Video Codec IP using Discrete Wavelet Transform

Sangeeta Satish Rao
ECE Department,PES University
Bangalore,India
sangeetasrao1@gmail.com

Gursimran Kaur Narula
ECE Department,PES University
Bangalore,India
simran.kaur8858@gmail.com

Ridhima Sudhir
ECE Department,PES University
Bangalore,India
ridhima.sudhir@gmail.com

Sanjana S
ECE Department,PES University
Bangalore,India
sanjana25399@gmail.com

Rajeshwari B
ECE Department,PES University
Bangalore,India
rajeshwari@pes.edu

Bajrangbali B
ECE Department,PES University
Bangalore,India
bajarangreddy@pes.edu

***Abstract*—In today's world, due to the ongoing pandemic, all our interactions have been restricted to occur from behind screens. Thus, our use of online streaming services, e-lectures, and video conferences have increased manifold and there is a strong need for better video and image compression techniques that can compress data while maintaining a good level of image quality. In our paper, we attempted to find the best transform and encoding method to provide a decent compression while retaining the quality. First, we attempted to gain a thorough understanding of current compression algorithms and their flaws, while also considering a new approach to compression in the form of Wavelets. To get better results, we used different versions of compression techniques on different sections of the files. Once a proper working design was established, the design was to be converted to a model which could be easily translated to a machine level language by an Electronic Design and Automation tool like Xilinx Vivado and HLS. We then tried parallelising some of the loops and blocks in our code to achieve a speed up of around 2s. The decoded frame we obtained seemed to resemble the initial frame to a good extent and for comparison with other compression algorithms, various metrics were considered. Finally, the Peak Signal-to-Noise Ratio and Compression Ratio were determined and the results were tabulated and we were able to achieve a CR of 21.85 maintaining a SSIM of 92.17% with PSNR 24.9dB**

***Index Terms*—Discrete Wavelet Transform, Discrete Cosine Transform, Video Codec, Zero Tree Wavelet Encoding**

I. INTRODUCTION

The usage of personal computers and mobile devices appears to be limitless with today's technology, allowing us to produce, distribute, and receive videos that capture an audience's attention in seconds. Videos are no longer merely something we watch; they have evolved into an indulgent experience in which we participate. Understanding the various video file formats and improvising with various procedures is thus required to ensure that they are generated in the best format and quality for their intended purpose and audience.

We first looked at different methods for video codec implementation. When compared to current approximation methods used in [1], the approximation used in this paper generates better compressed image quality while using the same level of complexity.

Also looking at the hardware implementation utilised in [2] resulted in less complexity, less memory usage, and less power consumption. Higher decomposition is necessary for higher compression. Using an FPGA to implement the method results in less memory usage and lower power consumption. DWT displays images in many resolutions. This paper shows how to use system c coding on the XILINX platform to create DWT and IDWT algorithms in a simple and quick way.

In [6] the EZW technique is used to encode the image in order to compress it into a high-accuracy bit stream. Instead of the four symbols utilised in Shapiro's EZW algorithm, this technique uses six symbols. For the bit stream of wavelet-based image compression, the EZW is an effective, fast, and computationally simple approach. A higher PSNR and compression ratio can be achieved by increasing the amount of symbols.

The study in [12] describes a video display system based on 1D SPIHT that makes two major contributions. It focuses on high-throughput hardware design. The suggested encoder and decoder have throughputs of 7.04 Gbps and 7.63 Gbps, respectively. Image quality is improved by 0.96dB thanks to the method utilised. By reducing dependencies, the improvement optimises SPIHT hardware utilisation. The compression ratio is reduced when schemes for resolving various dependencies in the frequency bands of the encoder and decoder are used.

