Wireless Capsule Endoscopy

Kushaagra Goyal and Abhishek Lal

A Dissertation presented at IIT Delhi
for the course EED411 (Major project)

Adviser: Dr. Basabi Bhaumik
Electrical Engineering, Indian Institute of Technology Delhi

January 2016

Abstract

The image compressor inside wireless capsule endoscope should have low power consumption,high compression ratio and good reconstructed image quality. Simple and hardware efficient schemes like Reversible Colour Transformation, 5/3 integer wavelet with lifting architecture, DPCM and adaptive Golomb encoder were used for image compression. We have also modified the architecture of serializer and made it specific to our needs so that it ran at 8 times higher frequency of the image compressor. In the previous works [1] [2] it used to run at 32 times the frequency of the image compressor. The algorithm was verified and tested at software level in Matlab. Register Transfer level code was implemented in verilog for the proposed image compressor. The proposed algorithm gives a 91.89 percent compression with a PSNR of 37.24 dB. The RTL is designed to operate on a 256 by 256 resolution image at 2 frames per second. RTL was also written for the proposed serialiser which runs at one fourth the frequency as compared to the designs proposed in [1] [2].

Acknowledgements

First of all, we would like to thank Dr. Basabi Bhaumik for her valuable guidance and support during the project. We would also like to thank Sanjeeva Kumar Jain (PhD) for helping us during the project. We would also like to thank the committe of examiners for reviewing our work and pointing out our potential pitfalls.

Kushaagra Goyal

Abhishek Lal

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

Wireless capsule endoscopy uses a miniature camera for seeing the entire digestive system.The whole system is designed inside a small capsule for minimum invasiveness. The capsule consists of a CMOS image sensor array, LED's, small battery and a RF transmitter [3].

The capsule weighs less than 4g and measures about 11mm in diameter and 26mm in length. It is made of plastic, is biocompatible and resistant to digestive fluids. It is used to examine the areas of small intestine which cannot be easily seen by other types of endoscopy such as colonoscopy or eosophagogastroduodenscopy [4]. Regarding the safety issues, it is considered to be completely safe except a few minor case where the capsule gets stuck inside the patient. In general, it comes out with the stool in 24-48 hours.

The entire WCE system consists of 4 parts: the capsule endoscope (CE), data receiving box, working station and application software. The capsule is ingested by the patient, the small camera inside takes images of the GI tract and the image compressor compresses the image and then it is passed on to the transmitter. The transmitter

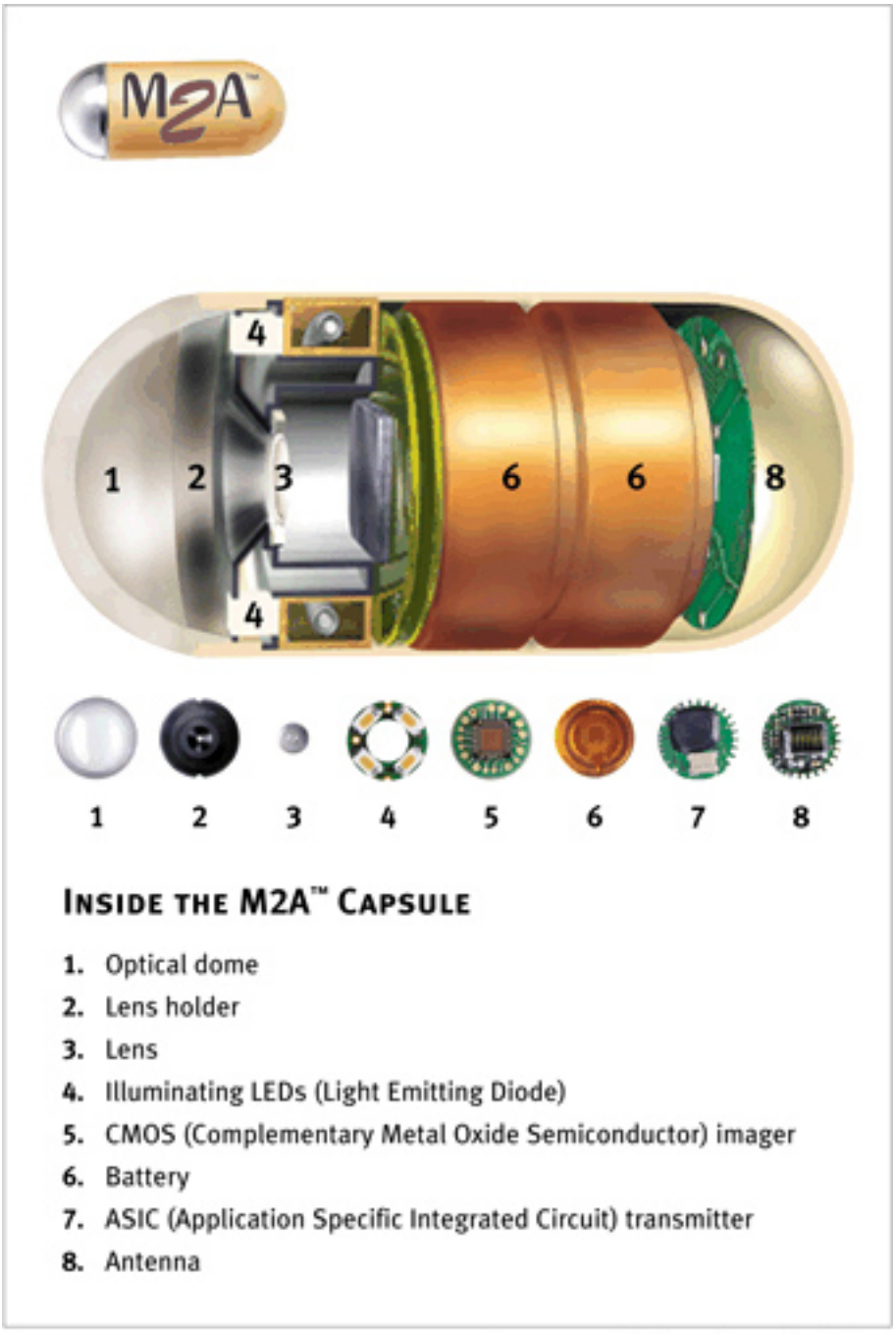

Figure 1.1: Wireless Endoscopy Capsule

transmits the bits to the receiver station outside. The capsule runs for about 8-10 hours and sends around 1,00,000 images. These images can then be analysed by the doctor to determine bleeding, ulcers or cancer. There are three main companies producing wireless CE systems by the approval of FDA. Another capsule by Chinese company has come into the market but its not approved by FDA. It is attractable due to its low price. A table of commercially available capsule is given in the figure 1.2. [3]

Table 1 Comparison between commercially available capsule endoscopy devices

Capsule	PillCam® SB 3 Given Imaging	EndoCapsule® Olympus America	MiroCam® Intromedic Company	OMOM® Jianshan Science and Technology
Size	Length: 26.2 mm Diameter: 11.4 mm	Length: 26 mm Diameter: 11 mm	Length: 24.5 mm Diameter: 10.8 mm	Length: 27.9 mm Diameter: 13 mm
Weight	3.00 g	3.50 g	3.25-4.70 g	6.00 g
Battery life	8 h or longer	8 h or longer	11 h or longer	6-8 h or longer
Resolution	340 × 340 30% better than SB2	512 × 512	320 × 320	640 × 480
Frames per second	2 fps or 2-6 fps	2 fps	3 fps	2 fps
Field of view	156°	145°	170°	140°
Communication	Radio Frequency Communication	Radio Frequency Communication	Human Body Communication	Radio Frequency Communication
FDA approval	Yes	Yes	Yes	No
Price per capsule	$500	$500	$500	$250

Figure 1.2: Commercially available Capsules

1.2 Motivation

Wireless capsule endoscopy is considered the best way to see the GI tract in a non-intrusive manner. But it has inherent drawbacks and limitations. Sometimes the capsule fails before the entire journey is completed. Also it has the problem of capsule retention. It is not so common, but there have already been few cases where retention of the capsule inside the lumen is observed and surgical intervention is desired. [3] . But the main need of the hour is reducing the power consumption of the capsule so that its battery lasts longer. Reducing the power consumption can also help in reducing the size of the capsule, as we can decrease the size of the batteries. Additional

features like manoeuvrability or controlled drug release can be introduced. Controlled drug release can help in easy treatment of GI diseases. The main power is consumed in the transmitter. Improving the image compressor can reduce the number of bits to be transmitted and thus the power consumption can be decreased.

These drawbacks were the main motivation behind this work. The image compressor and the transmitter are the main power consumers in wireless capsule endoscopy. An efficient image compressor can help in reducing power of both of these components. So in this work, we have designed an efficient image compressor as well as the interface to the transmitter which reconstructs the original image with a reasonable PSNR of 37.24 and a compression ratio of 91.89 percent. Furthermore the design of the interface between image compressor and transmitter is made efficient so that it is specifically designed for the algorithm and works at lower frequency as compared to the previous works.

1.3 Thesis Organisation

The thesis consists of six chapters. In the second chapter, we discuss an overview of image compression techniques previously employed for wireless capsule endoscopy. A brief summary of various concepts is also illustrated in this chapter. In the third chapter, we discuss the proposed algorithm for the image compressor. In the fourth chapter, we discuss the hardware architecture for the proposed image compressor and serializer. In the fifth chapter, we discuss the results of our proposed algorithm and compare it with the previous works. Finally a conclusion and a summary of our achievements is given.

Chapter 2

Literature Survey

2.1 Introduction

Image compression tries to remove redundancy and irrelevancy in the data in order to store the data in an efficient form. Generally there are three type of redundancies in an image coding redundancy, spatial redundancy, psychovisual redundancy.

Coding redundancy is attributed to the fact that we use exactly 8*3 bits to represent each pixel in an image. But since some values occur more than others, it could be beneficial if more frequently occurring bits are coded with less number of bits and less frequently occurring bits are coded with higher number of bits. Techniques like Huffman coding and Golomb-unary encoding are used for removing coding redundancy. This is called variable length coding as variable number of bits are used to encode each pixel.

Spatial redundancy corresponds to the fact that each pixel is related to its neighbours, i.e. a pixel can be estimated based on its neighbours. This type of redundancy is generally removed by taking transforms like DCT and DWT and thresholding. DPCM can also be used for exploiting spatial redundancy.

Psychovisual redundancy, Human eyes do not respond with equal sensitivity to

all visual information. They are more sensitive to lower frequencies than the higher frequencies in the visual spectrum. So we can exploit this by discarding data which is not perceptually significant.

