IMPLEMENTATION OF DWT ARCHITECTURE FOR VIDEO CODING

PROJECT REPORT
PHASE-II

Submitted by

GUNANANDHINI S

Register No: 15MAE004

in partial fulfillment for the requirement of award of the degree

of

MASTER OF ENGINEERING

in

APPLIED ELECTRONICS

Department of Electronics and Communication Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY
(An autonomous institution affiliated to Anna University, Chennai)

COIMBATORE-641049

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2017

BONAFIDE CERTIFICATE

Certified that this project report titled **"IMPLEMENTATION OF DWT ARCHITECTURE FOR VIDEO CODING"** is the bonafide work of **S. GUNANANDHINI [15MAE004]** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Prof. S. GOVINDARAJU
PROJECT SUPERVISOR
Department of ECE
Kumaraguru College of Technology
Coimbatore-641 049

SIGNATURE
Prof. K. RAMPRAKASH
HEAD OF PG PROGRAMME
Department of ECE
Kumaraguru College of Technology
Coimbatore-641 049

The candidate with **Register No. 15MAE004** is examined by us in the project viva-voce examination held on............................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First, I would like to express my praise and gratitude to the Lord, who has showered his grace and blessings enabling me to complete this project in an excellent manner.

I express my sincere thanks to the management of Kumaraguru College of Technology and Joint Correspondent **Shri Shankar Vanavarayar** for the kind support and for providing necessary facilities to carry out the work.

I would like to express my sincere thanks to our beloved Principal **Dr. R. S. Kumar,** Kumaraguru College of Technology, who encouraged me with his valuable thoughts.

I would like to thank **Prof. K. Ramprakash,** Head of PG programme, Electronics and Communication Engineering, for his kind support and for providing necessary facilities to carry out the project work.

I am greatly privileged to express my heartfelt thanks to my project guide **Prof. S. Govindaraju,** Department of Electronics and Communication Engineering, for his expert counseling and guidance to make this project to a great deal of success.

In particular, I wish to thank with everlasting gratitude to the project coordinator **Mrs. S. Nagarathinam,** Assistant Professor, Department of Electronics and Communication Engineering, throughout the course of this project work. I wish to convey my deep sense of gratitude to all teaching and non-teaching staff of ECE Department for their help and cooperation.

Finally, I thank my parents and my family members for giving me the moral support and abundant blessings in all of my activities and my dear friends who helped me to endure my difficult times with their unfailing support and warm wishes.

ABSTRACT

Wavelet transform techniques currently provide the most promising approach to high-quality image compression, which is essential for many real-world applications. DWT is used as basis for transformation in JPEG 2000 standard. DWT provides high quality compression at low bit rates. DWT performs better than DCT in the context that it avoids blocking artifacts which degrade the reconstructed images. However, DWT provides lower quality than JPEG at the Low compression rates. DWT requires longer compression time.

Discrete Wavelet Transform (DWT) based algorithm for video coding is proposed in this work. DWT provides good localization both in time and frequency domain and it has high performance than DCT. In this paper, a comparative analysis has been carried on Video compression using Haar, Daubechies and Bi-orthogonal wavelet. The redundancy of the DWT detail coefficients is reduced through quantization technique. It aims to attain minimum error while preserving the high peak signal to noise ratio (PSNR) and quality of the image in the acceptable range. Using the PSNR as a measure of quality, we show that Daubechies and Bi-orthogonal wavelet provides the better quality of video compared to Haar wavelet. At each level, performance is analyzed based on compression ratio, PSNR, MSE, SSIM, AD and NAE in between the frames.

The memory complexity is the most important issue for efficient realization of 2-D DWT in VLSI systems. It suggests to implement memory centric design strategy and based on that a convolution-based generic architecture for the computation of three-level 2-D DWT for Daubechies wavelet filters was proposed in this paper.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SYMBOL	ABBREVIATIONS
DWT	Discrete Wavelet Transform
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
2D	Two Dimension
3D	Three Dimension
RD	Relative data redundancy
HDL	Hardware Description Language
VHDL	Very High Level Design Language
VLSI	Very Large Scale Integration
FPGA	Field Programmable Gate Array
JPEG	Joint Photographic Expert Group
Daub	Daubechies wavelet
bi-ortho	Bi-Orthogonal wavelet
CR	Compression Ratio
PSNR	Peak Signal to Noise Ratio
MSE	Mean Square Error
SSIM	Structural Similarity Index Matrix
AD	Average Difference
NAE	Normalized Absolute Error
CT	Computation Time
PU	Processing Unit
PE	Processing Element

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF IMAGE COMPRESSION

The increasing demand for multimedia content such as digital images and video has led to great interest in research into compression techniques. The development of higher quality and less expensive image acquisition devices has produced steady increases in both image size and resolution, and a greater consequent for the design of efficient compression systems. Although storage capacity and transfer bandwidth has grown accordingly in recent years, many applications still require compression. In general, this thesis investigates video frame compression in the spatial domain. The main objective is to design a compression system suitable for processing, storage and transmission, as well as providing acceptable computational complexity suitable for practical implementation. The basic rule of compression is to reduce the numbers of bits needed to represent an image. In a computer, an image is represented as an array of numbers, integers to be more specific, that is called as digital image. The image array is usually two dimensional (2D), if it is black and white (BW) and three dimensional (3D) if it is colour image. Digital image compression algorithms exploit the redundancy in an image so that it can be represented using a smaller number of bits while still maintaining acceptable visual quality. Redundancy reduction aims at removing duplication from the signal source (image/video).

Factors related to the need for image compression include:

- Large storage requirements for multimedia data
- Low power devices such as handheld phones have small storage capacity
- Network bandwidths currently available for transmission
- Effect of computational complexity on practical implementation

Image compression standards bring about many benefits easier exchange of image files between different devices and applications, reuse of existing hardware and software for a wider array of products, existence of benchmarks and reference data sets for new and alternative developments.

1.2 PRINCIPLES BEHIND COMPRESSION:

Number of bits required to represent the information in an image can be minimized by removing the redundancy present in it. There are three types of redundancies such as spatial redundancy which is due to the correlation or dependence between neighbouring pixel values, spectral redundancy which is due to the correlation between different colour planes or spectral bands, temporal redundancy which is present because of correlation between different frames in images. Image compression research aims to reduce the number of bits required to represent an image by removing the spatial and spectral redundancies as much as possible. Data redundancy is of central issue in digital image compression. If n1 and n2 denote the number of information carrying units in original and compressed image respectively, then the compression ratio CR can be defined as CR=n1/n2.

Relative data redundancy RD of the original image can be defined as RD=1-1/CR.

Three possibilities arise here:

(1) If n1=n2, then CR=1 and hence RD=0 which implies that original image does not contain any redundancy between the pixels

(2) If n1>>n1, then CR→∞ and hence RD>1 which implies considerable amount of redundancy in the original image

(3) If n1<0 and hence RD→-∞ which indicates that the compressed image contains more data than original image

1.3 TYPES OF COMPRESSION:

Lossless versus Lossy compression: In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However lossless compression can only achieve a modest amount of compression. Lossless compression is preferred for archival purposes and often medical imaging, technical drawings, clip art or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. An image reconstruction following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discards redundant information. However, lossy schemes are capable of achieving much higher compression. Lossy methods are especially suitable for natural images such as photos in applications where minor loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visually lossless.

Predictive versus Transform coding:

In predictive coding, information already sent or available is used to predict future values, and the difference is coded. Since this is done in the image or spatial domain, it is relatively simple to implement and is readily adapted to local image characteristics. Differential Pulse Code Modulation (DPCM) is one particular example of predictive coding.

In Transform coding, first transforms the image from its spatial domain representation to a different type of representation using some well-known transform and then codes the transformed values. This method provides greater data compression compared to predictive methods, although at the expense of greater computational requirements.

1.3.1 COMPRESSION RATIO:

Data compression ratio is defined as the ratio between the uncompressed size and compressed size.

$$\text{Compression ratio} = \frac{\textit{Uncompressed data size}}{\textit{compressed data size}}$$

1.4 TRANSFORM CODING:

- Discrete Cosine Transform(DCT)
- Discrete Wavelet Transform(DWT)

1.4.1 DISCRETE COSINE TRANSFORM(DCT):

JPEG stands for the Joint Photographic Experts Group, a standards committee that had its origins within the International Standard Organization (ISO). JPEG provides a compression method that is capable of compressing continuous-tone image data with a pixel depth of 6 to 24 bits with reasonable speed and efficiency. JPEG may be adjusted to produce very small, compressed images that are of relatively poor quality in appearance but still suitable for many applications. Conversely, JPEG is capable of producing very high-quality compressed images that are still far smaller than the original uncompressed data. JPEG is primarily a lossy method of compression. JPEG was designed specifically to discard information that the human eye cannot easily see. DCT separates images into parts of different frequencies where less important frequencies are discarded through quantization and important frequencies are used to retrieve the image during decompression. DCT is used for transformation in JPEG standard. DCT performs efficiently at medium bit rates

Advantages of DCT

- ❖ It has been implemented in single integrated circuit
- ❖ It has the ability to pack most information in fewest coefficients
- ❖ It minimizes the block like appearance called blocking artifact that results when boundaries between sub-images become visible

Limitations of DCT

- ❖ Truncation of higher spectral coefficients results in blurring of the images, especially wherever the details are high
- ❖ Coarse quantization of some of the low spectral coefficients introduces graininess in the smooth portions of the images
- ❖ Serious blocking artifacts are introduced at the block boundaries, since each block is independently encoded, often with a different encoding strategy and the extent of quantization.

1.4.2 DISCRETE WAVELET TRANSFORM(DWT):

Wavelet Transform has become an important method for image compression. Wavelet based coding provides substantial improvement in picture quality at high compression ratios mainly due to better energy compaction property of wavelet transforms. Wavelet transform partitions a signal into a set of functions called wavelets. Wavelets are obtained from a single prototype wavelet called mother wavelet by dilations and shifting. The wavelet transform is computed separately for different segments of the time-domain signal at different frequencies.

1.4.2.1 SUBBAND CODING:

A signal is passed through a series of filters to calculate DWT. Procedure starts by passing this signal sequence through a half band digital low pass filter with impulse response. Filtering of a signal is numerically equal to convolution of the tile signal with impulse response of the filter.

A half band low pass filter removes all frequencies that are above half of the highest frequency in the tile signal. Then the signal is passed through high pass filter. After filtering half of the samples can be eliminated since the signal now has the highest frequency as half of the original frequency. The signal can therefore be subsampled by 2, simply by discarding every other sample. This constitutes 1 level of decomposition.

This decomposition halves the time resolution since only half the number of sample now characterizes the whole signal. Frequency resolution has doubled because each output has half the frequency band of the input. This process is called as sub band coding. It can be repeated further to increase the frequency resolution by the filter bank.

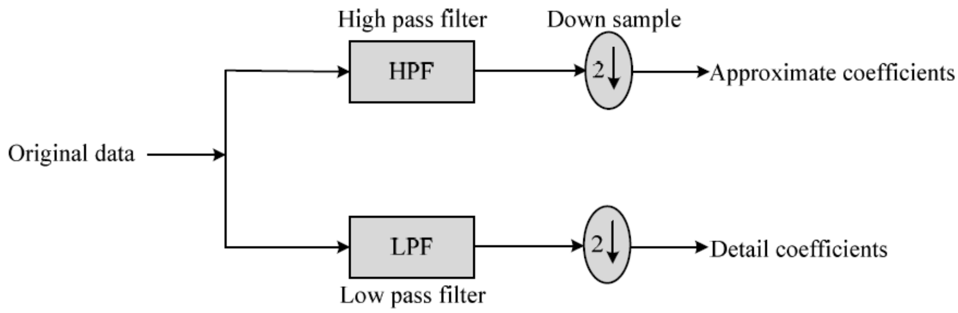


Fig. 1.1 Block diagram of 1-D forward DWT

The output of high pass and low pass filters are down sampled by 2. The output from low pass filter is an approximate coefficient and the output from the high pass filter is a detail coefficient. This procedure is one dimensional (1-D) DWT and Fig. 1.1 shows the schematics of this method.