For lossy image compression, the authors in [4] adjusted the embedded zero tree wavelet to obtain high compression in terms of PSNR and bitrate. They also demonstrated that the suggested method outperforms the set partitioning in hierarchical trees (SPIHT) and JPEG2000 algorithms. The proposed methodology improves the number of coefficients not to be encoded by using additional symbols, but the embedded zerotree wavelet has some downsides, including restricted gain compression performance and the presence of numerous redundant symbols

A. Motivation

The Covid-19 pandemic had brought the world to a standstill. While "normal life" seemed to be at bay, a new normal is shaping our world. Now that there appears to be a higher demand for multimedia content in the form of E-Classrooms, Webinars, Video Conferences, and other formats, there is a clear need for a compression algorithm that makes better use of our current storage technology while minimising the impact on our communication network. Codecs are used at every stage of the video production process, from shooting to editing to encoding our streaming media files for transmission. Especially in the current pandemic situation with cinemas closed and the work-from-home culture, people are switching to online streaming services like Netflix and Amazon Prime, so better codecs are required now more than ever. Multimedia files consume a large amount of disk space and take more time to load when compared to conventional document files. Hence newer compression techniques not only have to retain quality but also be able to decode the files to be accessible in time. Recognising this need, we propose to design and implement a video codec which will address these issues.

B. Proposed Methodology

- Technique selection: Compare various transforms (DCT, DWT-based) and choose the best one to convert the input image frame to a different domain.
- Optimization of Chosen Transform: Making improvements to the chosen transform for an enhanced design.
- Selection of Quantization parameters and Encoding techniques: Identifying a computationally less intensive quantization and encoding method that produces a good reconstructed image.
- Block Conversion: To convert the design to a hardware synthesizable block, the above-mentioned blocks would be translated from a higher-level language into an intermediate language.
- arallelisation and Optimisation: For efficient use of time and memory, we would be checking for possible parallel functioning of repetitive processes.
- Comparing our parameters with existing standards. Softwares we would be using - Matlab, Simulink,Vivado and Vitis.

II. Methodology

Video encoding is the method of transforming raw video into a specific format using various encoding methods in order to maximise quality while reducing file size. Video files are often compressed from gigabytes of data to megabytes for streaming. A video encoding technique known as codec is used to reduce the size of a video to a more manageable size. H.264/AVC (advanced video coding) is the most commonly used format for high-definition streaming video and video content delivery. It is the most recent industry standard for transmitting video content. Along with H.264, MP4 is used for improved compatibility since it is the most commonly adopted file format by different video players, editing applications, and is widely used on the internet.

A. Video codec

- Video encoding is the method of transforming raw video into a specific format using various encoding methods in order to maximise quality while reducing file size. Video files are often compressed from gigabytes of data to megabytes for streaming. A video encoding technique known as codec is used to reduce the size of a video to a more manageable size. H.264/AVC (advanced video coding) is the most commonly used format for high-definition streaming video and video content delivery. It is the most recent industry standard for transmitting video content. Along with H.264, MP4 is used for improved compatibility since it is the most commonly adopted file format by different video players, editing applications, and is widely used on the internet.

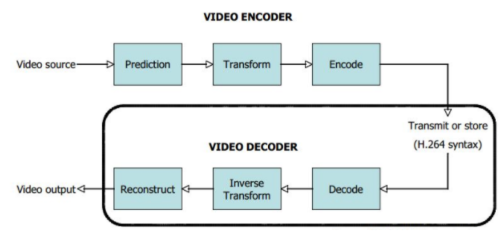

Fig. 1. H.264 codec block diagram.

- Predictor: Removes all redundant information from the video. The two kinds of redundancies are Temporal and Spatial redundancy . Since frames are frequently created with a very small time delay, bits of one frame can be approximated to bits of the previous ones; here, temporal redundancy comes into play. In frames with smaller pixels that can be approximated to larger pixels, spatial redundancy occurs.
- Transform: For energy compaction, the frame is transformed into a separate domain. Block-based and Image-based transformations are the two most common types of transforms. The block-based strategy partitions a single frame into smaller blocks, lowering the system's memory requirements. The image-based technique ignores the block approach and works on the whole picture. Types of Transforms:
- DCT (Discrete Cosine Transform): Naser Ahmed proposed the Discrete Cosine Transform (DCT), which is a block-based transform. It analyses small sections of the image and converts them to the frequency domain, allowing for more effective quantization and data compression. The DCT technique is modified in the H.26X and MPEG specifications to further minimise the size.
- DWT (Discrete Wavelet Transform): A DWT is a wavelet transform of discretely sampled wavelets. It keeps track of the frame's frequency as well as its position. The following are examples of wavelets used in DWT: HAAR Wavelet: compares input values and stores the difference before moving the sum. The Daubechies Wavelet is a wavelet function that produces fine discrete samples of an implicit main wavelet function. Dual Tree Complex Wavelet Transform: solves the problems of low direc-

