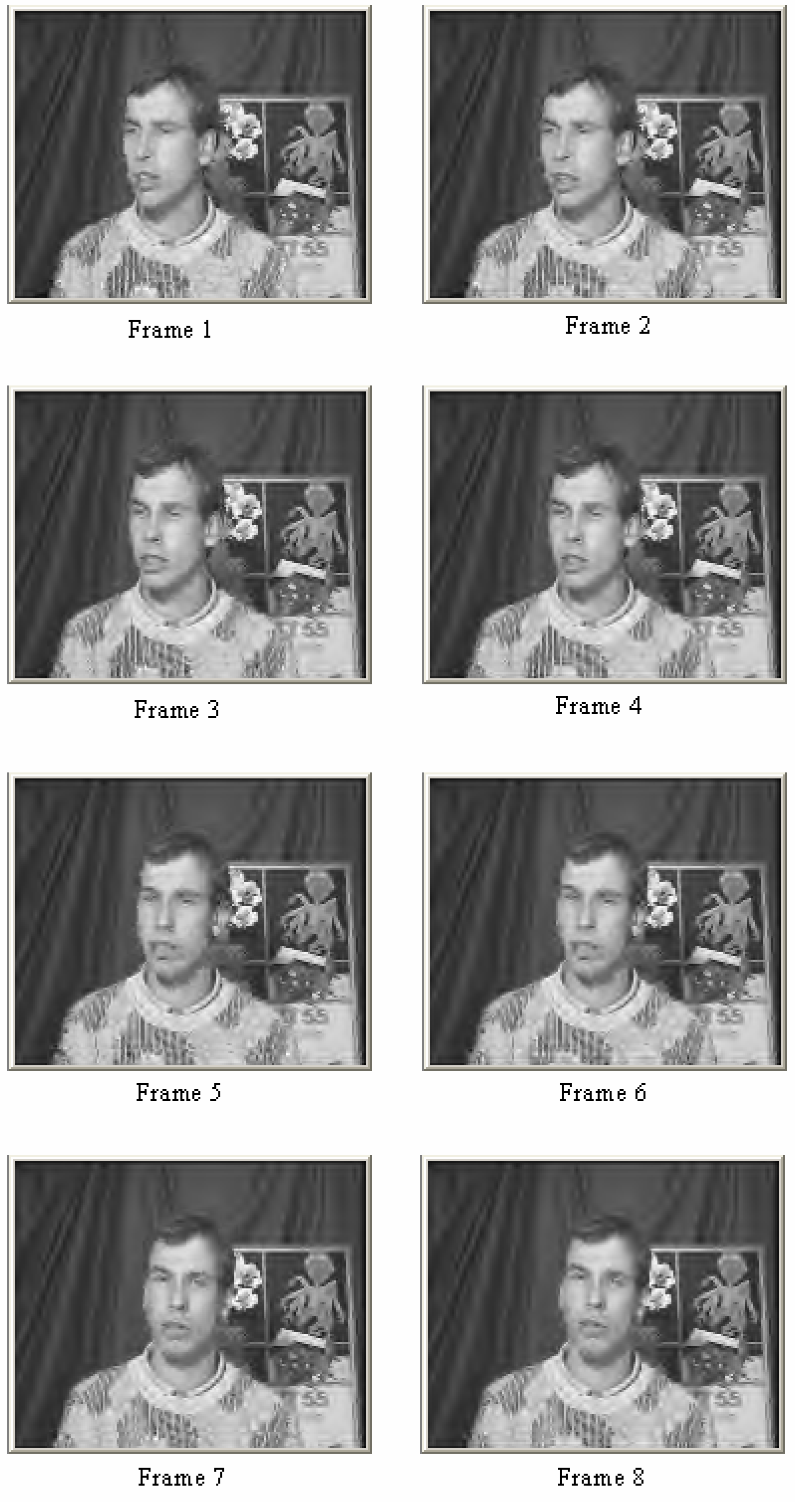
Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.44 Reconstructed “*Conference*” Video (test case: 2, L-3, CN: 6336)



Frame 1

Frame 2

Frame 3

Frame 4

Frame 5

Frame 6

Frame 7

Frame 8

Figure 3.45 Reconstructed "*Conference*" Video (test case: 4, L-3, CN: 6336)

CHAPTER 4

WAVELET BASED CODING OF VIDEO

4.1 Introduction

In Chapters 2 and 3, we reconstructed our test signals by using their most significant coefficients, which constitutes only a small fraction of the transform coefficients. While the significant coefficients are processed without any loss, insignificant ones are ignored in the synthesis operation. Although it is not a true expression of *compression ratio*, the fraction of number of significant coefficients to total number of transform coefficients can be taken as a measure of signal compression via WT. In practice, the *compression ratio* (R) is indicated as the ratio of the size of data in compressed form (C) to the original size (O). Sizes of both compressed and original data are given in terms of bits.

The bit representation of wavelet coefficients is achieved by using a recent wavelet coding algorithm, called "*Set Partitioning in Hierarchical Trees*" (SPIHT) [19]. It is originated from "*Embedded Zerotree Wavelet*" (EZW) image coding technique [18]. The basic principles of both can be summarized in three steps: (1) partial ordering of the transformed image elements by magnitude, with transmission of order by a subset partitioning algorithm that is duplicated at the decoder, (2) ordered bit plane transmission of refinement bits, and (3) exploitation of the self similarity of the image wavelet transform across different scales. Being a more recent technique, SPIHT partitions the subsets of wavelet coefficients and conveys their *significance* information in a different way than the EZW.

We applied the 2-D version of SPIHT coding technique (see Appendix C) to our test videos. The number of bits used in sorting and refinement passes when preparing bit planes is counted. Then, processing delays are estimated while encoding and decoding the code words. Finally, we evaluated the 2-D SPIHT technique on a three dimensional decomposed video signals.

4.2 Principles of Embedded Wavelet Image Coding

Embedded wavelet image coding techniques address the following two problem: (1) obtaining the best signal (image) quality for a given bit rate, and (2) accomplishing this task in an embedded fashion, i.e., in such a way that all encodings of the same image at lower bit rates are embedded at the beginning of the bit stream for the target bit rate. In other words, the decoder can cease decoding at any point in the bit stream, and still produce exactly the same image that would have been encoded at the bit rate corresponding to the truncated bit stream [18]. The problem is important in many applications, particularly for progressive transmission.

Both EZW and 2-D SPIHT were developed to generate an embedded bit stream with no apparent sacrifice in image quality. They utilize a technique to predict the absence of significant information across scales/levels by exploiting the self-similarity inherent in images or video frames. Using scalar quantization followed by entropy coding, the probability of the occurrence of the most likely symbol is extremely high. Typically, a large fraction of the bits available is allocated to encode the *significance map*, or the binary decision as to whether a wavelet coefficient has a zero or non-zero quantized value. It follows that a significant improvement in encoding the significance map translates into a significant improvement in compression.

4.2.1 Significance Map

To accomplish the binary significance map, a data structure called "*zerotree*" or "*list of insignificant set*" is defined. The *zerotree* is based on hypothesis that if a wavelet coefficient at a coarse scale is *insignificant* with respect to a threshold *T* (its magnitude is less than or equal to *T*), then all wavelet coefficients in the same spatial location at finer scales are likely to be *insignificant* with respect to *T*. More specifically, in a wavelet decomposed system, with the exception of the highest frequency subbands, every coefficient at a given level can be related to a set of coefficients at the next finer level of similar orientation. The coefficients at the coarse scale are called the "*parent node*", and all coefficients corresponding to the same spatial-temporal location at the next finer level of similar orientation are called the "*child nodes*". For a given *parent node*, the set of all coefficients at all finer scales of similar orientation corresponding to the same location are called *descendants*. Similarly, for a given child node, the set of coefficients at all coarser scales of similar orientation corresponding to the same location are called *ancestors*. For a wavelet subband decomposition, the parent child dependencies are shown in Figure 4.1.

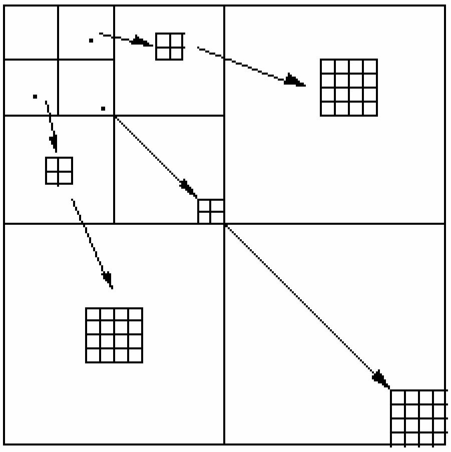


Figure 4.1 Parent-Child Tree/Set

Scanning of tree coefficients is performed in such a way that no child node is scanned before any of its parent nodes. Given a threshold level to determine whether or not a coefficient is significant, a node is said to be a "*zerotree root*" if (1) the coefficient has insignificant magnitude, (2) the node is not the *descendant* of a *root*, i.e. it is not completely predictable from a coarser scale, and (3) all of its descendants are insignificant. A *zerotree root* is an indication that the *insignificance* of the coefficients at finer scales is completely predictable. Zerotrees reduce the cost of encoding the binary significance map using self similarity. The major advantage of using zerotrees is that as the subband in which a zerotree root occurs, increases in coarseness (decrease in frequency), the number of predicted coefficients increases exponentially. Non-zero details can be isolated by immediately eliminating large insignificant regions from consideration.

4.2.2 Hierarchical Quantizer

The hierarchical quantizer takes the coarse to fine philosophy that is inherent in the wavelet transform representation and applies it to amplitude. Several quantization stages, each with a different threshold, are used as illustrated in Figure 4.2. Coefficients whose magnitudes are larger than the threshold are considered significant. The key to achieving an embedded coding scheme is to start with a large threshold, not smaller than half the magnitude of the largest coefficient, and code the binary significance map using the zerotree structure. The generation of the portion of the bit stream for binary significance map is called the "*dominant/sorting pass*". In order to aid embedding, it is useful to include the sign of significant values in the portion of the bit stream for the significance map.

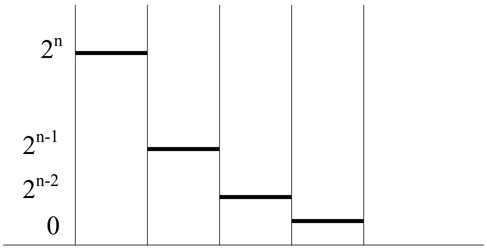
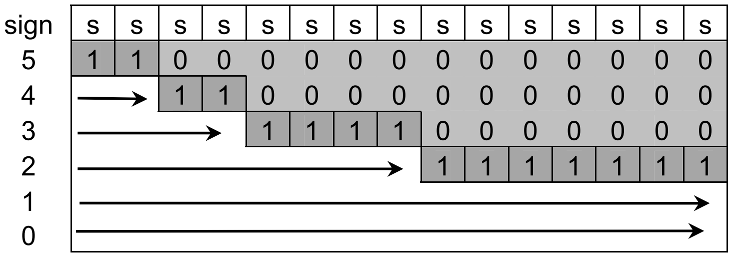


Figure 4.2 Four-Stage Quantizer

At each successive quantization stage, the threshold size is halved and those coefficients previously found to be significant are scanned again, whereby an additional bit of precision is encoded. The generation of the portion of the entropy coded bit stream that adds precision to a coefficient already known to be significant is called the "*subordinate/refinement pass*". After the subordinate pass is complete, those coefficients not yet found to be significant are processed by another *dominant pass* at the new smaller threshold. The process continues until the allocated bits (the bit budget) are used.

4.2.3 Order of Bits

Information in the bit stream is intended to be encoded in the order of importance. Coefficients of the WT are coded using a multi-stage coding system, where each stage uses a progressively smaller threshold to determine the *significance* as discussed in the previous section. At each smaller threshold, new entries of coordinates of significant coefficients are appended to a list such that coordinates of significant coefficients found at one threshold always precede the coordinates of significant coefficients found at smaller thresholds. As a result, the portion of the bit stream corresponding to a *subordinate pass* contains bits arranged in the order of importance. Refinement of coefficients occurs in the order determined by the list. Thus, coefficients that were found to be significant in the first *dominant pass*, i.e., those with larger magnitudes, are always refined in *subordinate passes* before coefficients that were first found to be significant in the following *dominant passes*. The binary representation of magnitude ordered coefficients (MOC) is illustrated in Figure 4.3.



sign	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	→		1	1	0	0	0	0	0	0	0	0	0	0	0
3	→				1	1	1	1	0	0	0	0	0	0	0
2	→								1	1	1	1	1	1	1
1	→														
0	→														

Figure 4.3 Magnitude Ordered Coefficients

The initial ordering of coefficients in the order of importance is set by the convention and is known to both the encoder and the decoder. In general, coefficients with larger magnitudes are supposed to be more important than coefficients with smaller magnitudes, regardless of the scale in which the coefficients occur. For two coefficients having equal reconstruction magnitudes the relative order of importance is determined by the original convention, whereby coordinates of the lower frequency (coarser resolution) coefficients precede coordinates of the higher frequency (finer resolution) coefficients. Information identified at a larger threshold in an earlier pass, including significance map entries and refinements of significant coefficients, is always encoded in the bit stream before information identified at a smaller threshold in a later pass. Since prior to the first dominant pass, the reconstruction values are all initialized to zero as shown in the upper right corner of the bit plane in Figure 4.3. The encoding can cease at any time and the resulting bit stream contains all lower rate encodings. This ability to cease encoding anywhere is extremely useful in systems that are either rate-constrained or distortion-constrained [18].

4.3 Encoding of Spatially Decomposed Video

The spatial only decomposition is illustrated in Chapter 2 while studying the 2-D WT of the "*Lena*" image. To encode the spatially decomposed video, the GOF structure is treated as separate images regardless of the temporal relation. Following the 2-D WT of the video signal, the encoding is performed on individual subband frames in the spatial domain. The encoding process employs an embedded image coding technique, called 2-D SPIHT. The SPIHT algorithm makes the bit plane ready for the transmission as illustrated in Figure 4.3. There are three outputs to the decoder in the encoding process: (1) the maximum bit count needed to hold the maximum wavelet coefficient, (2) the number, coordinates, and signs of coefficients for the current quantization step, (3) the refinement data bits belonging to the coefficients whose information are previously sent to the decoder. The first output determines the depth of the bit plane and thus helps initialization of the magnitude ordered coefficients (MOC) matrix for the decoder. Information about the coefficients which are subject to transmission in the next "*refinement pass*" is sent to the decoder as the second output. Since a special sorting algorithm is used to output the information, it is called "*sorting pass*". Following that information, the refinement data bits are sent. They belong to the coefficients whose sorting information is transmitted in the previous *sorting pass*. A detailed description of the encoding algorithm is explained in Appendix C.

In practical implementations, output bits are usually sent to the receiver via a transfer channel. To simulate a real world application, we saved the output bits in the text files. Three different text files are created to keep track of three different outputs for each frame in the GOF structure. Therefore, the encoding process ends up with 24 files (8 frames × 3 files / frame) consisting of all output bits. Keeping the output bits in separate files helps better analyzing the test results. The maximum bit count is especially useful in realization of the decomposition level. In Figures 3.3 and 3.19, we pointed out the maximum available coefficients at different decomposition levels. A pixel with a gray scale value of 255 contributes to the creation of a wavelet coefficient with a maximum magnitude of $[(\sqrt{2})^2]^{l} \times 255$ at the desired level, *l*. This, in turn, leads to an increase of the number of bits for the greatest magnitude (in the transformed signal). The depth of bit plane increases by 1 recursively at each decomposition level. While the depth increases, the length of the refinement bit packages (arrows in Figure 4.3) becomes shorter. This also shows that the signal de-correlation at deeper levels improves. More detailed features are exploited and the energy of the signal is retained by the coefficients with greater magnitudes.

The most significant data bits are not sent to the receiver. The maximum bit count together with the information about the number of coefficients for the current quantization step does actually provide the receiver with that information. The refinement bits include information about all bit positions of a coefficient except the most significant one. The coordinates of the coefficients are not explicitly sent to the receiver either. Instead of actual location information, the significance information is provided. The encoder scans a parent-child tree set and investigates the significance information for a coefficient in that tree. It transfers the comparison results with respect to the current quantization step to provide the information about the location of a significant coefficient. The sign information for significant coefficients is explicitly sent together with the significance information in the *sorting pass*. While the bit value 1 is sent out for negative valued coefficients, 0 is sent out for the positive ones.

The *sorting pass* is followed by the *refinement pass* where the data bits are prepared to be sent in the order of importance. The data bits prepared for the current quantization step belong to all magnitude ordered coefficients sorted in all previous *sorting passes*. For example, if the current quantization step is 2^4, the data bits at the 4th bit location (from the least significant position, 0) of all coefficients greater than or equal to 2^4 are transferred. If the encoding is ceased after sending all outputs for the current quantization step (2^4), all coefficients in the bit plane will be lack of bit values whose positions range from 0 to 3. While the coefficients whose magnitudes are less than 2^4 (they are subject to sorting in the next *sorting passes*) remain 0 as initialized (in the bit plane) at the beginning of the encoding, all coefficients which have been sent so far are lack of refinement information existing from the first 3 bit positions.

The above mentioned method is applied to the two test videos, "*Table Tennis*" and "*Conference*" introduced in Chapter 3. Both videos have QCIF resolution and a total number of 8 frames × 144 × 176 pixels / frame × 8 bits / pixel = 1622016 bits. Two test cases are generated related to the wavelet filters used in the spatial decomposition: (1) the encoding of videos transformed by Haar filters, and (2) the encoding of videos transformed by Daubechies-9/7 filters. The number of bits in the compressed form is calculated by summing bits spent to yield all outputs of the 2-D SPIHT. Because the maximum bit count for the maximum coefficient is much smaller than the total number of bits generated while the *sorting* and *refinement passes*, we ignored its contribution to the total number of bits in the compressed form. Therefore, C is the summation of bits only in *sorting* and *refinement passes*.

The encoding is ceased at different quantization steps, where the threshold values of 2^0, 2^4, 2^5, and 2^6 are applied to realize different compression ratios. The number of bits spent to encode each frame is measured in both test cases. To avoid messiness, data is only collected in the first and second level decompositions, which even ended up over 200 measurements. The results of the "*Table Tennis*" video are summarized in Tables 4.1 and 4.2. The C_1/O is the ratio of the number of *refinement bits* to the total number of bits in the original form, and C_2/O is the ratio of the total number of bits spent in both *refinement* and *sorting passes* to the number of bits in the original form. All threshold values except 2^0 are selected so as to reconstruct the video at around a PSNR of 30 db. Corresponding to the perfect reconstruction, all data bits excluding the zero valued coefficients are sent to the receiver in the case of the threshold value 2^0.

Table 4.1 Total Refinement Bit Count ("*Table Tennis*", 2-D only)

filter	threshold (≥)	level-1	C_1/O	level-2	C_1/O
Haar	2^0	597326	0.368	402281	0.248
	2^4	185161	0.114	66303	0.040
	2^5	131464	0.081	48375	0.029
	2^6	79789	0.049	33619	0.020
Db-9/7	2^0	620323	0.382	437831	0.269
	2^4	188986	0.116	73941	0.045
	2^5	132742	0.081	52775	0.032
	2^6	79163	0.048	34555	0.021

Table 4.2 Total Sorting Bit Count ("*Table Tennis*", 2-D only)

filter	threshold (≥)	level-1	C_2/O	level-2	C_2/O
Haar	2^0	804107	0.864	734194	0.700
	2^4	344463	0.326	170519	0.146
	2^5	286434	0.257	108001	0.066
	2^6	241098	0.197	80450	0.070
Db-9/7	2^0	900979	0.937	852186	0.795
	2^4	404353	0.365	230411	0.187
	2^5	341377	0.292	161643	0.132
	2^6	286802	0.225	124196	0.097

Although we take only the luminance components of the test videos into the encoding process, the original signals have QCIF resolution in 4:1:1 YUV system. The color information carried by the two chrominance components (U&V)

is totally ignored. The color information can be regarded as spectral redundancy and its elimination results in a compression ratio of 1:1.5. All tables showing the bit counts and compression ratios, C_i/O, including Tables 4.1 and 4.2 do not reflect the contribution of the elimination of the spectral redundancy.

The total bit counts in Tables 4.1 and 4.2 refer to the number of bits used to encode 8 frames. The results give us information about the effect of decomposition level, threshold value, and filter type on the signal compression. The *sorting* and *refinement* bit counts at level-2 are less than those at level-1 for all threshold values and filter types. At deeper levels, more coefficients and data bits remain below the current quantization step (threshold), yielding less refinement bits generated in the given *refinement pass*. Moreover, increasing the decomposition level causes more spatial orientations to appear at finer scales. This, in turn, increases the length of the parent-child tree. Correspondingly, the number of coefficients in a tree set also increases. Although it is not always a true assumption, the following statement is usually proven by experimental results: if a root (parent) of a tree set is found to be insignificant, its descendants are also insignificant. In the case that the assumption holds, the sorting algorithm works more efficiently since a great number of coefficients in the tree set is sorted out at once. Otherwise, more bits need to be used to sort individual coefficients in the tree in an inefficient way.

The threshold values show their effects more at deeper levels. The ratio of the bit counts at level-1 to those at level-2 increases as the threshold value increases. While the ratio is below 1:1.5 for the threshold value of 2^0, it reaches 1:3 when threshold is set to 2^6. As the threshold increases, less coefficients are selected to be significant, which decreases the number of bits spent to sort and refine the coefficients. The higher the threshold value is, the lower the total number of bits spent for *refinement* and *sorting passes*. Although the compression ratio increases in parallel with the increase of the threshold value, the reconstructed signal suffers from the lack of refinement information as will be seen in the decoding process. Unfortunately, the majority of the total bit budget is used to determine the significance map in *sorting passes* at all threshold values. Although it is not a common occurrence, we observed that the encoding method spends more bits to sort out a single significant child coefficient among all the other insignificant ones in the tree set. Therefore, the number of bits spent in *sorting passes* results in approximately 2 or 3 times that of the *refinement passes*.

For a given threshold, both the sorting and refinement bits in the test case (2) are more than those in the case (1). There are two reasons affecting the total number of bits spent in the Daubechies-9/7 case adversely: (1) the increased number of coefficients after the decomposition by Daubechies-9/7 filters, and (2) lack of spatial orientation existing in the spatially decomposed videos by the Daubechies-9/7 filters. In Chapter 3, we mentioned about the newly introduced wavelet coefficients around the image (frame) boundary after spatially decomposed video signals. The increased number of coefficients is directly related to the increase in both sorting and refinement bits. The length of the refinement bit package in a pass (arrows in Figure 4.3) increases in parallel with the increase of the width of the magnitude ordered bit plane. Similarly, the number of sorting bits also increases in parallel with the increase in the number of coefficients. Supplementary bits are spent to sort out additional coefficients.

The lack of spatial orientation is another reason affecting the efficiency of two-dimensional SPIHT algorithm adversely on spatially decomposed videos by the Daubechies-9/7 filters. The offspring of a parent node is determined by a group of four coefficients lying at the same spatial orientation at the finer level. That way, all descendants are determined and all coefficients in all subbands are contained in the predefined parent-child tree sets once the leaf nodes in the finest scale are included. Although constructing the tree sets is simply achieved in the Haar case, the procedure is not applicable straightaway for the Daubechies-9/7 case. Since the matrix dimensions increase unevenly after the decomposition by Daubechies-9/7 filters, defining a tree set with the coefficients lying in the same spatial orientation at finer levels gets difficult. Especially the coefficients around the coarsest subband frame boundaries suffer from the ownership of a tree set. Although the total refinement bit counts are not affected, the existence of suffering coefficients increases the number of total bit count generated in *sorting passes*.

The difference between the numbers of refinement bits in the two test cases gets smaller as the threshold value increases. Since the number of refinement bits is directly related to the number of coefficients, it can be noted that the number of coefficients around the threshold value of 2^0 is more in the case of spatially decomposed videos by Daubechies-9/7 filters than by Haar filters. This is in fact consistent with the inference in Chapter 3 that signals decomposed by Daubechies-9/7 filters yield more coefficients close to zero. This was experimentally revealed by retesting the entropy analysis with the "*Conference*" video in Chapter 3.

The "*Conference*" video has more spatial frequency components than the "*Table Tennis*". The Daubechies-9/7 filters are utilized to accomplish a better signal decorrelation for the "*Conference*" video in Chapter 3. Here, from the compression point of view, we compared the total bit counts for the *refinement* and *sorting passes* for the "*Conference*" video with that of the "*Table Tennis*". For this comparison, we only considered the test cases with threshold values 2^0 and 2^5. The resultant bit counts are summarized in Tables 4.3 and 4.4.

Table 4.3 Total Refinement Bit Count ("*Conference*", 2-D only)

filter	threshold (≥)	level-1	C_1/O	level-2	C_1/O
Haar	2^0	593752	0.366	437351	0.269
	2^5	105885	0.065	44328	0.027
Db-9/7	2^0	590302	0.363	436440	0.269
	2^5	105132	0.064	43909	0.027

Table 4.4 Total Sorting Bit Count ("*Conference*", 2-D only)

filter	threshold (≥)	level-1	C_2/O	level-2	C_2/O
Haar	2^0	823028	0.873	740901	0.726
	2^5	340580	0.275	177442	0.136
Db-9/7	2^0	913027	0.926	835404	0.784
	2^5	381953	0.300	206640	0.154

The results in Tables 4.3 and 4.4 give us similar information about the encoding of the "*Conference*" video with those derived for the "*Table Tennis*". The effect of decomposition level, threshold value, and filter type on the signal compression can be analyzed. The *sorting* and *refinement* bit counts at level-2 are less than those at level-1 for all threshold values and filter types. The threshold values show their effects more at deeper levels. The ratio of the bit counts at level-1 to those at level-2 increases as the threshold value increases. All these results are in parallel with the results in Tables 4.1 and 4.2.

A different output that needs attention is obtained about the performances of filter types. The gap existing between the numbers of refinement bits for both test cases (with Haar and Daubechies-9/7 filters) for the "*Table Tennis*" video diminishes for the "*Conference*" video. The refinement bit count in the second test case of the "*Table Tennis*" video is, in general, greater than those in the first test case at all threshold values and decomposition levels. However, the refinement bit count in case (2) of the "*Conference*" video falls behind that of case (1). Because the decrease in the number of refinement bits is not much and just slightly less than those in case (1), the ratios indicated by C_l/O are almost the same for both test cases.

Another point to note about the numbers of the refinement and the sorting bits between the "*Table Tennis*" and "*Conference*" videos is the less number of refinement bits and the more number of sorting bits in Tables 4.3 and 4.4. Because the wavelet decomposition of the latter test video is not as successful as the "*Table Tennis*", the entropy of its wavelet coefficients is higher than that of the "*Table Tennis*" video. Since the accumulation of the coefficients around zero is not much (the energy concentration in the coarsest scale is less), the length of the refinement bit packages (arrows in Figure 4.3) is less at higher threshold values. This can be observed by the bit counts at the threshold values of 2^5 in Tables 4.1 and 4.3.

Inherently, the WT supports the ordering of coefficients. The creation of coefficients with higher magnitudes in finer scales makes the sorting algorithm run inefficiently for the "*Conference*" video. The sorting algorithm suffers from the unevenly distributed wavelet coefficients while placing them in the order of importance.