Image compression can be lossy or lossless. In lossless compression, the reconstructed image is identical to the original image. We generally exploit only statistical redundancy and the compression achieved is low. With lossy compression, we can achieve very high compression rates as we utilise both spatial as well as psychovisual redundancies. But the reconstructed image is not identical to the original image. For an efficient lossy compression algorithm, the reconstructed image should seem perceptually indifferent to the human eye.

It is impractical to use state of the art image and video compression algorithms like JPEG2000, MPEG for wireless capsule endoscopy due to their computational complexity and memory requirements. It is necessary to design an image compressor which is computationally efficient, consumes low power, requires minimal memory and achieves high compression. Lots of research has gone into it and various architecture have cropped up in the scientific literature. The previous works based on this use 2D-DCT or 2D-DWT [5] for removing spatial redundancy in the images. They operate on blocks of size 4*4 or 8*8. These approaches require storing multiple rows of image before they start processing on the block. The memory component consumes large silicon area and requires a lot of power. Recently, DPCM based approaches which do not utilise any transforms have come up in the literature. These do not require any storage area and hence are efficient in terms of power. The approach mentioned in [1] utilizes colour transformation along with DPCM and static golomb encoding to achieve a reasonable compression rate off 80 percent with a good PSNR. Recently in [2], Khan et als approach was modified to include adaptive golomb and quantization to increase the compression. This approach achieved very high compres-

sion rates (90 percent) with a good PSNR. In the next few subsection, we are going to discuss the techiques which are commonly used for image compression for wireless capsule endoscopy.

2.2 Reversible Colour Transform

Image contains R, G, B components. Generally these R, G, B components are highly correlated in any image. Colour transformations are used to separate the information into luminance and chrominance components.The luminance component contains the essential information in the image. In a lossy image compressor, chrominance components can be either subsampled or quantised.
According to [2], the transformation specified in JPEG2000 , RGB to YUV, is ideal for endoscopy images.

$$Y = floor((R + 2G + B)/4) \tag{2.1}$$

$$U = R - G \tag{2.2}$$

$$V = G - B \tag{2.3}$$

This transformation effectively decouples the information. Y component contains the luminance value whereas the U and V components contain the chrominance component. U and V components can be subsampled or quantised without a significant loss of information. Main information about the image is contained in the Y component. Moreover this is a fully reversible transform, so there is no loss of information. [6]

2.3 Frequency Transforms

The image is transformed to a different domain where compression can be done efficiently. This approach utilises the spatial redundancy. Fourier, discrete cosine and wavelet transforms are used to convert the image into frequency domain. In the frequency domain, high frequency components are generally dropped and the low frequency components are retained. In Images, 2D DCT or DWT need to be used to remove 2D spatial interdependency but this requires storing a few row of pixels, which increases the memory consumption. To reduce the memory consumption, we propose to use 1D DWT in our image compressor. This will only utilise 1D spatial redundancy but it will not consume any memory and will be power efficient.

2.3.1 DWT (Discrete Wavelet Transform)

Discrete Wavelet Transform is used for resolution of signals with localisation in both time and space. In contrast to DCT or Fourier, where we completely go into frequency domain, in DWT we have some information on time as well as frequency. In it, a series of filters are applied to resolve a signal into high and low frequency components. As shown in the figure 2.1 [7], the input signal is passed through a level of high pass and low pass filters to produce wavelet coefficients at various levels. The choice of filters used differentiates among the different type of wavelet transforms.

In general, for hardware implementation it could be computationally expensive as we require memory to store filter coefficients and multiplier operations are required. Later Swelden [8], introduced the lifting scheme architecture for computing wavelet transforms. Furthermore, integer wavelets came into use which can be computed efficiently in hardware through simple shift and addition operations. A list of integer wavelets can be seen in [9].

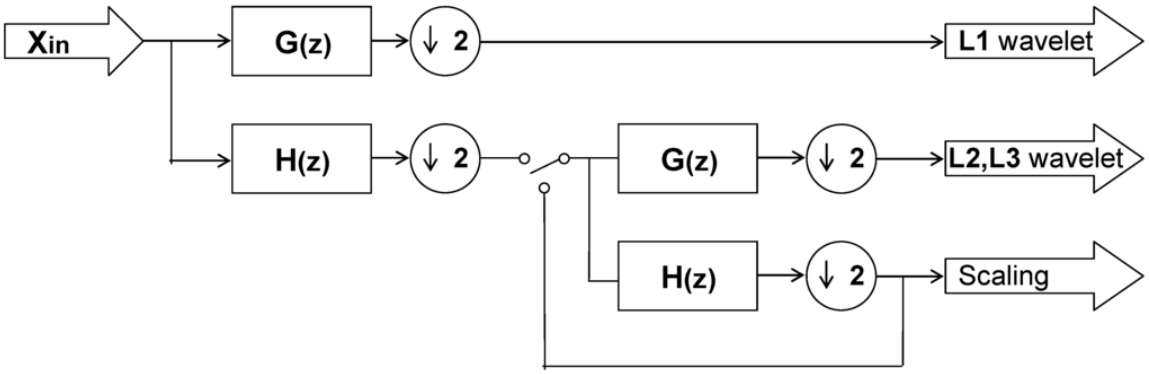

Figure 2.1: 2 Level 1D Wavelet Decomposition

For our work we used 5/3 integer wavelet. A lifting scheme architecture was used for implementing the wavelet transform. S[n] is the low pass output whereas D[n] is the high pass output.

$$S0[n] = X[2n] \tag{2.4}$$

$$D0[n] = X[2n+1] \tag{2.5}$$

$$D[n] = D0[n] - floor(1/2 * (S0[n+1] + S0[n])) \tag{2.6}$$

$$S[n] = S0[n] + floor(1/4 * (D[n] + D[n-1]) + 1/2) \tag{2.7}$$

The lifting scheme architecture is shown in figure 2.2 [8].

As compared to the conventional convolutional architecture for computing DWT, lifting architecture required significantly less number of shifting and addition operations.

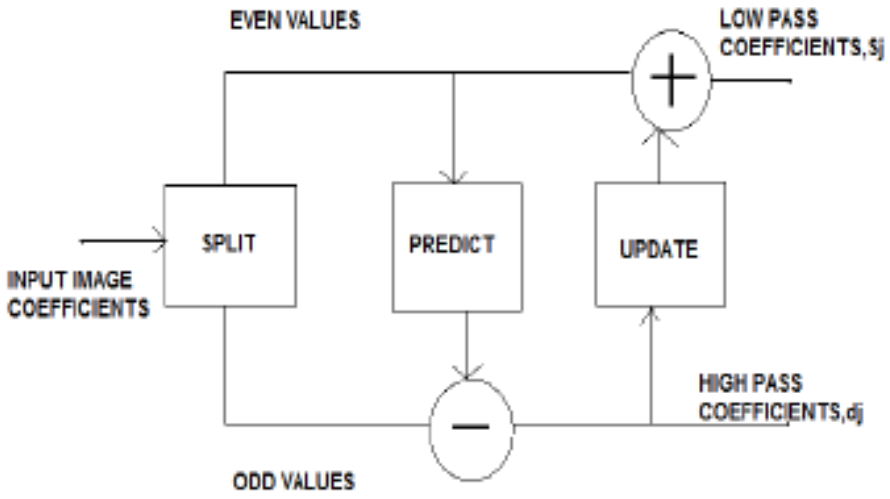

Figure 2.2: Lifting Wavelet Architecture

2.4 Variable Length Coding

It is much more efficient than fixed length coding as each symbol is encoded with different number of bits. Symbols with higher frequency must be encoded with lower number of bits, whereas symbols with lower frequency can be encoded with higher number of bits. Huffman encoding is the optimal algorithm used for this. For hardware implementation purposes, Golomb-unary encoder is used as it can be implemented efficiently in hardware. It is very efficient for geometric distributions.

2.4.1 Golomb Encoding

In golomb encoding, number are divided into groups of m. Symbols in same group have code words of equal length. Whereas groups with smaller symbol values have shorter codes. An example of golomb encoding with m=5 is given in figure 2.3 [10]

The number to be encoded is divided by m. The quotient is encoded using unary encoding, whereas the remainder is binary coded. For hardware implementations,

n	q	r	code
0	0	0	000
1	0	1	001
2	0	2	010
3	0	3	0110
4	0	4	0111

n	q	r	code
5	1	0	1000
6	1	1	1001
7	1	2	1010
8	1	3	10110
9	1	4	10111

n	q	r	code
10	2	0	11000
11	2	1	11001
12	2	2	11010
13	2	3	110110
14	2	4	110111

Figure 2.3: Example of Golomb Code with m=5

m is taken of the form 2 power k, so that division operation becomes bit shifting. Deciding this parameter m is very crucial for efficient encoding and high compression. So an adaptive golomb encoder is used which calculates the value of m depending on the context variables. Adaptive golomb can increase compression as compared to static parameter encoding. The algorithm for the adaptive golomb is shown below.

A = A0;

N = 1;

while (pixel)

K = max(0,ceil(log2(A/2*N))

Encode X using golomb with 2 power k as parameter

If(N = Nmax)

A = A/2 , N = N/2

Else

A = A + X, N = N+1

End

End

On comparing the performances in kinde et al's approach and khan et al approach, it can be seen that adaptive golomb can lead to higher compression, with small additional hardware. With adaptive golomb, we shall need to maintain extra context variables A and N.

2.5 DPCM (Differential Pulse Code Modulation)

DPCM (Differential Pulse Code Modulation) is used to reduce the spatial redundancy in the image. In it each pixel, Pij is estimated by the previously encoded pixels. Various prediction functions are mentioned which utilise the pixels Pi-1, j , Pi-1, j-1 and P i ,j-1 for estimating the current Pi,j pixel. In order to avoid storing a row of pixels in the hardware implementation, 1D estimation is preferred where Pi,j = P i,j-1 . The error is dX which is given by the actual pixel value minus the prediction. Usually these dX values are very small and are centered around zero. These dX values can be very efficiently encoded using golomb rice encoder. The prediction can be improved by using complicated prediction functions, but they increase the hardware complexity. This coupled with a golomb rice encoder is a lossless step and is very commonly used in a lossless image compressor algorithm.