tional selectivity and shift variance in higher dimensions than the second dimension.
- SPIHT (Set Partitioning In Hierarchical Trees): It is a wavelet decomposition image compression algorithm that takes advantage of the inherent similarity between the subbands.
- Discrete Sine Transform (DST): acts on real data with odd symmetry, analogous to the DCT. It is combined with DCT in HEVC prediction to focus spatial residual signals into low-frequency components.
- Quantize: Quantization is performed to approximate or round off any value to a specified level. Quantizer converts analog signal to its digitized value by discretizing the signal. Two types of quantization are: non-uniform quantization and uniform quantization. For quantizing a signal we need to mention the step size which is the difference between two levels. Uniform quantization has equal step size for the discrete signal, whereas non-uniform quantization has unequal step size.
- Colour Quantization: This reduces the number of colours used in an image. Colour Quantization algorithms include, Nearest Colour Algorithm and Median Cut Algorithm.
- Frequency Quantization: Reduces the amount of information required by ignoring high frequency components. This can be done with the help of Quantization Matrices.
- Encode: The frames or blocks are converted into a binary form. Predictive, variable length, and arithmetic encoding are the three methods of encoding. The difference between the present pixel value and the previous pixel value is transmitted using predictive encoding. For improved compression, variable encoding employs statistics. Each symbol corresponds to a codeword of variable length and a fixed number of bits. For correct reconstruction, a codebook must be available at the decoder. Arithmetic encoding converts a whole file into a single decimal number from a sequence of symbols. At each iteration, the input symbols are processed one by one.
- Decode: The encoded image is converted back to an uncompressed bitmap which can then undergo the inverse transform and then displayed on the screen. This involves the exact reverse of the steps involved in encoding the image.
- Parallelisation and optimisation: In order to achieve efficiency, we use various HLS Pragma functions. The HLS tool includes pragmas for optimising the design, reducing latency, improving throughput performance, and reducing the RTL code's area and device resource utilisation. These pragmas can be directly introduced to the kernel source code.

B. *Video compression techniques*

Multiple frames make up a video. We may view the video as a motion of pictures due to the fast shifting between frames. Smooth motion is achieved by using a frame rate of 20 or more frames per second. Frames per second range from 20 to 24 depending on the method used. Each frame is made up of pixels, each of which has a distinct colour and light value. These pixels could be identified individually, resulting in a high-resolution video, or clustered into blocks, resulting in a lower-resolution video. As a result, the higher the number of pixels per unit area, the better the video quality. The image-to-image estimation is the foundation of video compression. The first image is an I-frame, which is a high-resolution reference image. This first picture is used as a reference in the subsequent frames. A P-frame is an image projected from a single reference image, while a B-frame is an image predicted bidirectionally from one past and one future. There is little space available since many devices are forced to share the network with other data-intensive applications. Video compression methods are used to reduce the high bit rate in order to solve this problem. The compression ratio(CR) is used to evaluate their accuracy . Lower bandwidth consumption occurs with the higher the compression ratio. But increasing compression, on the other hand, does cause the image to degrade.

Some Video/Image Compression Standards:
- HEVC(H.265): H.265 is the successor to AVC and offers data compression and video quality improvements of 25% to 50%. Integer DCT and DST transforms are used by HEVC for different block sizes. A brief description of the original pixel is used to replace the redundant parts of the frame after comparison.
- AVC(H.264): The most widely used video compression technology is Advanced Video Coding, which is based on block oriented, motion compensated- DCT coding.
- MPEG: It was developed by Moving Picture Experts Group Licensing Administrator (MPEG LA) in the early 1990s.MPEG algorithms compress data into tiny bits that can be conveniently transmitted, then the data is decompressed precisely and rapidly to enable high-fidelity reconstruction.
- JPEG stands for "Joint Photographic Experts Group" and is a compression standard for images.. This method is lossy and uses the DCT transform method. In the spatial domain, an image is a property of x and y.. JPEG uses 2 levels of the DCT as a single step to provide a spatial frequency response with a domain function indexed by u and v.