Experiments show that the time estimation for encoding a test video is around 5 seconds in “Debug” mode over a Pentium® 4 processor with a speed of 1.6 GHz. An appropriate way to estimate processing times is to select the least time delay as reference and to estimate all relative times in seconds. Since the selected reference time is the same for all relative time estimations including those in Tables 3.2 and 3.10 in Chapter 3, the estimations summarized in Tables 4.5 and 4.6 are also comparable with the wavelet decomposition times in Chapter 3. The results given in Tables 4.5 and 4.6 are the relative times needed to complete the encoding of both test videos at different threshold values. By “*encoding*”, we only mean the total processing times for all *sorting* and *refinement passes*. The estimations indicate the sum of all processing times spent to encode the 8 frames.

Table 4.5 Relative Encoding Times (“*Table Tennis*”, 2-D only)

filter	threshold (≥)	level-1	level-2
Haar	2^0	1.60	1.66
	2^4	1	1.13
	2^5	0.9	1
	2^6	0.5	0.7
Db-9/7	2^0	1.35	1.25
	2^4	0.7	0.75
	2^5	0.46	0.60
	2^6	0.36	0.36

Table 4.6 Relative Encoding Times (“*Conference*”, 2-D only)

filter	threshold (≥)	level-1	level-2
Haar	2^0	1.46	1.86
	2^5	1	1.13
Db-9/7	2^0	1.33	1.40
	2^5	0.52	0.62

The threshold values shown in Tables 4.5 and 4.6 are the most recent values that the current quantization step have as the encoding continues. The duration that the encoding lasts is determined by the selected threshold value where the encoder is presumed to cease running. The relative times decrease gradually as the threshold value increases from 2^0 to 2^6. As an example, the time spent to encode the video at threshold value of 2^0 is the maximum and about 5 times that at 2^6. Another point to note is that the time is mostly spent in *sorting passes*. The gap between the instruction sets of the sorting and refinement phases is too large such that the times needed to accomplish the encoding are mostly determined by the processing times in the *sorting passes*.

The increased processing time at level-2 seems to contradict with the results given in Tables 4.1 - 4.4 where the number of bits generated at level-2 is less than those at level-1. However, the time difference between the two levels is mostly related to the way we implement the SPIHT algorithm. There are additional instructions to dynamically allocate variables for additional offspring and children to complete the parent-child tree at level-2.

4.4 Decoding of Spatially Decomposed Video

The decoding process is exactly the same as the encoding process in reverse order. The decoder follows the same scanning method of coefficients with the encoder. As the data bits are accepted, they are placed into the proper location to make the coefficient matrix ready any time for the reconstruction of the signal.

The resolution of the original video signal is deterministic to both encoder and decoder at the beginning of the transfer. To illustrate, we used "*Table Tennis*" and "*Conference*" videos which have the QCIF resolution with the dimensions of 8 frames × 144 × 176 pixels / frame and 8-bit gray level pixel values. Besides the signal characteristics, the wavelet filter type used in the decomposition and synthesis processes are also known to both coders. The information about the video signal and the wavelet filters is crucial for the decoder since it allocates the required memory space to maintain the wavelet coefficients prior to the transfer. The first output of encoder is the maximum bit count for the maximum wavelet coefficient. It determines both the initial threshold value of the current quantization step and the shape of the magnitude ordered coefficient matrix as illustrated in Figure 4.3. For example, if the magnitude of the greatest coefficient is $[(\sqrt{2})^2]^2 \times 255$ (= 1020), the threshold value is set to 2^9 (= 512) with a maximum bit count of 9 excluding the sign bit information. The initial setting is for both encoder and decoder. Once the depth of the MOC matrix is known to the decoder, the shape of the matrix can be determined from size of the total memory.

The decoder resets all elements (coefficients) in the MOC matrix to zero before receiving any information about the significance map and the coefficient data itself. While receiving the second output of the encoder, the decoder scans the same parent-child tree set and the coefficients in it as the encoder and accepts their significance information. It receives 1 if the scanned tree set or a coefficient inside is a significant one, otherwise 0. That way, all coefficients mapping the current quantization step (found significant for the current threshold value) are sorted out by the decoder. Besides the significance information, it also accepts the sign information for the significant coefficients. The decoder knows that the negative and positive values are indicated by 1 and 0, respectively. After the significant coefficients are sorted out for the current (and initial) quantization step, the decoder replaces the most significant bits of the significant coefficients in the MOC matrix. For the maximum bit count (9) given in the illustration above, all coefficients greater than or equal to 2^9 are sorted out and their number of occurrence are determined together with their sign information. Using that information, the decoder places 1's to the most significant bit position, 9, in the MOC matrix which is previously initialized to 0 at the very beginning.

The refinement bits are not expected for the initial quantization step. After performing the *sorting pass* for the initial quantization step, the decoder updates its quantizer with the new threshold value simultaneously with the encoder. Continuing with the example above, the maximum bit count is decremented by 1 and the new threshold value becomes 2^8 (= 256) for the current quantization step. For this new quantization step, the *sorting pass* is repeated to sort out new significant coefficients in the interval $[2^8, 2^9)$ and to output their signs. The same as the previous *sorting* *pass*, the decoder places 1's to the most significant bit location, 8, of the new significant coefficients in the MOC matrix together with their sign bits. There is an important distinction that needs attention about the sorting algorithm. The 2-D SPIHT does sort out the coefficients mapping the current quantization step but not places them in the magnitude order. Considering the example above, all wavelet coefficients are processed concerning the two magnitude intervals: $[2^9, 2^{10})$ and $[2^8, 2^9)$. The sorting algorithm investigates the coefficients if they belong to either of the intervals regarding their magnitudes. That way, the coefficients are somehow sorted but there is no clear sorting mechanism to place the coefficient in either interval in the order of magnitudes. Therefore by "*sorted out*", we mean the determination of the coefficients that only map to the current quantization step.

Once performing the *sorting pass* for all parent-child tree sets, the *refinement pass* is executed for the current quantization step (threshold value). In our example above, the first *refinement pass* is only performed after the *sorting pass* is completed for the interval $[2^8, 2^9)$ (or, for the threshold value, 2^8). The received refinement bits belong to the coefficients that are contained in $[2^9, 2^{10})$. These coefficients are sorted out in the previous *sorting pass* and the decoder does only have their most significant bit values up to now.

The sorting and refinement bits are accepted as the *sorting* and *refinement passes* are performed repeatedly for each quantization step. At each repetition, the maximum bit count is decremented by 1. When it becomes 1, threshold value at the current quantization step is set to 2^0, and the coefficients with unity magnitudes are sorted out. The refinement bits at the 0th location of the coefficients in the interval, $[2^1, 2^2)$ are placed in the MOC matrix. They are the last output bits transferred by the encoder. All the remaining elements (bits for the coefficients) in the MOC matrix have zero values as initialized at the beginning. The encoder neither spends any bits for the zero valued coefficients nor transfers extra bits for the significant ones. After the transmission is completed, the decoder has all the bit information for all coefficients and the location information of each coefficient in the MOC matrix.

Both the encoder and the decoder may stop running at any quantization step. It is the encoder's concern that the execution should be stopped at any moment so as to achieve an expected signal compression or a low bit rate in the transfer channel. On the other hand, the decoder's concern is about the reconstructed signal quality and the memory capacity. In fact, it is mostly the decoder's choice to reconstruct a video signal at desired quality. The decoder is only able to assure the quality of the reconstructed signal to be equal to that of the compressed signal prior the transmission in the best case when it executes exactly the same encoding procedure in reverse order. Once the MOC matrix is filled with the required bit information, the coefficients are moved to its exact location in the subband frames. To do so, the decoder uses the location information of each coefficient already available since the *sorting passes*. Following the coefficients' replacements, the reconstruction of the signal is performed by the inverse WT.

In Tables 4.7 and 4.8, we summarized the average PSNRs after the 2-D wavelet synthesis operation for the two test videos, "*Table Tennis*" and "*Conference*", whose encoding results are given in the previous section. While the former video is reconstructed at 4 different threshold values (2^0, 2^4, 2^5, 2^6), the latter is only reconstructed at the thresholds 2^0 and 2^5. The reconstruction test cases are the same as the encoding test cases: (1) the reconstruction of the video using Haar filters and (2) reconstruction of the video using Daubechies-9/7 filters.

Table 4.7 Average PSNR ("*Table Tennis*", 2-D only)

filter	threshold (≥)	level-1	level-2
Haar	2^0	54.58	56.24
	2^4	32.01	32.86
	2^5	27.58	29.56
	2^6	23.02	26.07
Db-9/7	2^0	58.80	58.72
	2^4	32.05	32.67
	2^5	27.62	29.35
	2^6	23.06	25.74

Table 4.8 Average PSNR ("*Conference*", 2-D only)

filter	threshold (≥)	level-1	level-2
Haar	2^0	54.55	56.23
	2^5	26.75	27.75
Db-9/7	2^0	58.79	58.72
	2^5	26.71	27.65

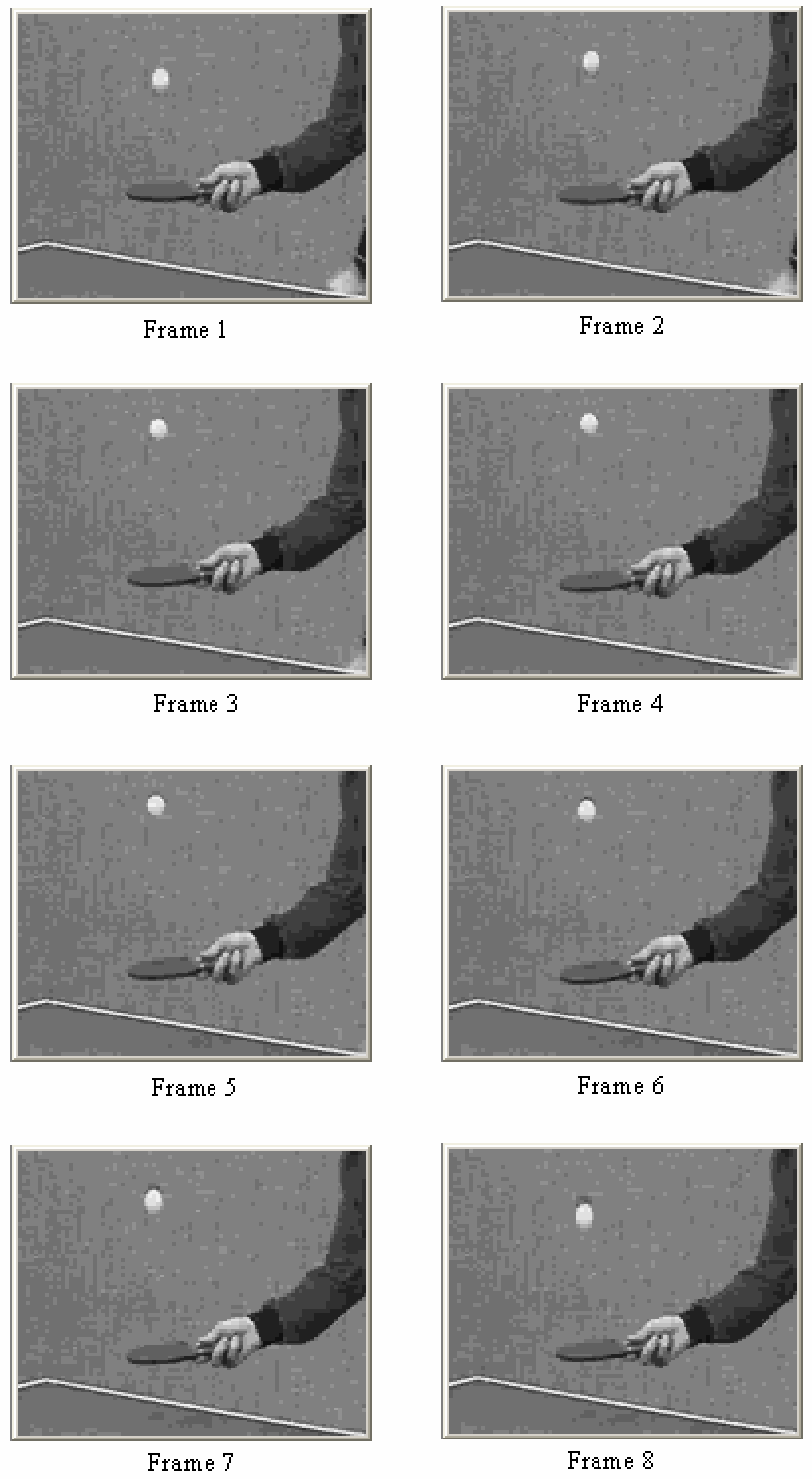
Two comments can be derived directly from the PSNR results: (1) the signal quality (PSNR) increases when decoding starts from a deeper decomposition level, and (2) the increased threshold value leads to the information loss, causing a worse reconstruction. The decomposition levels affect the quality of the reconstructed signals. There is a significant improvement at level-2 in the signal quality (compared to level-1). The improvement is up to 3 dB and the best improvement is achieved at the threshold value 2^6 for the first test case of the "*Table Tennis*" video. The higher *compression ratios* together with the fact that the signal quality improves significantly when the reconstruction is performed at deeper levels do indicate the great contribution of the decomposition level to the employment of the wavelet based video coding techniques.

The highest PSNR is obtained in case of 2^0, corresponding to the perfect reconstruction. Although the infinitely perfect reconstructions are expected for the case of 2^0, a measurable signal to noise ratio is obtained due to the information loss in two phases: (1) at the beginning of the encoding procedure, and (2) at the end of the synthesis procedure. A straightway to prepare the magnitude ordered bit plane is to round the floating valued wavelet coefficients to the nearest integers at the initialization phase of the encoding. Although it is not compulsory to round the coefficients, it eases the implementation and omits dealing with the definition of a decimal type common to both encoder and decoder.

Once the synthesis operation is completed, another rounding is performed on the pixel values. The wavelet synthesis operation generates the floating valued coefficients (pixel values) of the reconstructed signal. The rounding is necessary to represent the reconstructed video signals in 8-bit gray scale resolution. Since the information loss due to the rounding is not much, the PSNRs are very high. There is a PSNR difference between the case of 2^0 and the rest of the threshold values.

In general, the PSNRs for the two test cases are close to each other. The maximum difference of approximately 0.3 dB occurs at the threshold value of 2^6 for the Haar case. The advantage of the Daubechies-9/7 filters is only realized at the threshold value of 2^0 where the improvement for the reconstructed videos is about 3 db. However, the reconstruction around 58 dB by Daubechies-9/7 filters does not have a significant advantage from the compression and the observer's point of view: The compression ratio is the smallest for the case of 2^0. Moreover, the quality difference among the videos reconstructed with PSNRs over 35 dB is hardly perceivable by the human eye.

We only displayed the reconstructed videos decoded for the threshold case 2^5 for all test cases and videos in Figures 4.4 - 4.11. The smoothing properties of the Daubechies filters can be realized by the naked eye by comparing Figures 4.4 - 4.7 with 4.8 - 4.11, respectively. As opposed to Daubechies-9/7 filters, Haar filters cause rectangular regions to appear where spatial frequency changes occur in the reconstructed frames.



Frame 1

Frame 2

Frame 3

Frame 4

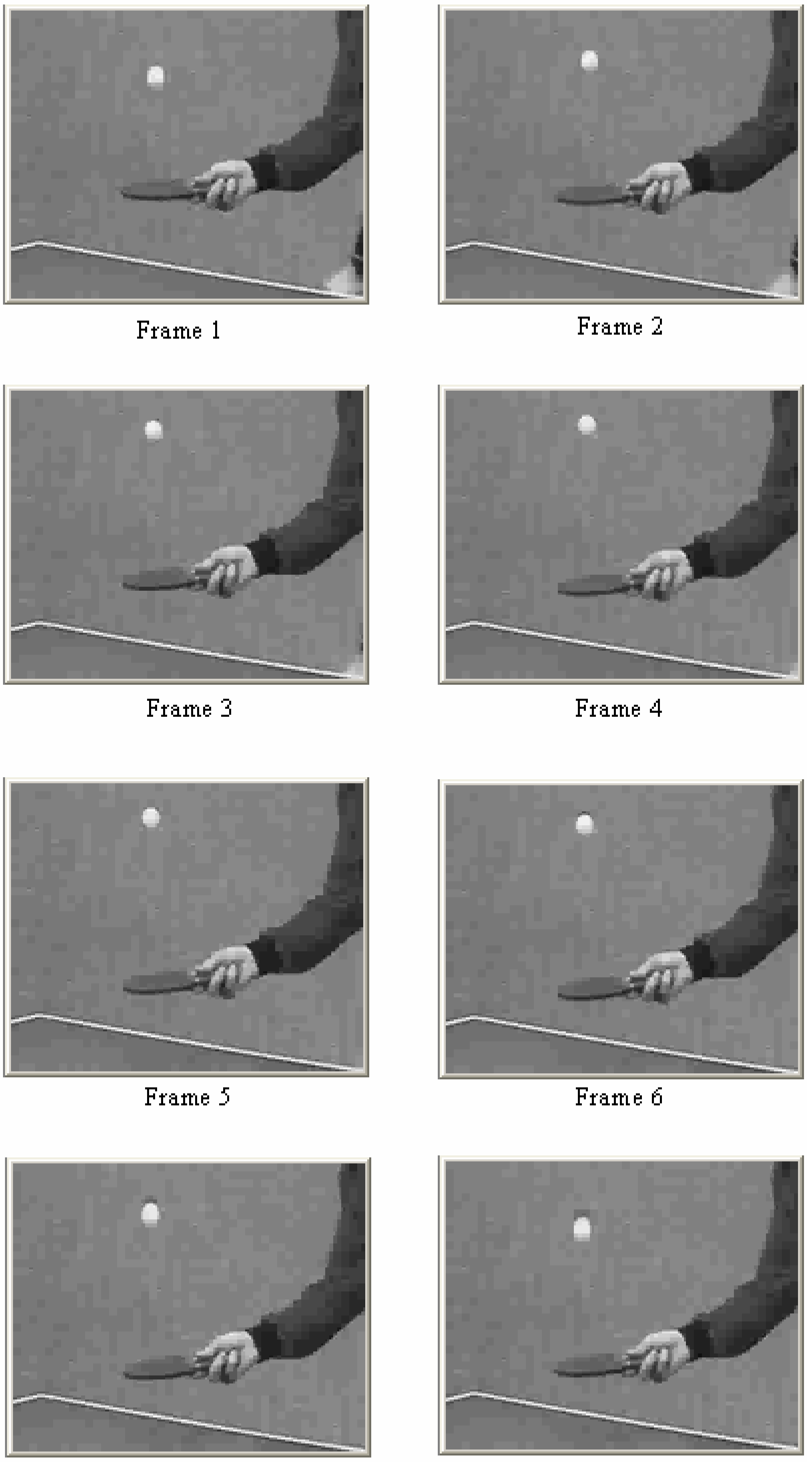
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.4 Reconstructed "*Table Tennis*" Video (2-D Haar, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

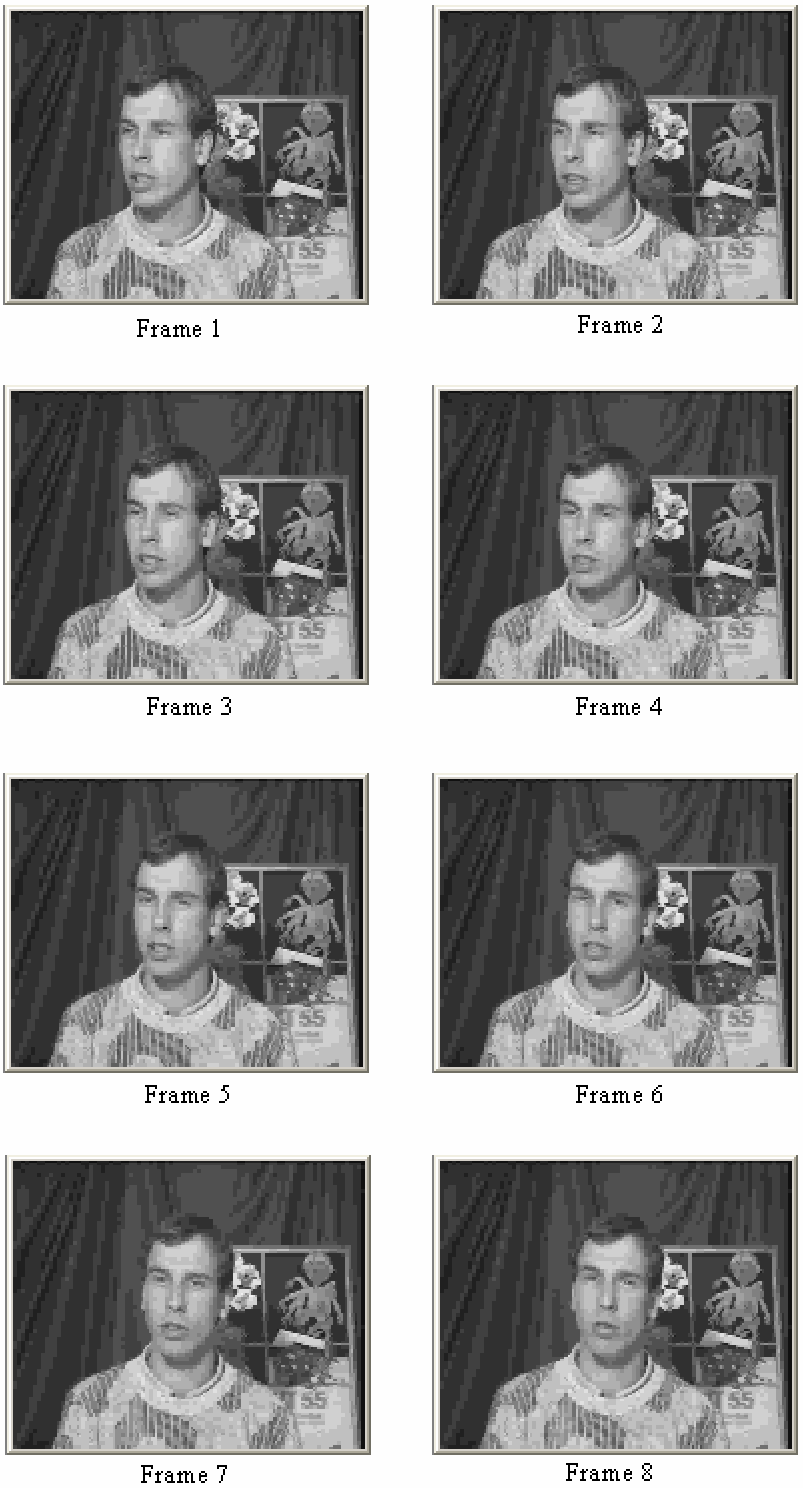
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.5 Reconstructed “*Table Tennis*” Video (2-D Haar, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

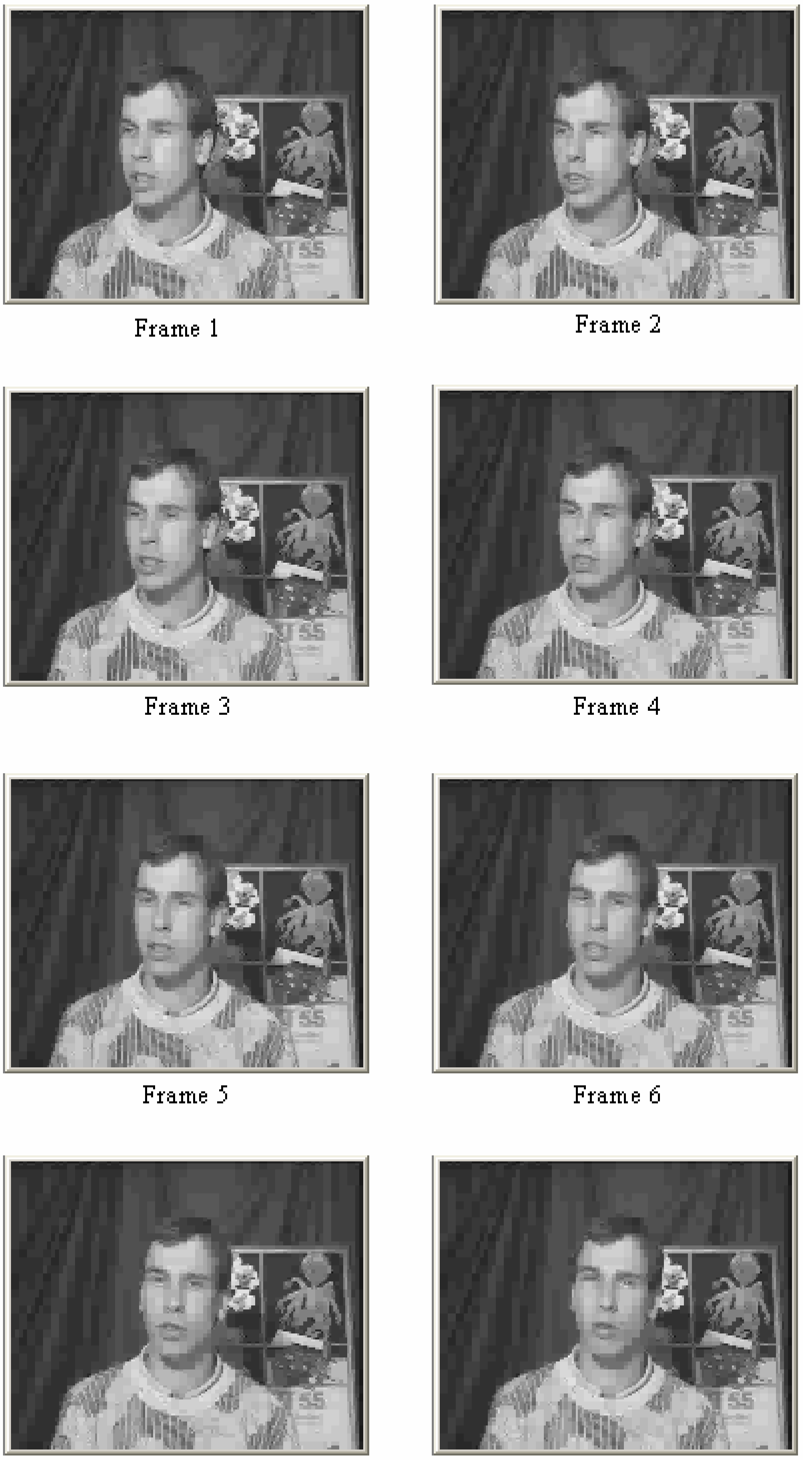
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.6 Reconstructed "*Conference*" Video (2-D Haar, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