2.6 Conclusion

Khan [1] proposed an image compressor for capsule endoscopy which required almost zero memory. He used RGB to YUV conversion. He subsampled the U and V components, and used DPCM along with static golomb encoding. He was able to achieve around 82 percent compression with a reasonable PSNR of 40. Furthermore it had very low hardware complexity and consumed very little power. The block diagram for his approach is shown in figure 2.4. Kinde et al in his approach slightly modified the approach of khan et al, where he introduced quantisation and used

an adaptive golomb instead of the static golomb encoder. This approach produced better results with a compression of nearly 90 percent and PSNR of almost 40. Both the approaches used negligible memory in their implementation and are hardware efficient. The block diagram is also shown in figure 2.5.

Various other approaches have also been proposed which can achieve compression upto 92 percent [5] using transforms like DCT but these approaches require buffer memory and thus consume a lot of power. In our approach, we modified the algorithm utilised by Kinde et al and Khan et al, and introduced an additional 2 level 1D DWT step. Our algorithm gives a compression of 91.9 percent with a PSNR of 37.24. Furthermore our approach does not require much additional hardware.

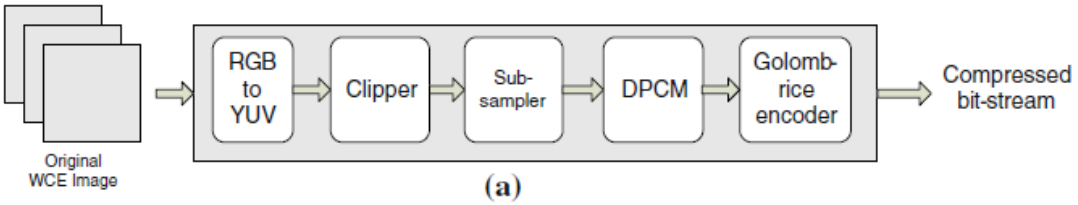

Figure 2.4: Block Diagram of Compressor Proposed by Khan

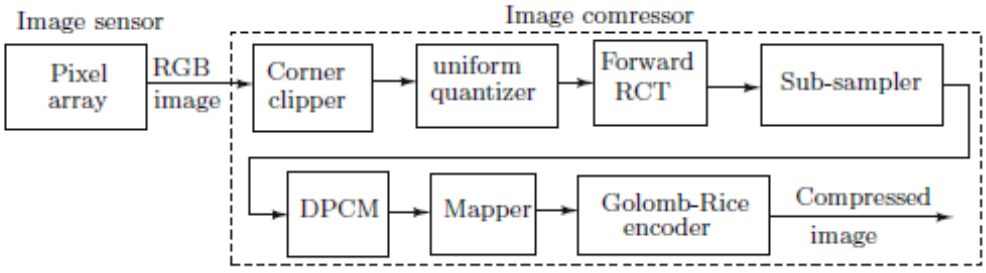

Figure 2.5: Block Diagram of Compressor Proposed by Kinde

Chapter 3

Proposed Design

3.1 Introduction

As seen in the literature survey, the designs proposed by khan et al [1] and Kinde et al [2] are power effiicient as they use zero buffer memory and achieve good compression. We propose to further enhance the algorithm by using 2-level 1D DWT instead of subsampling.We hope to increase the PSNR of the reconstructed images by this algorithmic optimisation. Furthermore, the architecture for the interface between image compressor and serialiser proposed in [2] is very inefficient and consumes a lot of power. We propose to decrease the maximum length of code produced by golomb rice to 16 instead of the usual 32 used in [2].This will reduce the frequency of operation of serializer to half of its previous value. Also we propose a new architecture for the serialiser which utilises the specific properties of the image compressor algorithm, so that the operating frequency further reduces by half. Through this optimisation, our interface to the transmitter will work at one-fourth the frequency. The block diagram of the complete image compressor and decompressor is shown in the figures 3.1 and 3.2.

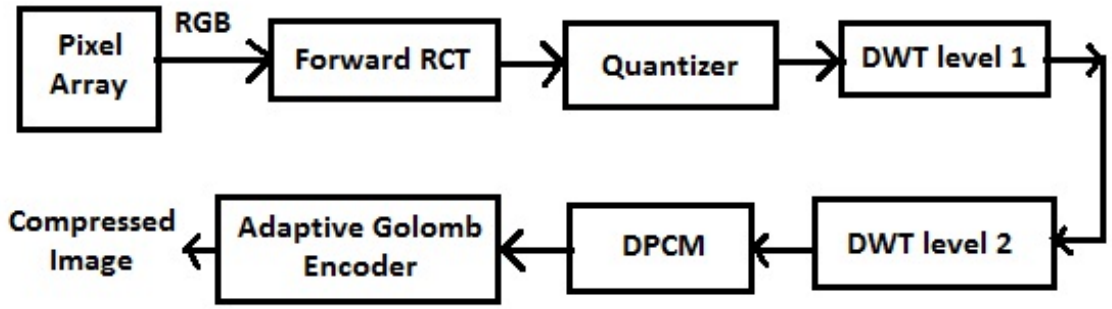

Figure 3.1: Block Diagram Image Compressor

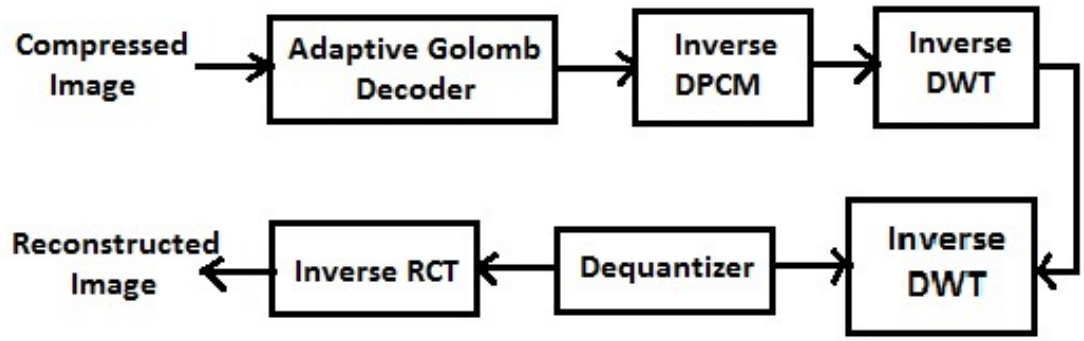

Figure 3.2: Block Diagram Decompressor

3.2 Forward RCT Transform

We can see from figure 3.3 that the values of Red, Green and Blues are correlated. In most of the images of gastro-intestinal tract, red is the most dominant colour. Green and Blue have lower intensity. Thus we can safely say that if we find the difference between Green and Blue channels we will get a signal with very less information. It will be a low pass signal as Green and Blue are highly correlated. Thus we used RGB to YUV colour space conversion to de-correlate the image.

From table 3.1 it can be easily seen that the Red has more energy content (since it has maximum mean) compared to Green and Blue. It can also be seen that the standard deviation of the data has been reduced after the RCT thereby increasing

coding redundancy. From table 3.2 we can see that Red, Green and Blue components are very highly correlated. The main aim of RCT is to de-correlate the image. From table 3.3 we can see that the new components YUV are less correlated compared to RGB.

In figure 3.3 we have plotted the R, G and B values of the pixels along the middle row. We can see that Green and Blue components have almost similar energy content. Figure 3.4 contains Y, U and V values along the middle row. By comparing figure 3.3 and figure 3.4 we can see that the difference in the components of YUV is significantly more compared to RGB.

Table 3.1: Mean and Standard Deviation of RGB and YUV

Metric	R	G	B	Y	U	V
Mean	106.59	73.21	62.59	78.54	33.45	10.76
Std	53.08	34.98	39.26	41.09	18.62	07.45

Table 3.2: Correlation between RGB

Correlation Matrix	R	G	B
Red	1	0.9626	0.9159
Green	0.9626	1	0.9864
Blue	0.9159	0.9864	1

Table 3.3: Correlation between YUV

Correlation Matrix	Y	U	V
Y	1	0.6814	0.6599
U	0.6814	1	0.8554
V	0.6599	0.8544	1

The Y corresponds to the intensity value of the image while U and V represents the chrominance components. The value of Y varies from 0 to 255 whereas the values of U and V varies from -255 to 255. Thus Y requires 8 bits and U,V requires 9 bits to encode.

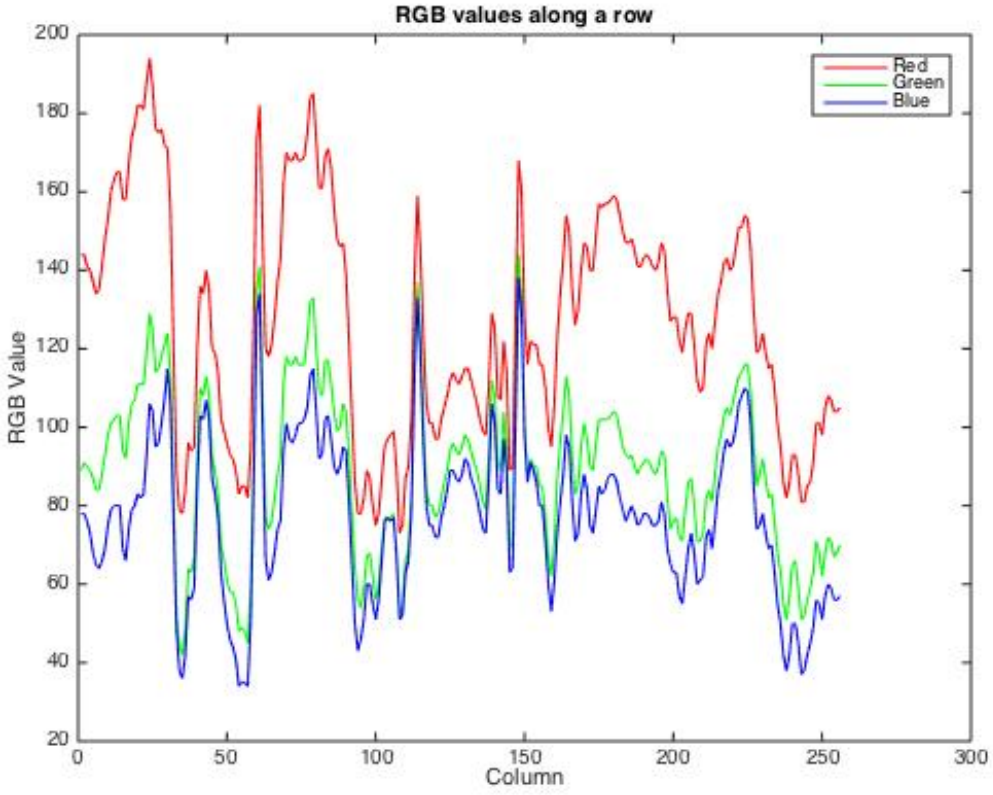

Figure 3.3: RGB Values In Middle Row

3.3 Quantization

The quantization plays a very important role in the compression ratio. Increasing the quantization level increases compression at the cost of image quality. To make it hardware efficient we quantize by 2 power n since it is equivalent to ignoring the last n least significant bits. The quantization formula used is

$$Q(x,n) = floor((floor(\frac{x}{2^{n-1}}) + 1)/2), if x < (255 - 2(n-1)) \tag{3.1}$$

$$floor(x/2^n), otherwise \tag{3.2}$$

In our design we have used n=3. This maps -4 to 3 to 0, 4 to 11 to 1 and so on. Quantization is another important way to improve coding redundancy. Comparing