C. *Steps in methodology*

- Selection of transform: Video compression can be implemented with two main techniques: Discrete Cosine Transform and Discrete Wavelet Transform. Discrete Cosine Transform transforms a signal/image from the spatial domain to the frequency domain.
  DCT divides an image into two cosine functions that are added together. Lower frequencies, which are present in the top left corner of the DCT, account for a large portion of the signal energy in most photographs. Since the lower right values reflect higher frequencies, compression is done because they are insignificant enough to be ignored

without causing much distortion. The input is given as an 8x8 integer array. The grey scale level of each pixel is stored in this array; each of the pixels have levels ranging from 0(black) to 255(white). The result would be as follows:
Computing 2D DCT by factoring to a series of1D DCTs: (i) 1-D DCT is applied Vertically to the columns (ii) 1-D DCT is applied Horizontally to the resulting vertical DCT. (iii) the DCT can be applied Horizontally and then Vertically as well.

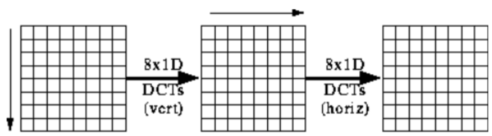

Fig. 2. 2-D DCT

The two dimensional DCT of an MxN matrix which is denoted by A and its inverse transform are given as follows:
Discrete Wavelet Transform, in simple words, transforms an image into pixels. The signal is decomposed into mutually orthogonal sets of wavelets using this transform. It contains information on both the frequency and position components. A signal's DWT is determined by passing it through a number of filters.A convolution of the input frame and an impulse response low pass filter occurs:
At the same time, a high-pass filter decomposes the signal The HPF provides us with information coefficients and the LPF provides us with the approximation coefficients. These two filters are referred to as a quadrature mirror filter and thus have to be similar to each other.
According to Nyquist's law, half of the samples can be discarded because half the signal's frequencies have been excluded. So subsampling the low-pass filter output by 2 and passing it through an LPF again followed by a HPF with a cut-off frequency equal to half of the previous filter., i.e.
Since the signal is characterised only by half of the filter's output, the time resolution has been halved. And since each output of the filter has only half of the input's frequency band, the frequency resolution has been increased by two. The outputs of the LPF are decomposed again with high pass and low pass filters and then down-sampled to improve the frequency resolution even further. This is interpreted as a binary tree. A filter bank is the name given to the tree.

- Transformation Optimization:
Here, we choose the most suitable transformation, by comparing the compression ratio, peak-to-signal noise ratio, and hardware feasibility based on the transformations' efficiency. Matlab implementation of the transform and quantization is performed and a suitable encoding algorithm is used for both the DCT and DWT transforms.