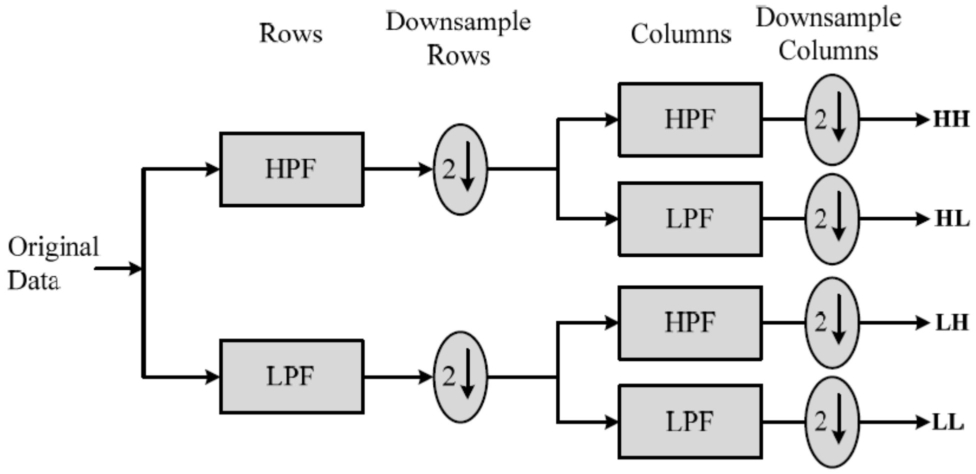


Fig. 1.2 Block diagram of 2-D forward DWT

In case of 2-D DWT, the input data is passed through set of both low pass and high pass filter in two directions, both rows and columns. The outputs are then down sampled by 2 in each direction as in case of 1-D DWT. The complete process is illustrated in Fig. 1.2. As shown in Figure 1.2, output is obtained in set of four coefficients LL, HL, LH and HH. The first alphabet represents the transform in row whereas the second alphabet represents transform in column. The alphabet L means low pass signal and H means high pass signal. LH signal is a low pass signal in row and a high pass in column. Hence, LH signal contain horizontal elements. Similarly, HL and HH contains vertical and diagonal elements, respectively.

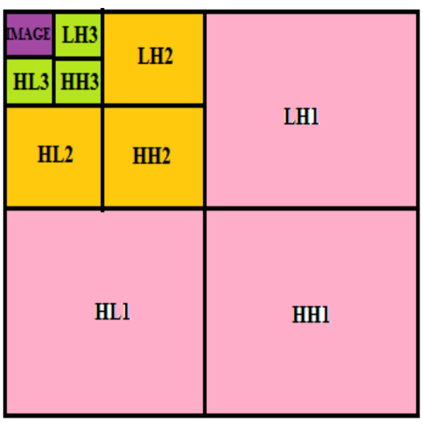


Fig. 1.3 Illustration of Forward DWT

In DWT reconstruction, input data can be achieved in multiple resolutions by decomposing the LL coefficient further for different levels as shown in Fig. 1.3. In order to reconstruct the output data, the compressed data is up-sampled by a factor of 2. The signal is further passed through the same set of high pass and low pass filter in both rows and columns. The entire reconstruction procedure is shown in Fig. 1.4.

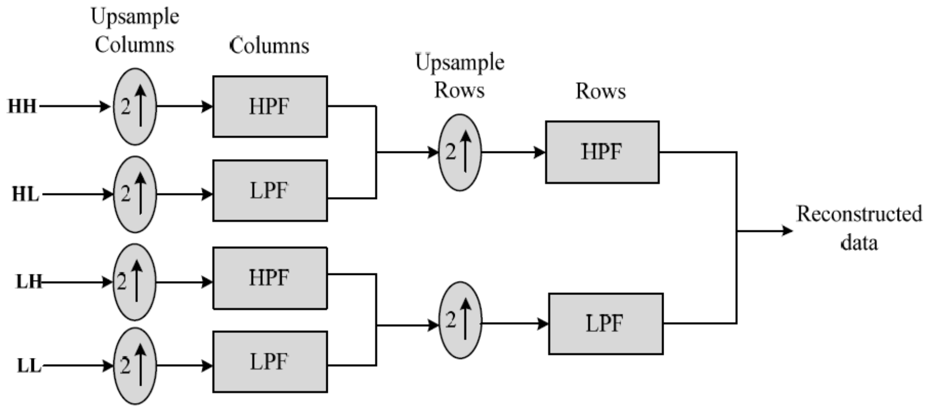


Fig. 1.4 Block diagram of 2-D inverse DWT

Each block is then passed through the two filters: high pass filter and low pass filter. The first level decomposition is performed to decompose the input data into an approximation and the detail coefficients. After obtaining the transformed matrix, the detail and approximate coefficients are separated as LL, HL, LH, and HH coefficients. All the coefficients are discarded, except the LL coefficients. The process continues for one more level. The coefficients are then divided by a constant scaling factor (SF) to achieve the desired compression ratio. Finally, for data reconstruction, the data is rescaled and padded with zeros, and passed through the wavelet filters.

Wavelet transform decomposes a signal into a set of basis functions. These basis functions are called wavelets. Discrete wavelet transform (DWT) which transforms a discrete time signal to a discrete wavelet representation. JPEG 2000 image compression stand makes use of DWT. DWT can be used to reduce the image size without losing much of the resolutions computed and values less than a pre-specified threshold are discarded. Thus, it reduces the amount of memory required to represent given image.

Advantages:

- Wavelet transform is applied to sub images, so it produces no blocking artifacts
- Allows good localization both in time and spatial frequency
- Higher compression ratio

- ❖ Higher performance

Limitations:

- ❖ The cost of computing DWT as compared to DCT may be higher.
- ❖ The use of larger DWT basis functions or wavelet filters produces blurring and ringing noise near edge regions in images or video frames
- ❖ Longer compression time
- ❖ Lower quality than JPEG at low compression rates

1.5 CLASSIFICATION OF VLSI DWT ARCHITECTURES:

TWO-DIMENSIONAL (2-D) discrete wavelet transform (DWT) is widely used in image and video compression. Several architectures have been suggested for efficient VLSI implementation of 2-D DWT for real-time applications. These structures are mostly of two types. Those are (i) convolution based and (ii) lifting-based.

1.5.1 LIFTING BASED SCHEME

Favourable characteristics compared with the convolution-based methods

- ❖ Low demand of arithmetic resources
- ❖ Memory-efficient in-place computation
- ❖ Inherent parallelism

On the other hand, the lifting based architectures suffer from a long critical path

1.5.2 CONVOLUTION BASED SCHEME

Analysis of both lifting and the convolution-based designs,

- ❖ Appropriate scheduling of multilevel decomposition could have lower complexity than the lifting-based design
- ❖ Memory savings offered by the convolution-based scheme is significantly higher than the saving of arithmetic components.

1.6 VIDEO SOURCE CODING BASICS

A digital image or a frame of digital video typically consists of three rectangular arrays of integer-valued samples, one array for each of the three components of a tri-stimulus colour representation for the spatial area represented in the image. Video coding often uses a colour representation having three components called Y, Cb, and Cr. Component Y is called luma, and represents brightness. The two Chroma components Cb and Cr represent the extent to which the colour deviates from gray toward blue and red, respectively. Because the human visual system is more sensitive to luma than Chroma, often a sampling structure is used in which the Chroma component arrays each have only one fourth as many samples as the corresponding luma component array (half the number of samples in both the horizontal and vertical dimensions). This is called 4:2:0 sampling. The amplitude of each component is typically represented with 8 bits of precision per sample for consumer-quality video. The two basic video formats are progressive and interlaced.

A frame array of video samples can be considered to contain two interleaved fields, a top field and a bottom field. The top field contains the even-numbered rows 0, 2... H/2-1 (with 0 being top row number for a frame and H being its total number of rows), and the bottom field contains the odd-numbered rows (starting with the second row of the frame). When interlacing is used, rather than capturing the entire frame at each sampling time, only one of the two fields is captured. Thus, two sampling periods are required to capture each full frame of video. We will use the term picture to refer to either a frame or field. If the two fields of a frame are captured at different time instants, the frame is referred to as an interlaced frame, and otherwise it is referred to as a progressive frame.

1.7 PROJECT OBJECTIVE

The Project aim is

1) To obtain high quality compression at low bit rates
2) To allow good localization both in time and spatial frequency domain
3) To provide the efficient realization of 2-D DWT in VLSI systems

1.8 SOFTWARE USED

- MATLAB R2014a
- Modelsim-Altera 6.3e

1.8.1 MATLAB

The MATLAB is high-performance language for technical computing integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modelling and analysis, and computational biology.

MATLAB provides a number of features for documenting and sharing your work. We can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

Features include:

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java™, COM, and Microsoft Excel
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration

The MATLAB System

The MATLAB system consists of the following parts for the implementation.

(i) Desktop Tools and Development Environment

This part of MATLAB is the set of tools and facilities that help you use and become more productive with MATLAB functions and files. Many of these tools are graphical user interfaces. It includes: MATLAB desktop and Command Window, an editor and debugger, a code analyser, and browsers for viewing help, the workspace, and folders.

(ii) Mathematical Function Library

This library is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

(iii) The Language

The MATLAB language is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object oriented programming features. It allows both "programming in the small" to rapidly create quick programs you do not intend to reuse.

(iv) Graphics

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

(v) External Interfaces

The external interfaces library allows you to write C/C++ and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), for calling MATLAB as a computational engine, and for reading and writing MAT-files.

1.8.2 MODELSIM:

ModelSim PE, our entry-level simulator, offers VHDL, Verilog, or mixed-language simulation. Coupled with the most popular HDL debugging capabilities in the industry, ModelSim PE is known for delivering high performance, ease of use, and outstanding product support. Model Technology’s award-winning Single Kernel Simulation (SKS) technology enables transparent mixing of VHDL and Verilog in one design. ModelSim’s architecture allows platform independent compile with the outstanding performance of native compiled code. An easy-to-use graphical user interface enables you to quickly identify and debug problems, aided by dynamically updated windows. For example, selecting a design region in the Structure window automatically updates the Source, Signals, Process, and Variables windows. These cross linked ModelSim windows create a powerful easy-to-use debug environment. Once a problem is found, you can edit, recompile, and re-simulate without leaving the simulator. ModelSim PE fully supports the VHDL and Verilog language standards. You can simulate behavioral, RTL, and gate-level code separately or simultaneously. ModelSim PE also supports all ASIC and FPGA libraries, ensuring accurate timing simulations. ModelSim PE provides partial support for VHDL 2008.

1.8.2.1 A More Intelligent GUI

An intelligently engineered GUI makes efficient use of desktop real estate. The intuitive arrangement of interactive graphical elements (windows, toolbars, menus, etc.) makes it easy to view and access the many powerful capabilities of ModelSim. The result is a feature rich GUI that is easy to use and quickly mastered. ModelSim redefined openness in simulation by incorporating the Tcl user interface into its HDL simulator. Tcl is a simple but powerful scripting language for controlling and extending applications.