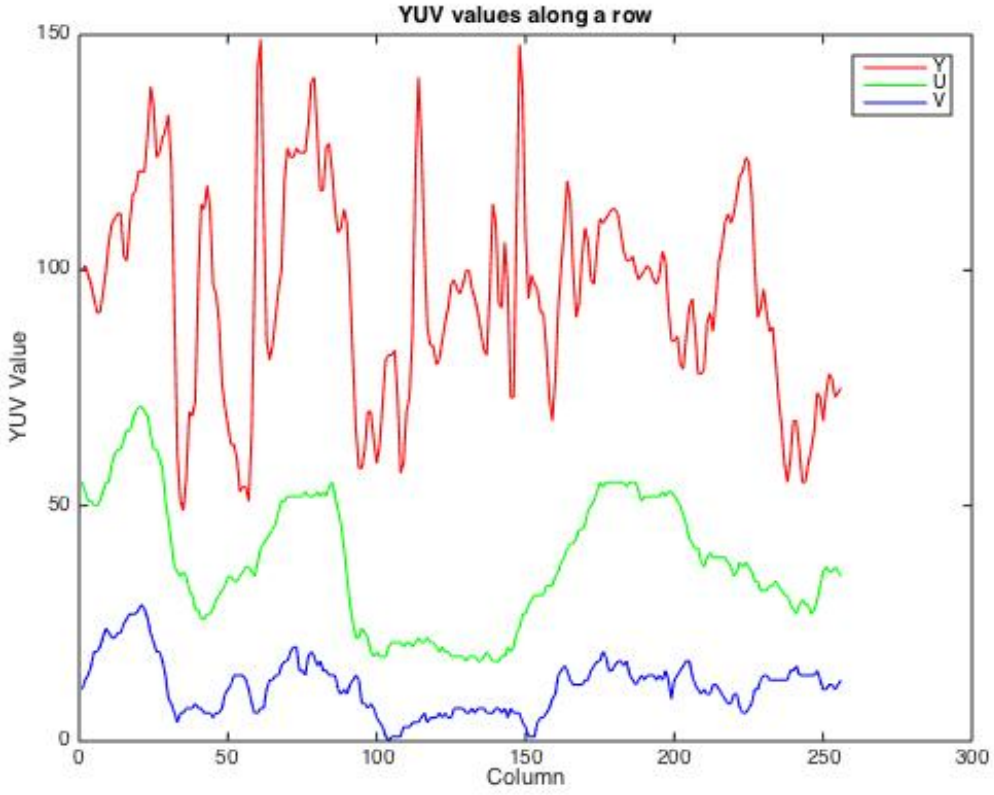

Figure 3.4: YUV Values In Middle Row

table 3.1 and table 3.4 we can see that the standard deviation of the quantized YUV components has been reduced significantly.

Table 3.4: Mean and Standard Deviation of Quantised YUV

Metric	Y	U	V
Mean	10.5466	4.4497	1.4257
Std	4.4973	2.0485	0.9138

3.4 Discrete Wavelet Transform

The amount of information present in U and V is significantly less. Therefore it is not necessary to send entire data of U and V. In the previous work by Khan et al [1] and Kinde et al [2] they have sub-sampled U and V components. But as they

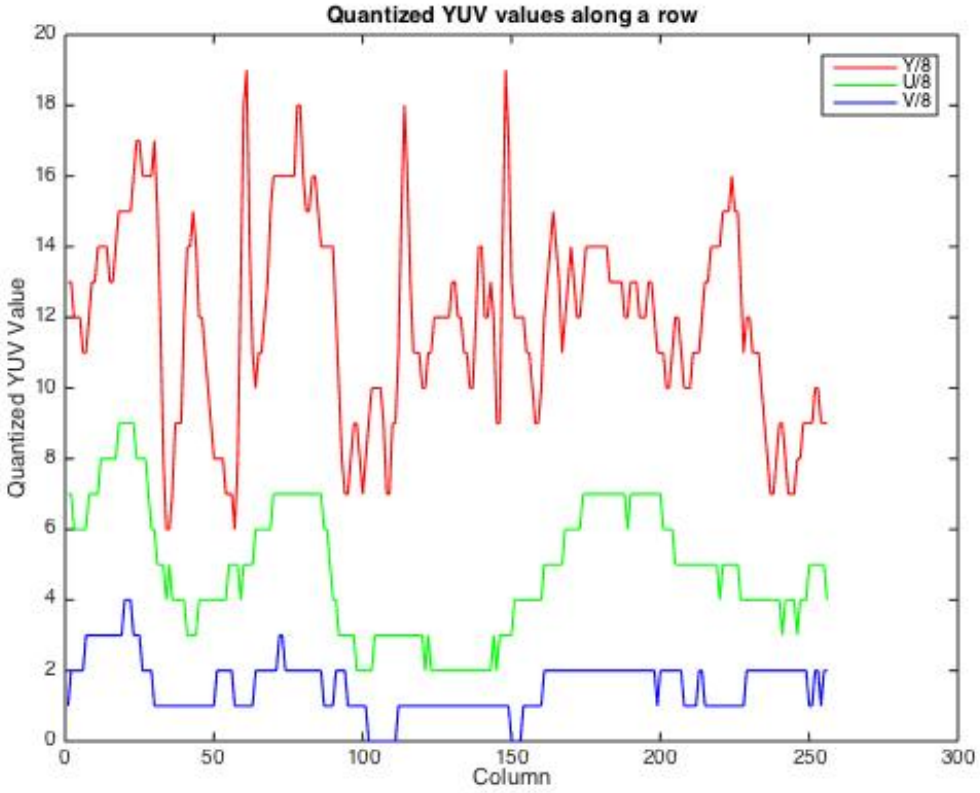

Figure 3.5: Quantised YUV on Middle Row

are indiscriminately removing information their PSNR varies over images. A better approach to this problem would be to first take wavelet transform of the image and then remove the high frequency component. The end result will contain only half of the information which is equivalent to sub sampling by 2 but at the same time maintaining the PSNR. So instead of subsampling by 4 we took 2-level DWT of the U and V components.

Figure 3.6 shows that most of the energy content of U and V after first level DWT is present in its lower frequency components. Therefore, it is safe to ignore the higher frequency components. Figure 3.7 shows that most of the energy content of U and V after second level DWT is present in its lower frequency components. Therefore, it is safe to ignore the higher frequency components. Figure 3.8, 3.9, 3.10, 3.11 shows

the histogram if DPCM is applied directly after RGB, Forward RCT, Quantization and 2 level DWT respectively. Comparing these figures, we can see that histogram is narrowed the most after quantization step. We can also see in figure 3.11 that applying DWT has not affected the spatial redundancy property as zero is the most occurring value.

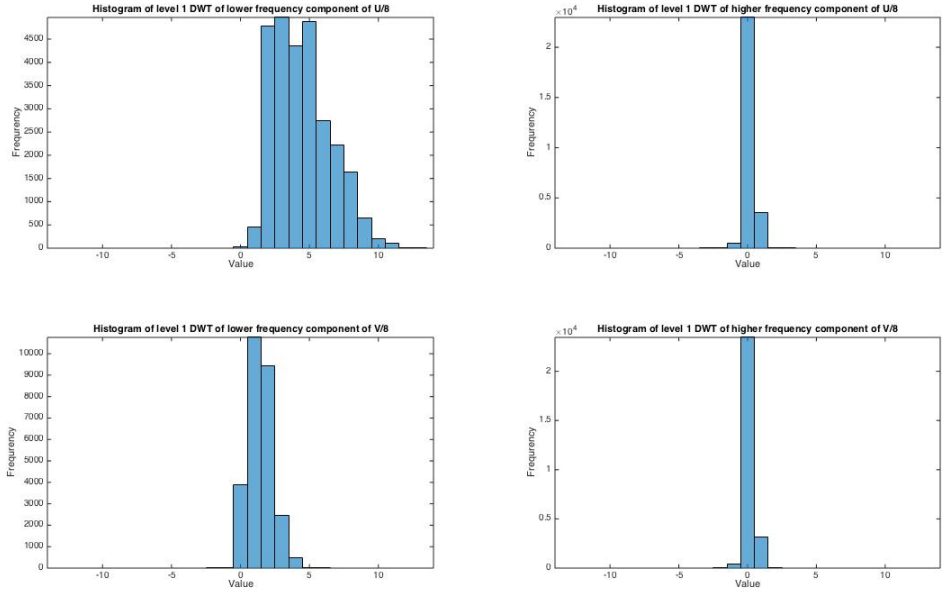

Figure 3.6: Histogram of DWT Components

3.5 DPCM

DPCM exploits the spatial redundancy in an image. In DPCM a current pixel is represented by its difference from previous pixel. Since there is no drastic changes in the value of two consecutive pixels the value of DPCM is usually small. There can be both 1D and 2D predictive schemes but since 2D predictive scheme would requires large amount of storage in hardware implementation so we only considered 1D DPCM.