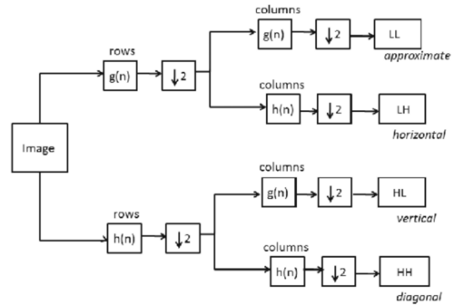

Fig. 3. 2-D DWT

Following this, decoding is done and the reconstructed image frames are compared.
For DCT we got a PSNR of 30.1849, SSIM of 0.9136 and a Compression Ratio of 1.6277. For DWT we got a PSNR of 17.0381, SSIM of 0.9136 and a Compression Ratio of 6.31.
Selection of quantization parameters and encoding technique:
Quantization:
For DCT: A macroblock approach is used in DCT wherein, after the transformed image is obtained, it is broken down into smaller blocks. Each such block is then processed with a quantisation matrix based on a selection of parameters. Some of these parameters depend on the part of the image that needs to be retained. For example, if the top left parts of each macroblock need to be retained the quantisation matrix is designed accordingly.
For DWT: Similarly various methods are employed in DWT to reduce bits in each pixel. Some of them include dropping alternate bits in the HH, HL, LH bands and dropping every third alternate bit in the LL band. Or setting a fixed value for a different range of pixel values thereby reducing bits. Based on the amount of information in each band, the corresponding number of bits can be dropped to reduce the number of bits.
In Simulink, the quantizer block can be used to define a particular quantization interval, using which each signal value can be mapped to the respective quantized values. After quantization, a smooth input signal can take on a stair-step shape. This equation mathematically describes the round-to-nearest method:
Encoding: The main objective of the encoding is to efficiently code the significant coefficients in the sub-bands. 1) Zero Tree Wavelet technique: Here, a fixed value is decided to be the parameter for a particular pixel value to be considered "significant". If the value is considered to be significant, it is added to the encoding matrix, otherwise it is dropped. This is done by first looking at the pixel value in a location say "X" for the LL band. If it is significant, it is added to the encoding matrix E and the pixel value at location X of the LH

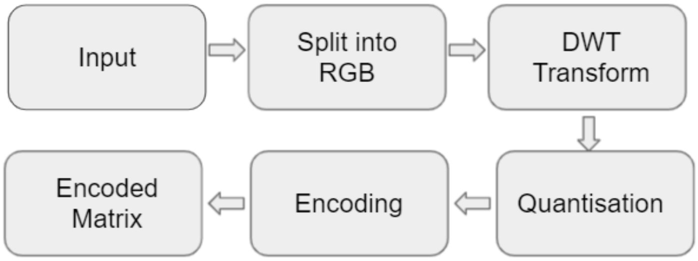

Fig. 4. Encoding Block Diagram

Fig. 5. Encoding Output Diagram

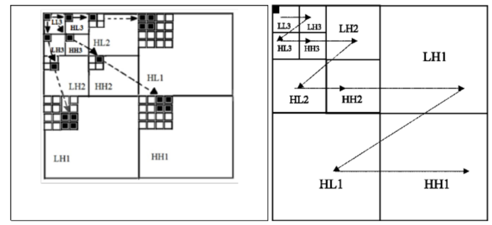

Fig. 6. Zero Tree Wavelet Encoding Technique

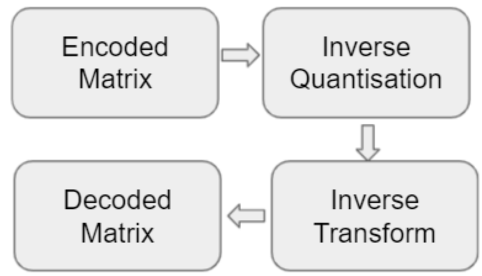

Fig. 7. Decoding Block Diagram

band is checked next. Similarly if this is "significant", it is added to the encoding matrix E and the pixel value of location X in HL band and then HH band is checked. The iteration then moves on to the next location X+1 starting from the LL band. Whenever a value fails to be significant, a "0" is added to the encoding matrix and the checking is stopped for the subsequent bands and the checking begins at the next location of the LL band. Since most of the coefficients in the tree will be zero, there is a significant reduction in the amount of space occupied. Hence, we can achieve a very good compression using this technique To further improve our compression ratio, an alternate approach used in our implementation involved getting the coefficients only from the LL and LH bands to be a part of the encoding technique and hence the encoding matrix, and completely dropping all the bits from the HL and HH bands. This was done knowing that the most important bits are contained in the LL band whereas the HL, LH and HH bands contain the vertical, horizontal and diagonal features of the image, depth perception and overall sharpness. The image frame after reconstruction bore a good resemblance to the original image frame and this method was further analysed in the HLS implementation.