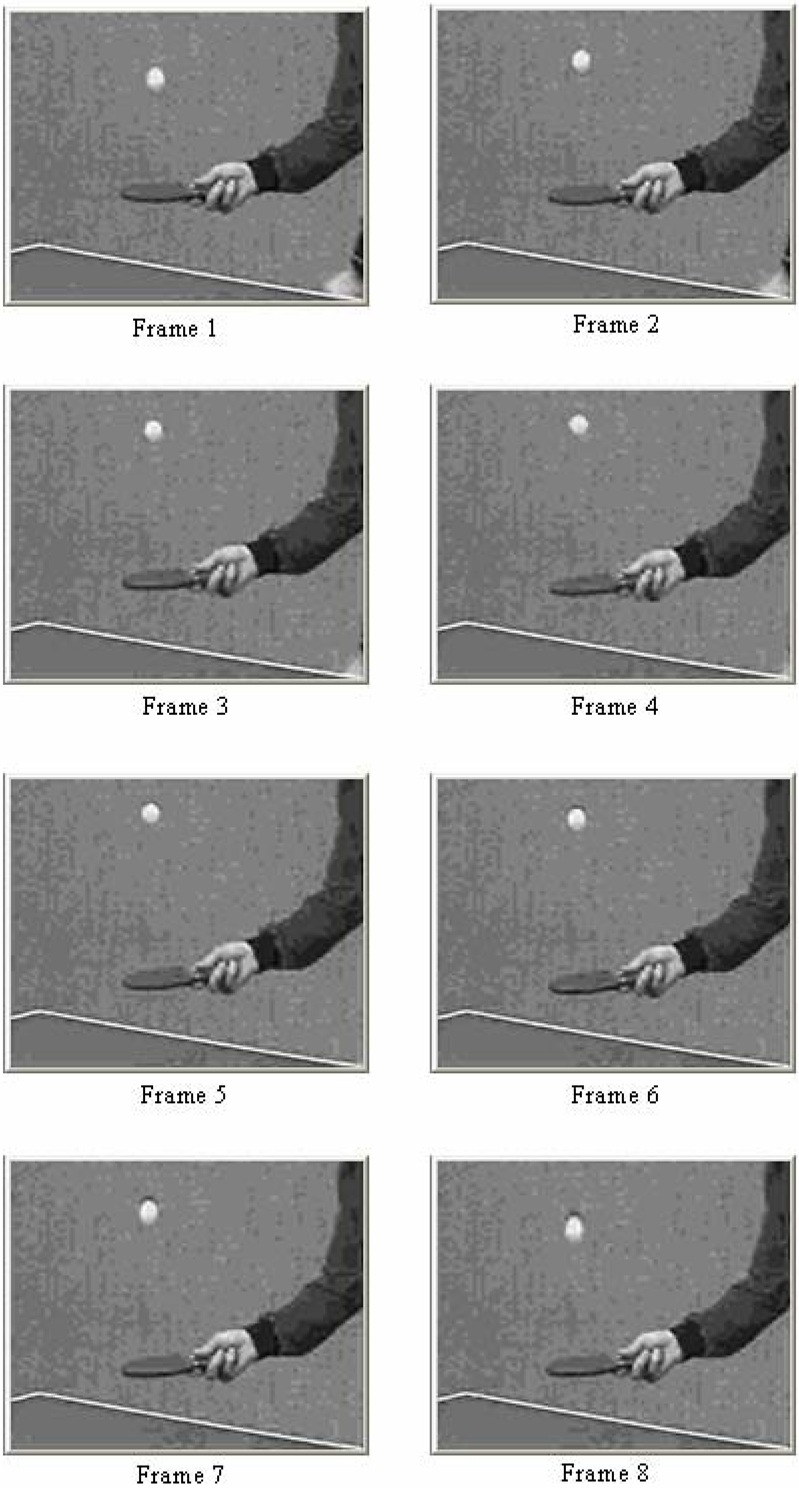
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.7 Reconstructed “*Conference*” Video (2-D Haar, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

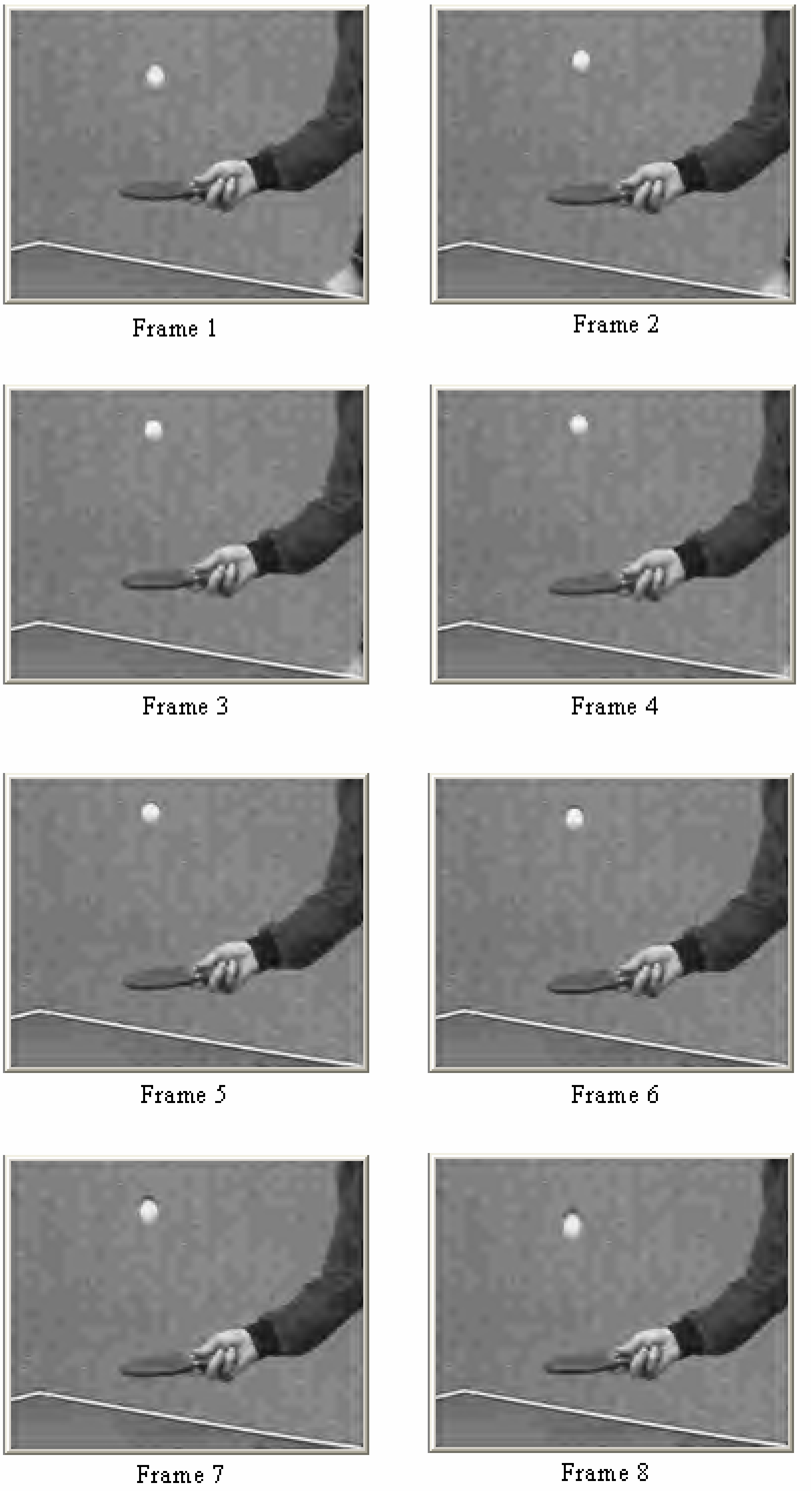
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.8 Reconstructed "*Table Tennis*" Video (2-D Db-9/7, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

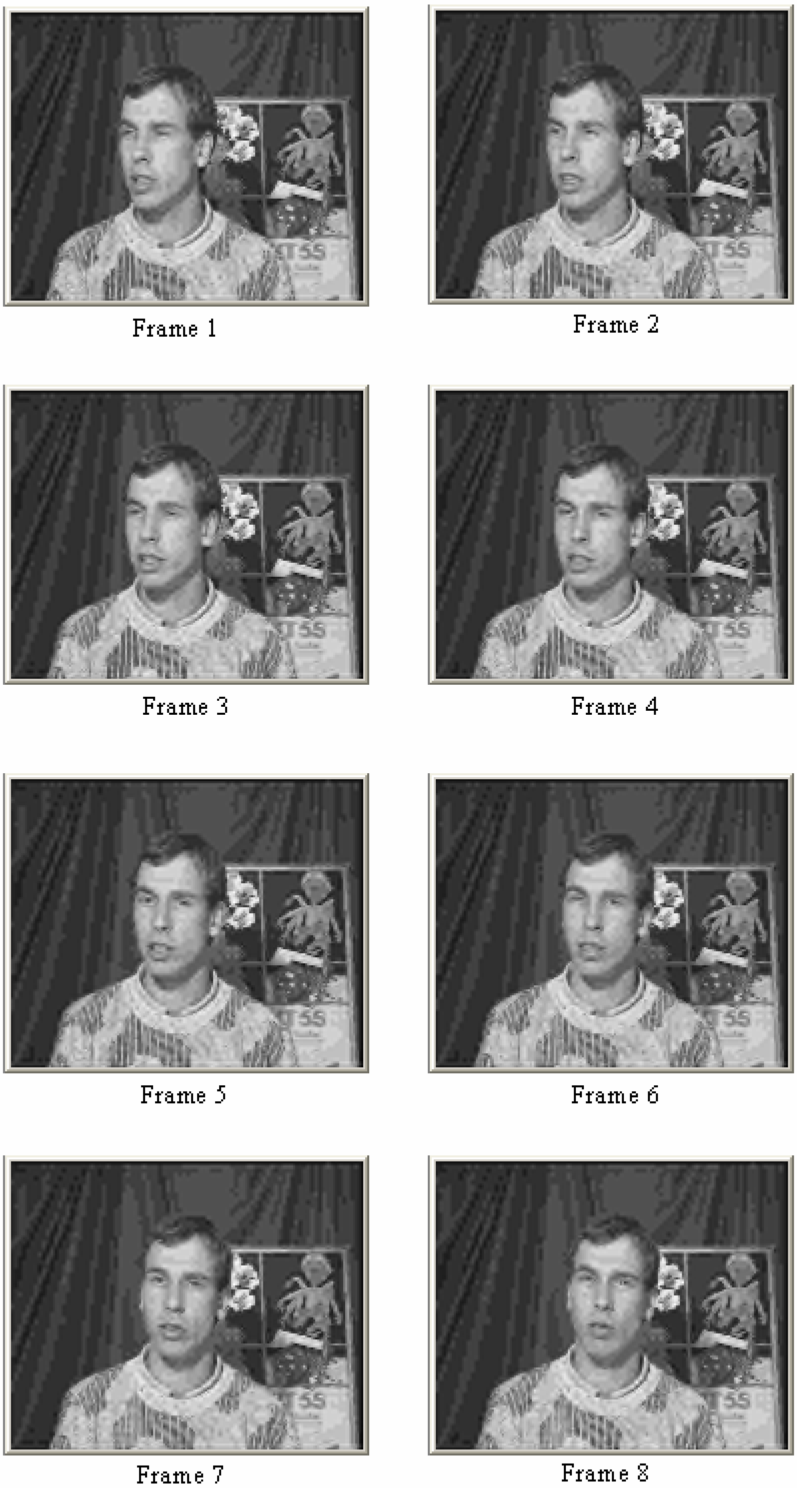
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.9 Reconstructed “*Table Tennis*” Video (2-D Db-9/7, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

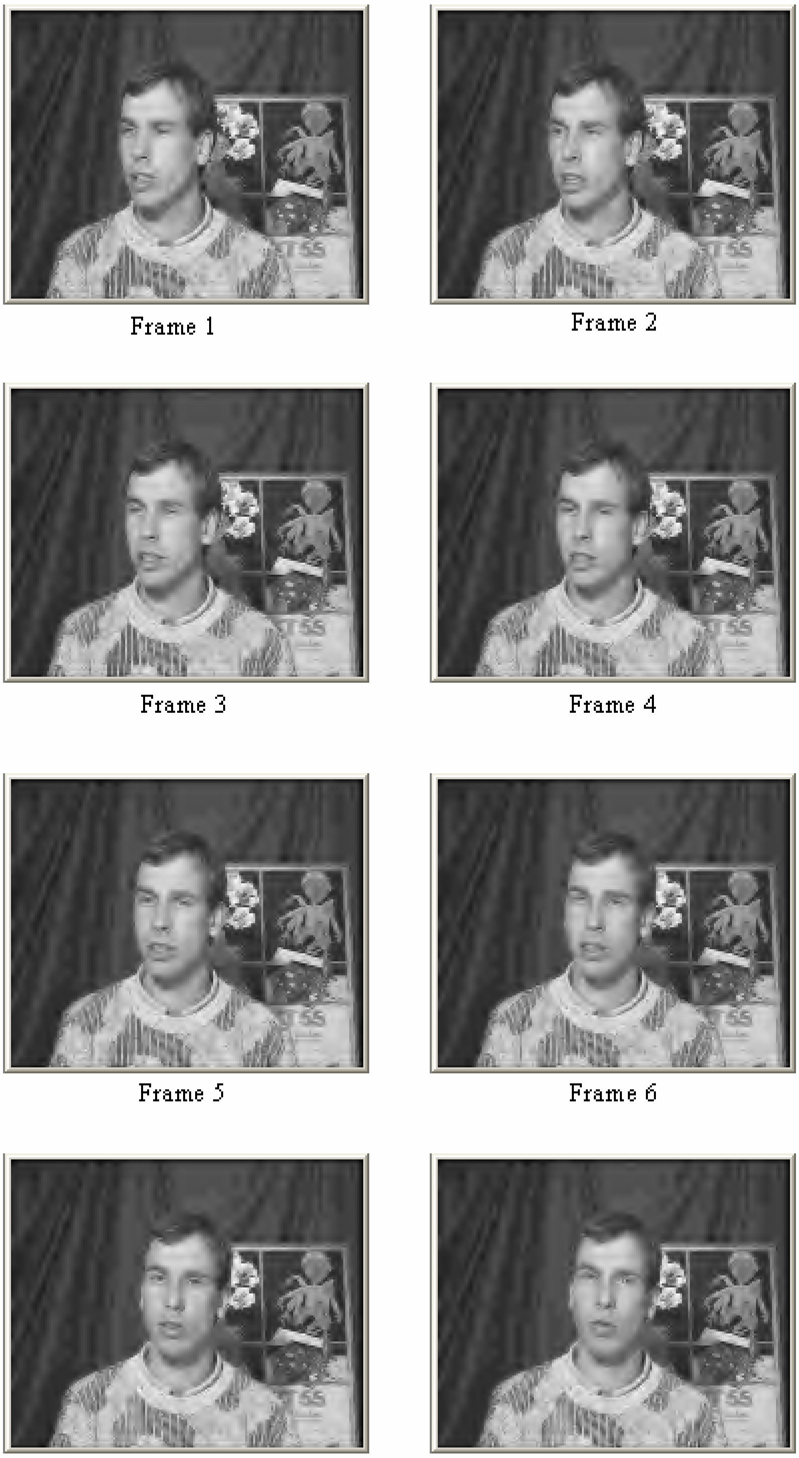
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.10 Reconstructed "*Conference*" Video (2-D Db-9/7, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.11 Reconstructed “*Conference*” Video (2-D Db-9/7, L-2, $T = 2^5$)

4.5 Encoding of Spatial-Temporal Decomposed Video

The spatial-temporal decomposition of the video signals is studied in Chapter 3, where a detail analysis of the transform coefficients, are performed for the two test videos, "*Table Tennis*" and "*Conference*". In Chapter 3, reconstruction of the video is performed using only a minority of the coefficients. This section deals with the encoding of the spatial-temporal decomposed video to indicate the practical representation of the compression ratio. The GOF structure of 8 frames is first three dimensionally decomposed. Following the 3-D WT of the video signal, the encoding is performed by employing an embedded image coding technique, called 2-D SPIHT (see Appendix C). Naturally, the 2-D SPIHT is applied to individual subband frames in the spatial domain. Since a three dimensional decomposition is performed, the temporal relation among the individual frames are also taken into account in the encoding procedure.

The application of the 2-D SPIHT algorithm to spatial-temporal subband frames is exactly the same as its application on the individual images (frames) mentioned in Section 4.3. An important point that needs attention is the different spatial levels to which the individual frames in the GOF structure are decomposed. A one-level 3-D wavelet decomposition of the GOF structure produces 4 spatial subbands for all frames. A further decomposition results in 4 spatial subbands newly generated in the temporal low subband frames. Thus, a two-level spatial-temporal decomposition of the GOF structure produces two types of subband frames considering the spatial decomposition: (1) two-level spatially decomposed temporal subband frames, and (2) one-level spatially decomposed temporal subband frames. Only the number of spatial decompositions that each frame has is taken into account when applying the 2-D SPIHT to a 3-D decomposed video. Different number of spatial decompositions changes the depth of the parent-child tree in the *sorting* and *refinement passes*.

No matter which decomposition level a frame has, the encoder produces the same outputs explained in Section 4.3. The output bits are saved in text files. Three different text files are created to store the three different outputs for each frame. The encoding process ends up with 24 files (8 frames × 3 files / frame) that contain all the information about the GOF structure.

Being an encoder output, the maximum bit count increments more at deeper levels of the 3-D decomposition compared to that in the spatial-only decomposition. For example, a pixel with a gray scale value of 255 contributes to the creation of a wavelet coefficient with a maximum magnitude of $[(\sqrt{2})^3]^{\,l} \times 255$ at the desired level, *l*. This, in turn, leads to an increase of the bit count for the greatest magnitude in the transformed signal. The depth of the bit plane increases by 2 at level-1, 3 at level-2 and 5 at level-3. While the depth increases, the length of the refinement bit packages (arrows in Figure 4.3) becomes shorter. This also shows the improvement for the signal de-correlation at deeper levels. More detailed features are exploited and the energy of the signal is kept by coefficients with greater magnitudes.

We applied the 2-D SPIHT technique to the "*Table Tennis*" and the "*Conference*" video whose encodings are already introduced in Section 4.3. There are four test cases related to the wavelet filters used in the spatial decomposition and motion compensation applied on the temporal decomposition. Four decomposition test cases are generated to apply the encoding process: (1) 3-D Haar decomposition, (2) MC 3-D Haar decomposition, (3) 3-D Haar & Daubechies-9/7 decomposition, and (4) MC 3-D Haar & Daubechies-9/7 decomposition. Because the maximum bit count for the maximum coefficient is much smaller than the total number of bits generated while the *sorting* and *refinement passes*, we ignored its contribution to the total number of bits in the compressed form.

The encoding is ceased at different quantization steps, where threshold values of 2^0, 2^4, 2^5, and 2^6 for the "*Table Tennis*" video and 2^0, 2^5 for the "*Conference*" video are applied to get different compression ratios. In all test cases, the number of bits spent to encode each frame is measured. Over 400 measurements are performed to collect the bit counts for both test videos. The results of the "*Table Tennis*" video are summarized in Tables 4.9 - 4.12 and those of the "*Conference*" video are summarized in Tables 4.13 - 4.16. The C_l/O is the ratio of the number of *refinement bits* to the total number of bits in the original form, and C_2/O is the ratio of the total number of bits spent in both the *refinement* and *sorting passes* to the number of bits in the original form.

Table 4.9 Total Refinement Bit Count ("*Table Tennis*", 3-D Haar)

case	threshold (≥)	level-1	C_1/O	level-2	C_1/O
No MC	2^0	371073	0.228	238914	0.147
	2^4	103904	0.064	25729	0.015
	2^5	75144	0.046	17206	0.010
	2^6	48661	0.030	12112	0.007
MC	2^0	367134	0.226	234787	0.144
	2^4	103225	0.063	24876	0.015
	2^5	75043	0.046	17101	0.010
	2^6	48612	0.029	12119	0.007

Table 4.10 Total Refinement Bit Count ("*Table Tennis*", 3-D H + Db-9/7)

case	threshold (≥)	level-1	C_1/O	level-2	C_1/O
No MC	2^0	379889	0.234	248332	0.153
	2^4	106164	0.065	28802	0.017
	2^5	76528	0.047	19226	0.011
	2^6	48663	0.030	12821	0.007
MC	2^0	379754	0.234	250220	0.154
	2^4	105629	0.065	28692	0.017
	2^5	76318	0.047	19226	0.011
	2^6	48603	0.029	12808	0.007

The estimation of the actual compression ratio requires the number of bits needed to encode the motion vectors (MVs) and the total number of bits in the encoding procedure. The bitwise representation of the MVs is omitted and therefore the estimations given in Tables 4.9 - 4.16 do not include the number of bits spent to encode the MVs.

The total bit counts and the compression ratios in Tables 4.9 - 4.12 give us information about the effect of decomposition levels, threshold values, filter types, and motion compensation on the signal compression. The effects of the decomposition levels, the threshold values, and the filter types are similar to those obtained in Section 4.3 such that the sorting and refinement bit counts at level-2 are less than those at level-1 for all threshold values and filter types. The ratio of bit counts at level-1 to those at level-2 increases as the threshold increases. While the ratio is above 1:1.5 for 2^0, it reaches 1:4.5 when the threshold is set to 2^5. Although the compression ratio increases in parallel with an increase in the threshold value, the reconstructed signal suffers from the lack of refinement information. The majority of the total bit budget is used up to determine the significance map in the sorting passes at all threshold values. The number of bits spent in the sorting passes results in approximately 3 to 14 times that of the refinement passes.

Table 4.11 Total Sorting Bit Count ("*Table Tennis*", 3-D Haar)

case	threshold (≥)	level-1	C_2/O	level-2	C_2/O
No MC	2^0	915776	0.793	883007	0.691
	2^4	402289	0.312	302016	0.202
	2^5	316285	0.241	202035	0.135
	2^6	246252	0.181	133240	0.089
MC	2^0	918306	0.792	889548	0.693
	2^4	400343	0.310	303751	0.202
	2^5	315854	0.240	205090	0.136
	2^6	246632	0.182	137032	0.091

Table 4.12 Total Sorting Bit Count ("*Table Tennis*", 3-D H + Db-9/7)

case	threshold (≥)	level-1	C_2/O	level-2	C_2/O
No MC	2^0	999564	0.850	978644	0.756
	2^4	445000	0.339	342814	0.229
	2^5	353731	0.265	238498	0.158
	2^6	276942	0.200	162862	0.108
MC	2^0	1005108	0.853	993258	0.766
	2^4	445778	0.339	353299	0.235
	2^5	354186	0.265	247960	0.164
	2^6	277076	0.200	171401	0.113

For a given threshold, both the sorting and refinement bits in the test cases (3) & (4) are more than those in the cases (1) & (2). The same reasons explained in Section 4.3 affect the total number of bits spent in the cases (3) & (4) adversely: (1)

The newly introduced wavelet coefficients around the image (frame) boundary after the decomposition by Daubechies-9/7 filters lead to an increase in the sorting and refinement bits. (2) Since the matrix dimensions increases unevenly in the videos spatially decomposed by the Daubechies-9/7 filters, defining a tree set with the coefficients lying in the same spatial orientation at finer levels gets difficult.

The difference between the numbers of refinement bits of the test cases that employ the Haar filters and the Daubechies-9/7 filters gets smaller as the threshold value increases in Tables 4.9 – 4.12. The decomposition of the "*Table Tennis*" video by the Daubechies-9/7 filters yield more coefficients closer to zero than by the Haar filters. Having seen the advantages of the Daubechies-9/7 filters on the decomposition of the "*Conference*" video in Chapter 3, we also encode the "*Conference*" video to compare the total bit counts with that of the "*Table Tennis*". In this comparison, we only considered the test cases with threshold values 2^0 and 2^5. The results are summarized in Tables 4.13 - 4.16.

Table 4.13 Total Refinement Bit Count ("*Conference*", 3-D Haar)

case	threshold (≥)	level-1	C_1/O	level-2	C_1/O
No MC	2^0	413584	0.254	298781	0.184
	2^5	67817	0.041	18653	0.011
MC	2^0	394938	0.243	278268	0.171
	2^5	67748	0.041	18436	0.011

Table 4.14 Total Refinement Bit Count ("*Conference*", 3-D H + Db-9/7)

case	threshold (≥)	level-1	C_1/O	level-2	C_1/O
No MC	2^0	406581	0.250	289633	0.178
	2^5	67095	0.041	18191	0.011
MC	2^0	394425	0.243	280185	0.172
	2^5	67017	0.041	18571	0.011

According to the results in Tables 4.13 and 4.14, the difference between the numbers of refinement bits of the Haar and Daubechies-9/7 test cases existing for the "*Table Tennis*" video diminishes for the "*Conference*" video. Even, the refinement bit counts in cases (3) & (4) of the "*Conference*" video falls behind that of the cases (1) & (2) except the MC case with the threshold value of 2^0 at level-2.

Table 4.15 Total Sorting Bit Count ("*Conference*", 3-D Haar)

case	threshold (≥)	level-1	C_2/O	level-2	C_2/O
No MC	2^0	907591	0.814	865899	0.718
	2^5	336999	0.249	233392	0.155
MC	2^0	926934	0.814	889790	0.720
	2^5	346575	0.255	245338	0.162

Table 4.16 Total Sorting Bit Count ("*Conference*", 3-D H + Db-9/7)

case	threshold (≥)	level-1	C_2/O	level-2	C_2/O
No MC	2^0	1001905	0.868	960479	0.770
	2^5	380288	0.275	269717	0.177
MC	2^0	1010660	0.866	979585	0.776
	2^5	379765	0.275	280235	0.184

Another point to note about the difference of the refinement and sorting bit counts between the "*Table Tennis*" and "*Conference*" videos is the decreased refinement bit counts and the increased sorting bit counts in Tables 4.13 - 4.16. Because the wavelet decomposition of the latter test video is not as successful as the "*Table Tennis*", the entropy of its wavelet coefficients is higher than that of the "*Conference*" video. Since the accumulation of the coefficients around zero is not much, or in other words, the energy concentration in the coarsest scale is less, the length of the refinement bit packages (arrows in Figure 4.3) is less at higher threshold values. This can be realized by observing the bit counts at the threshold values of 2^5 in Tables 4.9 - 4.10 and 4.13 - 4.14. Inherently, the WT supports the ordering of the coefficients. The creation of coefficients with higher magnitudes in finer scales also makes the sorting algorithm run inefficiently for the "*Conference*" video. The sorting algorithm suffers from the unevenly distributed wavelet coefficients while placing them in the order of importance.

There are common inferences for both test videos considering the effects of the motion compensation: (1) the number of refinement bits is less in the MC test cases; on the other hand, (2) the number of sorting bits is more. The entropy analysis in Chapter 3 revealed that MC technique results in more coefficients to accumulate around zero. Having very small magnitudes, these coefficients appear at the end of the MOC matrix. Because of their small or zero magnitudes, they are usually represented by the least significant bits (in the 0th bit location) in the MOC matrix. The application of a threshold to the refinement bits causes more of these coefficients (having zeroed or unity magnitudes) to be ignored in the MC cases. Because the decrease in the refinement bits is not significant, the ratios, C_1/O do not show a noticeable improvement in the tables above. Besides the insufficient improvement in the compression of the refinement bits, the sorting passes are influenced adversely by the MC either. The MC may change the magnitudes of the coefficients in the regions where "*unconnected*" or "*double connected*" samples are found, in such a way that the spatial orientation of a frame at finer levels is lost. If the coefficients that are marked as descendants in a tree set are found to be significant regardless of the insignificance of the parent node cause the sorting algorithm to work inefficiently. In that case, more bits need to be spent to sort out the individual coefficients in the tree. The overall *compression ratio* is determined by the increase in the sorting bits. Thus, the compression ratios in the MC test cases are slightly less than the No MC cases.

Another important point to be paid attention is the great difference between the encoding results of the 3-D and 2-D decomposed videos. The refinement bit counts given in Table 4.1 are more than those given in the No MC test cases of Tables 4.9 - 4.11. Especially, the order of the difference gets larger at level-2. The ratio is above 1:2.5 for the threshold value of 2^6 for both filter types. The compression ratio in the refinement bits of the 3-D decomposed "*Table Tennis*" video is approximately twice that of its 2-D decomposition. The most important reason for the decrease in the number of the refinement bits is the success of the 3-D wavelet decomposition in the concentration of the energy of the signal in the coarsest scale or in the accumulation of the coefficients closer to zero. How the temporal decomposition in addition to the spatial decomposition does improve the decorrelation performance is already studied in detail in Chapter 3. Since the energy of the signal is mostly represented by the coefficients in the temporal low subband frames, the estimated refinement bit counts for the entire GOF structure lessen especially at high threshold values. The accomplishment becomes clearer as the decomposition goes further down.