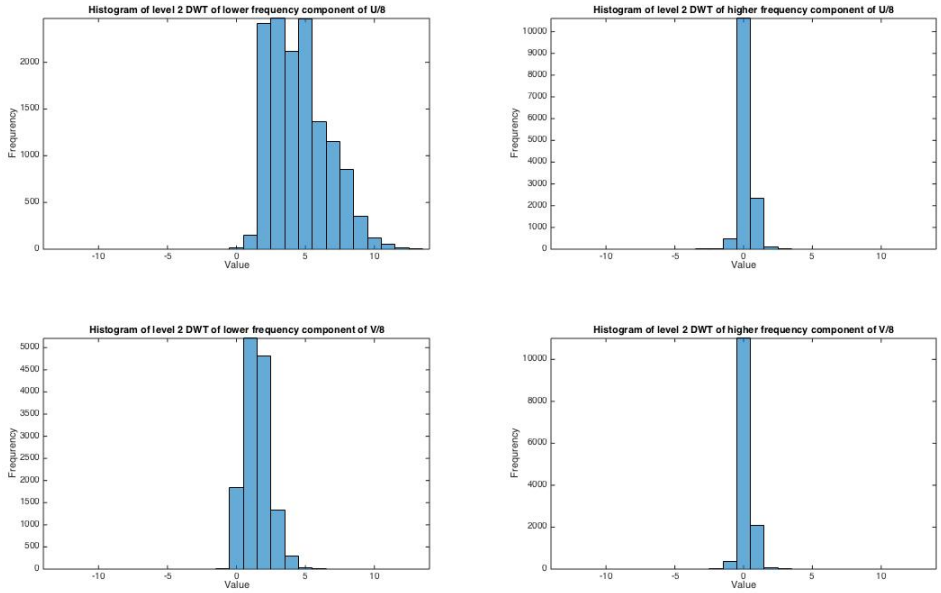

Figure 3.7: Histogram of 2nd Level DWT Components

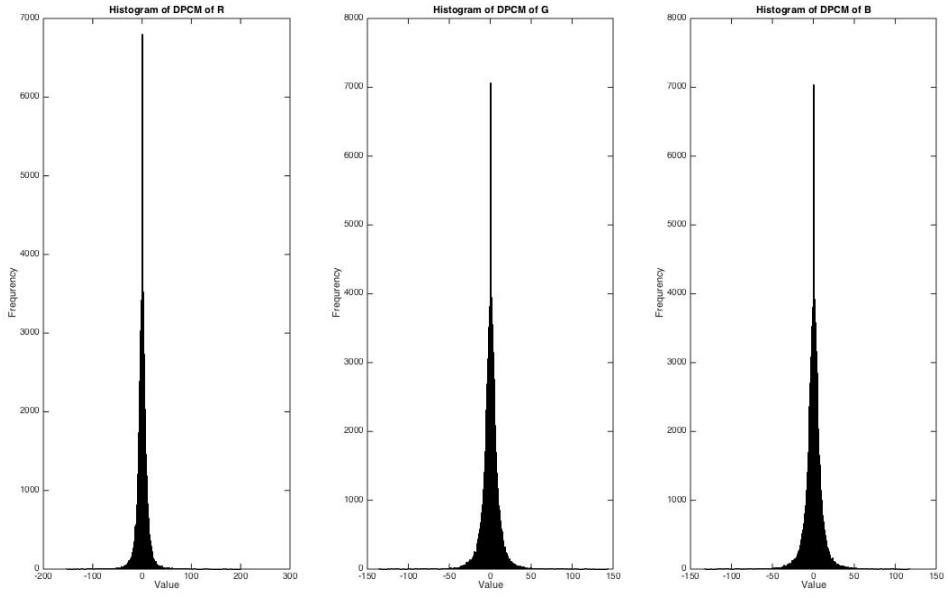

Figure 3.8: Histogram of DPCM of RGB

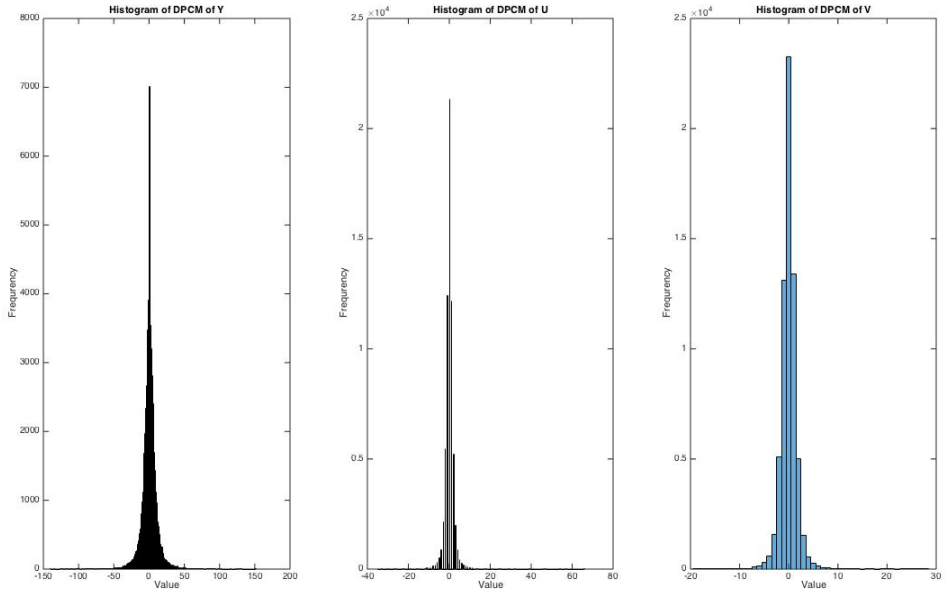

Figure 3.9: Histogram of DPCM of YUV

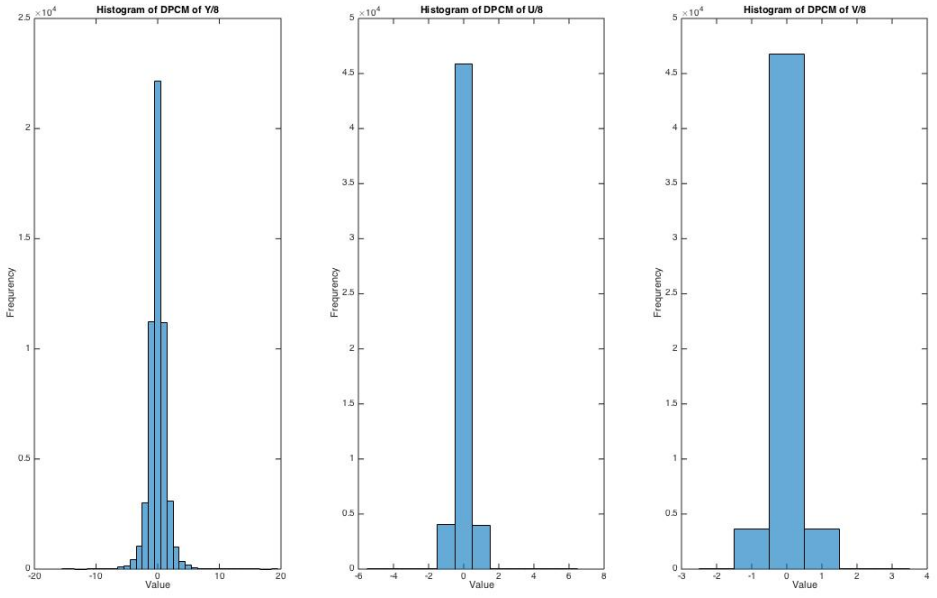

Figure 3.10: Histogram of DPCM of Quantised YUV

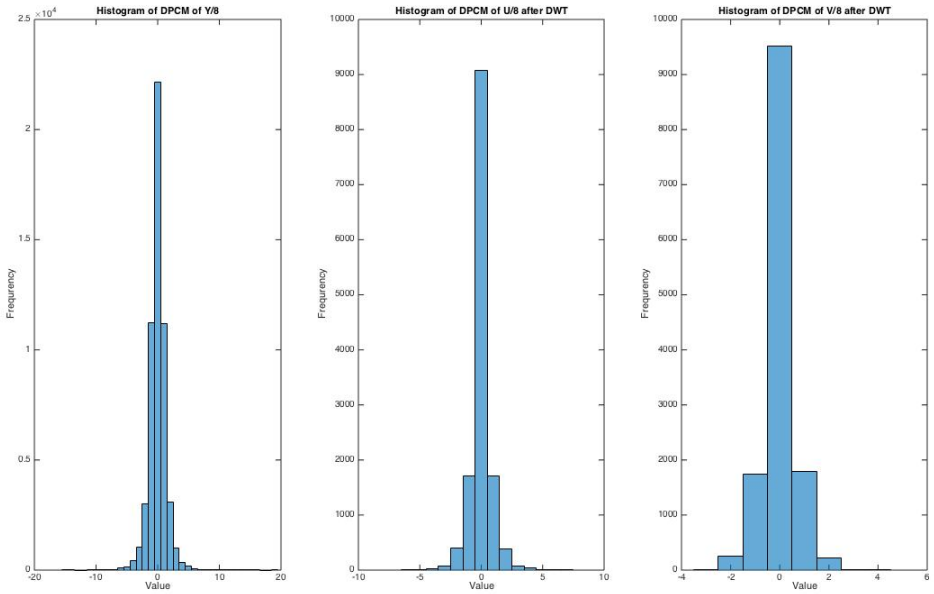

Figure 3.11: Histogram of DPCM of Quantised YUV obtained after DWT step

3.6 Corner Clipper

The lens takes circular images but outputs a square image. Therefore, the corners of the square image have no data. The pixel value of those corner parts are zero. But the proposed Adaptive Golomb Rice Encoder will encode it in 1 bit per pixel which reduces our compression ration. Since cropping the image in an exact circular fashion is very hardware expensive we can crop the sides with a straight line. The cropping technique is discussed by Khan [1] in details.

3.7 Conclusion

The design proposed in this thesis achieves both higher percentage compression and good PSNR. Although it came at a price of extra hardware complexity but we have actually managed to save a considerable amount of power in the RF transmitter. We

have exploited unique properties of GI images to make our computation more efficient. The algorithm is designed keeping in mind the simplicity of hardware implementation. Now in the following section we will discuss about the hardware implementation of our proposed algorithm.

Chapter 4

Hardware Architecture

4.1 Introduction

The hardware implementation is done using verilog. The behavioral verification was done using Xilinx Vivado. The power, area, frequency of operation will be calculated using Synopsys Design Vision with 130 nm CMOS technology (UMC HS library). There were total of 6 modules under the main module named Compressor along with a testbench module. The modules are Control Unit, Camera Interface, Forward RCT, Quantizer, Discrete Wavelet Transform, Adaptive Golomb Encoder and Serializer.

4.2 Modules

4.2.1 Control Unit

Control Unit is the heart of the design and generates control signals for the entire design. It takes clock and reset and maintains three counters. The first counter maintains the row and column number of the current pixel being processed. The row and column numbers both lie in between 0 to 255. The second counter maintains whether the output data is corresponding to Y or U or V. It is a modulo 3 counter.

The third is a modulo 12 counter which helps in determining whether the output generated is valid or not. It also helps in updating the registers of DWT module by providing the information whether to update or not. Modulo 12 counter is used for a very specific reason. Since we have taken two level DWT transform of the U and V components of forward RCT data, the data is not available at every clock edge but rather every fourth U or V cycle. But since Y is sent as it is, its data is available at every Y cycle. Reset values of the registers are tuned depending on how many cycles after reset is low does the output comes from the camera.