1.8.2.2 Verilog 2001/System Verilog

ModelSim PE now fully supports IEEE 1364-2001, including System Verilog design language features. System Verilog is an Accellera standard that provides new constructs for modeling at higher levels of abstraction.

1.8.2.3 Memory Window

Allows flexible viewing and changing of memory locations. VHDL and Verilog memories are auto extracted in the GUI allowing powerful search, fill, load and save functionality. Memory Window allows pre-loading of memories thus saving the time-consuming step of initializing sections of your simulations just to load memories. All functions are available via the command line allowing their use in scripting.

Source Window Templates and Wizards

VHDL and Verilog templates and wizards allow you to quickly develop HDL code without having to remember the exact language syntax. All the language constructs are available with a click of a mouse. Easy-to-use wizards step you through creation of more complex HDL blocks. The wizards show you how to create parameterizable logic blocks, test bench stimuli, and design objects. The source window templates and wizards benefit both novice and advanced HDL developers with time-saving shortcuts. VHDL and Verilog templates and wizards allow you to quickly develop HDL code without having to remember the exact language syntax. All the language constructs are available with a click of a mouse. Easy-to-use wizards step you through creation of more complex HDL blocks.

Platform and Standards Support

ModelSim PE supports both VHDL and Verilog and accelerates VITAL functions, procedures and timing checks. ModelSim PE runs on Windows XP, Vista and 7.

ModelSim PE Features:

- Partial vhdl support 2008
- Transaction wlf logging support in all languages including VHDL
- Windows 7 support
- Secure ip support
- System c option
- RTL and Gate-Level Simulation
- Integrated Debug
- Verilog, VHDL and System Verilog Design
- Mixed HDL Simulation option.
- Code coverage option
- Enhanced debug option
- Windows 32 bit.

ModelSim PE Benefits:

- Cost effective HDL Simulation.

- ❖ Intuitive GUI efficient interactive debug.
- ❖ Integrated project management simplifies managing project data
- ❖ Easy to use outstanding technical support
- ❖ Sign-off support for popular ASIC libraries

1.9 VHDL

VHDL (Very high speed integrated circuit Hardware Description Language) is a hardware description language used in electronic design automation to describe digital and mixed-signal systems such as field-programmable gate arrays and integrated circuits. VHDL can also be used as a general purpose parallel programming language. VHDL is commonly used to write text models that describe a logic circuit. Such a model is processed by a synthesis program, only if it is part of the logic design. A simulation program is used to test the logic design using simulation models to represent the logic circuits that interface to the design. This collection of simulation models is commonly called a test bench.

VHDL has file input and output capabilities, and can be used as a general-purpose language for text processing, but files are more commonly used by a simulation test bench for stimulus or verification data. There are some VHDL compilers which build executable binaries. In this case, it might be possible to use VHDL to write at test bench to verify the functionality of the design using files on the host computer to define stimuli, to interact with the user, and to compare results with those expected. However, most designers leave this job to the simulator. It is relatively easy for an inexperienced developer to produce code that simulates successfully but that cannot be synthesized into a real device, or is too large to be practical. One particular pitfall is the accidental production of transparent latches rather than D-type flip-flops as storage elements. One can design hardware in a VHDL IDE (for FPGA implementation such as Xilinx ISE, Altera Quartus or Mentor Graphics HDL Designer) to produce the RTL schematic of the desired circuit. After that, the generated schematic can be verified using simulation software which shows the waveforms of inputs and outputs of the circuit after generating the appropriate test bench. To generate an appropriate testbench for a particular circuit or VHDL code, the inputs have to be defined correctly. For example, for clock input, a loop process or an iterative statement is required. A final point is that when a VHDL model is translated into the "gates and wires" that are mapped onto a programmable logic device such as a CPLD or FPGA, then it is the actual hardware being configured, rather than the VHDL code being "executed" as if on some form of a processor chip.

CHAPTER 2
LITERATURE SURVEY

INTRODUCTION

Video is a series of still images which are called frames. The consumers using digital video increasing day by day, so video compression is necessary to reduce the size. Video compression has two important benefits. First, it makes it possible to use digital video in transmission and storage environments that would not support uncompressed video for example current Internet throughput rates are insufficient to handle uncompressed video in real time. A DVD can only store a few seconds of uncompressed video so video storage would not be practical without video and audio compression. Second video compression enables more efficient use of transmission and storage resources. If a high bit rate transmission channel is available, then it is more attractive proposition to send a high resolution compressed video or multiple compressed video channels than send a single, low resolution, uncompressed stream. Earlier DCT was used for image compression applications, but it has several shortcomings such as blocking artifact and bad subjective quality images are restore at high compression ratio.

Discrete wavelet transform (DWT) is widely used in image and video compression. Several architectures have been suggested for efficient VLSI implementation of 2-D DWT for real-time applications. The last two decades, the DWT has gained establishing role in signal processing and image processing applications because of their ability to decompose the signal into different sub bands with both time and frequency information. DWT also has features like progressive image transformation, ease of compressed image manipulation, region of interest coding etc.

IMAGE AND VIDEO COMPRESSION USING DISCRETE WAVELET TRANSFORM MATLAB RESULTS

K Sureshraju and P. Sreekanth [1] proposes DWT method for image and video compression. Image quality known by the PSNR and CR values after image reconstruction using IDWT algorithm. Video which consists 120 frames. Video compression techniques are of prime importance for reducing the amount of information needed for picture sequence without losing much of its quality, judged by human viewers. Discrete Wavelet Transform (DWT) is used to achieve the compression for image and it extended to series of images which is nothing but a video.

DESIGN AND IMPLEMENTATION OF LIFTING BASED 2D DISCRETE WAVELET TRANSFORMING FPGA

A. Hasna and Jayaraj. U. Kidavu [2] this paper proposes a highly pipelined and distributed VLSI architecture of lifting based 2D DWT with lifting coefficients represented in fixed point [2:14] format. Compared to conventional architectures, the proposed highly pipelined architecture optimizes the design which increases significantly the performance speed. The design raises the operating frequency, at the expense of more hardware area. In this paper, initially a software model of the proposed design was developed using MATLAB.

VIDEO COMPRESSION USING HYBRID DCT-DWT ALGORITHM

L. Escalin Tresa1 and M. Sundararajan [3] shows the proposed method uses a Hybrid DWT-DCT algorithm on motion compensated frame by taking the advantages of both the method. DWT (Discrete Wavelet Transform) and DCT (Discrete Cosine Transform) are the most commonly video compression techniques. DCT has high energy compaction and requires less computational resources, DWT on the other hand is a multiresolution transformation. But the compression ratio that can be achieved is low. The performance of the proposed method can be evaluated using compression ratio, PSNR and mean square error.

THREE-DIMENSIONAL VIDEO COMPRESSION USING SUBBAND/WAVELET TRANSFORM WITH LOWER BUFFERING REQUIREMENTS

Hosam Khalil, Amir F. Atiya and Samir Shaheen [4] shows that 3-D coding of image sequences can be achieved in the true sense of temporal direction decomposition but with much less buffering requirements. For a practical coder, this can be achieved by introducing an approximation to the way the transform coefficients are encoded. Applying wavelet decomposition using some types of filters may introduce edge errors, which become more prominent in short signal segments. Solution for this problem for the Daubechies family of filters also shown.

MOTION ESTIMATION AND MOTION COMPENSATED VIDEO COMPRESSION USING DCT AND DWT

Thazni Aziz and D. Raveena Judie Dolly [5] shows the Full Search strategies which is used to reduce computation. Video compression techniques are used to reduce the redundancy in video data. DCT (Discrete Cosine transform) and DWT (Discrete Wavelet Transform) are used. One advantage of DCT is to find the match of low frequency values then it can be increased by comparing the higher frequency value. The goal of wavelet based compression is to store video data in a little space. Hence, the performance is analyzed based on compression ratio and PSNR values using these two techniques.

HEVC VIDEO COMPRESSION USING DWT AND BLOCK MATCHING ALGORITHM

Anusha Dandu and EscalinTresa [6] proposed a video transcoder based on Intraframe redundancy and it is achieved using various techniques of DWT, Quantization and Entropy coding. Temporal redundancy, also known as Inter-frame redundancy is achieved using this technique. It is observed that when the motion is compensated using frame based block matching method and also there by reducing the matched blocks the resultant video is reduced significantly. However, these blocks resemble constant over the frames until any motion is detected in following frames. When the blocks size is further increased to achieve greater compression ratio it is observed the video quality is compromised and seen blurred.

VLSI IMPLEMENTATION OF INTEGER DCT ARCHITECTURES FOR HEVC IN FPGA TECHNOLOGY

M. Devendra and P. Giribabu [7] this paper deals with an energy and area efficient VLSI architecture of an HEVC-compliant inverse transform and dequantization engine is presented. The implementation of a pipelining scheme to process all transform sizes at a minimum throughput of 2 pixel/cycle with zero-column skipping for improved throughput is shown. We use data gating in the 1-D Inverse Discrete Cosine Transform engine to improve energy-efficiency for smaller transform sizes. A high-density SRAM-based transpose memory is used for an area-efficient design.

HAAR WAVELET BASED APPROACH FOR IMAGE COMPRESSION AND QUALITY ASSESSMENT OF COMPRESSED IMAGE

Kamrul Hasan TalukderI and Koichi Harada [8] applies the 2D discrete wavelet transform and estimate the detail matrices from the information matrix of the image. The reconstructed image is synthesized using the estimated detail matrices and information matrix provided by the Wavelet transform. The quality of the compressed images has been evaluated using some factors like Compression Ratio (CR), Peak Signal to Noise Ratio (PSNR), Mean Opinion Score (MOS), Picture Quality Scale (PQS) etc**.**

PEAK SIGNAL-TO-NOISE RATIO BASED ON THRESHOLD METHOD FOR IMAGE SEGMENTATION

Farshid Pirahansiah and Siti Norul Huda Sheikh Abdullah [9] proposes adaptive threshold method, based on the peak signal-to-noise ratio (PSNR), has the potential to be applied in OCR. The proposed algorithm achieves competitive results in standard, printed, and handwritten images. The algorithm achieves better results compared with previous methods. However, it produced slightly worse results compared to newer methods, such as multi-level thresholding. Recently, PSNR has been widely used as a stopping criterion in multilevel threshold methods for segmenting images. Alternatively, we have applied the PSNR as a criterion to determine the most suitable threshold value. The comparison of the proposed method with state-of-the-art multilevel and multi-threshold methods are shown.

IMAGE COMPRESSION BASED UPON WAVELET TRANSFORM AND A STATISTICAL THRESHOLD

Ahmed A. Nashat and N. M. Hussain Hassan [10] the paper deals with Haar wavelets as the basis of transformation functions. Haar wavelet transformation is composed of a sequence of low pass and high pass filters, known as filter bank. The redundancy of the DWT detail coefficients is reduced through thresholding and further through Huffman encoding. The proposed threshold algorithm is based upon the statistics of the DWT coefficients. The quality of the compressed images has been evaluated using some factors like Compression Ratio, (CR), and Peak Signal to Noise Ratio, (PSNR). Experimental results demonstrate that the proposed technique provides sufficient higher compression ratio compared to other compression thresholding techniques.