Contrary to the relation between the numbers of the refinement bits, the number of sorting bits is more in the 3-D decomposed "*Table Tennis*" video than in its 2-D decomposed case. There are two reasons that increase the sorting bit counts and correspondingly affect the compression ratios (C_2/O) adversely: (1) the different spatial decomposition level that each frame has after the 3-D WT and (2) the averaged intensity (magnitude) of the pixels (coefficients) after the temporal decomposition. Two-level spatial-temporal decomposition of the GOF structure produces 4 two-level spatially decomposed and 4 one-level spatially decomposed temporal subband frames. Only the number of spatial decompositions of the individual frames is taken into account when applying the 2-D SPIHT. Because the latter 4 frames are only one-level spatially decomposed, both the sorting and refinement bit counts are more than the former 4 frames. Besides, the averaged intensity (magnitude) of the pixels (coefficients) after the temporal decomposition fails the following assumption which is crucial for the SPIHT algorithm to work well: If a root of a tree set is found to be insignificant, its descendants are also insignificant.

4.6 Decoding of Spatial-Temporal Decomposed Video

The decoding process follows the same as the encoding process in reverse order. The decoder scans the same coefficients with the encoder. As the data bits are accepted, they are placed into the proper location to make the coefficient matrix ready for the reconstruction of the signal. The resolution of the original video signal is deterministic to both the encoder and the decoder at the beginning of the transfer. The decoding algorithm and the information transfer are the same as that in Section 4.4.

In Tables 4.17 and 4.20, we summarized the average PSNRs after the 3-D wavelet synthesis operation of the two test videos, "*Table Tennis*" and "*Conference*", whose encoding results are given in the previous section. While the former video is reconstructed at 4 different threshold values (2^0, 2^4, 2^5, 2^6), the latter is only reconstructed at the thresholds 2^0 and 2^5. The reconstruction test cases are the same as the encoding test cases: (1) the reconstruction of the video using the Haar filters, (2) the MC reconstruction of the video using the Haar filters, (3) the reconstruction of the video using the Daubechies-9/7 filters, and (4) the MC reconstruction of the video using the Daubechies-9/7 filters.

Table 4.17 Average PSNR ("*Table Tennis*", 3-D Haar)

case	Threshold (≥)	level-1	level-2
No MC	2^0	59.22	59.18
	2^4	33.80	34.73
	2^5	29.32	31.24
	2^6	24.26	28.10
MC	2^0	59.08	59.02
	2^4	33.81	34.73
	2^5	29.44	31.43
	2^6	24.36	28.41

Table 4.18 Average PSNR ("*Table Tennis*", 3-D H + Db-9/7)

case	threshold (≥)	level-1	level-2
No MC	2^0	58.78	58.74
	2^4	33.77	34.62
	2^5	29.46	31.20
	2^6	24.28	27.94
MC	2^0	58.65	58.57
	2^4	33.76	34.59
	2^5	29.50	31.22
	2^6	24.33	28.01

Three comments can be derived directly from the PSNR results: (1) the signal quality (PSNR) increases when decoding starts from a deeper decomposition level, (2) the increased threshold value leads to the information loss, causing a worse reconstruction, (3) the 3-D reconstruction of both videos gives better results as compared to the PSNRs in Tables 4.7 and 4.8. The first two comments are the same as those derived after the 2-D reconstruction of the same videos in Section 4.4. There is a significant improvement at level-2 in the signal quality. The improvement is up to 4 db, and the best is achieved at the threshold value 2^6 for the second test case of the "*Table Tennis*" video.

Table 4.19 Average PSNR ("*Conference*", 3-D Haar)

case	threshold (≥)	level-1	level-2
No MC	2^0	59.18	59.11
	2^5	28.13	29.18
MC	2^0	58.99	58.89
	2^5	28.40	29.57

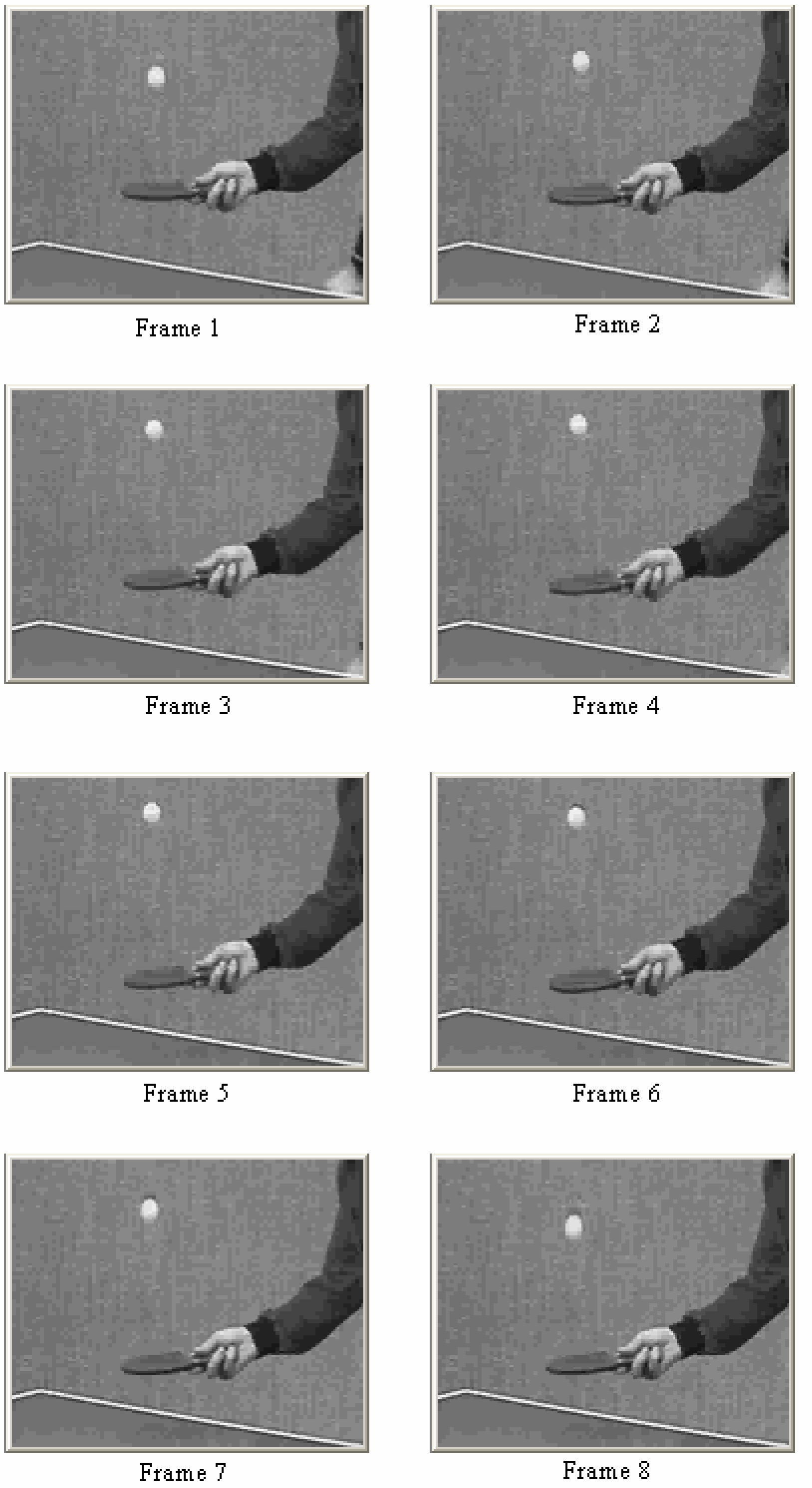
Table 4.20 Average PSNR ("*Conference*", 3-D H + Db-9/7)

case	threshold (≥)	level-1	level-2
No MC	2^0	58.77	58.75
	2^5	28.21	29.27
MC	2^0	58.60	58.52
	2^5	28.45	29.53

The highest PSNR is obtained in case of 2^0, corresponding to the perfect reconstruction. A measurable signal to noise ratio is obtained due to the information loss by the rounding operations at the beginning of the encoding and at the end of the synthesis processes. The rounding operation eases the implementation and omits dealing with the definition of a decimal type common to both the encoder and the decoder. The PSNR results reveal us that the reconstructed "*Table Tennis*" video gives slightly better PSNRs in the Haar test cases. Using the Daubechies-9/7 filters is only advantageous in the reconstruction of the "*Conference*" video. However, the benefit is very little up to 0.4 dB at higher threshold values. On the other hand, the motion compensation seems to be advantageous at high threshold values for both videos, but the profit is very little. The maximum PSNR improvement is 0.4 db, which is achieved at level-2 in both videos for the Haar test cases.

The most significant advantage of using the 3-D reconstruction over the 2-D reconstruction is its contribution to the PSNR. In general the PSNR improvement between Tables 4.7 – 4.8 and the No MC test cases in Tables 4.17 – 4.18 is 2 db. However, there are extreme cases where this difference can reach 4 dB in the Haar test cases at the threshold value 2^0.

We displayed the reconstructed “*Table Tennis*” video decoded for the threshold values 2^5 and 2^6, and the “*Conference*” video decoded for only the threshold value 2^5 in Figures 4.12 - 4.34. The smoothing properties of the Daubechies-9/7 filters can be seen by the naked eye in the reconstructed frames. As opposed to the Daubechies-9/7 filters, the Haar filters cause rectangular regions to appear where spatial frequency changes occur in the reconstructed frames.



Frame 1

Frame 2

Frame 3

Frame 4

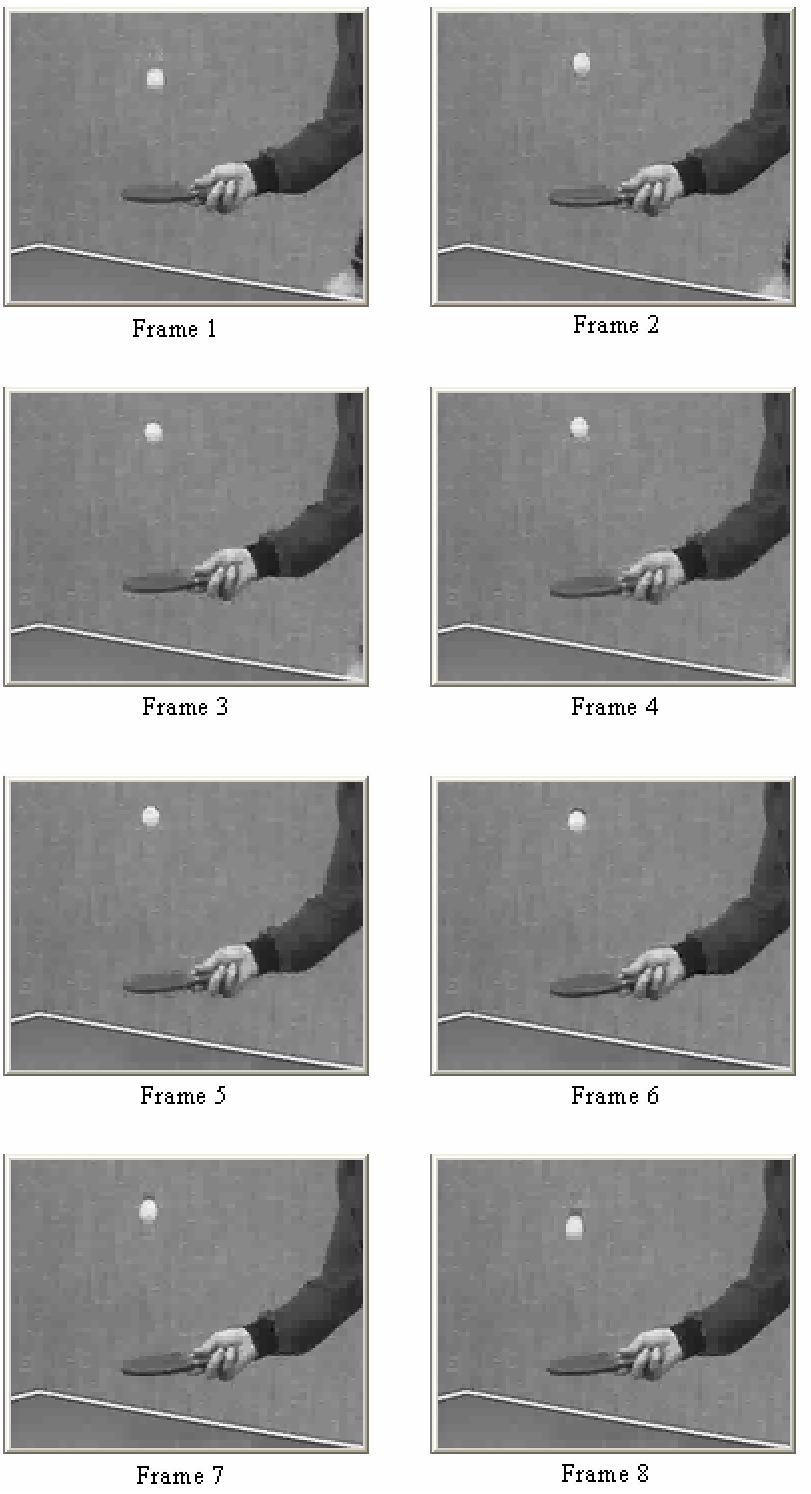
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.12 Reconstructed “*Table Tennis*” Video (3-D Haar, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

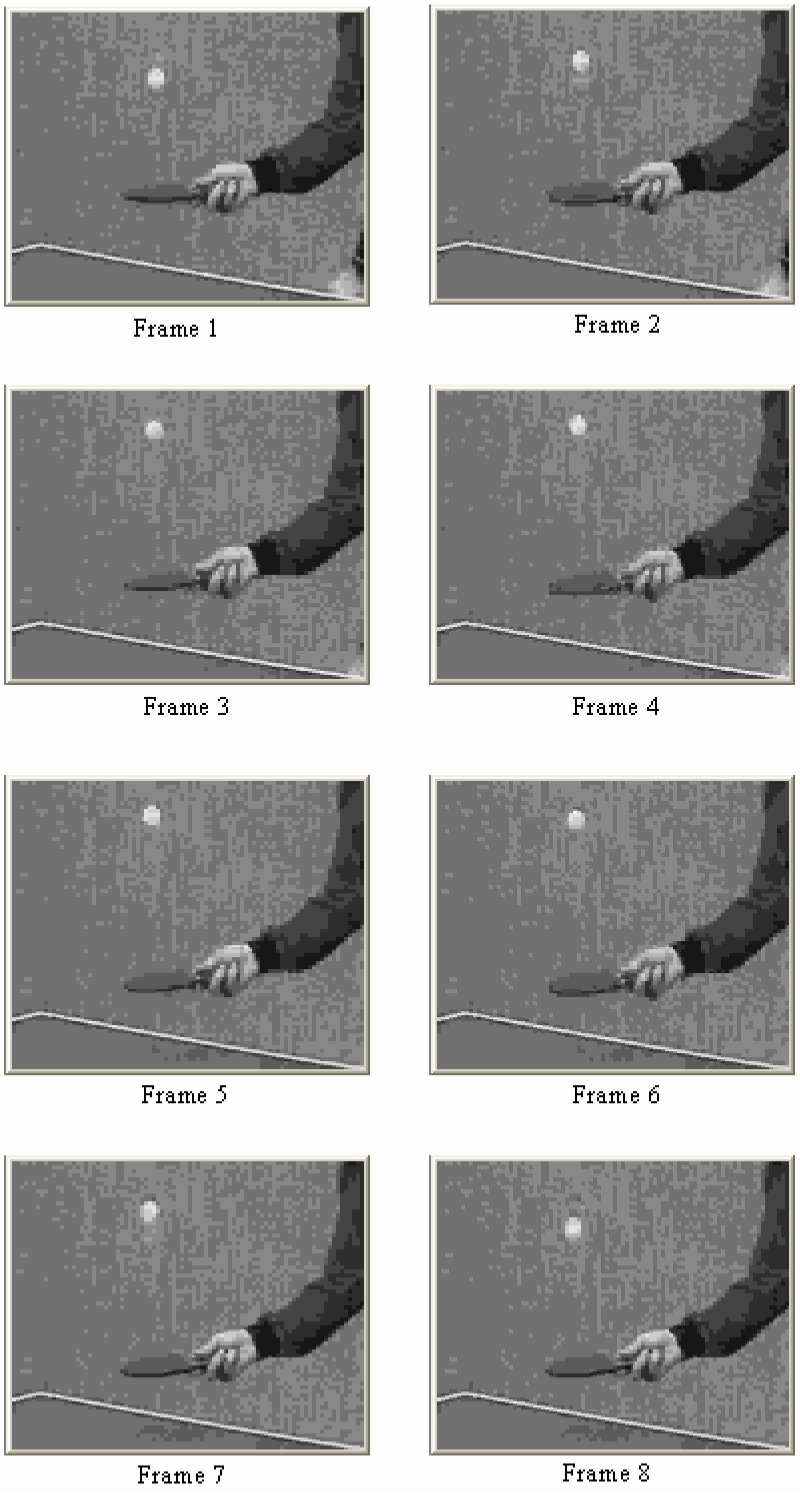
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.13 Reconstructed “*Table Tennis*” Video (3-D Haar, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

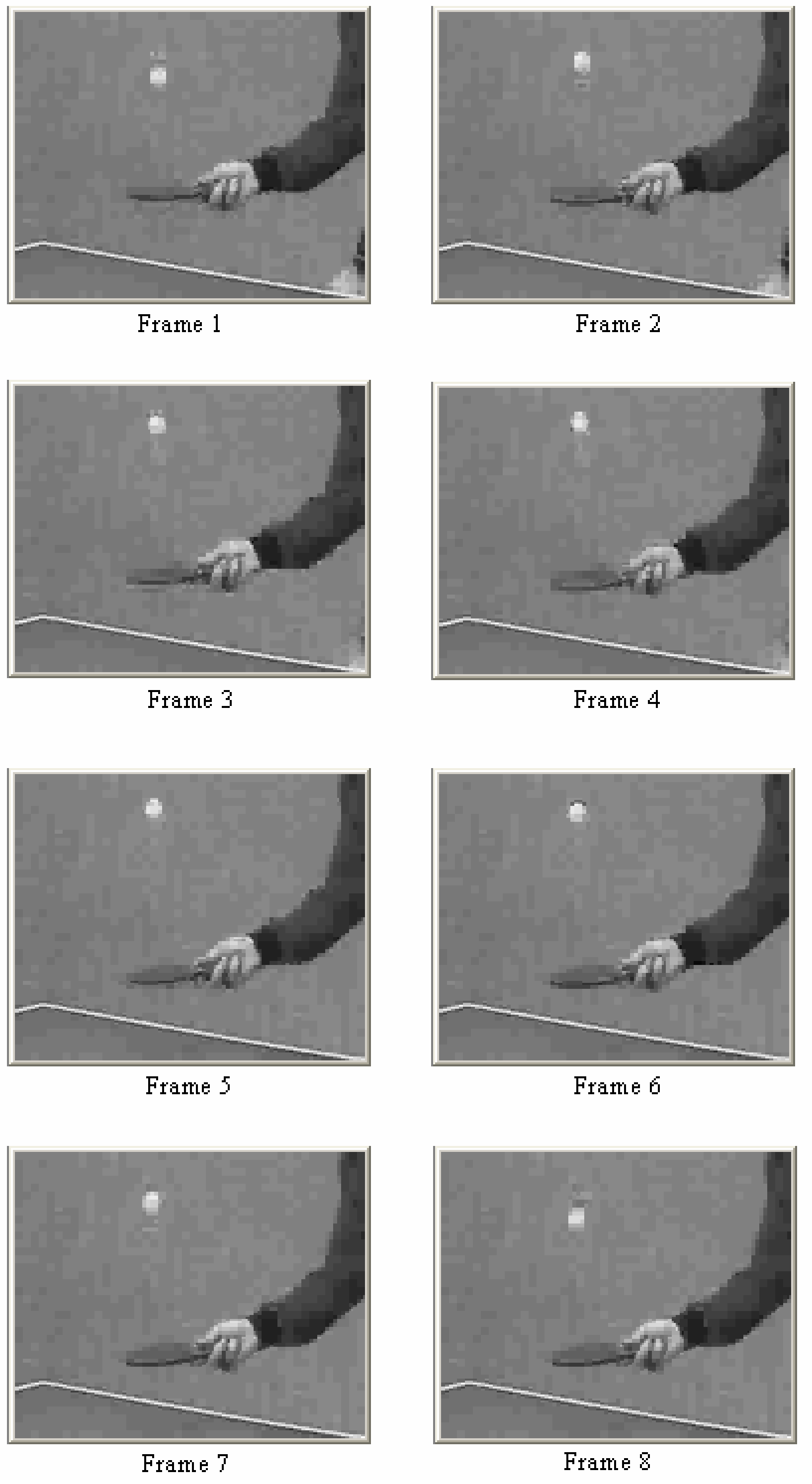
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.14 Reconstructed “*Table Tennis*” Video (3-D Haar, L-1, $T = 2^6$)



Frame 1

Frame 2

Frame 3

Frame 4

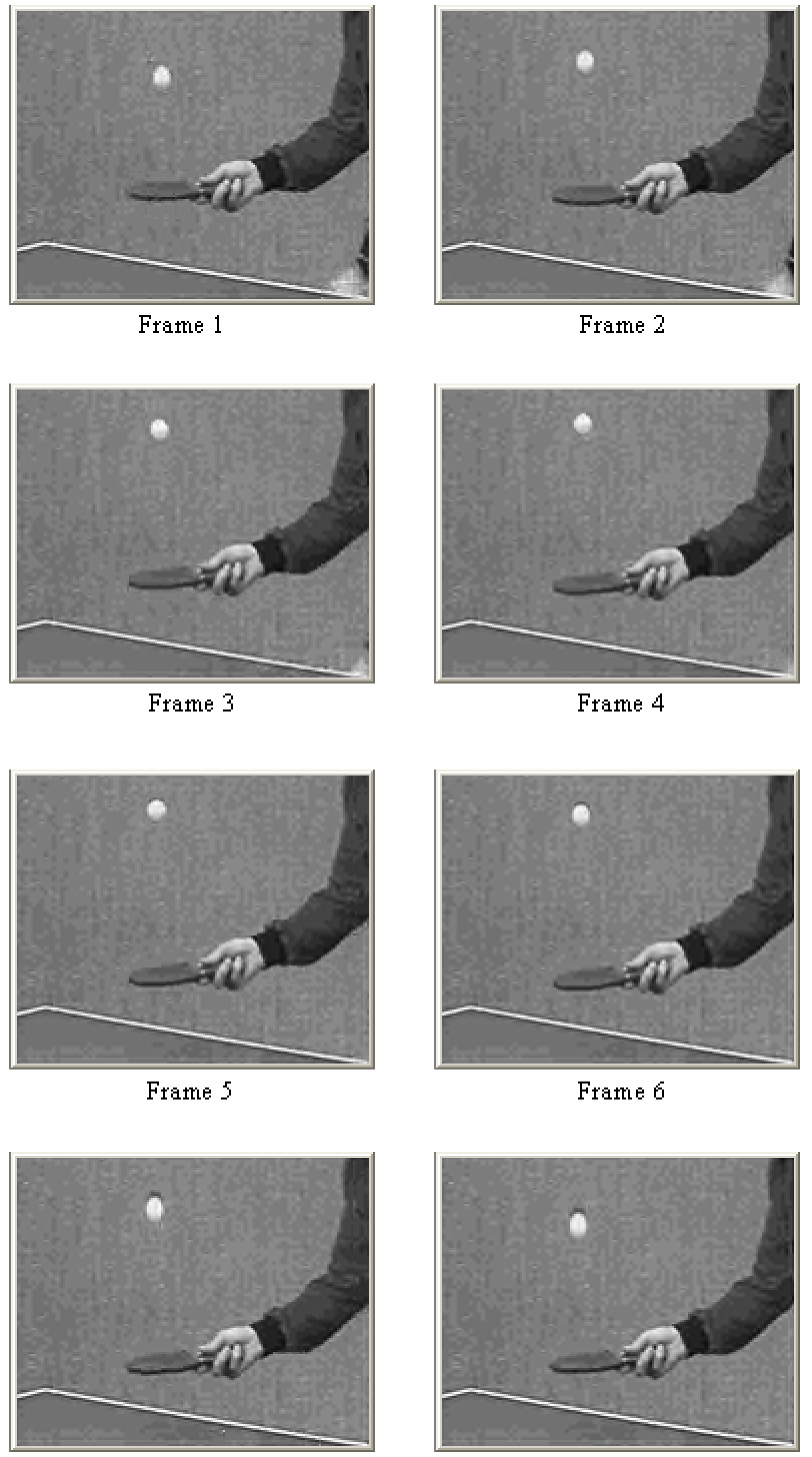
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.15 Reconstructed "*Table Tennis*" Video (3-D Haar, L-2, $T = 2^6$)



Frame 1

Frame 2

Frame 3

Frame 4

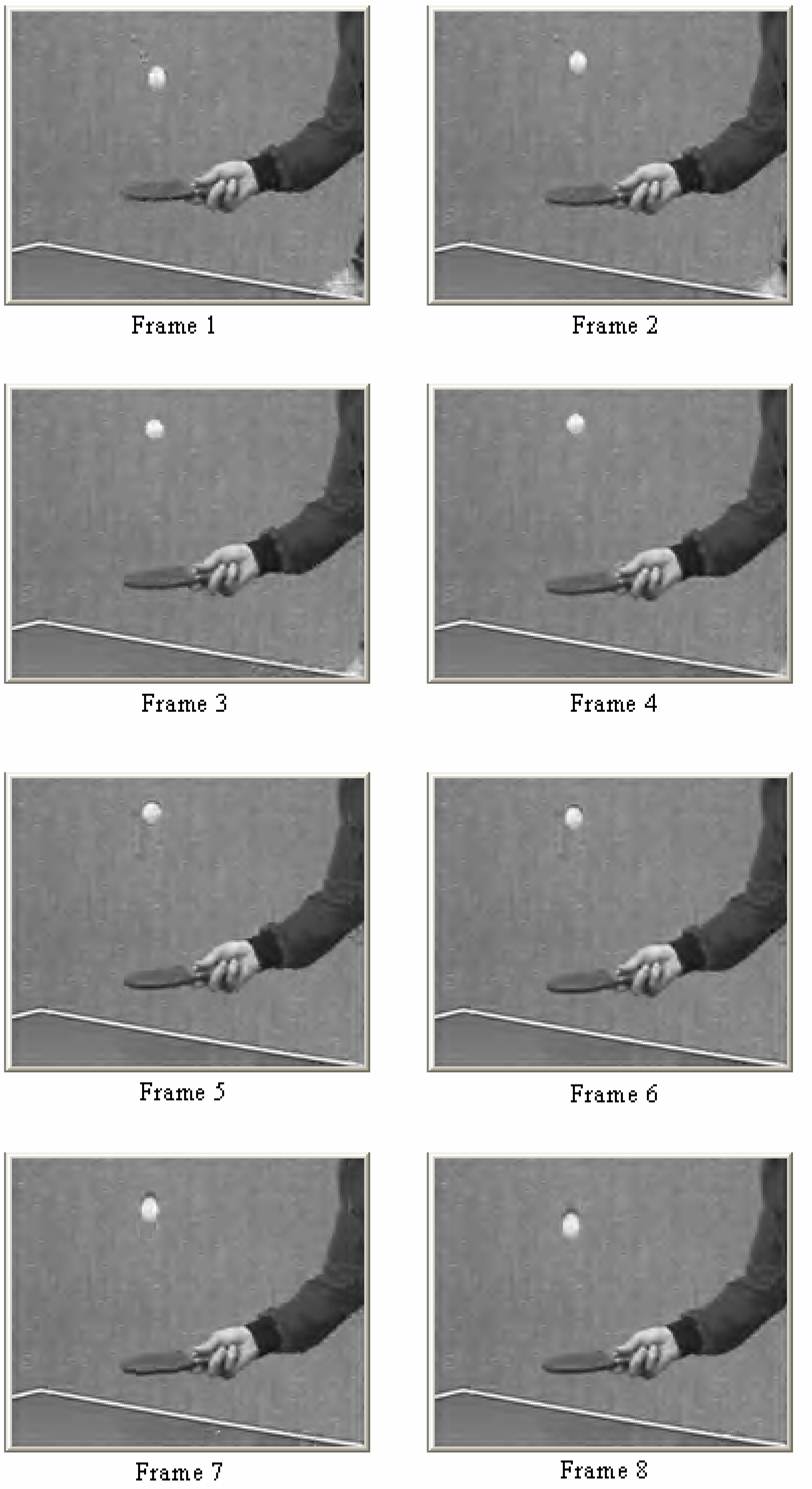
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.16 Reconstructed "*Table Tennis*" Video (MC 3-D H + Db-9/7, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