4.2.2 Corner Clipper

The corner clipper module takes row number and column number generated by the control unit as input and determines whether the pixel lies in corner clipped region or not. The module contains 4 comparators and an adder and subtractor module.

4.2.3 Camera Interface

This module connects cameras output to compressors input. It stores the RGB values of current and previous pixels. While it takes input the current pixel it outputs the pixel stored previously. It therefore uses six 8-bit registers and three 8-bit multiplexers. Reset values of all the registers are 0.

4.2.4 Forward RCT

This module takes the RGB values of a pixel and performs colour space transformation from RGB to YUV. The module is purely combinational and the input to this changes after every 3 clock cycles.

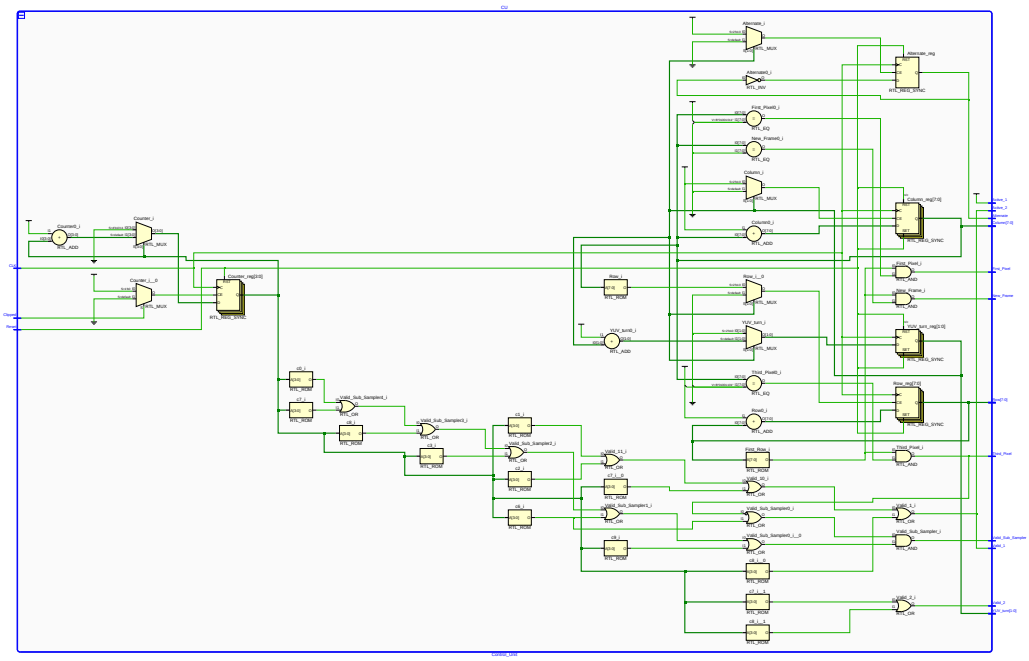

Figure 4.1: Schematic of Control Unit Synthesised By Xilinx

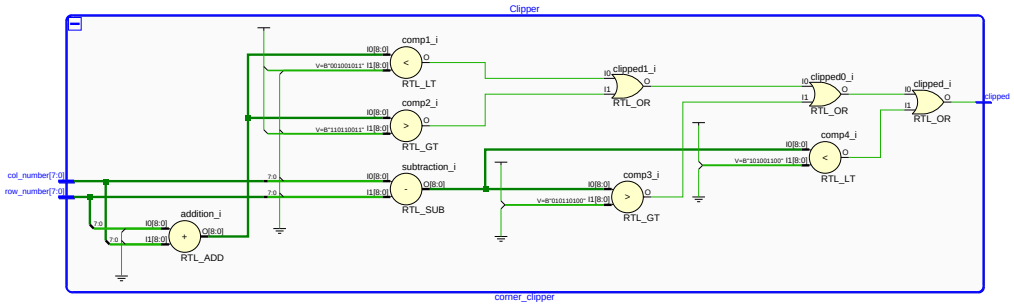

Figure 4.2: Schematic of Clipper Synthesised By Xilinx

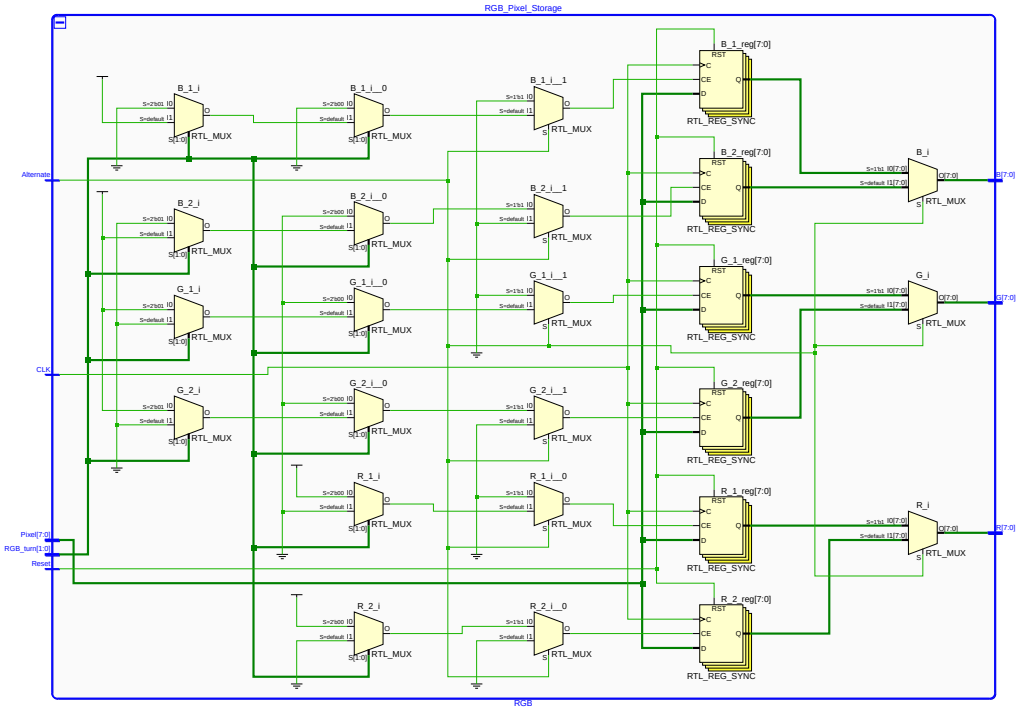

Figure 4.3: Schematic of Camera Interface Synthesised By Xilinx

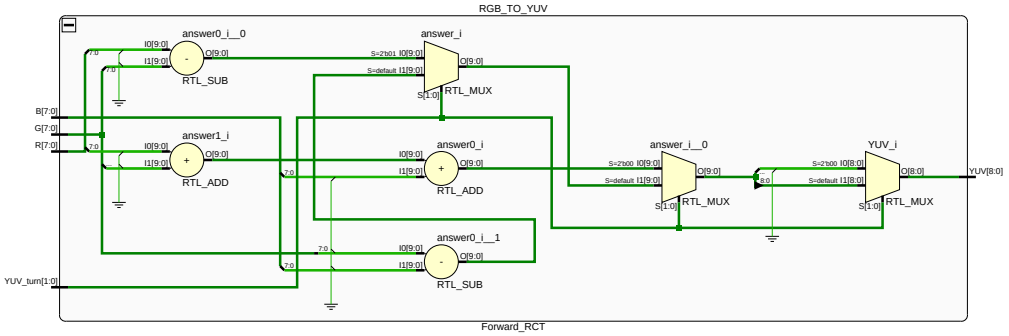

Figure 4.4: Schematic of Forward RCT Synthesised By Xilinx

4.2.5 Quantizer

This combinational module takes the YUV output of the forward RCT block and performs 3-bit quantization. The quantization implemented in hardware is floor((floor(x/4) + 1)/2). x varies form -255 to 255. Special exception has been made to map x form 252-255 to 31.The quantizer consists of an incrementer, 7-bit comparator and a 6-bit multiplexer.

4.2.6 Discrete Wavelet Transform

This module takes a signal as input and outputs only the low frequency component of the DWT transform. A 5/3 integer wavelet with lifting architecture is used. Reset value of all the registers are 0.

4.2.7 Serialiser

The Serializer as proposed by [1] [2] works at 32 times higher frequency than that of compressor. Because of this the Serializer becomes the major power consuming module. The first way to reduce this is to put an upper cap on maximum code word length of adaptive-Golomb. The limit which we put in our design is 16 instead of 32. This reduced the frequency requirement by half with minimal drop in percentage compression. We have already discussed earlier that we dont get output at every clock edge for U and V components. Us and Vs valid data only comes every fourth clock cycle. In the proposed design of the Serializer works at 8 times higher frequency. It employs a FIFO to store the incoming code word and effectively uses the idle U and V clock cycle to empty its queue. Furthermore, it can be easily seen that if we use a FIFO of size 40 bits then there will never be any bit loss via buffer overflow.