- Decoding technique and performance optimization:
  Decoding:
  Decoding was implemented as the reverse process of encoding. The encoded matrix was fed as the input, and first reverse bitshift quantization was performed. For example if the quantization technique used was bit shifting, each pixel value was bit shifted in the opposite direction by the same amount. Following this, inverse of the chosen transform was performed to get the decoded image. Performance optimization: By checking for any scope of parallelisation, we came across HLS Pragma functions and implemented the same which allowed us to reduce execution time and achieve greater speed-up. The following were some of the pragmas we tried implementing: pipeline, unroll, loop-flatten. Pipeline: It allows the loop's actions to be carried out in parallel, as seen in the diagram below.
  Unroll: It allows the loop to be unrolled completely or partially to improve data access and throughput. For each loop iteration, a copy of the loop body is created in the RTL when unrolling the loop fully, allowing the complete loop to be performed parallely. While partially unrolling a loop, a factor N needs to be provided which results in many copies of the loop and a reduction of number of iterations. Loop flatten: It takes one whole clock cycle to move from an outer loop to an inner loop and vice-versa in the RTL implementation. Nested loops can be flattened and optimised as a single loop. This saves clock cycles, potentially allowing for more loop body logic optimization.
- Conversion to hardware design block: After comparing various quantization techniques and encoding patterns, we choose the most suitable technique and simulate the video codec in HLS and hence implement this on the Zynq7000 FPGA board. Mainly, we need 3 blocks to give an input to the Zedboard: DRAM, Video DMA and AXI4 Stream.
- Hardware input process: The video signal is split into multiple frames which are stored in the frame buffer and

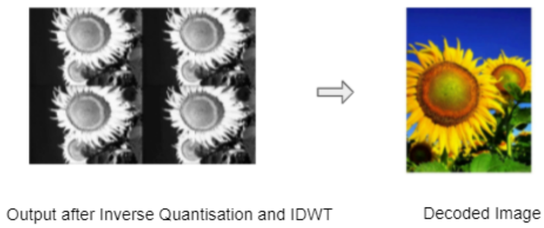

Fig. 8. Decoding Output Diagram

sent one after the other based on some signals. There are mainly two signals involved in this process: Start Of Frame (SOF) and End Of Line (EOL). If the resolution of a frame is XxY, then the frame is said to have X horizontal lines and Y vertical lines. The pixels of the frame are converted into bits line by line and it is then converted to hexadecimal format to feed into the FPGA. While sending the first pixel, SOF signal is made high and EOL is made high for every line that is passed. Once all the lines are transmitted, the SOF signal goes high again to indicate the start of the next frame. At the input of the board, SOF signal and EOL signal correspond to t-user and t-last respectively. The pixels are read by the slave of the AXI stream based on the s-ready and s-valid signals. The data read by the slave is then transferred to the master when the m-valid and m-ready signals are high. Each pixel is thus passed through the AXI stream interface at every positive edge of the clock cycle. We created the block diagram with the AXI IP and VDMA that passes the values of the frame buffer through the AXI interface to the Video IP. However, since we could not carry out the hardware implementation, we worked on a single frame and hence only a single SOF signal and 480 EOL signals were transmitted since the resolution of the video we used was 480x640.

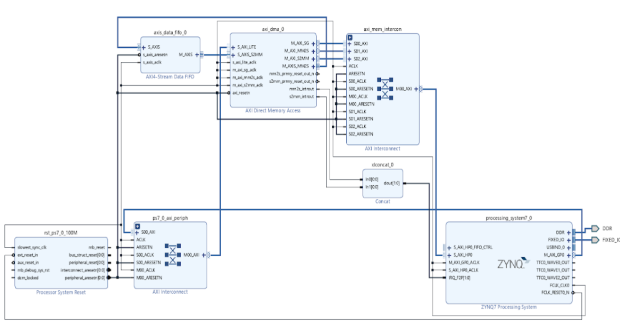

Fig. 9. Block Diagram of the AXI Interface

- Compare results with existing standards: With growing demand for higher quality videos, there has been a necessity for better compression techniques. High Efficiency Video Coding (HEVC) is the most recent generation of compression technology in the industry with higher capability than Advanced Video Coding (AVC). The compression ratio offered by HEVC is upto 1000:1. In order to produce better results, we need to evaluate our values with respect to industry standards and offer better compression.

III. Results

The results we obtained from our HLS implementation were compared with the compression standards and results from various papers. The total number of pixels occupied by an image is represented as MxN, where M is the number of columns (width) and N is the number of rows (height).