THE HAAR WAVELET AND THE BIORTHOGONAL WAVELET TRANSFORMS OF AN IMAGE

Teena Varma, Vidya Chitre and Dipti Patil [11] put an effort to achieve better understanding of the wavelet based methods, Haar and Biorthogonal wavelets are discussed on result oriented basis using MATLAB environment. Two different kinds of wavelet transform can be distinguished, a continuous and a discrete wavelet transform. The Haar transform and the Biorthogonal Wavelets are explained using example input image.

MEMORY-EFFICIENT HIGH-SPEED CONVOLUTION-BASED GENERIC STRUCTURE FOR MULTILEVEL 2-D DWT

Basant Kumar Mohantand and Pramod Kumar Meher [12] In this paper, he has proposed a design strategy for the derivation of memory-efficient architecture for multilevel 2-D DWT. Using the proposed design scheme, he has derived a convolution-based generic architecture for the computation of three-level 2-D DWT based on Daubechies (Daub) as well as biorthogonal filters. This is a major advantage when the structure is implemented for higher throughput. The structure has regular data-flow, small cycle period *TM* and 100% hardware utilization efficiency.

CHAPTER 3
METHODOLOGY

3.1 DISCRTE WAVELET TRANSFORM

The Forward and Inverse Discrete Wavelet Transform (FDWT/IDWT) has been widely used as an alternative to the existing time-frequency representations such as DFT and DCT. It has become a powerful tool in many areas, such as image compression and analysis, texture discrimination, fractal analysis, pattern recognition and so on. The recent and future developments of high definition digital video and the diversity of the terminals had led to consider a multi-resolution codec. Creating a multi-resolution pyramid of images at each level, just store the differences (residuals) between the image at that level and the predicted image from the next level. We can reconstruct the image by just adding up all the residuals. Main advantage is to store the residuals at an easy manner. It is proposed to use DWT for VIDEO CODING. As the DWT algorithms are computational intensive, VLSI implementations of the algorithms are preferred for real-time applications.

In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency and location information (location in time). 2-D Discrete Wavelet Transformation (DWT) converts the image from the spatial domain to frequency domain. In DWT, two channel filter bank is used. By applying the 1-D discrete wavelet transform along the rows of the image first, and then along the columns to produce 2-D decomposition of image, the wavelet transform decomposes the image into low-low, low-high, high-low and high-high frequency components as shown in figure 3.1. In case of a 2D image, an N level decomposition can be performed resulting in 3N+1 different frequency bands and it is shown in figure 3.1. These four components are referred to as approximation, horizontal, vertical and diagonal coefficients respectively because low-low frequency components contain average information whereas the other components contain directional information due to spatial resolution. Higher absolute values of wavelet coefficients in the high bands correspond to salient features such as edges or lines.

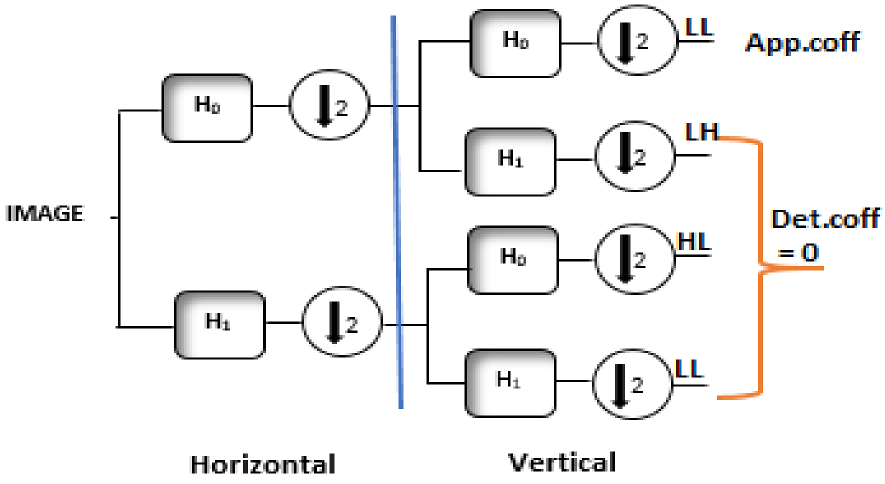


Fig. 3.1 Single level decomposition DWT

3.2 VIDEO CODING

Original Videos are first converted into frames which shows like a continuous of still images. Each frame undergoes various wavelet transforms. It transforms the input data into a format to reduce interpixel redundancies in the input image. Transform coding techniques use a reversible, linear mathematical transform to map. Image compression model and image decompression model the pixel values onto a set of coefficients, which are then quantized. The key factor behind the success of transform-based coding schemes is that many of the resulting coefficients for most natural images have small magnitudes and can be quantized without causing significant distortion in the decoded image. For compression purpose, higher the capability of compressing information in fewer coefficients, for that reason Discrete Wavelet Transform(DWT) have become the most widely used transform coding techniques. Thresholding in certain signals, many of the wavelet coefficients are close or equal to zero. Through threshold these coefficients are modified so that the sequence of wavelet coefficients contains long strings of zeros. In hard threshold, a threshold value is selected. Any wavelet whose absolute value falls below the tolerance is set to zero with the goal to introduce many zeros without losing a great amount of detail.

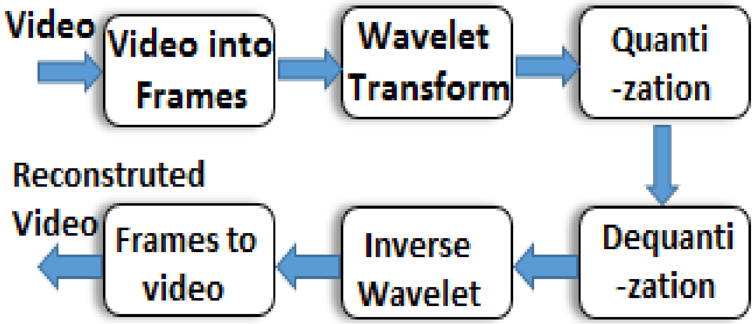


Fig. 3.2 Block diagram of DWT

Quantization is used to discard perceptibly insignificant information. It basically converts each real DCT coefficient to an integer by scaling it by a factor and then discarding the digits after the decimal point. For each coefficient, a scaling factor is chosen in such a way that there is no perceptible change even after discarding digits after the decimal point. Note that the inverse quantization cannot recover the original value completely. Quantization and inverse quantization can take anywhere between 3% and 15% of processor cycles and have only small memory requirement. Quantization converts a sequence of floating numbers to a sequence of integers. The simplest form is to round to the nearest integer. Another method is to multiply each number by a constant and then round to the nearest integer. Quantization is called lossy because it introduces error into the process

3.3 WAVELET USED

- Haar Wavelet
- Daubechies wavelets
- Bi-Orthogonal Wavelet

3.3.1 HAAR WAVELET

A Haar wavelet is the simplest type of wavelet. In discrete form, Haar wavelets are related to a mathematical operation called the Haar transform. The Haar transform serves as a prototype for all other wavelet transforms. One distinctive feature that the Haar transform enjoys is that it lends itself easily to simple hand calculations.

Haar transform has the orthogonal property which is used for analysing the frequency components present. By using haar wavelet, multiplications can be avoided. Paring up of input values may store the difference and sum can be processed to provide the next level, which leads to $2^{n-}1$ differences and a final sum. Low pass filter computes moving average and High pass filter computes moving difference of its input.

1D Transformation

A 1D image with a resolution of four pixels, having values [9 7 3 5]. Haar wavelet basis can be used to represent this image by computing a wavelet transform. To do this, first the average the pixels together, pairwise, is calculated to get the new lower resolution image with pixel values [8 4]. Clearly, some information is lost in this averaging process. We need to store some detail coefficients to recover the original four pixel values from the two averaged values. In above example, 1 is chosen for the first detail coefficient, since the average computed is 1 less than 9 and 1 more than 7. This single number is used to recover the first two pixels of our original four-pixel image. Similarly, the second detail coefficient is -1, since 4 + (-1) = 3 and 4 - (-1) = 5. Thus, the original image is decomposed into a lower resolution (two-pixel) version and a pair of detail coefficients. Repeating this process recursively on the averages gives the full decomposition.

Resolution	Averages	Detail coefficients
4	$[\,9 \quad 7 \quad 3 \quad 5\,]$	
2	$[\,8 \quad 4\,]$	$[\,1 \quad -1\,]$
1	$[\,6\,]$	$[\,2\,]$

Fig. 3.3 Decomposition to lower resolution

Thus, for the one-dimensional Haar basis, the wavelet transform of the original four-pixel image is given by [6 2 1 -1]. We call the way used to compute the wavelet transform by recursively averaging and differencing coefficients, filter bank. We can reconstruct the image to any resolution by recursively adding and subtracting the detail coefficients from the lower resolution versions.

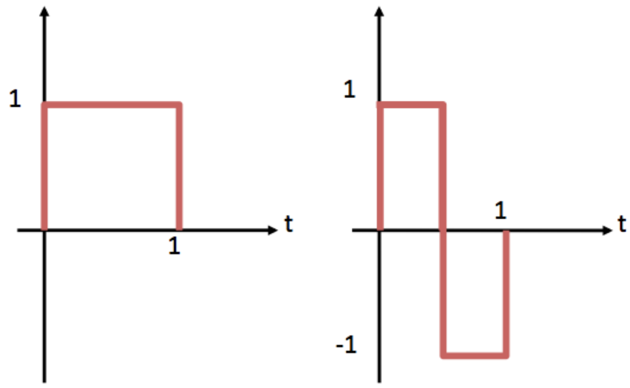


Fig. 3.4 Haar Scaling and Wavelet function

called scaling functions, and are usually denoted by the symbol Φ. A simple basis for V is given by the set of scaled and translated box functions

$$\phi_i^j(x) := \phi(2^j x - i) \quad i = 0, 1, 2\ldots\ldots 2^j - 1 \text{ where}$$

$$\phi(x) := \begin{cases} 1 & \text{for } 0 \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$

The wavelets corresponding to the box basis are known as the Haar wavelets, given by-

$$\Psi_i^j(x) := \Psi(2^j x - i) \quad i = 0, 1, 2\ldots\ldots 2^j - 1 \text{ where}$$

$$\Psi(x) := \begin{cases} 1 & \text{for } 0 \le x < 1/2 \\ -1 & \text{for } 1/2 \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$

Thus, the DWT for an image as a 2D signal will be obtained from 1D DWT. We get the scaling function and wavelet function for 2D by multiplying two 1D functions. The scaling function is obtained by multiplying two 1D scaling functions: $\varphi(x,y)=\varphi(x)\varphi(y)$. The wavelet functions are obtained by multiplying two wavelet functions or wavelet and scaling function for 1D. For the 2D case, there exist three wavelet functions that scan details in horizontal $\Psi^{(1)}(x,y)= \varphi(x)\Psi(y)$, vertical $\Psi^{(2)}(x,y)= \Psi(x)\varphi(y)$ and diagonal directions: $\Psi^{(3)}(x,y)= \Psi(x)\ \Psi(y)$. This may be represented as a four-channel perfect reconstruction filter bank. Now, each filter is 2D with the subscript indicating the type of filter (HPF or LPF) for separable horizontal and vertical components. By using these filters in one stage, an image is decomposed into four bands. There exist three types of detail images for each resolution: horizontal (HL), vertical (LH), and diagonal (HH). The operations can be repeated on the low low (LL) band using the second stage of identical filter bank. Thus, a typical 2D DWT, used in image compression, generates the hierarchical structure shown in Fig. 1.3.