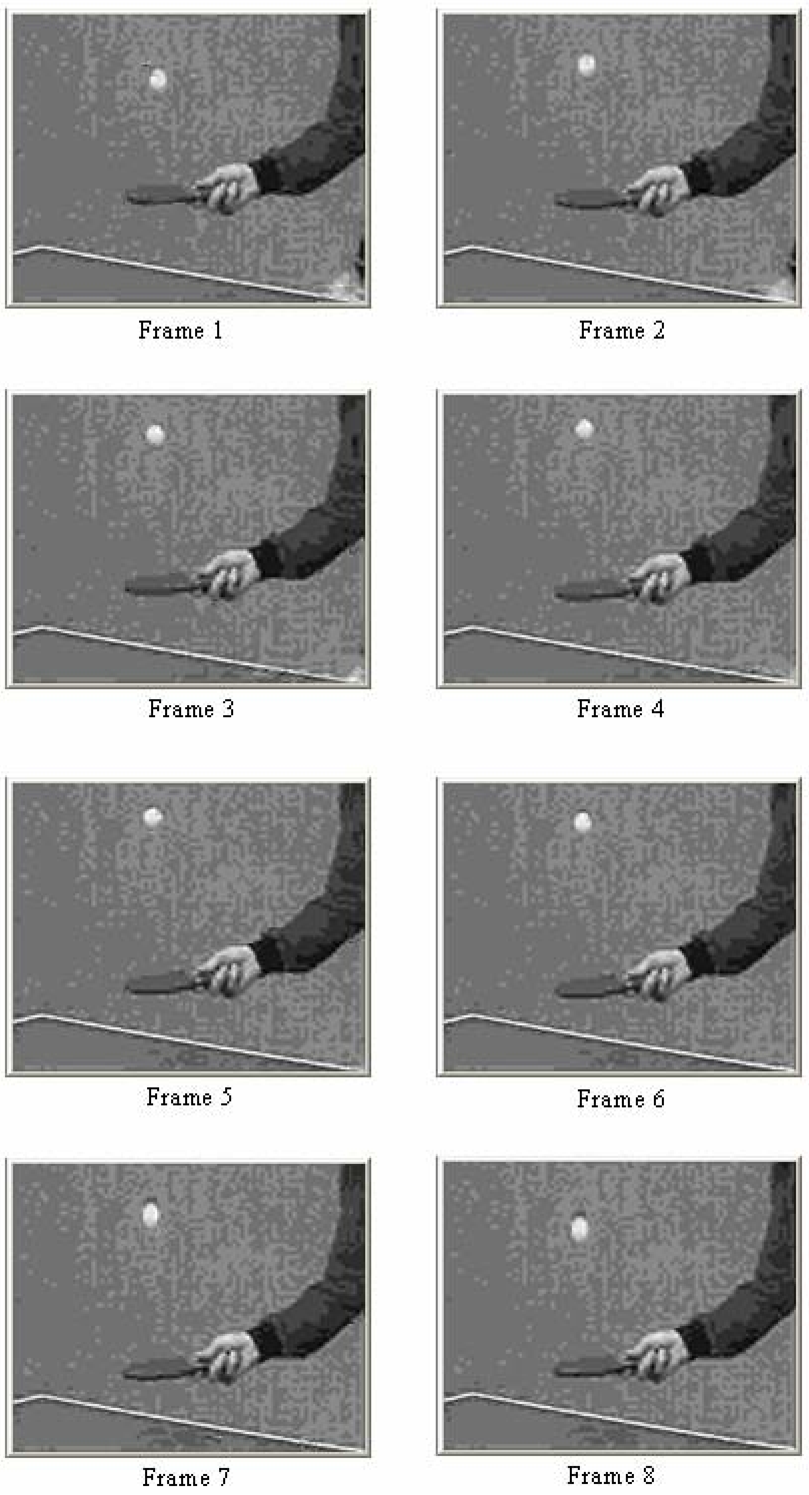
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.17 Reconstructed "*Table Tennis*" Video (MC 3-D H + Db-9/7, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

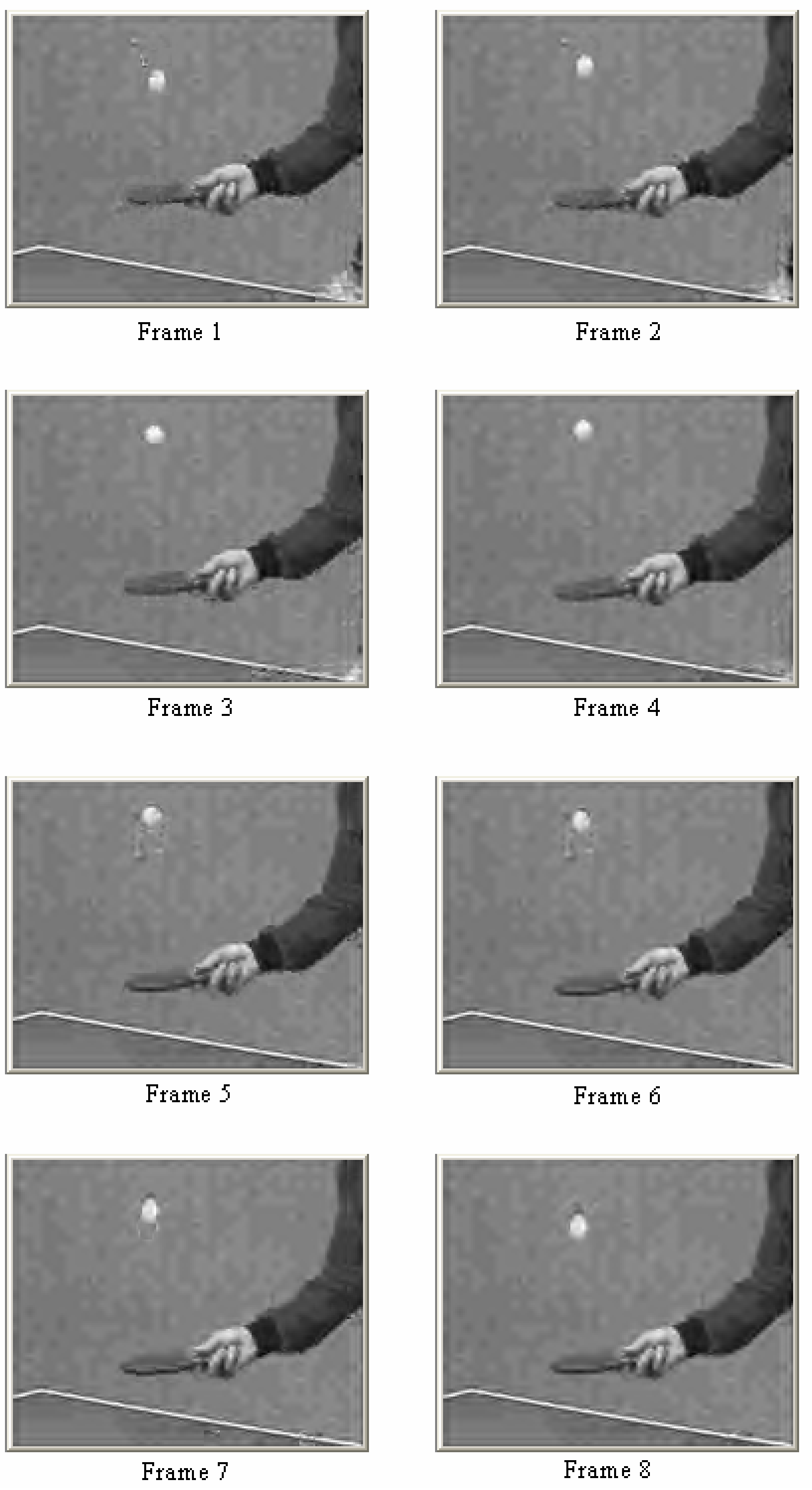
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.18 Reconstructed "*Table Tennis*" Video (MC 3-D H + Db-9/7, L-1, $T = 2^6$)



Frame 1

Frame 2

Frame 3

Frame 4

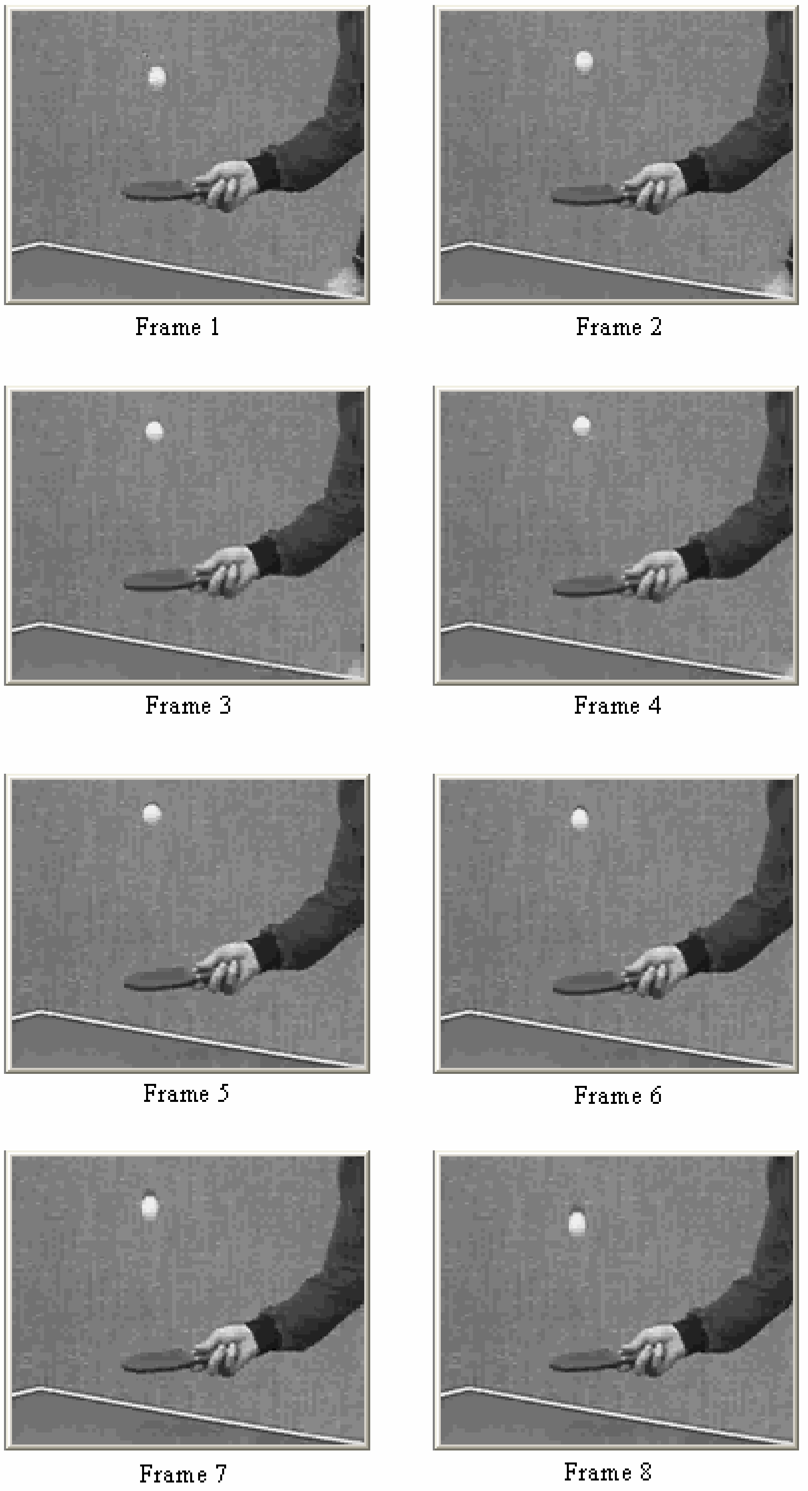
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.19 Reconstructed "*Table Tennis*" Video (MC 3-D H + Db-9/7, L-2, $T = 2^6$)



Frame 1

Frame 2

Frame 3

Frame 4

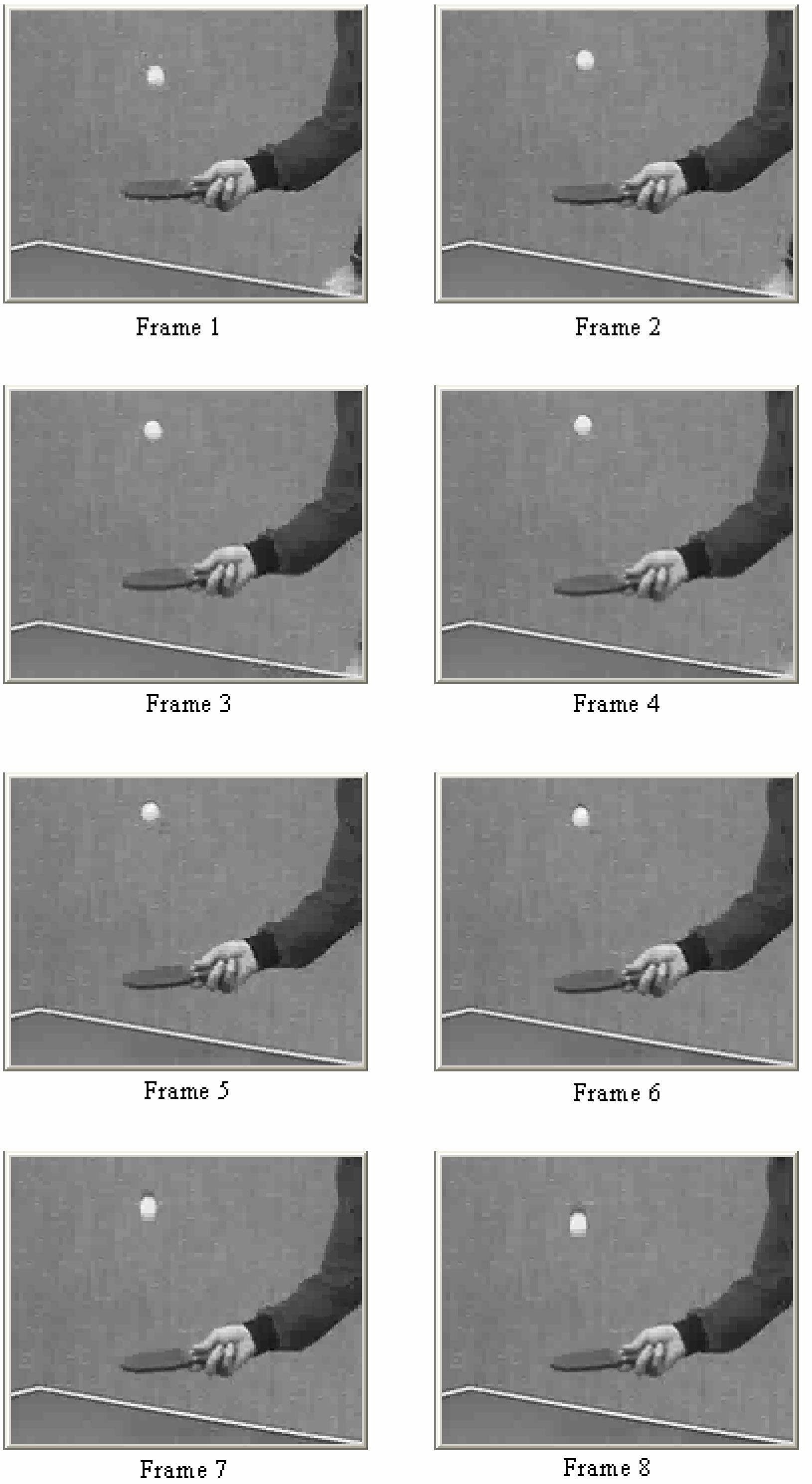
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.20 Reconstructed “*Table Tennis*” Video (MC 3-D Haar, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

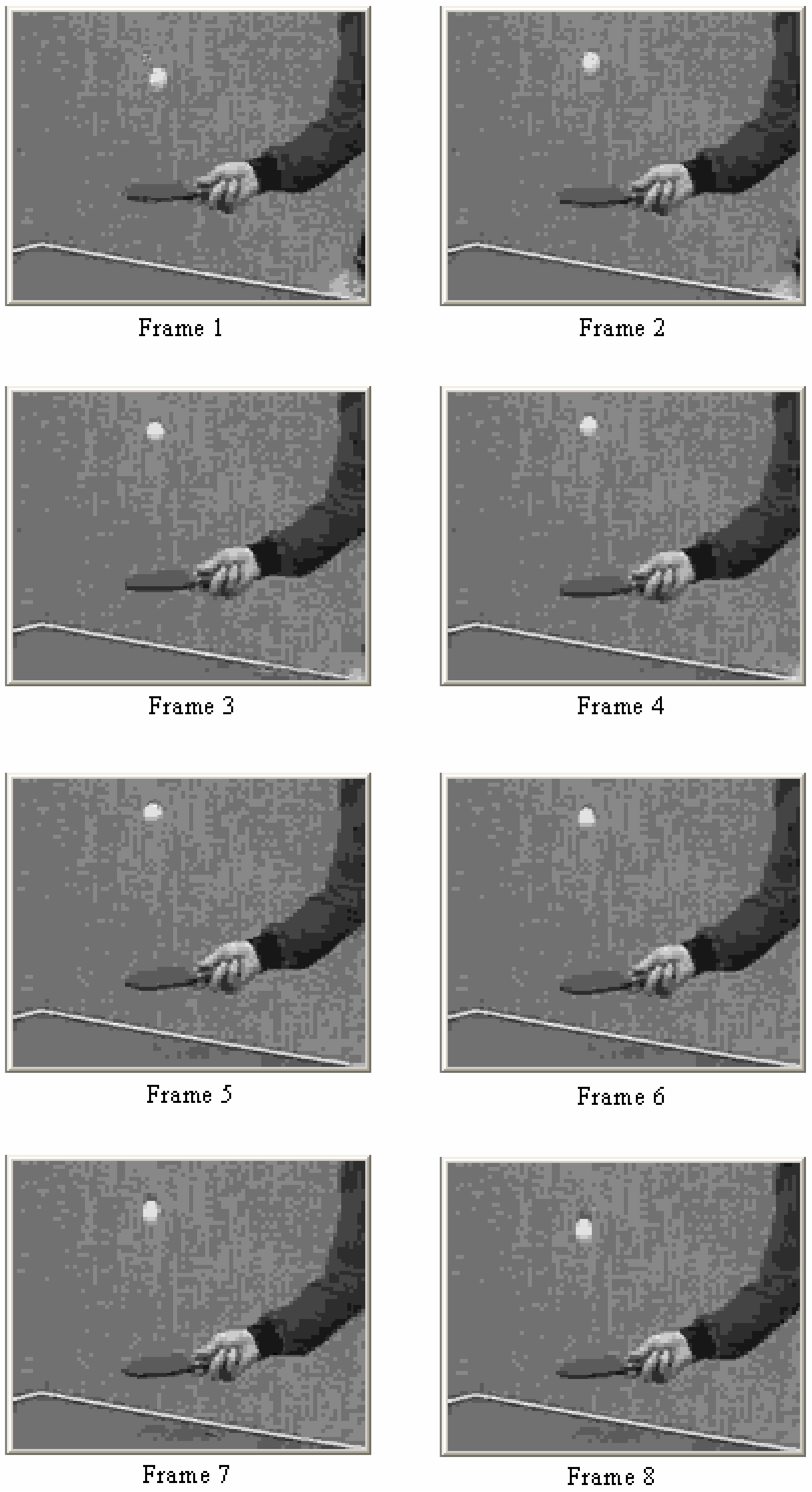
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.21 Reconstructed "*Table Tennis*" Video (MC 3-D Haar, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

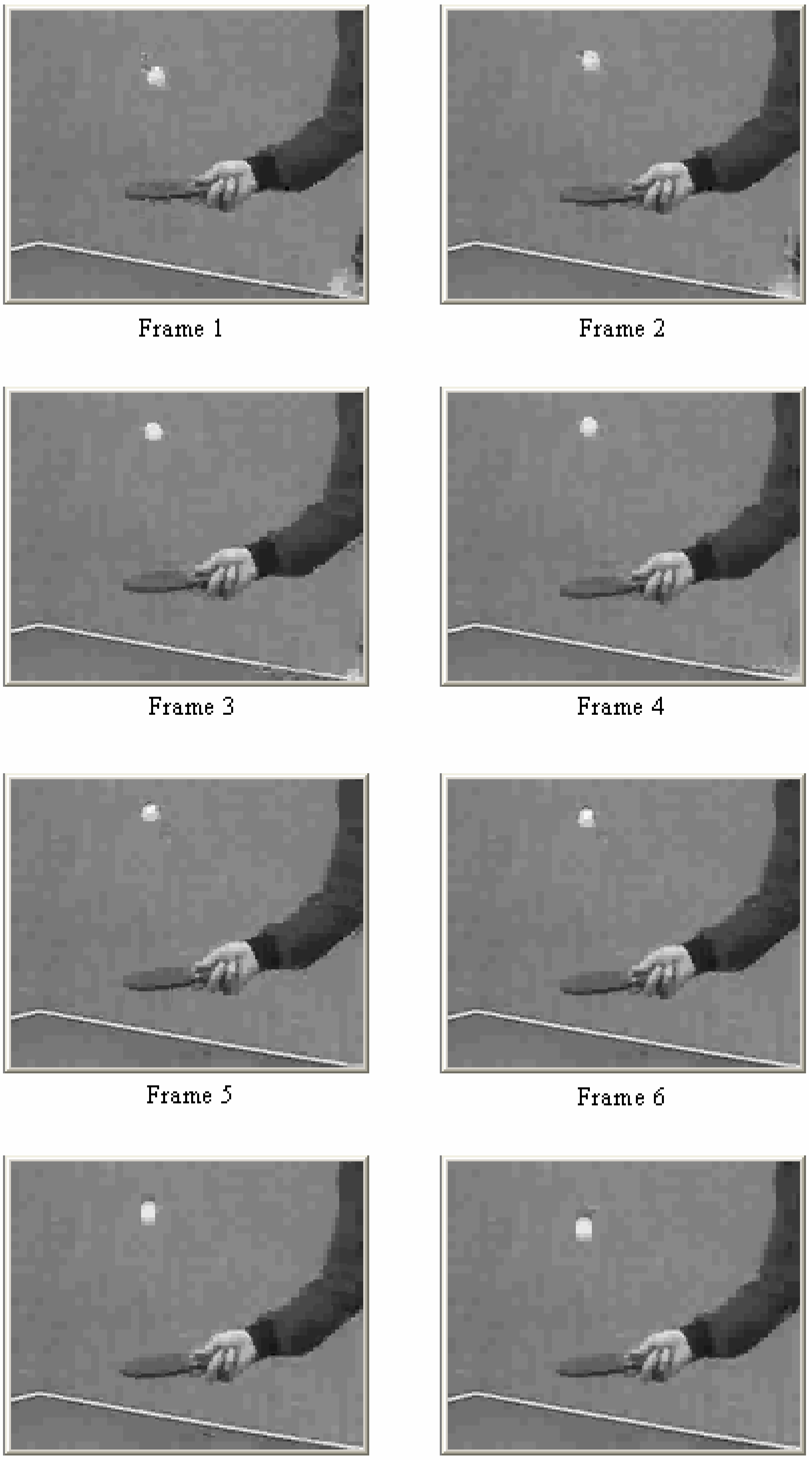
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.22 Reconstructed "*Table Tennis*" Video (MC 3-D Haar, L-1, $T = 2^6$)



Frame 1

Frame 2

Frame 3

Frame 4

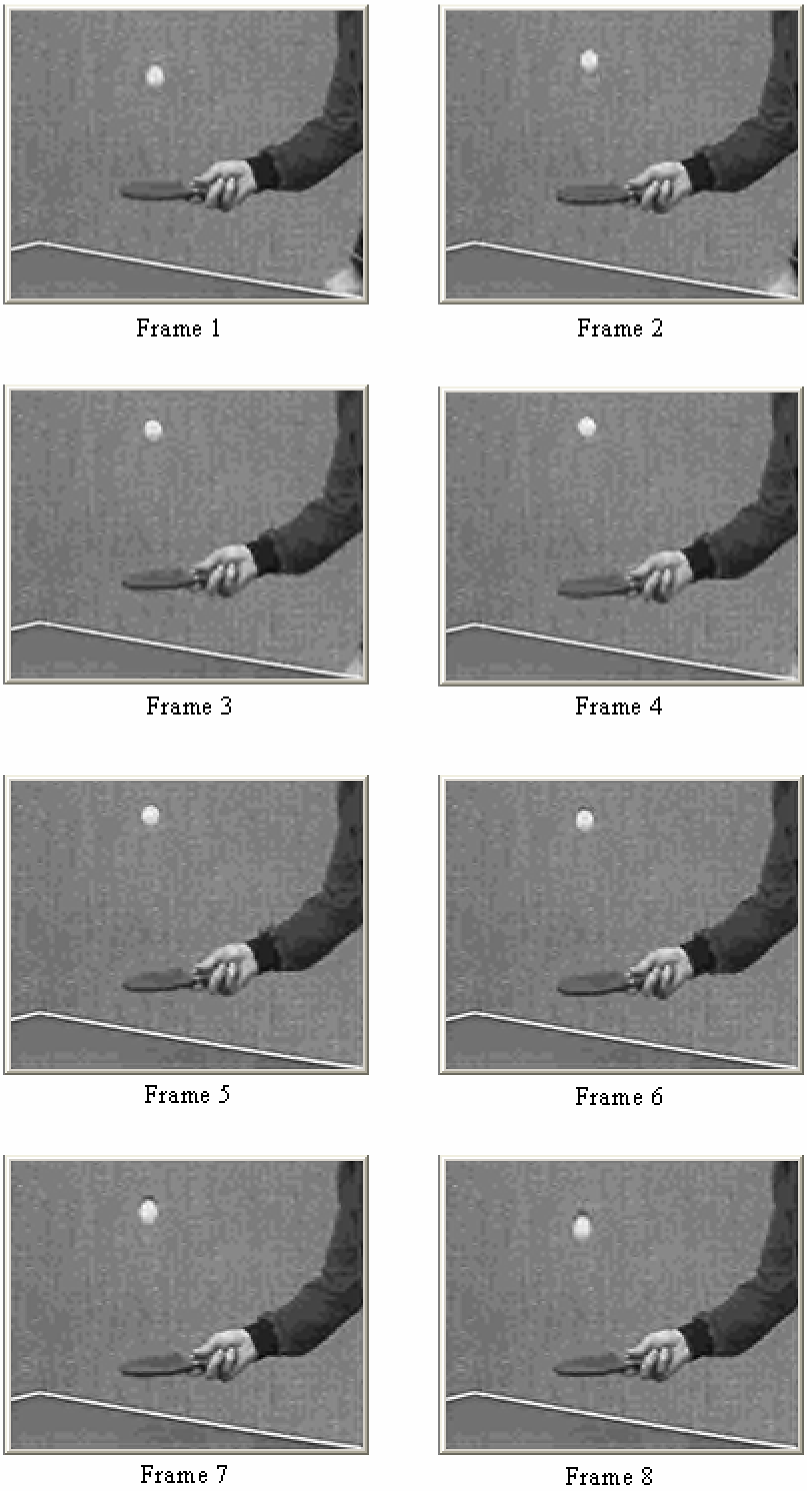
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.23 Reconstructed “*Table Tennis*” Video (MC 3-D Haar, L-2, $T = 2^6$)