The Serializer consists of a 40-bit FIFO, 40-bit barrel shifter, 3-bit counter, and a 6-bit counter which calculates the current buffer occupancy. The FIFO works normally

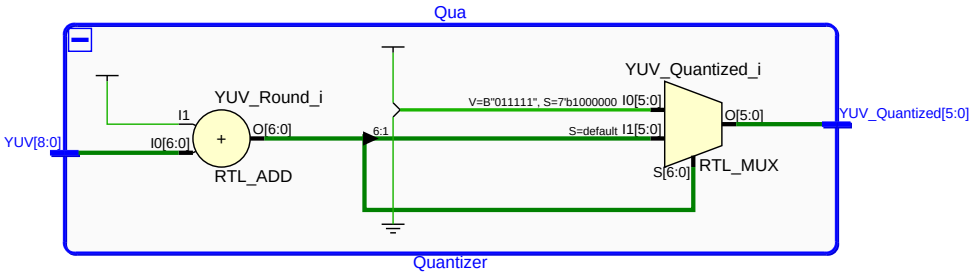

Figure 4.5: Schematic of Quantizer Unit Synthesised By Xilinx

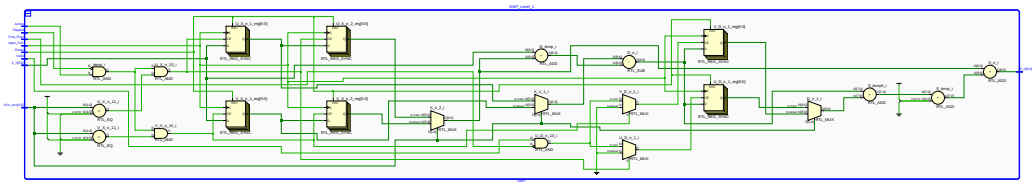

Figure 4.6: Schematic of First Level of DWT Synthesised By Xilinx

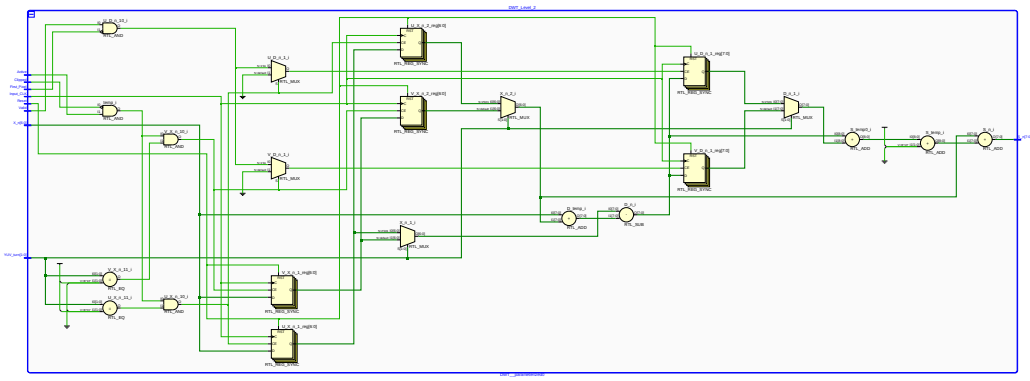

Figure 4.7: Schematic of second level of DWT Synthesised By Xilinx

as a shift register except when the 3-bit counter value is 0. When it is 0 the barrel shifter pushes Code Word in to the FIFO i.e. at every eighth clock cycle. The shift which Barrel Shifter must provide is decided by the value of registers which store buffers occupancy. Serial Clock is deactivated once the FIFO becomes empty. Reset value of all registers in the design are 0. The block diagram is shown in Fig 4.9.

.

4.3 Conclusion

The main power hogging modules in our design are DWT, Serializer and Adaptive-Golomb contributing to approximately 20 percent,20 percent and 30 percent respectively of the total design hardware. Several optimizations have been made to reduce the combinational logic as well as the register count in our design. The design is expected to be synthesized using UMC 130 nm HS library with Synopsys Design Compiler. The layout has to be generated using Cadence Encounter in near future.

Figure 4.8: Schematic of Serializer Unit Synthesised By Xilinx

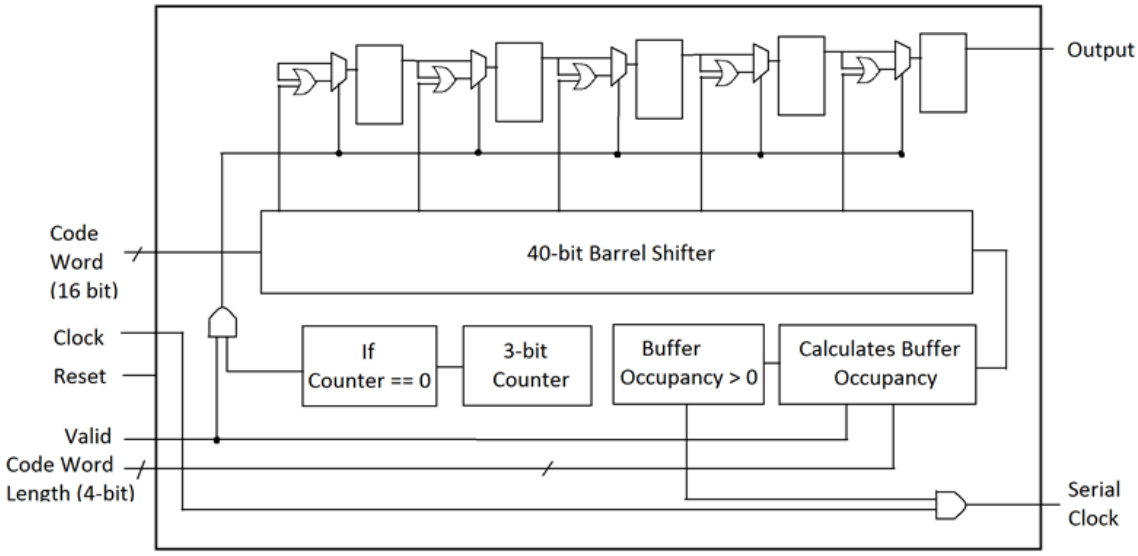

Figure 4.9: Proposed Serialiser Block Diagram

Chapter 5

Results

5.1 Performance Metric

The metric used to evaluate the performance of image compression algorithm is compression rate (CR) which is computed as :

$$CR = (1 - \frac{1}{compressionratio}) * 100 \tag{5.1}$$

Where compression ratio is defined as the ratio of original image size to compressed image size in bits.

Since we have used lossy compression techniques like quantization and discrete wavelet transform the reconstructed will never be exactly equal to the original image. The simplest and most widely used metric used to find the quality of the image with respect to original image is called PSNR (Peak Signal-to-Noise Ratio). In order to find PSNR we first compute the mean squared error given by

$$MSE = \frac{1}{3 * M * N} \sum_{z=1}^{3} \sum_{y=1}^{N} \sum_{x=1}^{M} (f(x,y,z) - F(x,y,z))^2 \tag{5.2}$$

Where M and N are image height and image width, f(x,y,z) is to original image and F(x,y,z) is the reconstructed image. The PSNR is computed as

$$PSNR = 10 * log10(\frac{255 * 255}{MSE}) \tag{5.3}$$

5.2 Sample Images

Sample Images are shown in figures 5.1, 5.2, 5.3

5.3 Comparison

Table 5.1: PSNR and Compression Comparison

Quantisation	PSNR	PSNR (min)	Percentage Compression
Q = 8 (Proposed)	37.24	36.39	91.89
Q = 4	41.56	40.78	89.22
Q= 2	45.82	45.26	85.00
Q = 1	54.14	52.60	80.09
Khan et al [1]	40.80	nil	81.94
Kinde et al [2]	40.66	nil	90.35

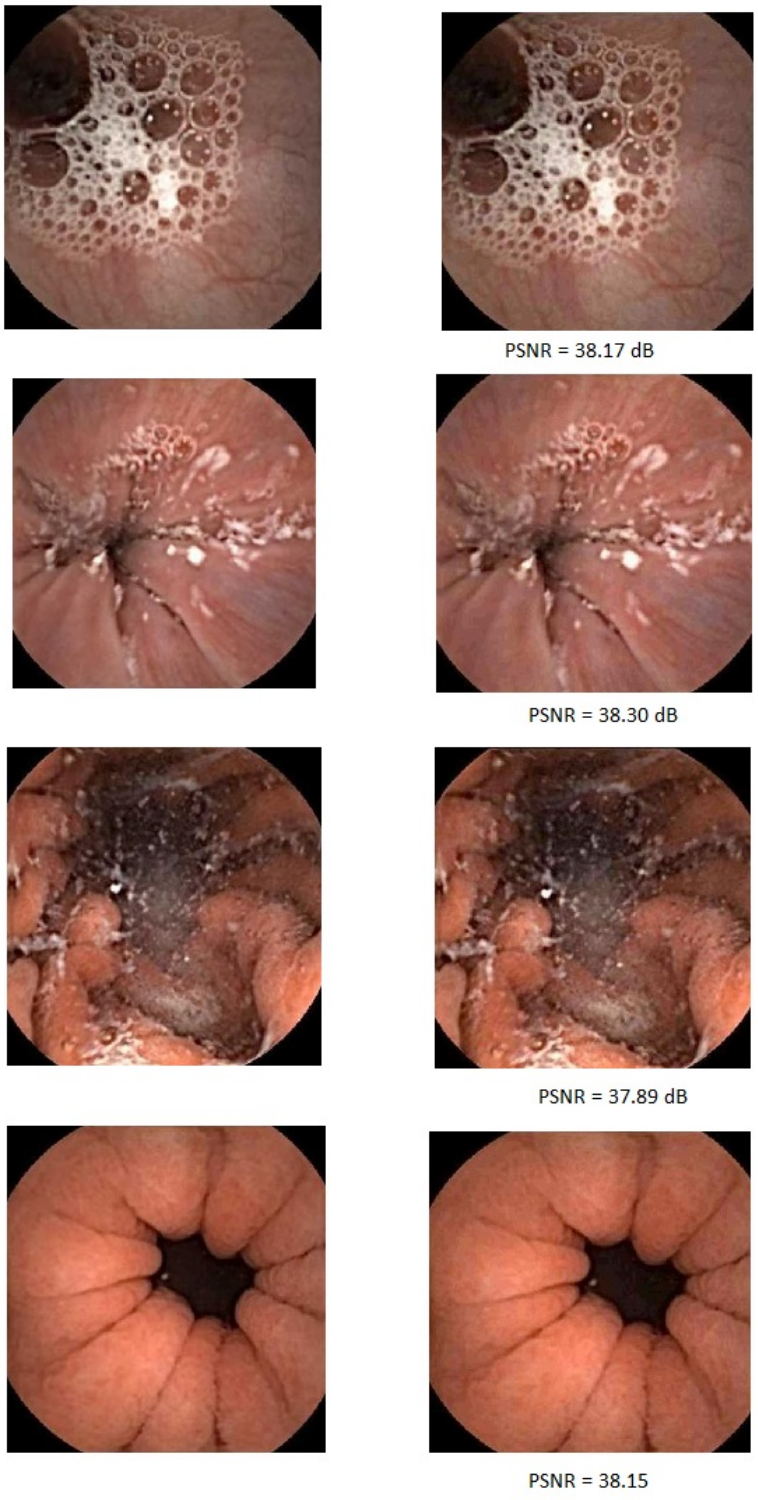

Figure 5.1: Original Image (Left) , Reconstructed Image (Right)