The scaling and wavelet functions are defined as

$$\phi(x) = h_0\phi(2x) + h_1\phi(2x - 1)$$

$$\psi(x) = h_1\phi(2x) - h_0\phi(2x - 1)$$

$$h_0 = h_1 = {}^{1}/_{\sqrt{2}}$$

The transformation of the 2D image is a 2D generalization of the 1D wavelet transformed. It applies the 1D wavelet transform to each row of pixel values. This operation provides us an average value along with detail coefficients for each row.

Next, these transformed rows are treated as if they were themselves an image and apply the 1D transform to each column. The resulting values are all detail coefficients except a single overall average co-efficient. In order to complete the transformation, this process is repeated recursively only on the quadrant containing averages.

3.3.2 DAUBECHIES WAVELETS

Daubechies wavelets are characterized in an indistinguishable way from the Haar wavelet by computing running midpoints and contrasts by means of scalar items with scaling sand wavelets functions. The Daub4 wavelet transform, like the Haar transform, can be extended to multiple levels as many times as the signal length can be divided by 2.

The difference between the Haar transform and the Daub4 transform lies in the way that the scaling signals and wavelets are defined. We shall first discuss the scaling signals. Let the scaling numbers α1, α2, α3, α4 be defined by

$$\alpha_1 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad \alpha_2 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \alpha_3 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad \alpha_4 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

Using these scaling numbers, the 1-level Daub4 scaling signals are

$$\begin{aligned}
\mathbf{V}_1^1 &= (\alpha_1, \alpha_2, \alpha_3, \alpha_4, 0, 0, \dots, 0) \\
\mathbf{V}_2^1 &= (0, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, 0, 0, \dots, 0) \\
\mathbf{V}_3^1 &= (0, 0, 0, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, 0, 0, \dots, 0) \\
&\vdots \\
\mathbf{V}_{N/2-1}^1 &= (0, 0, \dots, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4) \\
\mathbf{V}_{N/2}^1 &= (\alpha_3, \alpha_4, 0, 0, \dots, 0, \alpha_1, \alpha_2).
\end{aligned}$$

An important property of these scaling signals is that they all have energy 1. This is because of the following identity satisfied by the scaling number.

$$\alpha_1^2 + \alpha_4^2 + \alpha_3^2 + \alpha_4^2 = 1$$

It is clear that the above equation implies that each 1-level scaling signal has energy 1. To see that it also implies that each k-level scaling signal has energy 1 is more difficult. Another identity satisfied by the scaling numbers is

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = \sqrt{2}$$

Let the wavelet numbers β1, β2, β3, β4 be defined by

$$\beta_1 = \frac{1-\sqrt{3}}{4\sqrt{2}}, \quad \beta_2 = \frac{\sqrt{3}-3}{4\sqrt{2}}, \quad \beta_3 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \beta_4 = \frac{-1-\sqrt{3}}{4\sqrt{2}}$$

Notice that the wavelet numbers are related to the scaling numbers by the equations $\beta_1 = \alpha_4$, $\beta_2 = -\alpha_3$, $\beta_3 = \alpha_2$, and $\beta_4 = -\alpha_1$. Using these wavelet numbers, the 1-level Daub4 wavelets are defined by

$$\begin{aligned}
\mathbf{W}_1^1 &= (\beta_1, \beta_2, \beta_3, \beta_4, 0, 0, \dots, 0) \\
\mathbf{W}_2^1 &= (0, 0, \beta_1, \beta_2, \beta_3, \beta_4, 0, 0, \dots, 0) \\
\mathbf{W}_3^1 &= (0, 0, 0, 0, \beta_1, \beta_2, \beta_3, \beta_4, 0, 0, \dots, 0) \\
&\vdots \\
\mathbf{W}_{N/2-1}^1 &= (0, 0, \dots, 0, \beta_1, \beta_2, \beta_3, \beta_4) \\
\mathbf{W}_{N/2}^1 &= (\beta_3, \beta_4, 0, 0, \dots, 0, \beta_1, \beta_2).
\end{aligned}$$

The Daub4 wavelets all have energy 1. This is clear for the 1-level Daub4 wavelets, since

$$\beta_1^2 + \beta_2^2 + \beta_3^2 + \beta_4^2 = 1$$

It can also be shown that all k-level Daub4 wavelets have energy 1 as well.

$$\beta_1 + \beta_2 + \beta_3 + \beta_4 = 0$$

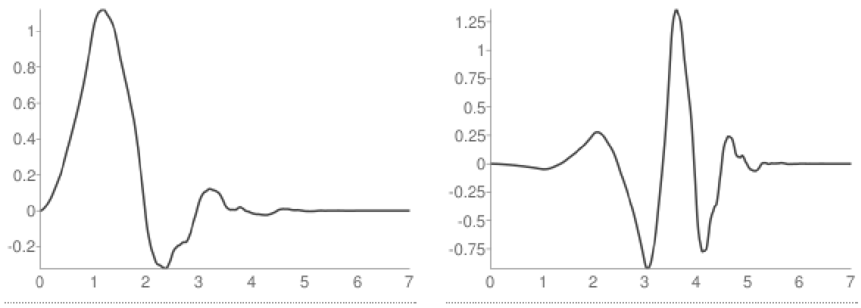


Fig. 3.5 Daubechies Scaling and Wavelet function

The scaling and wavelet functions are defined as

$$\phi(x) = h_0\phi(2x) + h_1\phi(2x-1) + h_2\phi(2x-2) + h_3\phi(2x-3) \quad (3)$$

$$\psi(x) = h_0\phi(2x-1) - h_1\phi(2x) + h_2\phi(2x+1) - h_3\phi(2x+2) \quad (4)$$

$$h_0 \approx 0.48296; h_1 \approx 0.83652; h_2 \approx 0.22414; h_3 \approx -0.12941$$

3.3.3 BI-ORTHOGONAL WAVELET

In filtering applications, none of the orthogonal filters except the haar filters provide the linear phase. The biorthogonal filters are designed to provide the symmetric property and the exact reconstruction by using two wavelet filters instead of one. Biorthogonal Wavelet transform is used for image denoising as it has the property of linear phase. The orthogonal filter of wavelet transform does not have the characteristics of linear phase, therefore the phase distortion will lead to the distortion of the image edge. This problem is reduced by the use of biorthogonal wavelet, as it contains spline wavelets. This will help in perfect reconstruction of the image by using Finite Impulse Response filters. This property is not present in orthogonal filters. In order to gain greater flexibility in the construction of wavelet bases, the orthogonality condition is relaxed allowing semi-orthogonal, biorthogonal or non-orthogonal wavelet basis. Biorthogonal Wavelets are families of compactly supported symmetric wavelets. The symmetry of the filter coefficients is often desirable since it results in linear phase of the transfer function. In the biorthogonal case, rather than having one scaling and wavelet function, there are two scaling functions that may generate different multiresolution analysis, and accordingly two different wavelet functions.

The dual scaling and wavelet functions have the following properties:
1. They are zero outside of a segment.
2. The calculation algorithms are maintained, and thus very simple.
3. The associated filters are symmetrical.
4. The functions used in the calculations are easier to build numerically than those used in the Daubechies wavelets.

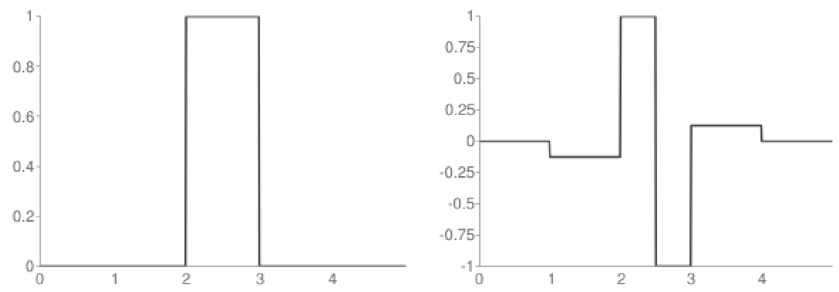


Fig. 3.6 Biorthogonal Scaling and wavelet function

3.4 IMAGE QUALITY METRICES

3.4.1 PEAK SIGNAL TO NOISE RATIO (PSNR):

Peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

PSNR is most commonly used to measure the quality of reconstruction of lossy compression codec (e.g., for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs, PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases it may not. One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content. The PSNR between reconstructed image and original image is calculated using following equations,

$$\boldsymbol{PSNR = 20 * log_{10}(255^2/MSE)}$$

Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, provided the bit depth is 8 bits, where higher is better. For 16-bit data typical values for the PSNR are between 60 and 80 Db.

3.4.2 MEAN SQUARE ERROR (MSE):

MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.

$$\boldsymbol{MSE = {}^{1}/_{m \times n} \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} [X(i,j) - Y(i,j)]^2}$$

Where 255 is the maximum pixel value in the grey-scale image and MSE is the average mean-squared error, as defined. Here, *X* and *Y* are the two compared images, the size of each being *M* × *N* pixels.

3.4.3 STRUCTURAL SIMILARITY INDEX MATRIX (SSIM):

SSIM index is a framework for quality assessment based on the degradation of structural information. For human visual system, a calculation of structural information difference can provide a good approximation to the image distortion perceived. The product of the illumination and the reflectance gives the luminance of the surface of an object. But the structures of the objects in the scene are independent of the illumination. SSIM index defines the structural information in an image as those attributes that represent the structure of objects in the scene, independent of the average luminance and contrast.

$$\mathrm{SSIM}(x,y)=\frac{\left(2\mu_x\mu_y+c_1\right)(2\sigma_{xy}+c_2)}{\left(\mu_{x2}+\mu_{y2}+c_1\right)+(\sigma_{x2}+\sigma_{y2}+c_2)}$$

Where, μ_x is the average of x

μ_y is the average of y

σ_{x2} is the variance of x

σ_{y2} is the variance of y

$\sigma_{xy}$is the covariance of x and y

3.4.4 AVERAGE DIFFERENCE (AE):

AD is simply the average of difference between the reference signal and compressed image. It is given by the equation,

$$AD=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(X(i,j)-Y(i,j)\right)$$

3.4.5 NORMALIZED ABSOLUTE ERROR:

The large value of *NAE* means that image is poor quality. *NAE* is defined as follows,

$$NAE=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\left|X(i,j)-Y(i,j)\right|\right)}{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(X(i,j)\right)}$$

3.5. CONVOLUTION BASED DESIGN

Lifting-based scheme has been considered more efficient compared with the convolution-based approach due to its lower arithmetic complexity. But, after a thorough analysis of both lifting and the convolution-based designs, we observe that, convolution-based scheme along with appropriate scheduling of multilevel decomposition could have lower complexity than the lifting-based design. The memory savings offered by the convolution-based scheme is significantly higher than the saving of arithmetic components. Also, we observe that, the existing folded and block-based designs have some inherent difficulties to reduce the memory complexity of the 2-D DWT structures. Although the block-based design of [13] eliminates the requirement of FB for the multilevel DWT computation and utilizes on-chip memory efficiently, it involves larger on chip memory and introduces significant overhead introduced by its input-interface units. To overcome the above difficulties, we propose here a scheme to derive a memory-efficient computing structure for multilevel 2-D DWT.

1) DWT levels are computed concurrently to avoid FB.

2) Convolution scheme is used for orthogonal as well as biorthogonal wavelet filters to derive maximum advantage of parallel data-access scheme.

3) Parallel data access is applied in each DWT level to reduce memory complexity of the overall structure.