First, we only consider a single image. We perform the transformation, quantization, encoding and decoding only for this image for various bitshift values. Observing raw compression ratio, overall compression ratio, SSIM and PSNR, we choose suitable bitshift values. These are then applied to a variety of other images and the average values of all these parameters is found.

TABLE I
Readings for different bit shift values for all 4 bands: LL,LH,HL,HH

Bit Shifts	**Metrics**		
LL,LH	***Raw CR***	***Image CR***	***PSNR and SSIM***
1,1,1,1	4.9691	14.536	PSNR:24.9978 SSIM:0.95997
1,2,2,3	5.4013	15.283	PSNR:24.7945 SSIM:0.95894
1,4,4,5	5.7775	15.463	PSNR:24.6874 SSIM:0.95674
2,3,3,5	6.9434	15.889	PSNR:22.8951 SSIM:0.94261
2,4,4,5	7.1143	16.340	PSNR:21.5784 SSIM:0.93582

TABLE II
Readings for different bit shift values for only LL, LH bands:

Bit Shifts	**Metrics**		
LL,LH	***Raw CR***	***Image CR***	***PSNR and SSIM***
1,1	5.3216	15.427	PSNR: 22.4200 SSIM: 0.93382
1,3	5.6990	15.651	PSNR: 22.3485 SSIM: 0.93199
2,3	6.9956	15.707	PSNR:22.2707 SSIM: 0.92939
2,5	7 .1509	16.635	PSNR: 21.9740 SSIM: 0.92469
2,9	7.1545	16.720	PSNR: 21.7424 SSIM: 0.92157
3,7	9.7111	16.887	PSNR:21.4654 SSIM:0.91257

Here too, it is observed that as the bit shift values increase, we see that more bits are dropped in the encoding matrix and the raw compression ratio increases. But in this case, the raw compression ratio is much higher than what was observed earlier. What is worth noting is that the SSIM values are in an extremely satisfactory range. Moreover, we calculate the compression ratio based on image size after suitably representing the decoded matrix in JPEG format. This gives us compression ratios in the range 15-17.

Considering the trade offs we notice that for bit shift values of 2 and 5 for the LL and LH bands the raw compression ratio is 7.15 and the SSIM is 92.7%.

Moving forward with 2 and 5 bit shift values for the LL and LH bands, we calculated the Compression Ratio, PSNR and

SSIM for different images as the encoding process depends on the significant coefficients of DWT- bands of each image, hence the encoded data size is not a constant for all images which in turn results in varying compression ratios.

TABLE III
READINGS FOR DIFFERENT IMAGES VALUES CONSIDERING 2 AND 5 BITSHIFTS IN LL AND LH BANDS:

Image	***Raw CR***	***Image CR***	***PSNR***	***SSIM***
Image1	7 .1509	16.635	21.9740	0.92469
Image2	7.1989	19.819	24.4669	0.91532
Image3	7.3068	25.249	28.9726	0.90895
Image4	6.9919	20.389	23.6213	0.89917
Image5	7.0846	27.510	28.4213	0.98950
Image6	5.7815	22.982	27.3184	0.98695
Image7	6.1397	20.344	22.5991	0.82855
Average	***6.8042***	***21.85***	***24.9729***	***0.92187***

TABLE IV
COMPARISON WITH REFERENCE PAPER

Paper	***Average CR***	***Average PSNR***
Reference Paper 7	16.03	27dB
Our algorithm	21.85	25dB

CONCLUSION

In this paper, we initially tried implementing DCT and DWT transforms on Matlab and on comparing we obtained a better raw compression ratio of 6.31 when implementing the DWT transform and 1.62 for DCT transform. On the other hand, DCT showed better results in terms of PSNR of value 30.18 and 17.03 for DWT. While DCT shows better readings as compared to DWT, the hardware implementation of DCT might prove to be harder than that of DWT. In DCT, the quantization requires that each frame be broken down into 8x8 macro blocks. Quantization is done to each macro block and then the frame is reconstructed from these quantized macroblocks. DWT quantizes the entire frame at once instead of dividing into smaller macro blocks and then quantizing. Hence, we proceeded with DWT transform for further processes.