Frame 1

Frame 2

Frame 3

Frame 4

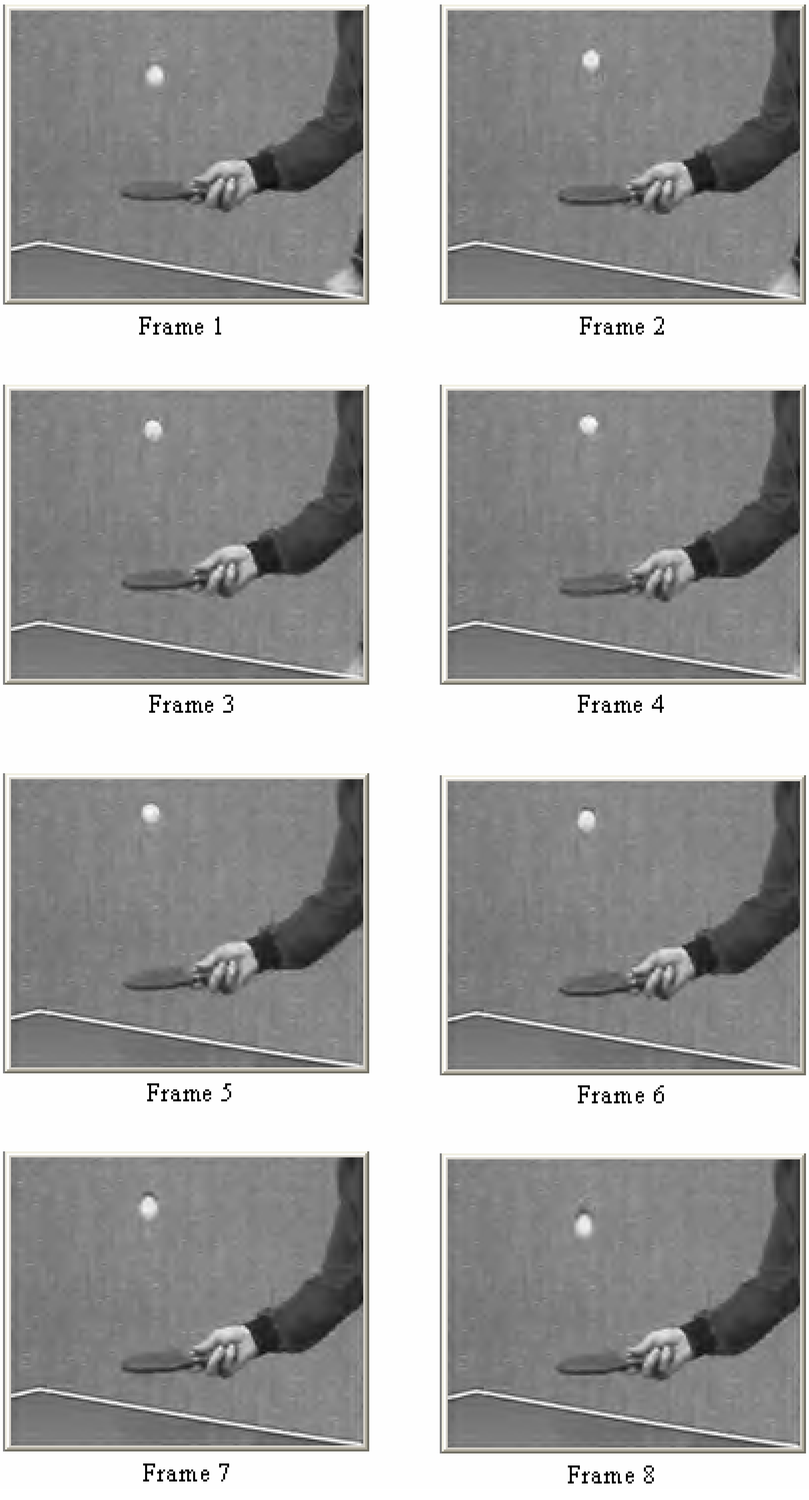
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.24 Reconstructed “*Table Tennis*” Video (3-D H + Db-9/7, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

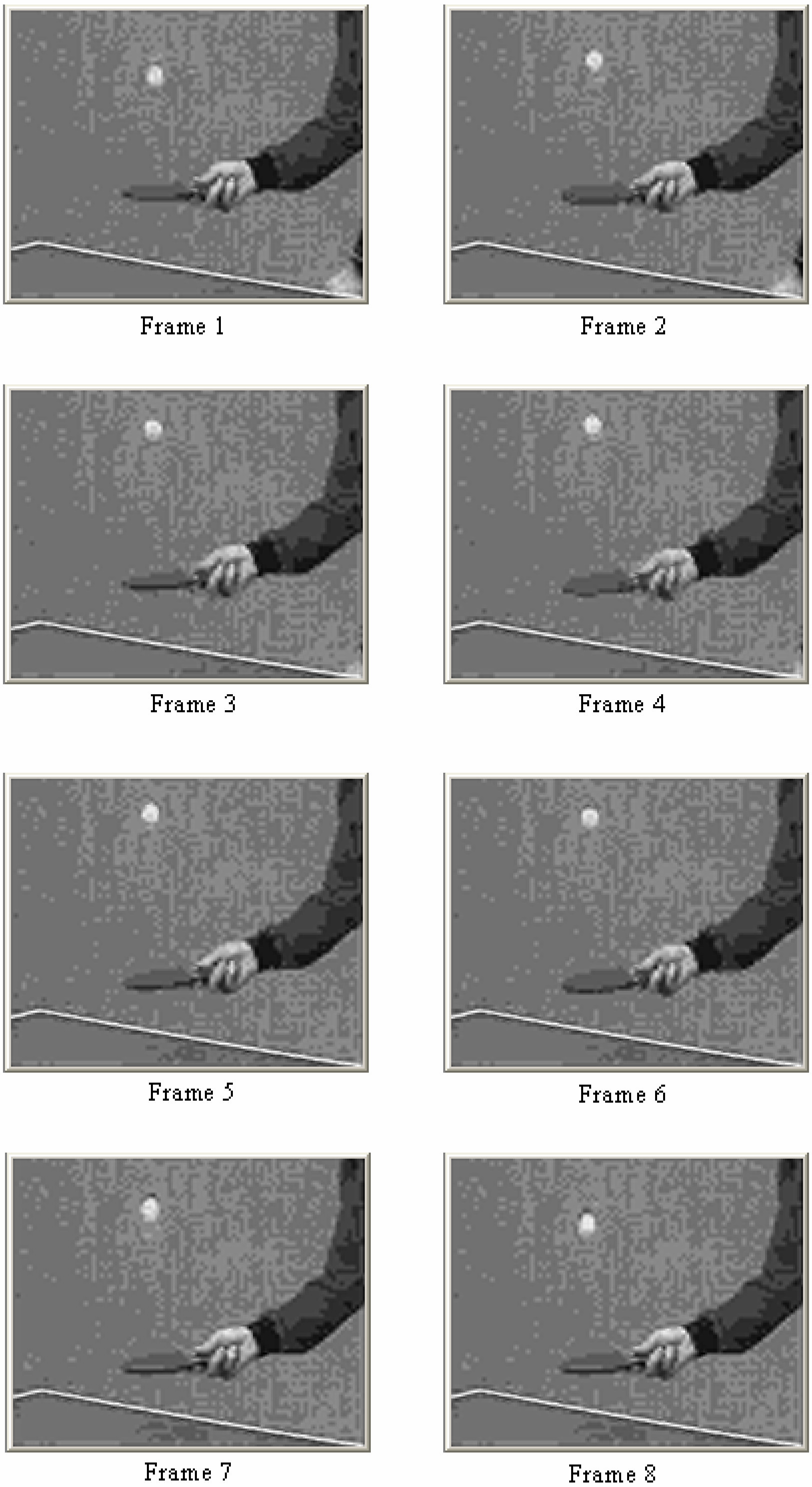
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.25 Reconstructed “*Table Tennis*” Video (3-D H + Db-9/7, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

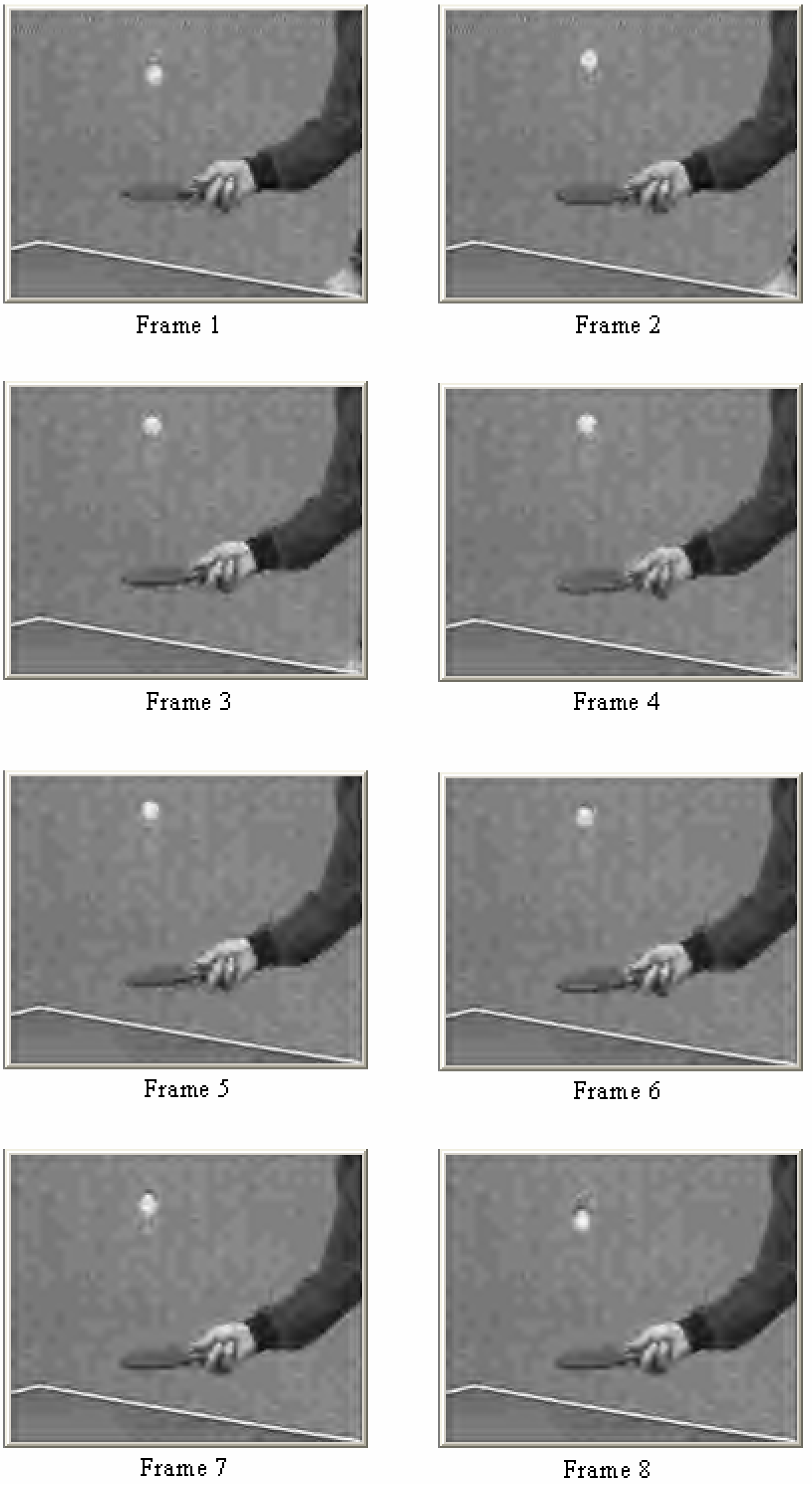
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.26 Reconstructed “*Table Tennis*” Video (3-D H + Db-9/7, L-1, $T = 2^6$)



Frame 1

Frame 2

Frame 3

Frame 4

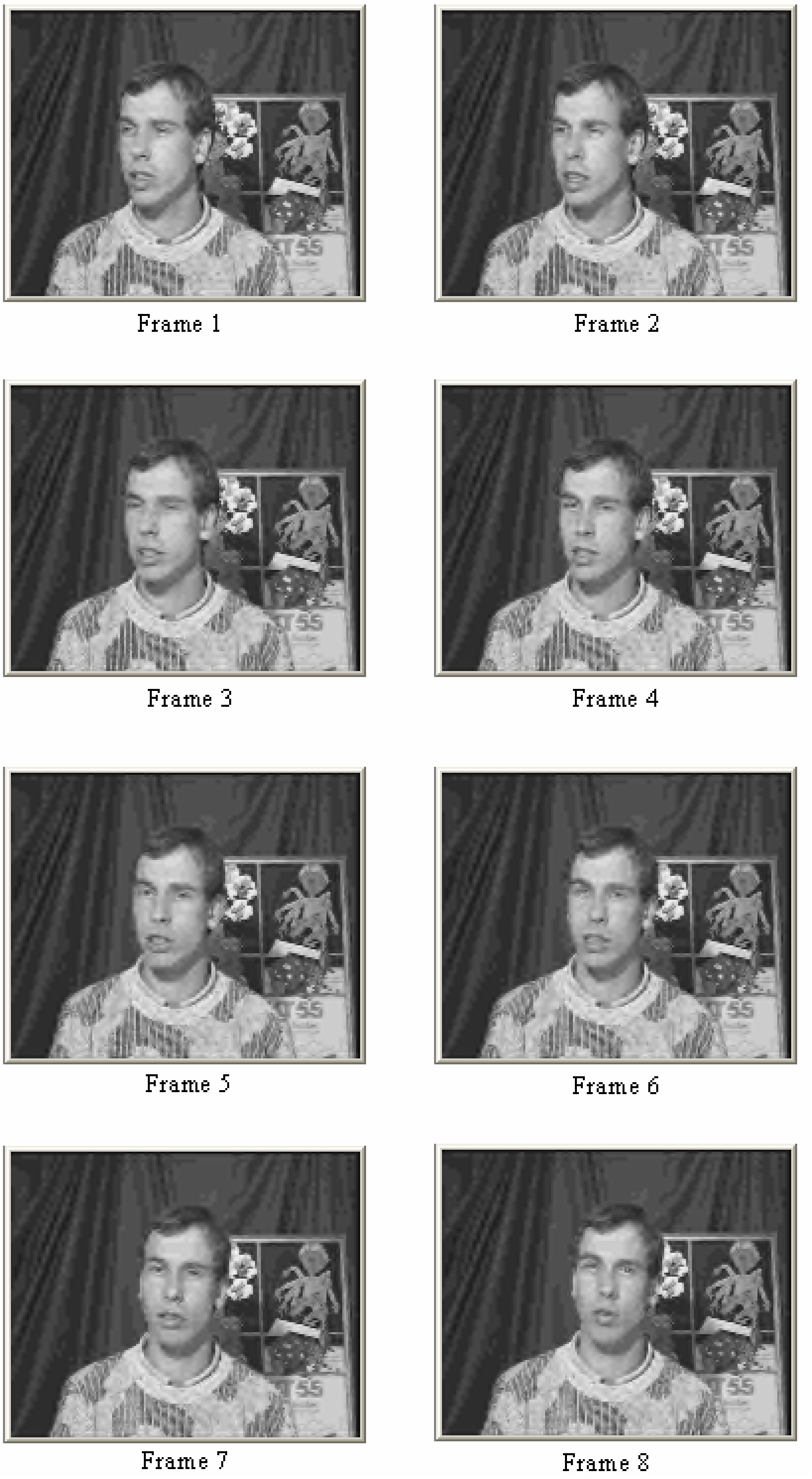
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.27 Reconstructed “*Table Tennis*” Video (3-D H + Db-9/7, L-2, $T = 2^6$)



Frame 1

Frame 2

Frame 3

Frame 4

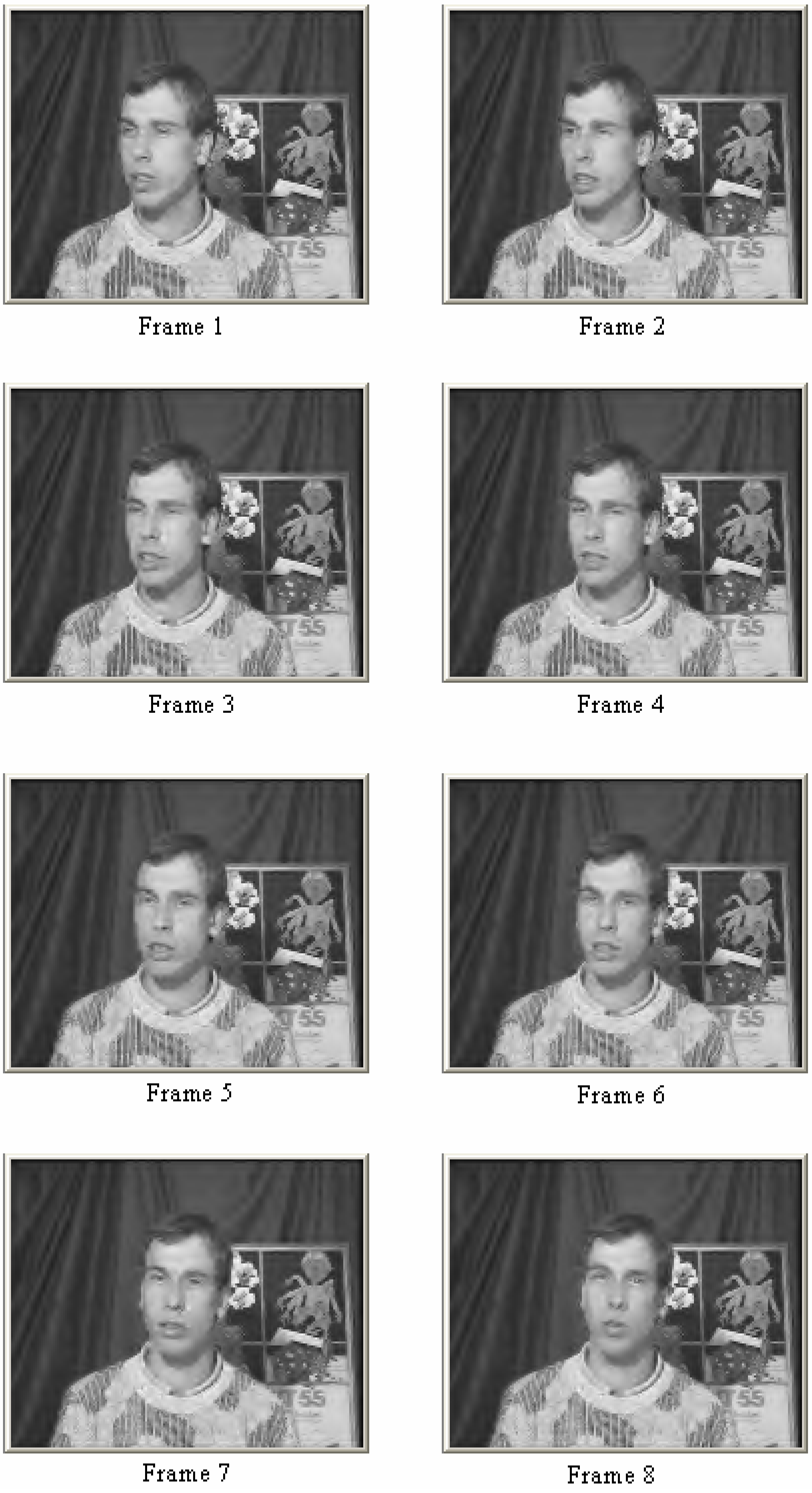
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.28 Reconstructed “*Conference*” Video (3-D H + Db-9/7, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

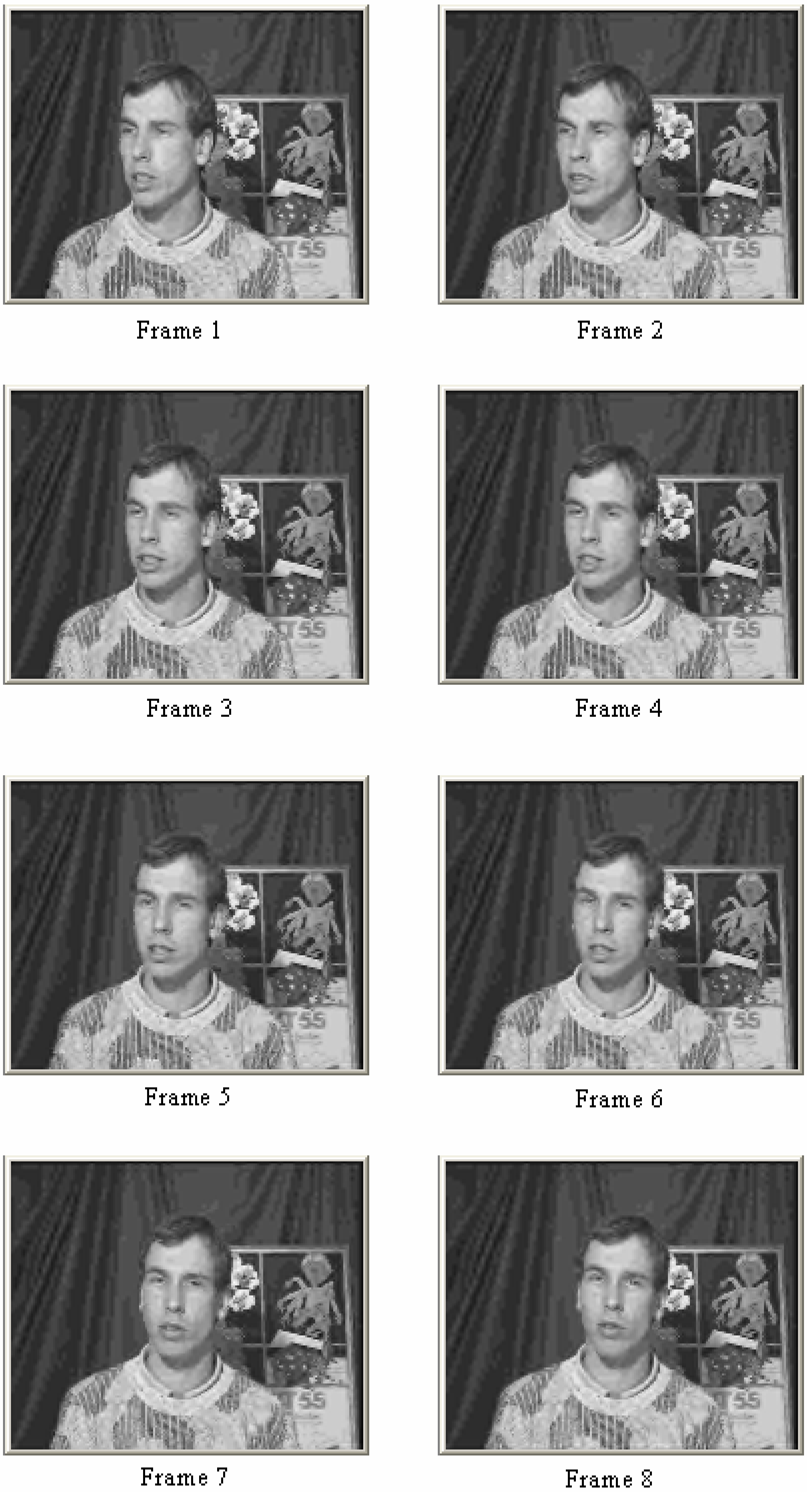
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.29 Reconstructed "*Conference*" Video (3-D H + Db-9/7, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

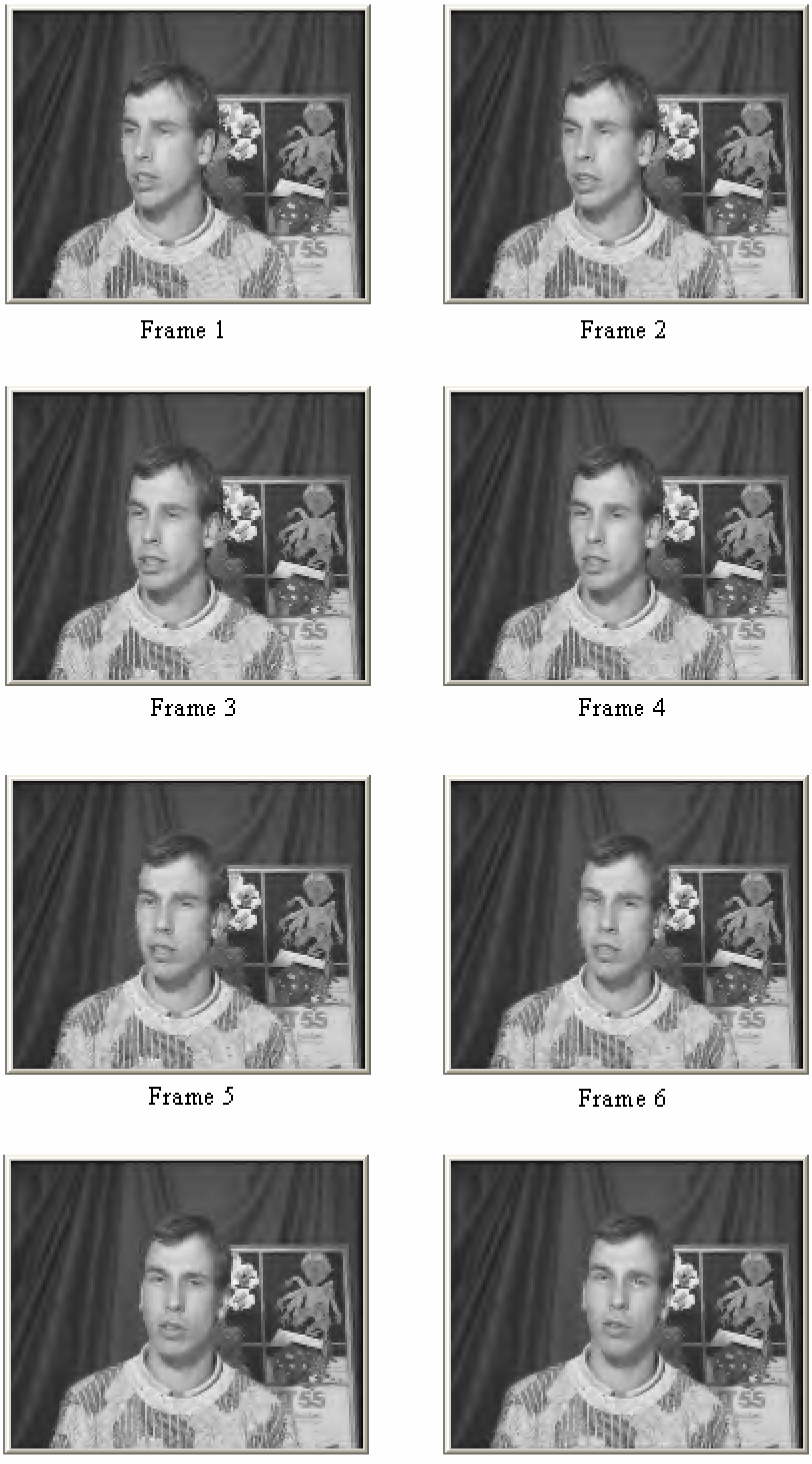
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.30 Reconstructed “*Conference*” Video (MC 3-D H + Db-9/7, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