3.5.1 HARDWARE ARCHITECTURE

Due to down-sampled filtering, the computational complexity after each level of decomposition steadily decreases by a factor of four. In order to achieve 100% HUE, hardware resource of the processing units (PUs) should be reduced by a factor of 4 after every DWT level. But we do not have any suitable straightforward approach to map the DWT computation of successive levels to the PU with one-fourth complexity. Maximum (100%) HUE may be achieved by introducing four times more parallelism in two-level DWT computation. Similarly, for three-level DWT, we need to have 16 times more parallelism which could be too high for many applications. Alternatively, if we assume hardware resource of a PU can be reduced up to one sub cell comprised of a pair of low-pass and a high-pass filters, then PU-*J* corresponding to *J*th level is comprised of one sub cell and processes two samples in every cycle for 100% HUE, where the row and column computations are time multiplexed. The overall throughput rate of PU-*J* is one sample per cycle. Based on these observations, we have derived a pipeline structure for three-level 2-D DWT. However, similar structures can be derived for higher DWT levels as well. The proposed structure is shown in fig. 3.7. It consists of three processing units (PUs), where PU-1, PU-2, and PU-3, respectively, perform computations of first-level, second-level and third-level.

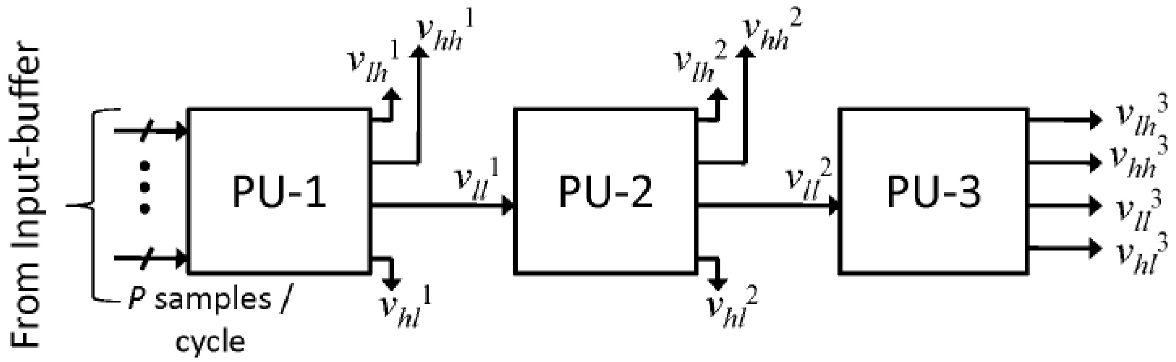


Fig. 3.7 Computation of three level 2D DWT

Structure of PU-1 is shown in Fig. 3.8. It consists of eight processing elements (PEs). Each input block is extended by (K−2) samples such that adjacent input blocks of a row are overlapped by (K − 2) samples, where K is the filter order. From the input matrix (**X**), extended input blocks **I** (m, i) are fed to PU-1 block-by-block in every cycle.

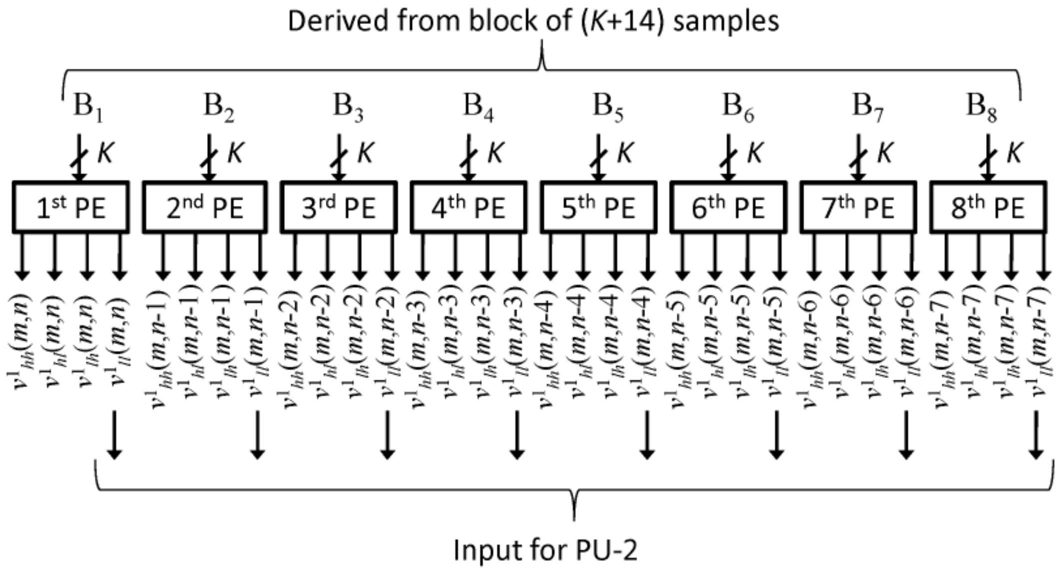


Fig. 3.8 Structure of PU-1

The input block **I** (m, i) corresponding to the mth row of (**X**) contains the samples $\{x(m, 16i), x(m, 16i + 1), \ldots, x(m, 16i + K + 12), x(m, 16i + K + 13)\}$, for $0 \le m \le M - 1$ and $0 \le i \le (N/16) - 1$. Suppose in the first cycle, the first input block of the first row of **X** is fed then during the second cycle, the first input block of the second row is fed to PU- 1, such that the first input blocks of all the M rows of **X** are fed in M cycles and in the next set of M cycles, second input blocks of all the M rows are fed to the structure.

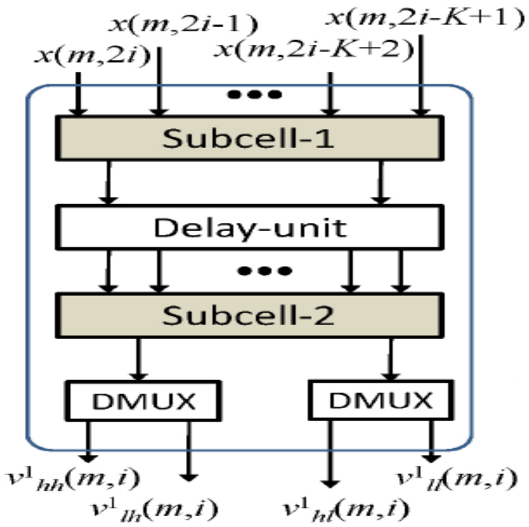


Fig. 3.9 a) Structure of PE

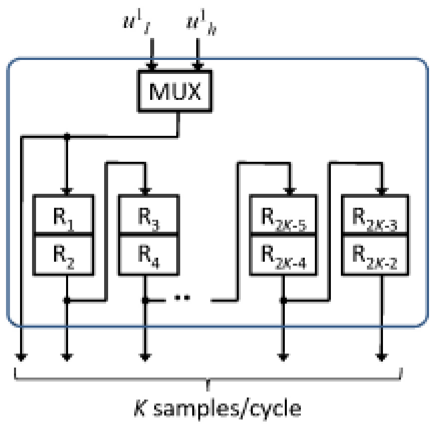


Fig. 3.9 b) Structure of delay-unit (DU-1)

The data vectors are fed in parallel to the PEs, such that $(q + 1)^{th}$ PE receives the data-vector. Structure of PE is shown in Fig. 3.9(a). It consists of a pair of identical sub cells (subcell-1 and subcell-2) and one delay-unit (DU-1). Subcell-1 performs the necessary DWT computation along the row-direction where K consecutive samples of a particular row constitute the data-vector.

3.5.1 STRUCTURE FOR ORTHOGONAL WAVELET(Daub4)

Structure of PE is shown in Fig. 3.8. It consists of a pair of identical Sub cells (subcell-1 and subcell-2) and one delay-unit (DU-1). Subcell-1 performs the necessary DWT computation along the row-direction where K consecutive samples of a particular row constitute the data-vector. Structure of the sub cell using orthogonal wavelet filters is shown in Fig. 3.9 (a). It consists of K multiplier-units (MUs), where each MU performs multiplications corresponding to a pair of low-pass ($h(k)$ and high-pass $g(k)$)

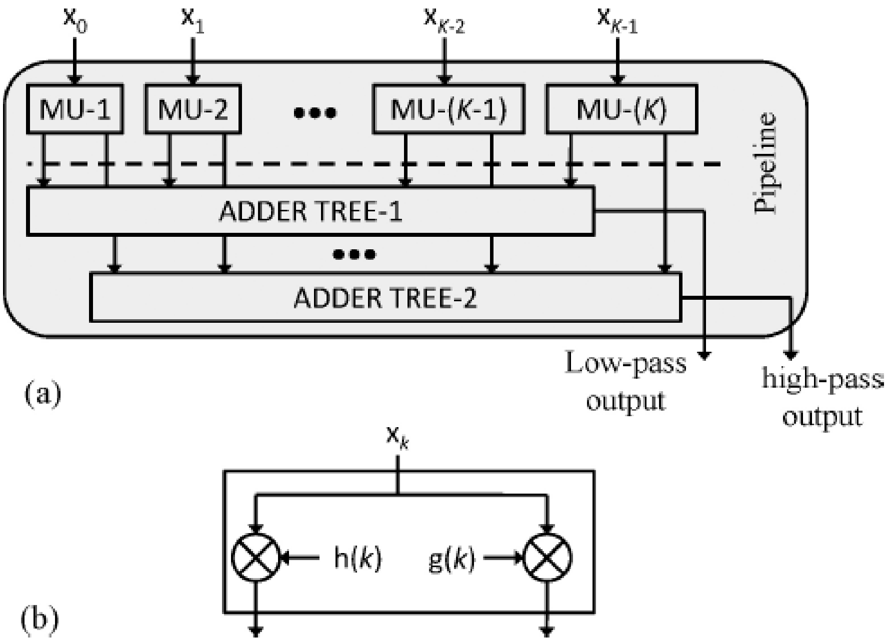


Fig. 3.9 (a) Structure of the sub cell for orthogonal wavelet filters.
(b) Structure of MU.

filter coefficients, for $0 \leq k \leq K - 1$. Internal structure of the $(k + 1)$th MU is shown in Fig. 3.9 (b). The low-pass and the high-pass partial results are added in two separate adder-trees to compute a pair of low-pass and high-pass filter outputs. A pair of DWT components corresponding to two adjacent columns of two sub bands (v^3_{ll}, v^3_{lh}) are obtained during even-numbered period of eight cycles. During the odd numbered sets of eight cycles, a pair of components of other two sub bands (v^3_{hl}, v^3_{hh}) of the same two columns are produced.

CHAPTER 4

RESULTS

4.1 VIDEO COMPRESSION

Videos entertainment and video communication are excessively used. But storage space required to the video is large amount of memory. If such videos have to send over transmission media then large requirement of transmission bandwidth. Therefore, Video compression is essential method for making video to transmittable size.

Video is series of the still image referred as frames. According to the persistence of vision property of the human eye, the scene formed on the retina will remain as it is for 60ms. If the frame duration is kept below the 60ms, then string of images formed on the eye will viewed as the video. The frames contain the large amount of the Redundant data bits. Thus, by discarding the redundant information compression can be achieved. The video compression using DWT achieve greater compression with less losses. There is inversely proportional relation between video quality and amount of compression achieved. If the compression ratio is more than quality of video decreases. Discrete Wavelet transform are mostly used methods for video compression.