Simulink is a graphical programming environment for modeling dynamic systems. We used simulink as a bridge between MATLAB implementation and HLS implementation. Using inbuilt block libraries present in Simulink, we used the graphical block diagramming tool to get the entire flow of codec running using both DCT and DWT transforms.

Moving forward with DWT transform, we implemented DWT transform on the image matrix by convolving the image matrix with a particular kernel to obtain the different bands. We implemented bit shifting and the Zero Tree Wavelet encoding on all 4 bands, we were able to obtain a compression ratio of 16 and a SSIM of 95% . But by performing the same with a bit shift of 2   5 on the LL and LH bands, we were able to achieve compression ratios of 21.85 and were able to obtain a PSNR of 24.9729 with a similarity index of 92% , which is comparable with the compression ratio of JPEG standard of 10. Further, using pragma functions to parallelize the convolution process, we were able to achieve a speedup of 2s.

REFERENCES

[1] M. Jridi and P. K. Meher, Scalable Approximate DCT Architectures for Efficient HEVC-Compliant Video Coding, in IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 8, pp. 1815-1825, Aug. 2017, doi: 10.1109/TCSVT.2016.2556578.

[2] G. P. Joshi and N. P. Bhosale, ”Video compression using DWT algorithm implemented on FPGA,” 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), Pune, 2017, pp. 31-35, doi: 10.1109/ICDMAI.2017.8073481.

[3] R. Karthikeyan and G. Hegde, ”High performance VLSI architecture for 3-D DWT (discrete Wavelet Transform),” 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2018, pp. 892-897, doi: 10.1109/ICISC.2018.8398929.

[4] Boujelbene, R., Boubchir, L. and Ben Jemaa, Y. (2019), “Enhanced embedded zerotree wavelet algorithm for lossy image coding”. IET Image Processing, 13: 1364-1374. doi:10.1049/iet-ipr.2018.6052

[5] B. E. Caroline, G. Sheeba, J. Jeyarani and F. S. R. mary, ”A reconfigurable DCT/IDCT architecture for video codec: A Review,” 2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT’12), Coimbatore, 2012, pp. 1-5, doi: 10.1109/ICCCNT.2012.6396054.

[6] R. George and Manimekalai M.A.P., ”A novel approach for image compression using zero tree coding,” 2014 International Conference on Electronics and Communication Systems (ICECS), 2014, pp. 1-5, doi: 10.1109/ECS.2014.6892611

[7] Uday Bhade, Sanjeet Kumar, Prashant dwivedy, Shahbaz Soofi, Avinash Ray,“Comparative study of DWT, DCT, BTC and SVD Techniques for Image Compression”, Proceeding International conference on Recent Innovations in Signal Processing and Embedded Systems (RISE -2017), 27-29 October 2017.

[8] Rania Boujelbene , Larbi Boubchir, Yousra Ben Jemaa, “Enhanced embedded zerotree wavelet algorithm for lossy image coding”, IET Image Processing, Vol 13 , Issue: 8, 2019

[9] Wladimir J. van der Laan, Jos B.T.M. Roerdink, Andrei C. Jalba, “Accelerating Wavelet-Based Video Coding on Graphics Hardware using CUDA”, Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis, 2009.

[10] B. He, Y. Lei, L. Jiawen, F. Jia, W. Hao and W. Hua, ”Hardware Design of Video Compression System Based on TMS320DM368,” 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2017, pp. 294-297, doi: 10.1109/IHMSC.2017.178.

[11] K. Kolarov, W. Lynch, B. Arrighi and B. Hoover, ”A fractional chip wavelet zero tree codec (WZT) for video compression,” Proceedings DCC’99 Data Compression Conference (Cat. No. PR00096), 1999, pp. 535-, doi: 10.1109/DCC.1999.785692.

[12] S. Kim, D. Lee, J. Kim and H. Lee, A High-Throughput Hardware Design of a One-Dimensional SPIHT Algorithm, in IEEE Transactions on Multimedia, vol. 18, no. 3, pp. 392-404, March 2016, doi: 10.1109/TMM.2015.2514196.