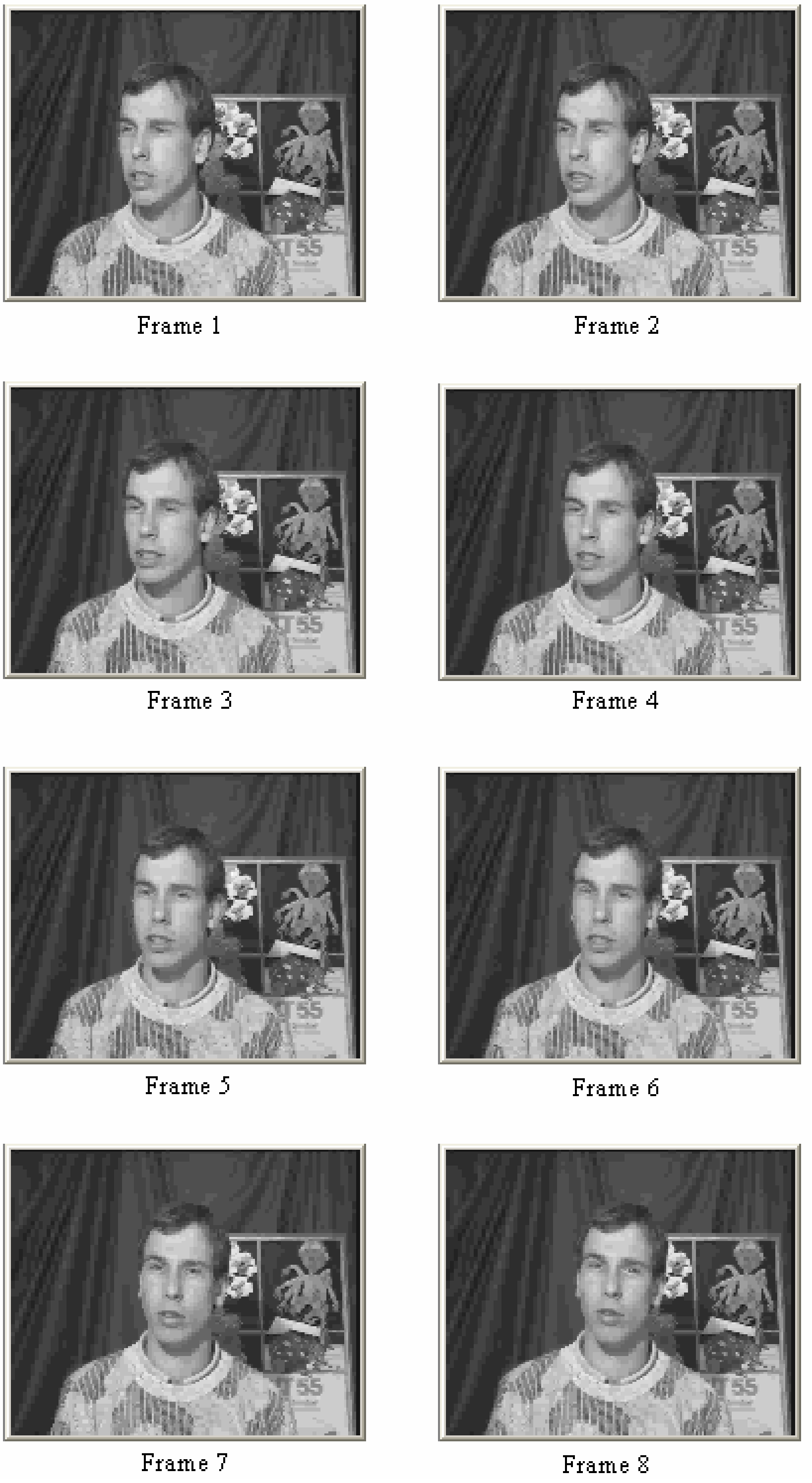
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.31 Reconstructed "*Conference*" Video (MC 3-D H + Db-9/7, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

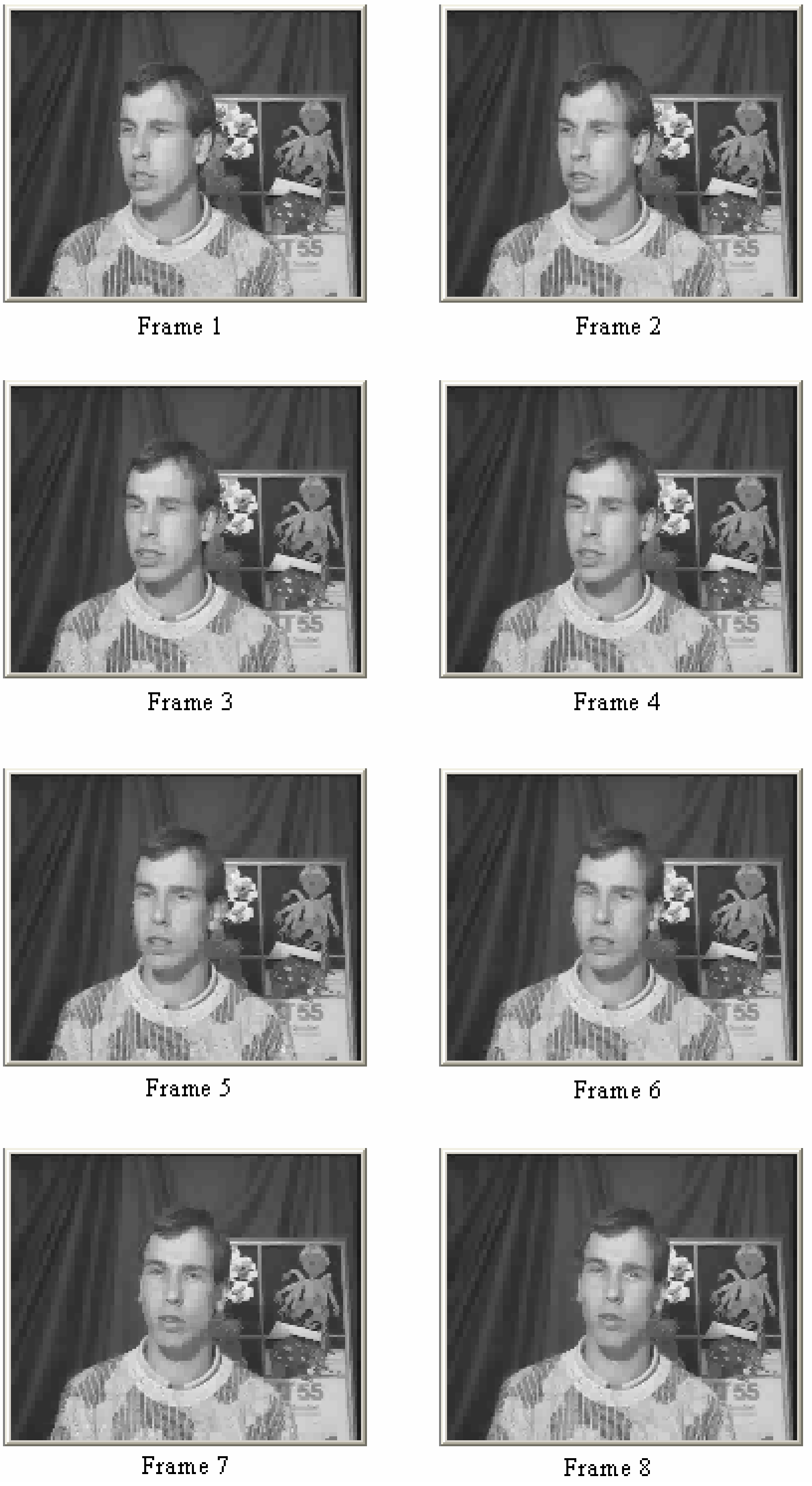
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.32 Reconstructed “*Conference*” Video (MC 3-D Haar, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

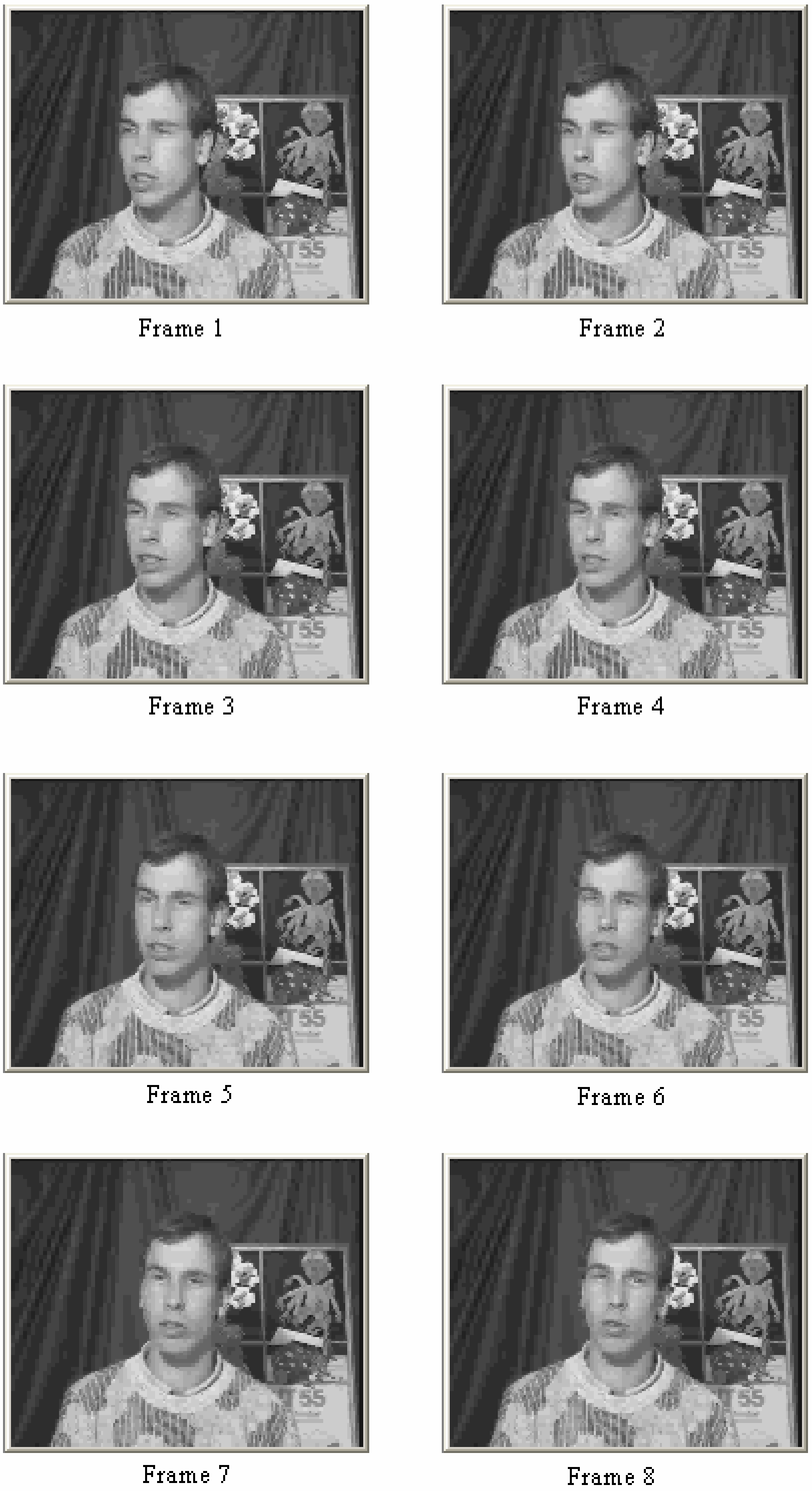
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.33 Reconstructed “*Conference*” Video (MC 3-D Haar, L-2, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

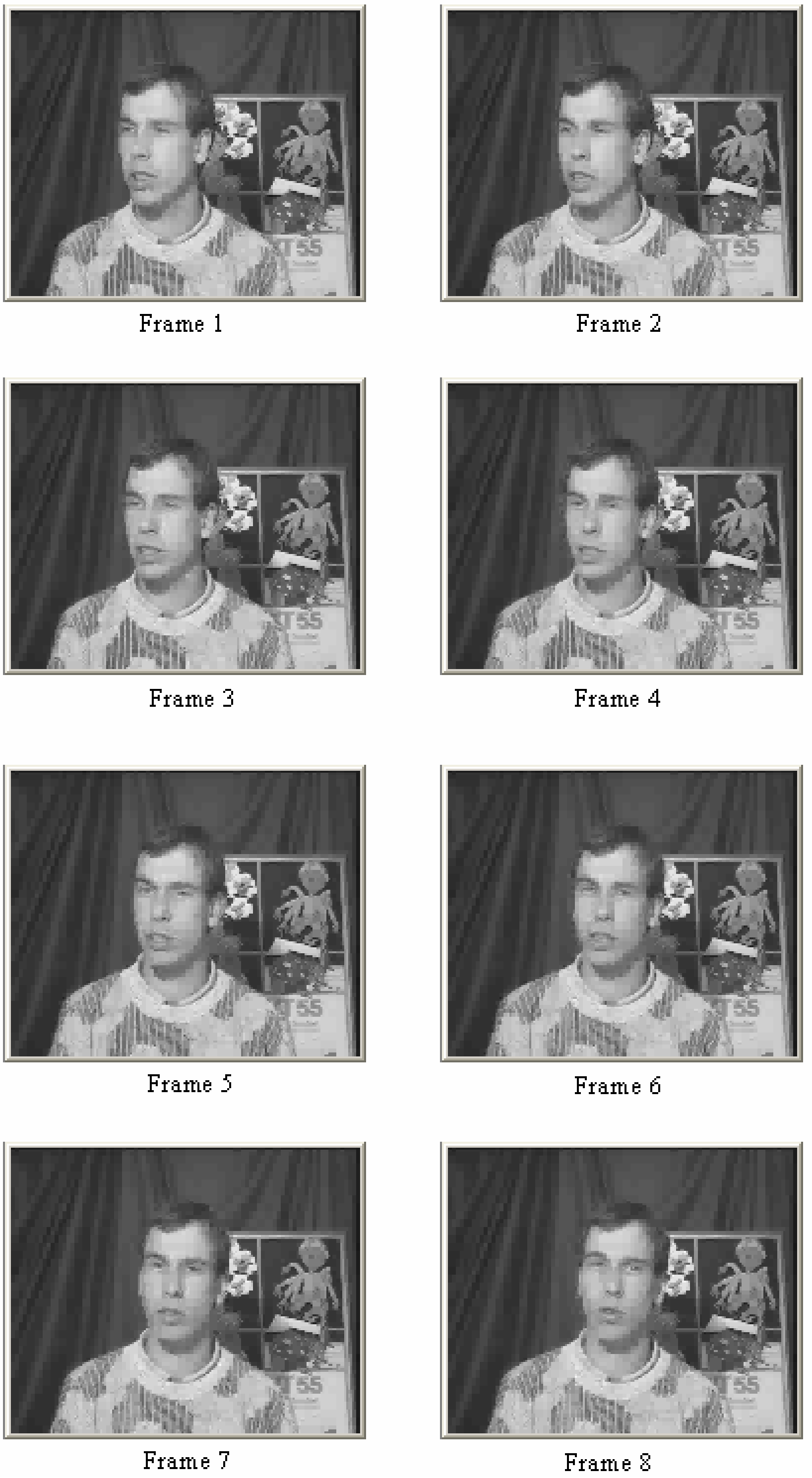
Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.34 Reconstructed "*Conference*" Video (3-D Haar, L-1, $T = 2^5$)



Frame 1

Frame 2

Frame 3

Frame 4

Frame 5

Frame 6

Frame 7

Frame 8

Figure 4.35 Reconstructed “*Conference*” Video (3-D Haar, L-2, $T = 2^5$)

CHAPTER 5

CONCLUSIONS

In this thesis, a low bit rate video coding system based on wavelet coding is studied. Besides our initial motivation to make use of the motion compensated wavelet based coding schemes, the other techniques that do not utilize the motion compensation in its coding procedures have also been analyzed on equal footing. The low bit rate requirements are satisfied by eliminating the redundancies existing in raw image and video signals. In redundancy removal procedures, the characteristics of the Human Visual System (HVS) are concerned to save the visual quality of the compressed signals. The experiments revealed that compression of the image/video signals without significant degradation of the visual quality is possible as they contain a high degree of redundant information not perceived by the HVS.

The properties of HVS and several redundancy types are cited in Chapter 2, where specifically the spatial redundancy due to correlation between neighboring pixels in images (frames) is investigated in detail. The removal procedure is achieved via the most common techniques: the DCT and the DWT. These techniques are tested on the "*Lena*" test image. The experiments not only help us to realize both the DCT and the DWT based modern image/video compression schemes but also provide us with comparative inferences. The noticeable *blocking artifacts* inherent in the DCT based reconstructed images vanish in the DWT based reconstructions without sacrificing the visual quality. Especially the images reconstructed from the deeper wavelet decomposition levels have higher PSNRs. Besides the advantages of multi-level decompositions, many alternative solutions to the problem of signal decorrelation are realized via different wavelet filters.

The practical use of the DWT as a tool in the spatial redundancy removal procedures for 2-D signals also leads us to utilize it for 3-D signals in Chapter 3.

The 3-D wavelet decomposition is computed by applying the 1-D WT along the coordinate axes of the video signal. The order of the axes is the time, the row and the column. The synonym "t + 2-D" is intentionally used in place of the "3-D" to emphasize the order of temporal (t) and spatial (2-D) decompositions. Before the realization of the "t + 2-D" scheme, the temporal only decomposition is studied separately. Specifically, the motion compensated (MC) filtering is fully examined using the two-tap Haar filter in the temporal domain. Both with and without MC temporal decomposition practices are joined with the spatial only decomposition technique to form the "t + 2-D" scheme as the next step. Four decomposition test cases are generated to analyze the video signals in three dimensions: (1) 3-D Haar decomposition, (2) MC 3-D Haar decomposition, (3) 3-D Haar & Daubechies-9/7 decomposition, and (4) MC 3-D Haar & Daubechies-9/7 decomposition. In these test cases, all temporal decompositions are performed using only the Haar filter, and either the Haar or the Daubechies-9/7 analysis filters are used for spatial decompositions. The coefficients are generated at 3 successive levels, and their distribution is analyzed by computing the entropies.

The "t + 2-D" analysis on the "*Table Tennis*" video signal evidently revealed the advantage of the spatial-temporal decomposition over the temporal only decomposition. The entropy of coefficients is computed below that computed in the temporal only case. The contribution of 1-level spatial-temporal decomposition to the accumulation of coefficients around zero (the insignificant energy interval) is only achieved by 3 successive MC temporal only decompositions. The multi-level spatial-temporal analysis discloses the *energy compaction property* of the wavelet transform even more. The Daubechies-9/7 filter and the insertion of the MC temporal filtering into the "t + 2-D" decomposition provide better signal decorrelation as well. The advantages of both are realized especially when used over the signals with larger dimensions. Since the scale and the resolution of the video signal lessen at deeper levels, the benefit from the longer tap filters and from the motion estimation (ME) are reduced. Unfortunately, there is a trade off between the operational complexity and the utilization of Daubechies-9/7 filter and motion compensated filtering. The necessity to maintain the additional number of wavelet coefficients and the newly generated motion vectors together with the dominant algorithmic complexity of the block matching ME approach increases the decomposition times.

The reconstruction of video signals is performed for all test cases in the analysis phase. The majority of the wavelet coefficients are assigned to zero and only the remaining minority is used prior the "t + 2-D" synthesis operations. The coefficients ignored belong to the high subbands and have smaller magnitudes close to zero. The Peak Signal to Noise Ratios (PSNRs) are computed as a measure of the quality of reconstructed signals. Consistent with the entropy analysis, all compositions starting from deeper levels yield better PSNRs. Experimental results reveal that the compositions from the first level do not give a satisfactory visual quality if a very small fraction of the coefficients are used in the reconstruction. On the other hand, the improvement from level-1 to level-2 is more than 20 dB and from level-2 to level-3 is about 2 db. The best case reconstruction is obtained in the MC Haar test case at level-3, and the improvement from the nearest case (Haar at level-3) is about 0.5 dB for the "*Table Tennis*" video signal.

Regarding the PSNRs, the compositions by the Haar filter turn out to be more advantageous than the Daubechies-9/7 filter at deeper levels. However, the latter is more beneficial when used over the signals with more complex spatial frequencies such as the "*Conference*" video. The experiments with the "*Conference*" video signal yield the best PSNR for the MC Daubechies-9/7 test case at level 3. The improvement from the nearest case (MC Haar at level-3) is about 0.1 db. Another point related to the Daubechies-9/7 filter is their smoothing effects on the reconstructed frames. As opposed to the Daubechies-9/7 filter, the Haar filter causes rectangular regions to appear where spatial frequency changes occur in the reconstructed frames. The difference between the two can be perceived by the naked eye.

Although the PSNR improvements are not significant, all cases using MC technique are in general better than those not utilizing it in the above experiments. In the cases where the highest PSNRs are obtained, the improvement due to MC decomposition is about 0.5 dB for the "*Table Tennis*" and 0.7 dB for the "*Conference*" video signals. Without motion compensation, the continuous play is lost when the reconstruction is performed with a very small fraction of the coefficients at level-1. It is perhaps the most significant advantage of utilizing the MC technique to save the continuity of video play between the successive frames.

The observations in Chapter 3 gave ideas about the "t + 2-D" analysis of video signals. Reconstructions using a certain fraction of the most significant coefficients provided intuitive results about the signal compression. A better quantitative expression about the compression ratio is introduced in Chapter 4. The bit representation of wavelet coefficients is achieved by using the "*Set Partitioning in Hierarchical Trees*" (SPIHT) algorithm. The word "2-D SPIHT" is intentionally used instead of "SPIHT" to emphasize its derivation as an embedded image coding algorithm. It partitions the subsets of wavelet coefficients and expresses their significance information. The significance order of the coefficients are determined and conveyed in the *sorting passes* and the data bits of coefficients are transmitted in the *refinement passes*.

The SPIHT is first applied to individual frames of the spatially (2-D only) decomposed videos. The spatial decompositions are obtained by the use of Haar and Daubechies-9/7 filters. For the 3-D decomposed videos, the SPIHT is applied to spatial-temporal subband frames. The four decomposition test cases are identical to those in Chapter 3 for the "t + 2-D" analysis. The data in compressed form is consists of the bit count representing the greatest coefficient, the refinement bits and the sorting bits. Only the number of refinement and sorting bits are taken into account when computing the compression ratios. Different thresholds (quantization step values) are set for both the encoder and the decoder, corresponding to the different bit rates in all test cases.

The experiments on the "*Table Tennis*" and "*Conference*" videos revealed that multi-level decompositions yield higher compression ratios. The number of sorting and refinement bits lessens at level-2 no matter the filter type is. It causes the number of sorting bits to decrease moderately; and the number of refinement bits to decrease significantly (up to 4 times) depending on the threshold value applied. Besides the multi-level decomposition, the threshold value is another factor which increases the amount of compression considerably. The ratio reaches 1:9 at level-2 in the case of greatest threshold value. Although the compression ratio increases in parallel with the increase of threshold value, the reconstructed signals suffer from the lack of refinement information. For all threshold values, the majority of the bits available are used to determine the significance map in the sorting passes. This reduces the fraction of available bits needed in the refinement passes drastically such that the number of sorting bits may become 2 to 14 times the refinement bits. Correspondingly, the overall encoding time is mostly determined by the times spent during the sorting passes.

In the MC test cases, the number of refinement bits is less and the number of sorting bits is more compared to other cases not using the MC. Because the decrease in refinement bits is not significant and the increase in sorting bits is relatively higher than the changes in the number of refinement bits, the compression ratio is mostly determined by the number of sorting bits. The MC filtering modifies the magnitudes of coefficients in the regions where "*unconnected*" or "*double connected*" samples are determined in such a way that the spatial orientation of a sample deviates at finer levels. Thus, the compression ratios in the MC test cases are slightly less than those in the other cases despite the fact that the motion vectors are not coded. The compression ratios obtained in the case of Daubechies-9/7 filter are lower than those obtained in the Haar filter case either. Besides the supplemental coefficients, the lack of spatial orientation in the videos spatially decomposed by the Daubechies-9/7 filter also affects the total number of bits used adversely. This adverse effect is especially valid for the sorting bits in the "*Table Tennis*" video signal, and diminishes for the "*Conference*" video signal.

An important point about the different encoding performances and hence the compression ratios is realized between the 3-D (t + 2-D) and 2-D decomposed videos. In the case of 3-D decompositions, the number of sorting bits increases with an order of 2, whereas the number of refinement bits decreases with an order of 10. Regardless of the large difference between the numbers of bits, the compression ratios do not alter much since the number of sorting bits is much more than the refinement bits. The most significant advantage of preferring the "t + 2-D" scheme is its contribution to PSNR improvement of up to 2 dB for the same compression ratios. Actually, the most important factors that affect the quality of the reconstructed (decoded) videos in either scheme are the multi-level signal decomposition and the high threshold value. The improvement in the signal quality from level-1 to level-2 is about 3 dB and 4 dB for the 2-D and "t + 2-D" analyses, respectively. They are achieved at the highest threshold value applied in the experiments. The contribution of the other factors like the filter type and the MC filtering is not much as the decomposition level and the threshold. The utilization of Daubechies-9/7 filter is slightly advantageous in decoding the "*Conference*" video. The benefit is up to 0.4 dB at high threshold values. On the other hand, the MC decomposition seems to be slightly beneficial for both videos only at high threshold values. The maximum PSNR improvement due to motion compensation is 0.4 dB, which is achieved at level-2 in the Haar test cases.

In this thesis, mostly the spatial and temporal redundancy elimination is of concern. The probable contribution of the spectral redundancy removal is briefly mentioned in Chapter 4; only the luminance components of the test video signals are processed. The experiments and the test cases reveal us that among many factors, the multi-level decomposition is the most important one in the wavelet based video compression and coding schemes. This study can be extended to comprehend the spectral and statistical redundancy removal operations as well. From the compression point of view, the improvements in the sorting mechanism of the "2-D SPIHT" would be more helpful. Moreover, the "3-D" version of the SPIHT waits in front of us to fully achieve an embedded video coding.

REFERENCES

[1] Collin E. Manning, http://newmediarepublic.com/dvideo/, 2004.

[2] JPEG Committee, “JPEG2000-Links”, http://www.jpeg.org , 2004.

[3] A. Secker and D. Taubman, “Motion Compensated Highly Scalable Video Compression Using an Adaptive 3D Wavelet Transform Based on Lifting,” in Proceedings of the IEEE International Conference on Image Processing, Thessaloniki, Greece, Oct. 2001.