4.1 INPUT VIDEOS:

Video is series of the still image referred as frames. Hockey video consist of 77 frames which has 26 frames per second as shown in Fig. 4. 1. The time taken to play this video up to 3 second’s. Rhinos video consist of 114 frames which has 16 frames per second. The time taken to play this video up to 7 second’s.

Fig. 4.1 Hockey video

Fig. 4.2 Rhinos video

Frame 1 Frame 4 Frame 8 Frame 12

Frame 24 Frame 48 Frame 64 Frame 77

Fig. 4.3 Original input video frame for Hockey

Frame 1 Frame 4 Frame 8 Frame 12

Frame 24 Frame 48 Frame 64 Frame 77

Fig. 4.4 Original input video frame for Rhinos

4.2 SIMULATION RESULT OF HAAR WAVELET:

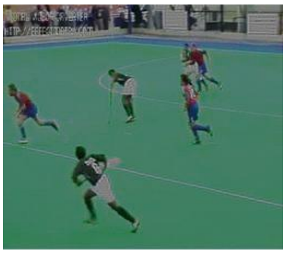
Video has to be first splitted into frames. Each and every frame is transformed by using different types of wavelets and the performance of each wavelet can be shown by using different quality metrics are tabulated.


Frame 1


Frame 4


Frame 8


Frame 12


Frame 24


Frame 48


Frame 64


Frame 77

Fig.4.5 Hockey Decompressed Video frame of Haar wavelet

After the performance of Three level decomposition and the wavelet compression, the decompressed video frame of Haar wavelet for both Hockey and Rhinos videos are shown in fig. 4. 5 and fig. 4. 6 respectively. The performance of the video can be analysed in between the frames among 77 and 114 frames of Hockey and Rhinos videos respectively.


Frame 1


Frame 4


Frame 8


Frame 12


Frame 24


Frame 48


Frame 64


Frame 77

Fig.4. 6 Rhinos Decompressed Video of Haar wavelet

Table 4.1 Performance metrics of Haar Wavelet

VIDEO	HOCKEY						RHINOS					
Quality Metrics	**MSE**	**PSNR (dB)**	**SSIM**	**AD**	**NAE**	**CT (s)**	**MSE**	**PSNR (dB)**	**SSIM**	**AD**	**NAE**	**CT (s)**
1	**0.4790**	**54.524**	**0.8**	**-0.075**	**0.0029**	**0.73**	**0.5626**	**53.1269**	**0.8**	**-0.126**	**0.0044**	**0.45**
4	**0.4785**	**54.533**	**0.8**	**-0.074**	**0.0029**	**1.78**	**0.5628**	**53.1243**	**0.8**	**-0.125**	**0.0045**	**1.28**
8	**0.4784**	**54.534**	**0.8**	**-0.073**	**0.0029**	**3.07**	**0.5628**	**53.1276**	**0.8**	**-0.124**	**0.0043**	**2.43**
12	**0.4790**	**54.534**	**0.8**	**-0.072**	**0.0028**	**4.52**	**0.5624**	**53.1291**	**0.8**	**-0.125**	**0.0043**	**3.55**
24	**0.4784**	**54.534**	**0.8**	**-0.07**	**0.0028**	**8.58**	**0.5627**	**53.1247**	**0.8**	**-0.126**	**0.0041**	**6.98**
48	**0.4788**	**54.527**	**0.8**	**-0.069**	**0.0029**	**16.4**	**0.5619**	**53.1374**	**0.8**	**-0.128**	**0.0039**	**13.7**
64	**0.4790**	**54.523**	**0.8**	**-0.068**	**0.0028**	**22.7**	**0.5617**	**53.1403**	**0.8**	**-0.142**	**0.0038**	**18.3**
77	**0.4789**	**54.525**	**0.8**	**-0.067**	**0.0028**	**26.1**	**0.5614**	**53.1449**	**0.8**	**-0.153**	**0.0038**	**22.7**
144	**-**	**-**	**-**	**-**	**-**	**-**	**0.5689**	**53.2815**	**0.8**	**-0.133**	**0.0037**	**33.1**

The Performance metrics such as Mean Square Error(MSE), Peak Signal to Noise Ratio(PSNR), Average Difference(AD), Normalized Absolute Error(NAE), Structural Similarity Index Measure and the Computation time are calculated for Haar wavelet and are tabulated as shown in Table. 4.1. Compression ratio obtained for Haar wavelet is 38.

4.3 SIMULATION RESULT OF DAUBECHIES 4 WAVELET:

Video has to be first splitted into frames. Hockey and Rhions videos consist of 77 and 144 frames respectively. Each and every frame is transformed by using daubechies 4 wavelet and the performance of each wavelet can be shown by using different quality metrics are tabulated.

Frame 1

Frame 4

Frame 8

Frame 12

Frame 24

Frame 48

Frame 64

Frame 77

Fig. 4.7 Hockey Decompressed Video frame of Daubchies wavelet

Frame 1

Frame 4

Frame 8

Frame 12

Frame 24

Frame 48

Frame 64

Frame 77

Fig.4.8 Rhinos Decompressed Video frame of Daubchies wavelet

After the performance of Three level decomposition and the wavelet compression, the decompressed video frame of Daubechies wavelet for both Hockey and Rhinos videos are shown in fig. 4.7 and fig. 4.8 respectively. The performance of the video can be analysed in between the frames among 77 and 114 frames of Hockey and Rhinos videos respectively.

Table 4. 2 Performance metrics of Daubechies Wavelet

Video	HOCKEY						RHINOS					
Quality Metrics	MSE	PSNR (dB)	SSIM	AD	NAE	CT(s)	MSE	PSNR (dB)	SSIM	AD	NAE	CT(s)
1	0.0731	71.006	0.9	0.045	0.0006	0.89	0.1420	65.195	0.9	0.044	0.0013	0.86
4	0.0616	72.412	0.9	0.010	0.0015	1.82	0.1400	65.212	0.9	0.042	0.0013	1.73
8	0.0656	71.887	0.9	0.030	0.0006	2.99	0.1527	64.454	0.9	0.043	0.0013	2.85
12	0.0678	71.660	0.9	0.043	0.0006	4.15	0.1773	63.356	0.9	0.107	0.0016	3.93
24	0.0999	68.305	0.9	0.058	0.0010	8.35	0.1730	63.371	0.9	0.17	0.0014	6.25
48	0.1059	67.737	0.9	0.098	0.0007	14.26	0.1935	62.415	0.9	0.173	0.0015	13.32
64	0.1003	68.204	0.9	0.046	0.0006	18.78	0.1380	65.337	0.9	0.097	0.0011	17.65
77	0.0559	73.338	0.9	0.038	0.0005	22.2	0.1692	63.564	0.9	0.045	0.0012	21.27
114	-	-	-	-	-	-	0.1781	63.045	0.9	0.035	0.0015	31.14

The Performance metrics are calculated for Daubechies wavelet and are tabulated as shown in Table. 4.2 for both videos. Compression ratio obtained for Daubechies wavelet is 43.

4.4 SIMULATION RESULT OF BI-ORTHOGONAL WAVELET:

Video is series of the still image referred as frames. Hockey and Rhions videos consist of 77 and 144 frames respectively. Each and every frame is transformed by using daubechies 4 wavelet and the performance of each wavelet can be shown by using different quality metrics are tabulated.

The frames contain the large amount of the Redundant data bits. Thus, by discarding the redundant information compression can be achieved.


Frame 1


Frame 4


Frame 8


Frame 12

Frame 24 Frame 48 Frame 64 Frame 77

Fig. 4. 9 Hockey Decompressed Video frame of Bi-orthogonal wavelet

After the performance of Three level decomposition and the wavelet compression, the decompressed video frame of Bi-orthogonal wavelet for both Hockey and Rhinos videos are shown in fig. 4. 9 and fig. 4. 10 respectively. The performance of the video can be analysed in between the frames among 77 and 114 frames of Hockey and Rhinos videos respectively.

Frame 1 Frame 4 Frame 8 Frame 12

Frame 24 Frame 48 Frame 64 Frame 77

Fig. 4.10 Rhinos Decompressed Video frame of Bi-orthogonal wavelet

The Performance metrics are calculated for Bi-orthogonal wavelet and are tabulated as shown in Table. 4.3 for both videos. Compression ratio obtained for Bi-orthogonal wavelet is 43.

Table 4. 3. Performance metrics of Bi-orthogonal Wavelet

Video	HOCKEY						RHINOS					
Quality Metrics	MSE	PSNR (dB)	SSIM	AD	NAE	CT(s)	MSE	PSNR (dB)	SSIM	AD	NAE	CT(s)
1	0.0228	75.969	0.9	-0.007	0.00018	0.92	0.0291	73.882	0.9	-0.016	0.00019	0.88
4	0.0231	75.862	0.9	-0.007	0.00018	1.82	0.0230	76.312	0.9	-0.006	0.00016	1.77
8	0.0232	75.828	0.9	-0.007	0.00019	2.99	0.0234	75.749	0.9	-0.006	0.00015	2.88
12	0.0233	74.768	0.9	-0.0069	0.00019	4.14	0.0254	75.278	0.9	-0.009	0.00016	3.94
24	0.0232	76.434	0.9	-0.0067	0.00019	7.53	0.0226	74.325	0.9	-0.004	0.00014	7.20
48	0.0242	74.817	0.9	-0.0067	0.00021	14.32	0.0238	74.789	0.9	-0.008	0.00014	13.4
64	0.0229	76.929	0.9	-0.007	0.00021	18.59	0.0244	76.241	0.9	-0.009	0.00013	17.5
77	0.0241	76.520	0.9	-0.0069	0.00021	22.17	0.0233	74.764	0.9	-0.008	0.00011	20.9
114	-	-	-	-	-	-	0.0236	76.254	0.9	-0.008	0.00013	30.7

From the Table 4.1, 4.2 and 4.3, it inferred that Daubechies and Bi-Orthogonal wavelet tends to have moderate compression ratio, high gain and minimum error when compared to that of haar wavelet. So, in the succeeding work for implementation of video by Daubechies and Bi-Orthogonal can be used to get a better result.

4.5 IMPLEMENTATION RESULT

The system implementation has its difficulties in verification and debug stages. In order to overcome this, usage of many tools that are able to implement simulations at different levels of our system to define the features and performance of the system. The tools are Mentor Graphics Modelsim- Altera6.5e and tool from MATLAB(R2014a) which provides Link for ModelSim that has been used to do simulation between ModelSim and MATLAB simultaneously. Daubechies4(Daub4) wavelet was chosen for the implementation process based on the performance which was discussed in earlier section. Daub4 wavelet transform was simulated by using ModelSim- Altera 6.5e software.

Fig. 4.11 Input Image

First, specifications and preliminary design was performed using MATLAB. Then, the design is implemented in VHDL using a logic design environment for FPGAs such as ISE or ModelSim Importing of coefficients from an image into the text can be done by via MATLAB software. These text file from the MATLAB can be used by a ModelSim as input data. The image was shown in the fig 4.11. In order to achieve 100% HUE, hardware resource of the processing units (PUs) should be reduced by a factor of 4 after every DWT level. But we do not have any suitable straightforward approach to map the DWT computation of successive levels to the PU with one-fourth complexity. Maximum (100%) HUE may be achieved by introducing four times more parallelism in two-level DWT computation. Similarly, for three-level DWT as shown in Fig. 4.12, we need to have 16 times more parallelism which could be too high for many applications.