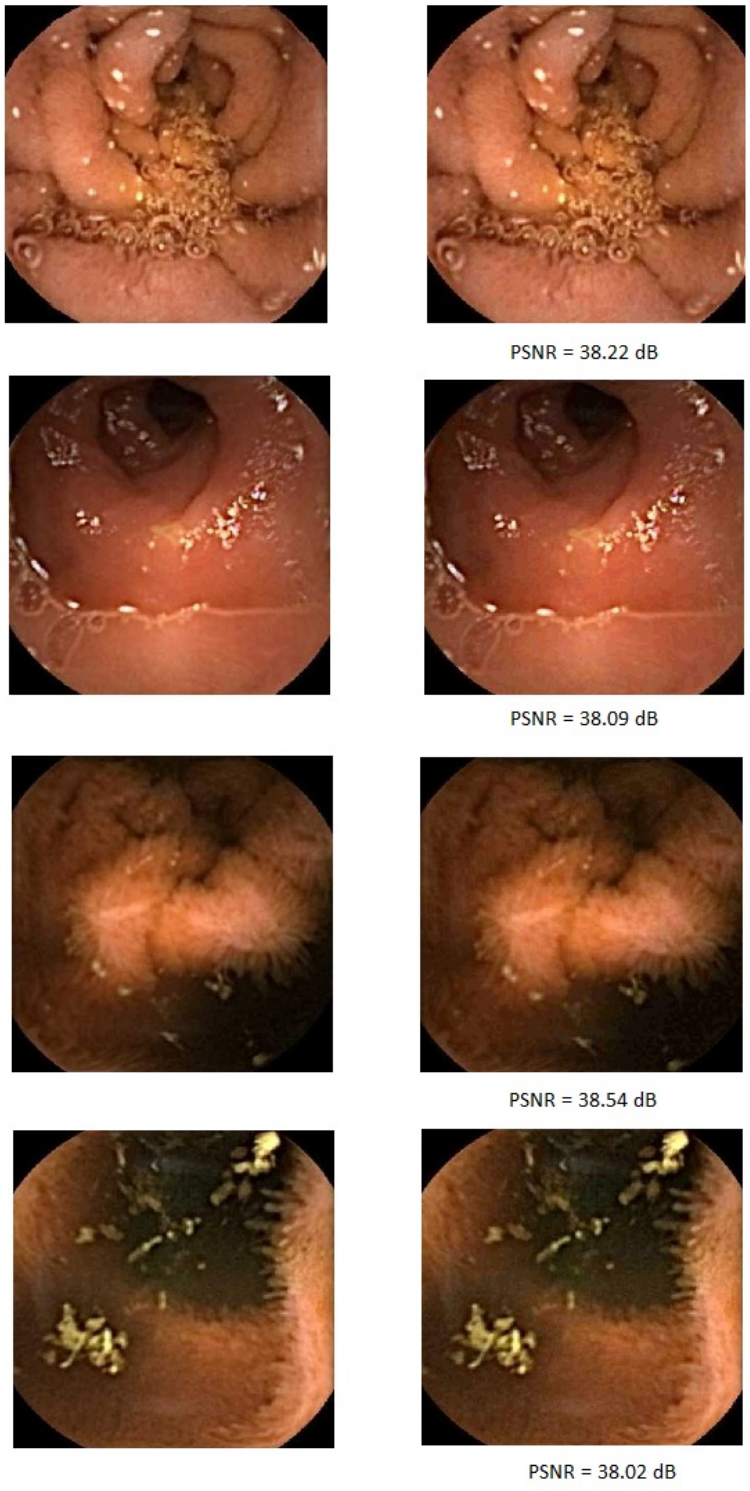

Figure 5.2: Original Image (Left) , Reconstructed Image (Right)

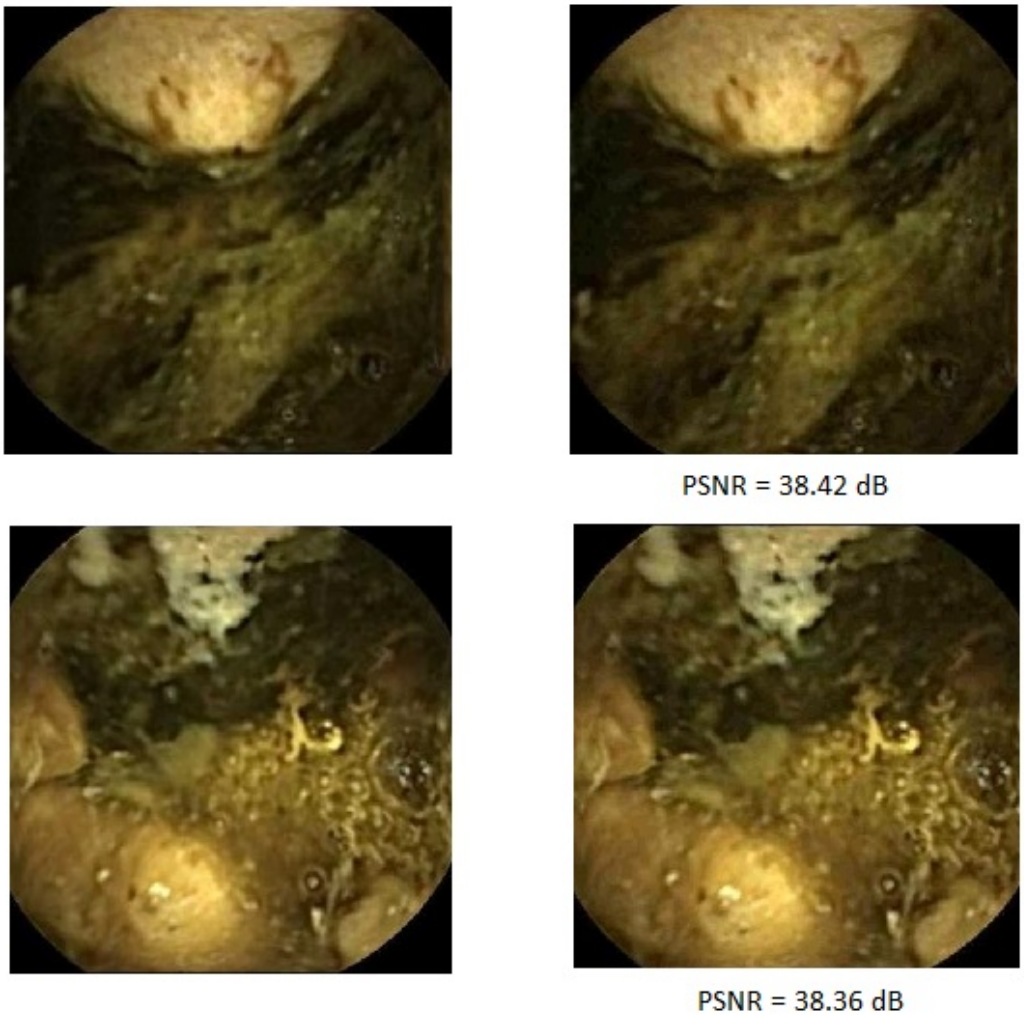

Figure 5.3: Original Image (Left) , Reconstructed Image (Right)

Chapter 6

Conclusion

6.1 Summary

The main goal of this project was to design a power efficient compressor module for the wireless capsule endoscopy. We experimented with several different techniques. We tried to increase the percentage compression keeping the PSNR above a threshold value of 35.

The images of gastro-intestinal tract show high correlation in blue and green. Furthermore, the energy content in luminance component is much more compared to chrominance component. To exploit these properties, the color plane used is YUV as proposed by the JPEG 2000 standard.

Quantization is used in the design which is a very effective way to achieve high compression. In the above design we were successfully able to quantize 3 bits without allowing the average PSNR to drop below 36.

Further sub-sampling of chrominance component could lead to further increase in compression ratio but it was affecting the PSNR. Therefore, we decided to first do a DWT of chrominance component and then again take the DWT transform of the lower frequency part. This 2-level DWT transform leads to same compression ratio

as subsampling by 4 but at same time enhances PSNR. For hardware efficiency a 5/3 integer wavelet with lifting scheme was used to compute the transform.
The data now is passed through DPCM which reduces the standard deviation and helps in better encoding of the data.

Finally, we use adaptive Golomb encoder which adjusts its parameters on the fly. It gives a better compression ratio compared to the static Golomb encoder. It tunes the parameters of the encoder based on the recent inputs. The max code limit of the of the encoder is set to 16 with Nmax=7. Limiting the max code limit to 16 instead of 32 is very helpful as it reduces the frequency of operation of serialiser to half. Limiting the code limit to 16 instead of 32 has a negligible effect on the compression ratio while reducing the power of serializer to nearly half. Furthermore to take advantage of the specific properties of the image compression algorithm, the design architecture of the serialiser was modified to that it can work at further half the frequency. Now the serialiser needs to work at 8 times the frequency of the image compressor in our work as compared to the pevious work by kinde where it worked at 32 times the frequency. This lead to nearly one-fourth the power consumption as compared to the state of the art architecture.

6.2 Future Work

- Currently we have assumed that demosaicing of the image is done on the camera itself. It may be possible that the camera gives mosaiced image as the output. For this case either we have to demosaic the image before the compressor or a separate algorithm has to be designed which could directly perform the compression of mosaiced image.

- Endoscopy is primarily used for polyp detection. Therefore, we can also do auto poly detection of images which can greatly help the doctors analyzing the

images. If this step has to be implemented, it has to be done at the receiver side where we can leverage high computational power.

Bibliography

[1] T. H. K. . K. A.Wahid, "Subsample based image compression for capsule endoscopy," *Real-Time Image Processing*, pp. 5–19, 2013.

[2] C. Kinde Ante Fante , Basabi, Bhaumik , Shouri, "Design and implementation of computationally efficient image compressor for wireless capsule endoscopy," *Circuits Systems and Signal Processing*, pp. 1–27, 2015.

[3] C. V. D. Bruaene, D. D. Looze, and P. Hindryckx, "Small bowel capsule endoscopy: Where are we after 15 years of use?," *World Journal of Gastrointestinal Endoscopy*, 2015.

[4] Wikipedia, "Capsule endoscopy."

[5] M. D. P. Turcza, "Lowpower fpga-based image processing core forwireless capsule endoscopy," *Sens. Actuators A Physics*, pp. 552–560, 2011.

[6] Wikipedia, "Jpeg2000."

[7] G. Images, "https://www.google.co.in/imghp."

[8] U. B. . D. A. Chilambuchelvan, "A detailed suvey on vlsi architectures for lifting based dwt for efficient hardware implementation," *International Journal of VLSI design and Communication Systems*, pp. 207–214, 2012.

[9] Z. J. . F. J. yun , Han Cheng-de, "The selection of reversible integer-to-integer wavelet transforms for dem multi-scale representation and progressive compression," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science*, pp. 1010–1024, 2008.

[10] J. Lang, "Source coding in digital communications ensc 861 http://www.sfu.ca/ jiel/courses/861/slides.htm."