[4] H. Khalil, A. F. Atiya, “Three-Dimensional Video Compression Using Subband/Wavelet Transform with Lower Buffering Requirements”, IEEE Trans. on Image Processing, vol. 8, pp. 762-773, Jun. 1999.

[5] A. Aksay, “Motion Wavelet Video Compression”, A master thesis submitted to Electrical and Electronics Engineering Dept. METU, Jan. 2001.

[6] A. M. Tekalp, “Digital Video Processing”, Prentice Hall, Inc, 1995.

[7] “MPEG-2 Video”, ITU-T Recommendation H.262-ISO/IEC 13818-2, Jan. 1995.

[8] “Video Coding for Low Bitrate Communication”, ITU-T Recommendation H.263, Dec. 1995.

[9] D. Lee Gall, “MPEG: A Video Compression Standard for Multimedia Applications”, Commun. ACM, vol. 34, no. 4, pp. 47-58, Apr. 1991.

[10] YQ. Zhang, S. Zafar, “Motion-Compensated Wavelet Transform Coding for Color Video Compression”, IEEE Trans. Circuits Syst. Video Tech., vol. 2, pp. 285-296, Sep. 1992.

[11] P.H. Westerink, J. Biemond, F. Muller, “Subband Coding of Image Sequences at Low Bit Rates”, Signal Processing: Image Commun., vol. 2, pp. 441-448, 1990

[12] S. A. Martucci, I. Sodagar, “A Zerotree Wavelet Video Coder”, IEEE Trans. Circuits Syst. Video Tech., vol. 7, pp. 109-118, Feb. 1997.

[13] Nicoulin *et al.*, “Image Sequence Coding Using Motion Compensated Subband Decomposition”, in Motion Analysis and Image Sequence Processing. Boston, MA: Kluwer, 1993.

[14] Jens-Rainer Ohm, “Three Dimensional Subband Coding with Motion Compensation”, IEEE Transactions on Image Processing, vol. 3, pp. 559-571, Sep. 1994.

[15] D. Taubman, A. Zakhor, “Multirate 3-D Subband Coding of Video”, IEEE Trans. on on Image Processing, vol. 3, pp. 572-588, Sep. 1994

[16] Seung-Jong Choi, John W. Woods, “Motion Compensated 3-D Subband Coding of Video”, IEEE Trans. on Image Processing, vol. 8, pp. 155-167, Feb. 1999.

[17] M. Antonini, M. Barlaud, , I. Daubechies, “Image Coding Using Wavelet Transform”, IEEE Trans. on Image Processing, vol. 1, pp. 205-220, Apr. 1992.

[18] J. M. Shapiro, “Embedded Image Coding Using Zerotrees of Wavelet Coefficients”, IEEE Trans. on Signal Processing, vol. 41, pp. 3445-3462, Dec. 1993.

[19] Amir Said, W. A. Pearlman, “A New, Fast, and Efficient Image Codec Based On Set Partitioning in Hierarchical Trees”, IEEE Trans. Circuits Syst. Video Tech., vol. 6, pp. 243-250, Jun. 1996.

[20] Beong-Jo Kim, Z. Xiong, W. A. Pearlman, “Low Bit-Rate Scalable Video Coding with 3-D Set Partitioning in Hierarchical Trees (3-D SPIHT)”, IEEE Trans. Circuits Syst. Video Tech., vol. 10, pp. 1374-1387, Dec. 2000.

[21] Z. Xiong, K. Ramchandran, M. T. Orchard, and Y.-Q. Zhang, “A Comparative Study of DCT and Wavelet-Based Image Coding”, IEEE Trans. Circuits Syst. Video Tech., vol. 9, pp. 692-695, Aug. 1999.

[22] W. B. Pennebaker, J. L. Mitchell, “JPEG Still Image Data Compression Standard”, New York: Van Nostrand Reinhold, 1992.

[23] J. W. Woods, J. R. Ohms, “Special Issue on Subband Wavelet Interframe Video Coding,” Signal Processing: Image Communication 19 (2004) 557-559

[24] R. Calderbank, I. Daubechies, W. Sweldens, B. Yeo, “Wavelet Transforms that Map Integers to Integers,” Applied and Computational Analysis, vol. 5, pp. 332-369, Jul. 1998.

[25] B. Pesquet-Popescu and V. Bottreau, “Three-Dimensional Lifting Schemes for Motion Compensated Video Compression,” in IEEE Int. Conf. on Acoustics, Speech and Signal Proc., Salt Lake City, UT, May 2001.

[26] T. Resert, K. Hanke, C. Mayer, “Enhanced Interframe Wavelet Video Coding Considering the Interrelation of Spatio-Temporal Transform and Motion Compensation”, Signal Processing: Image Communication 19 (2004) 617-635.

[27] G. Pau, C. Tillier, B. P. Papescu, H. Heimans, “Motion compensation and scalability in lifting-based video coding”, Signal Processing: Image Communication 19 (2004) 677-600.

[28] L. Lua, F. Wu, S. Li, Z. Xiong, Z. Zhuang, “Advanced motion threading for 3D wavelet video coding”, Signal Processing: Image Communication 19 (2004) 601-616.

[29] Y. Andreopoulos, A. Munteanu, J. Barbarien, M. V. Schaar, J. Cornelis, P. Schelkens, “In-band Motion Compensated Temporal Filtering”, Signal Processing: Image Communication 19 (2004) 653-673.

[30] M. Flierl, B. Girod, “Video Coding With Motion Compensated Lifted Wavelet Transforms”, Signal Processing: Image Communication 19 (2004) 561-575.

[31] X. Li, “Scalable Video Compression via Overcomplete Motion Compensated Wavelet Coding,” Processing: Image Communication 19 (2004) 637–651.

[32] J. W. Woods and G. Lilienfield, "A Resolution and Frame-Rate Scalable Subband/Wavelet Video Coder," IEEE Trans. Circuits and Systems for Video Technology, vol. 11, no.9, pp. 1035-1044, Sep. 2001.

[33] M. Vetterli, "Wavelets and Subband Coding", University of California at Berkley, Publications, pp.156, 1995.

[34] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Trans. Pattern Anal. Mach. Intel., vol. 11, Jul. 1989.

[35] Y. Meyer, *Wavelets: Algorithms and Applications,* Society for Industrial and Applied Mathematics, Philadelphia, 1993, pp. 13-31, 101-105

[36] Documentation of Intel® OpenCV Image Processing Library, http://www.sourceforge.net/projects/opencvlibrary, ver3.2, 2002.

[37] J. S. Slim, "Two-Dimensional Signal and Image Processing", Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, pp.613, 1990.

[38] L. Nachtergaele, B. Vanhoof, M. Peon, G. Lafruit, J. Bormans, I. Bolsens, "Implementation of a Scalable MPEG-4 Wavelet-Based Video Texture Compression System", IMEC, 1999.

[39] MPEG Committee, http://www.mpeg.org, 2004.

[40] A. Graps, "An Introduction to Wavelets," IEEE Computational Science and Engineering, vol. 2, num. 2, Summer 1995.

[41] Y. Meyer, "Wavelets: Algorithms and Applications", Society for Industrial and Applied Mathematics, Philadelphia, 1993, pp. 13-31, 101-105.

[42] D. S. Cruz, T. Ebrahimi, J. Askelof, M. Larsson, C. A. Christopoulos, "JPEG2000 Still Image Coding versus Other Standards", Applications of Digital Image Processing XXIII Proc. SPIE vol.4115, Dec. 2000.

APPENDIX A

COLOR SPACE CONVERSION

A.1 RGB ↔ YUV Color Space Conversion

RGB are primary colors. A color is produced by adding the three components, red, green, and blue. YUV is used in European TVs. Y is linked to the component of luminance, and U, V are linked to the component of chrominance. Equation B.1 shows conversion from RGB space to YUV and Equation B.2 shows YUV space to RGB.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{A.1}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.14 \\ 1 & -0.396 & 0.581 \\ 1 & 2.029 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \tag{A.2}$$

APPENDIX B

DISCRETE WAVELET TRANSFORM

B.1 Discrete Wavelet Analysis/Decomposition

The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. The Figure B.1 depicts a 1-level of decomposition of signal, $x[n]$ into different frequency bands.

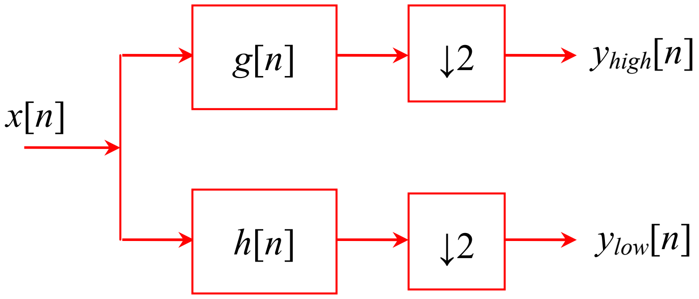


Figure B.1 1-Level Wavelet Analysis

In Figure B.1, the original signal $x[n]$ is first passed through the analysis filters $g[n]$ and $h[n]$ with half band highpass and half band lowpass filter characteristics respectively. After the filtering, the signal is sub-sampled by 2, simply by discarding every other sample. The $y_{high}[n]$ and $y_{low}[n]$ are the outputs of the decomposition of $x[n]$ and they are called the *wavelet coefficients*. They can mathematically be expressed as follows:

$$y_{high}[n] = \sum\nolimits_{k=-\infty}^{\infty} x[k].g[2n-k] \tag{B.1}$$

$$y_{low}[n] = \sum\nolimits_{k=-\infty}^{\infty} x[k].h[2n-k] \tag{B.2}$$

For the reconstruction, the DWT coefficients are up-sampled by 2, passed through the synthesis filters $g'[n]$, and h'[n]. The then the outputs of the filters are added. The procedure is depicted in Figure B.2.

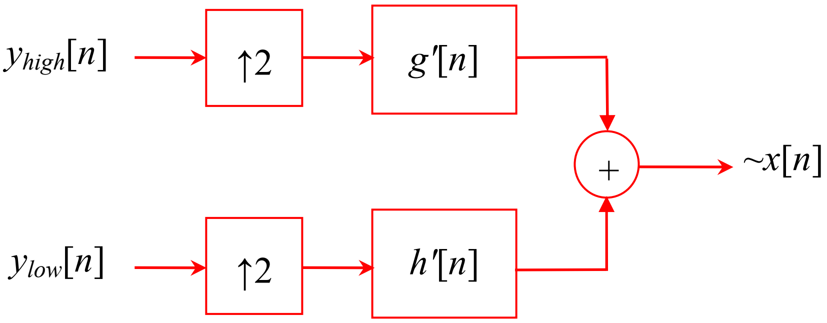


Figure B.2 1-Level Wavelet Synthesis

Therefore, the reconstruction formula becomes:

$$\sim x[n] = \sum_{k=-\infty}^{\infty} \left[\left(y_{high}[k].g[-n+2k] \right) + \left(y_{low}[k].h[-n+2k] \right) \right] \tag{B.3}$$

The 1-D analysis and sythesis schemes shown in Figures B.1 and B.2 can be successively repeated for further decompositions and compositions as illustareted in Figures B.3 and B.4, respectively. This way multi-level analysis and syntehesis of a signal is achieved.

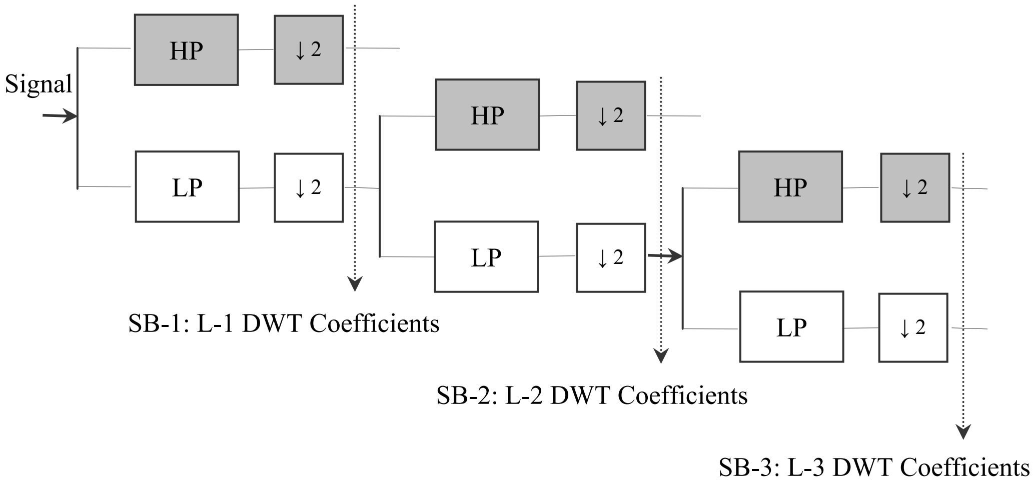

Signal
HP
↓2
LP
↓2
HP
↓2
LP
↓2
HP
↓2
LP
↓2
SB-1: L-1 DWT Coefficients
SB-2: L-2 DWT Coefficients
SB-3: L-3 DWT Coefficients

Figure B.3 3-Level Wavelet Analysis

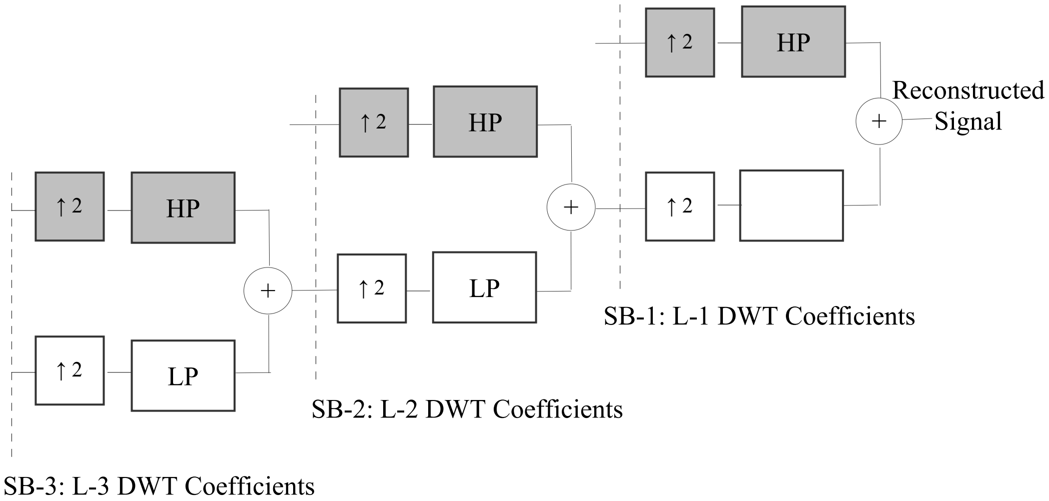

↑2
HP
↑2
LP
+
↑2
HP
↑2
LP
+
↑2
HP
↑2
+
Reconstructed Signal
SB-1: L-1 DWT Coefficients
SB-2: L-2 DWT Coefficients
SB-3: L-3 DWT Coefficients

Figure B.4 3-Level Wavelet Synthesis

APPENDIX C

TWO-DIMENSIONAL (2-D) SPIHT ALGORITHM

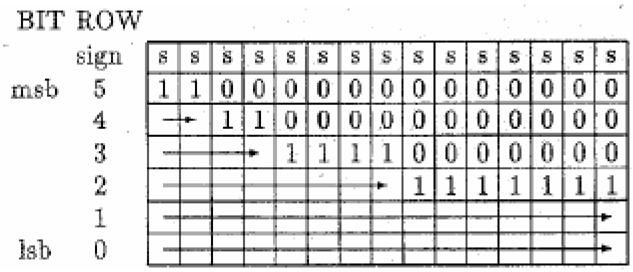
The SPIHT technique is first introduced as a new, fast, and efficient image codec based on set partitioning in hierarchical trees and its detailed explanation can be found in [19]. Regarding the implementation, we summarized this study below.

C.1 Transmission of the Coefficient Values

Let us assume the wavelet transform coefficients, $c_{\eta(k)}$ are ordered according to the minimum number of bits required for its magnitude binary representation, that is, ordered according to a one-to-one mapping η: $I \rightarrow I^2$, such that

$$\left\lfloor \log_2 \left| c_{\eta(k)} \right| \right\rfloor \geq \left\lfloor \log_2 \left| c_{\eta(k+1)} \right| \right\rfloor \text{ for } k = 1,\ldots, N. \tag{C.1}$$

Figure C.1 shows the schematic binary representation of a list of magnitude-ordered coefficients. Each column k in Figure C.1 contains the bits of $c_{\eta(k)}$. The bits in the top row indicate the sign of the coefficients. The rows are numbered from the bottom up, and the bits in the lowest row are the least significant.



BIT	ROW															
	sign	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
msb	5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	→		1	1	0	0	0	0	0	0	0	0	0	0	0
	3	→				1	1	1	1	0	0	0	0	0	0	0
	2	→								1	1	1	1	1	1	1
	1	→														
lsb	0	→														

Figure C.1 Binary Representation of the Magnitude Ordered Coefficients

Now, let us also assume that, (besides the ordering information), the decoder also receives the numbers μ_n corresponding to the number of coefficients such that $2^n \leq | c_{i,j} | < 2^{n+1}$. In the example of Figure C.1, we have $\mu_5 = 2$, $\mu_4 = 2$, $\mu_3 = 4$, etc. Since the transformation is unitary, all bits in a row have the same information content, and the most effective order for progressive transmission is to sequentially send the bits in each row, as indicated by the arrows in Figure C.1. Note that, because the coefficients are in decreasing order of magnitude, the leading "0" bits and the first "1" of any column do not need to be transmitted, since they can be inferred from the μ_n and the ordering.

The progressive transmission method outlined above can be implemented with the following algorithm to be used by the encoder.

Algorithm I:

1) Output $n = \lfloor \log_2 (\max_{(i,j)} \{|c_{i,j}|\}) \rfloor$ to the decoder;

2) Output μ_n, followed by the pixel coordinates $\eta(k)$ and sign of each of the μ_n coefficients such that $2^n \leq | c_{\eta(k)} | < 2^{n+1}$ (**sorting pass**);

3) Output the nth most significant bit of all the coefficients with $| c_{i,j} | \geq 2^{n+1}$ (i.e., those that has their coordinates transmitted in previous sorting passes), in the same order used to send the coordinates (**refinement pass**);

4) Decrement n by 1, go to **Step 2**.

The fact that this coding algorithm uses uniform scalar quantization may give the impression that it must be inferior to other methods that use nonuniform and/or vector quantization. However, this is not the case: the ordering information makes this simple quantization method very efficient. On the other hand, a large fraction of the "*bit budget*" is spent in the sorting pass, and the sophisticated coding methods are needed.

C.2 Set Partitioning Sorting Algorithm

One of the main features of the SPIHT method is that the ordering data is not explicitly transmitted. Instead, it is based on the fact that the execution path of any algorithm is defined by the results of the comparisons on its branching points. So, if the encoder and decoder have the same sorting algorithm, then the decoder can duplicate the encoder's execution path if it receives the results of the magnitude comparisons, and the ordering information can be recovered from the execution path.

One important fact used in the design of the sorting algorithm is that not all coefficients are sorted. Actually, the algorithm simply selects the coefficients such that $2^n \leq |c_{i,j}| < 2^{n+1}$ with n decremented in each pass. Given n, if $|c_{i,j}| \geq 2^n$ then we say that a coefficient is significant; otherwise it is called insignificant.

The sorting algorithm divides the set of pixels into partitioning subsets T_m and performs the magnitude test

$$\max_{(i,j)\in T_m} \{|c_{i,j}|\} \geq 2^n \tag{C.2}$$

If the decoder receives a "no" to that answer (the subset is insignificant), then it knows that all coefficients in T_m are insignificant. If the answer is "yes" (the subset is significant), then a certain rule shared by the encoder and the decoder is used to partition T_m into new subsets $T_{m,l}$ and the significance test is then applied to the new subsets. This set division process continues until the magnitude test is done to all single coordinate significant subsets in order to identify each significant coefficient.

To reduce the number of magnitude comparisons (message bits), a set partitioning rule that uses an expected ordering in the hierarchy (defined by the subband pyramid) is defined. The objective is to create new partitions such that subsets expected to be insignificant contain a large number of elements, and subsets expected to be significant contain only one element.

To clarify the relationship between magnitude comparisons and message bits, the following function is used

$$S_n(T) = \begin{cases} 1, \max\limits_{(i,j)\in T}\left\{\left|c_{i,j}\right|\right\} \geq 2^n, \\ 0, \quad otherwise \end{cases} \tag{C.3}$$

is used to indicate the significance of a set of coordinates *T*. To simplify the notation of single pixel sets, we write $S_n(\{(i, j)\})$ as $S_n(i, j)$.

C.2 Spatial Orientation Trees

Normally, energy of an image signal is mostly concentrated in the low frequency components. Consequently, the variance decreases as we move from the highest to the lowest levels of the subband pyramid. Furthermore, a spatial self-similarity between subbands has been observed, and the coefficients are expected to be better magnitude ordered if we move downward in the pyramid following the same spatial orientation.

A tree structure, called *spatial orientation tree*, naturally defines the spatial relationship on the hierarchical pyramid. Figure C.2 shows how the spatial orientation tree is defined in a pyramid constructed with recursive four-subband splitting. Each node of the tree corresponds to a pixel and is identified by the pixels'

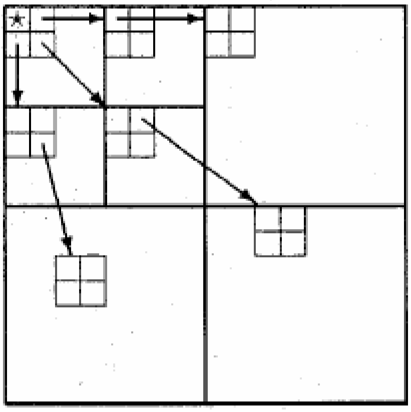


Figure C.2 Parent-Offspring Dependencies in the Spatial-Orientation Tree

coordinate. Its direct descendants (offspring) correspond to the pixels of the same spatial orientation in the next finer level of the pyramid. The tree is defined in such a way that each node has either no offspring (the leaves) or four offspring, which always form a group of 2×2 adjacent pixels. In Figure C.2, the arrows are oriented from the parent node to its four offspring. The pixels in the highest level of the pyramid are the tree roots and are also grouped in 2×2 adjacent pixels. However, their offspring branching rule is different, and in each group, one of them (indicated by the star in Figure C.2) has no descendants.

The following sets of coordinates are used to present the coding method:

- $O(i, j)$: set of coordinates of all offspring of node (i, j);
- $D(i, j)$: set of coordinates of all descendants of the node (i, j);
- H: set of coordinates of all spatial orientation tree roots (nodes in the highest pyramid level);
- $L(i, j)$: $D(i, j) – O(i, j)$.

For instance, except at the highest and lowest pyramid levels, we have

$$O(i, j) = \{(2i, 2j), (2i, 2j+1), (2i+1, 2j), (2i+1, 2j+1)\} \qquad \text{(C.4)}$$

The parts of the spatial orientation trees are used as the partitioning subsets in the sorting algorithm. The set partitioning rules are simply following:

1. The initial partition is formed with the sets $\{(i, j)\}$ and $D(i, j)$, for all (i, j) Є H.
2. If $D(I, j)$ is significant, then it is partitioned into $L(i, j)$ plus the four single-element sets with (k, l) Є $O(i, j j)$.
3. If $L(i, j)$ is significant, then it is partitioned into the four sets $D(k, l)$ with (k, l) Є $O(i, j)$.

C.3 Coding Algorithm

In practice, the significance information is stored in three ordered lists, called list of insignificance sets (LIS), list of insignificance pixels (LIP), and list of significant pixels (LSP). In all lists, each entry is identified by a coordinate *(i, j)*, which in the LIP and LSP represents individual pixels, and in the LIS represents either the set $D(i, j)$ or $L(i, j)$. To differentiate between them, we say that a LIS entry is of type *A* if it represents D and of type *B* if it represents $L(i, j)$.

During the sorting pass (see Algorithm I), the pixels in the LIP – which were insignificant in the previous pass – are tested, and those that become significant are moved to the LSP. Similarly, sets are sequentially evaluated following the LIS order, and when a set is found to be significant it is removed from the list and partitioned. The new subsets with more than one element are added back to the LIS while the single-coordinate sets are added to the end of the LIP or the LDP depending whether they are insignificant or significant, respectively. The LSP contains the coordinates of the pixels that are visited in the refinement pass.

Below, the encoding algorithm is presented in its entirety. It is essentially equal to Algorithm I, but uses the set-partitioning approach in its sorting pass.

Algorithm II:

1) Initialization:

Output $n = \lfloor \log_2(\max_{(i,j)}\{|c_{i,j}|\}) \rfloor$; set the LSP as an empty list, and add the coordinates *(i, j)* Є *H* to the LIP, and only those with descendants also to the LIS, as type *A* entries.

2) Sorting Pass:

```
2.1)  For each entry $(i, j)$ in the LIP do:
      2.1.1)  Output $S_n(i, j)$;
      2.1.2)  If $S_n(i, j) = 1$, then move $(i, j)$ to the LSP and output the sign of $c_{i,j}$;
```

```
2.2)    For each entry $(i, j)$ in the LIS do:
        2.2.1)  If the entry is of type $A$ then
                - Output $S_n(D(i, j))$;
                - If $S_n(D(i, j)) = 1$ then

                        * For each $(k, l) \in O(i, j)$ do:
                                - Output $S_n(k, l)$;
                                - If $S_n(k, l) = 1$ then add $(k, l)$ to the LSP and output the sign of
                                  $c_{k,l}$;
                                - If $S_n(k, l) = 0$ then add $(k, l)$ to the end of the LIP;

                        * If $L(i, j) \neq \emptyset$ then move $(i, j)$ to the end of the LIS, as an entry of type $B$,
                          and go to Step 2.2.2; otherwise, remove entry $(i, j)$ from the LIS;

        2.2.2)  If the entry is of type $B$ then
                - Output $S_n(\mathrm{L}(k, l))$;
                - If $S_n(\mathrm{L}(k, l)) = 1$ then

                        * Add each $(k, l) \in O(i, j)$ to the end of the LIS as an entry of type $A$;

                        * Remove $(i, j)$ from the LIS.
```

3) Refinement Pass:

For each entry (i, j) in the LSP, except those included in the last sorting pass (i.e., with same n), output the nth most significant bit of $| c_{i,j} |$;

4) Quantization-Step Update:

Decrement n by 1 and go to **Step 2**.

One important characteristic of the algorithm is that the entries added to the end of the LIS in Step 2.2, are evaluated before that same sorting pass ends. So, when we say "for each entry in the LIS" we also mean those that are being added to its end.

Note that in Algorithm II, all branching conditions based on the significance data S_n – which can only be calculated with the knowledge of $c_{i,j}$ – are output by the encoder. Thus, to obtain the desired decoder's algorithm, which duplicates the encoder's execution path as it sorts the significant coefficients; simply replace the words "output" by "input" in Algorithm II. Comparing the algorithm above to Algorithm I, it is seen that the ordering information $\eta(k)$ is recovered when the coordinates of he significant coefficients are added to the end of the LSP; that is, the coefficients pointed by the coordinates in the LSP; that is the coefficients pointed b the coordinates in the LSP are sorted as in Equation C.1. But note that whenever the decoder inputs data, its control lists (LIS, LIP, and LSP) are identical to the ones used by the encoder it outputs that data, which means that the decoder recovers the ordering from the execution path. It is easy to demonstrate that coding and decoding have the same computational complexity.