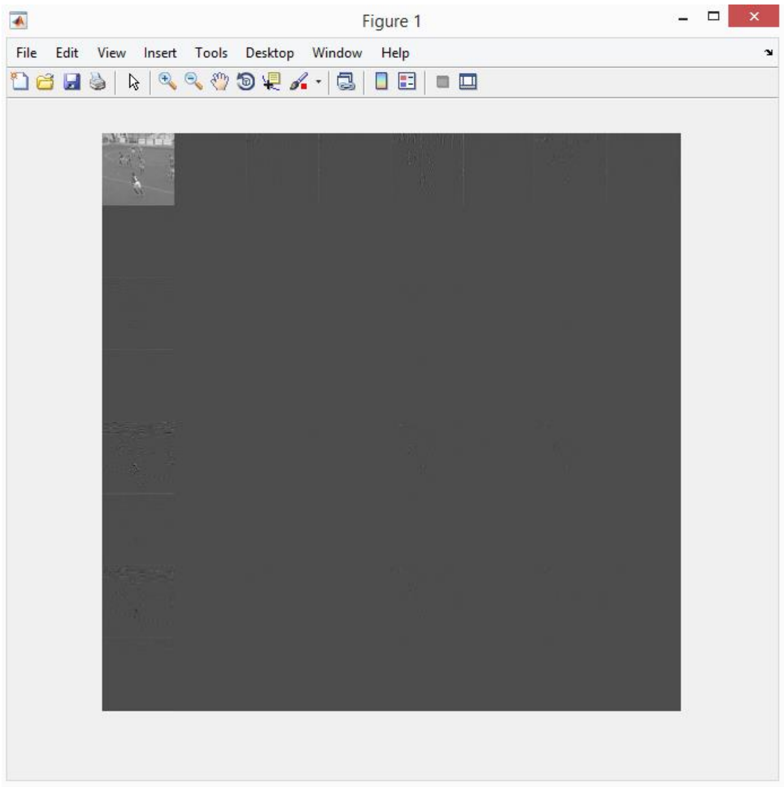


Fig. 4.12 MATLAB result

4. 5. 1 MODELSIM RESULT for 512x 512 image

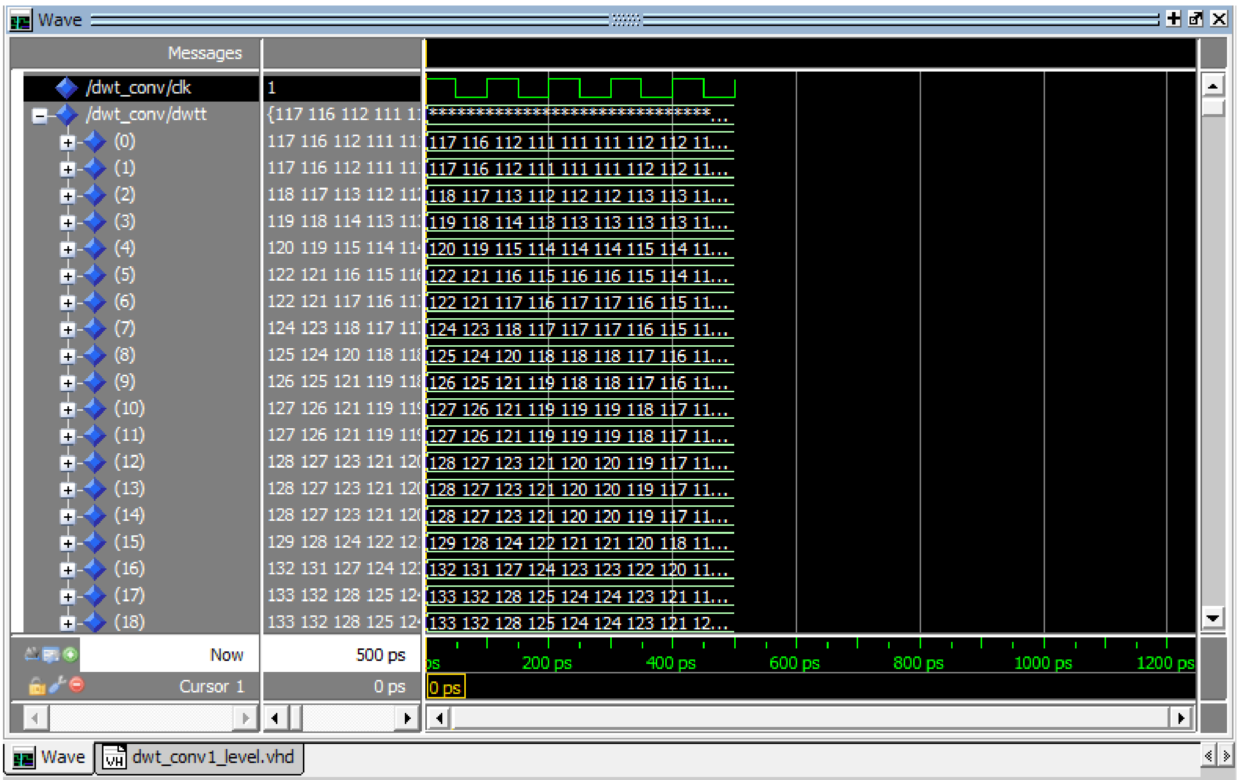


Fig. 4.13 Simulation result of Daubechies 4 wavelet

The Simulation result of Daubechies 4 wavelet by ModelSim was shown in fig 4.13. A pair of DWT components corresponding to two adjacent columns of two sub bands (v^3_{ll}, v^3_{lh}) are obtained during even- numbered period of eight cycles. During the odd numbered sets of eight cycles, a pair of components of other two sub bands (v^3_{hl}, v^3_{hh}) of the same two columns are produced as discussed in chapter 3.

CHAPTER 5
CONCLUSION

5.1 CONCLUSION

Thus, the performance of various wavelets transform has been analysed by using Matlab software. From the analysis of simulated results DWT shows better performance compared with DCT. Video compression using Haar, Daubechies and Bi-orthogonal wavelets as the basis function along with the quality metrics of decompressed video frame have been obtained in this research for the size of 512x512. Video compression using wavelet transforms results shown improved compression ratio as well as image quality. While comparing with Haar wavelet, Daubechies and Bi-orthogonal wavelet provides minimum error and acceptable PSNR range which indicates that the technique works well in video compression. The quality of compressed video has been measured based on the metrics values like compression ratio, PSNR, MSE, SSIM, AD and NAE.

The memory complexity is the most important issue for efficient realization of 2-D DWT in VLSI systems. It suggests to implement memory centric design strategy and based on that a convolution-based generic architecture for the computation of three-level 2-D DWT for Daubechies wavelet filters. The proposed structure does not involve FB. This is used as a major advantage when the structure is implemented for higher throughput rate. The implementation of multilevel 2-D DWT using Daubechies filters for high-performance image processing applications.

5.2 FUTURE WORK

The future work of this project can be extending to implement the DWT architecture for video coding based on a convolution-based generic architecture for 2-D DWT using various wavelet filters for different frame rates and sizes.

REFERENCES

[1] K Sureshraju and P. Sreekanth, “Image and Video Compression using Discrete Wavelet Transform Matlab Results,” International Journal of Engineering Research & Technology (IJERT), Vol. 3 Issue 10, October 2014.

[2] A. Hasna And Jayaraj. U. Kidavu, “Design and Implementation of Lifting Based 2d Discrete Wavelet Transforming Fpga,” IJAICT, Vol. 1 Issue 3, July 2014.

[3] L. Escalin Tresa1 and M. Sundararajan, “Video Compression Using Hybrid DCT-DWT Algorithm,” IRJET, Vol. 03, Issue 05, May 2016.

[4] Hosam Khalil, Amir F. Atiya and Samir Shaheen, “Three-Dimensional Video Compression Using Subband/Wavelet Transform with Lower Buffering Requirements,” IEEE Transactions on Image Processing, Vol. 8, No. 6, June 1999.

[5] Thazni Aziz and D. Raveena Judie Dolly, “Motion Estimation and Motion Compensated Video Compression Using DCT and DWT,” IJETAE Vol 2, Issue 12, December 2012.

[6] Anusha Dandu and EscalinTresa, “Hevc Video Compression using DWT and Block Matching Algorithm,” ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 9, May 2015.

[7] M. Devendra and P. Giribabu, “VLSI implementation of Integer DCT Architectures for HEVC in FPGA Technology” IJOEET, Vol. 2, Issue 5, June 2015

[8] Kamrul Hasan TalukderI and Koichi Harada, “Haar Wavelet Based Approach for Image Compression and Quality Assessment of Compressed Image,” IAENG International Journal of Applied Mathematics, Feb 2007.

[9] Farshid Pirahansiah and Siti Norul Huda Sheikh Abdullah, “Peak Signal-To-Noise Ratio based on threshold method for Image Segmentation,” Journal of Theoretical and Applied Information Technology, Vol. 57 No.2, November 2013.

[10] Ahmed A. Nashat and N. M. Hussain Hassan, "Image Compression Based upon Wavelet Transform and a Statistical Threshold," International Conference on Optoelectronics and Image Processing, 2016
[11] Teena Varma, Vidya Chitre and Dipti Patil, “The Haar Wavelet and the Biorthogonal Wavelet Transforms of an image,” International Journal of Engineering Research and Applications (IJERA), VNCET-30 Mar’12

[12] Basant Kumar Mohantand Pramod Kumar Meher 2013, “Memory-Efficient High-Speed Convolution-based Generic Structure for Multilevel 2-D DWT,” IEEE Transactions on circuits and systems for Video Technology, Vol. 23, No. 2, February 2013.

[13] Anilkumar Katharotiya1, Swati Patel and Mahesh Goyani, “Comparative Analysis between DCT & DWT Techniques of image Compression,” Journal of Information Engineering and Applications, Vol 1, No.2, 2011.

[14] Chao Chengand Keshab K. Parhi, “High-Speed VLSI Implementation of 2-DDiscrete Wavelet Transform,” IEEE Transactions on Signal Processing, Vol. 56, No. 1, January 2008

[15] Monika Rathee and Alka Vij, “Image compression Using Discrete Haar Wavelet Transforms,” International Journal of Engineering and Innovative Technology (IJEIT), Volume 3, Issue 12, June 2014.

[16] Chao Chengand Keshab K. Parhi, “High-Speed VLSI Implementation of 2-DDiscrete Wavelet Transform,” IEEE Transactions on Signal Processing, Vol. 56, No. 1, January 2008.

[17] Anurag Mahajan, and Pramod Kumar Meher, “Area- and Power-Efficient Architecture for High Throughput Implementation of Lifting 2-DDWT”, IEEE Trans. Circuits Syst. Express Briefs, vol. 59, no. 7, July 2012.

[18] Yusong Hu and Ching Chuen Jong, “A Memory-Efficient Scalable Architecture for Lifting-Based Discrete Wavelet Transform,” IEEE Transactions on circuits Syst– II: Express Briefs, Vol. 30, No. 2, December 2013.

LIST OF CONFERENCES

- Presented a paper titled **"IMPLEMENTATION OF DWT ALGORITHM FOR VIDEO CODING"** in **IEEE** sponsored **4th International Conference on Innovations in Information, Embedded and Communication Systems(ICIIECS'17)** on 17th and 18th March 2017 at Karpagam college of Engineering, Coimbatore.

- Presented a paper titled "**VARIOUS DWT ALGORITHM FOR MONOCHROME IMAGE AND VIDEO"** in **IETE** sponsored **NATIONAL CONFERENCE on Emerging Trends in Information, Communication, VLSI design & Embedded system (ICVE 2k17)** on 9th & 10th March 2017 at Adhiyamaan college of Engineering, Hosur.