**Compe 565,** **Semester 2021**

**HW 4: Multimedia Communication System**

**Prepared by**

Ethan Nagelvoort (821234668)

E-mail: enagelvoort@sdsu.edu

Electrical and Computer Engineering

San Diego State University

**Table of Contents**

* List of figures …………………………………………………………………………………….2
* Introduction ……………………………………………………………………………………....3
* Procedural Section ………………………………………………………………………………. 4

1. Decoder …………………………………………………………………………..4
2. Encoder …………………………………………………………………………..4
3. Motion Estimation ………………………………………………………………4

* Results…………………………………………………………………………………………….. 5
* Conclusion………………………………………………………………………………………..14
* References ……………………………………………………………………………………….14

**List of figures**

Figure 1: Procedure of code in block diagram format

Figure 2: Original and Reconstructed frame 6, I frame

Figure 3: Motion Vectors form frame 6 and 7

Figure 4: Motion Vector for Frames 7 and 8

Figure 5: Motion Vector for Frames 8 and 9

Figure 6: Motion Vector for Frames 9 and 10

Figure 7: Original, Difference, and Reconstructed P Frame 7

Figure 8: Original, Difference, and Reconstructed P frame 8

Figure 9: Original, Difference, and Reconstructed P frame 9

Figure 10: Original, Difference, and Reconstructed P frame 10

**Introduction**

MPEG Video encoding and decoding allows a programmer to predict a future frame in the video and reconstruct frames as well. This process involves a lot of moving parts since there are a lot to both the encoding and decoding process. In encoding, the programmer has to subsample the frame, perform DCT coefficients, and then quantize it as well. Then in decoding, the programmer has to inverse quantize, inverse DCT, and upsample to reconstruct the image back into its regular rgb size. After decoding, the programmer performs a motion estimation and prediction to help determine an image motion vector.

In this home work, I utilized MatLab to manipulate a walking video’s sixth through tenth frame to go through the encoding and decoding process. I did this by separating the different processes into functions. For example, I have a linear interpolation function for upsampling and an inverse DCT function to help with the decoding process. There are three big parts to this code; encoding, decoding, and the motion estimation. These three parts follow the diagram below, excluding entropy, mux, and buffer.

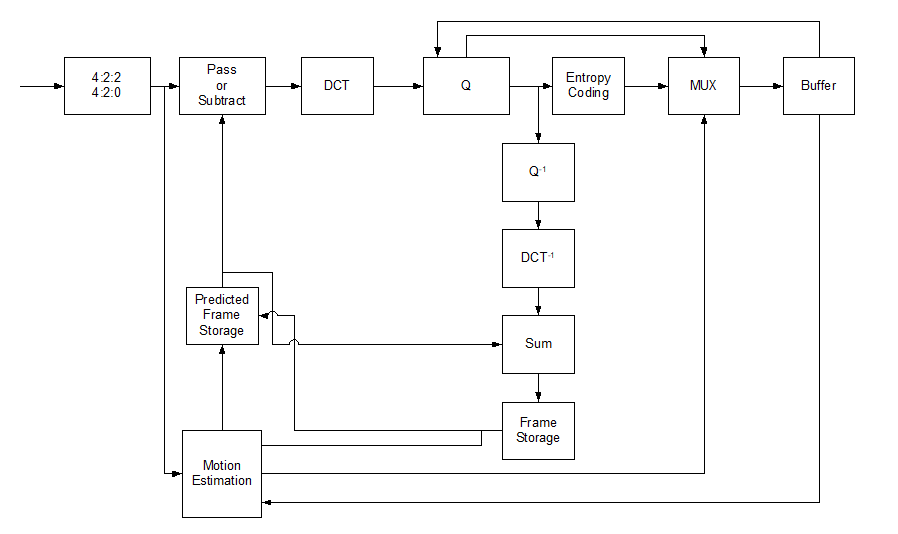


Figure 1: Procedure of code in block diagram format

**Procedural Section**

1. Encoder

In the encoder part of the code, I first converted the video frames to ycbcr and then subsampled frames 6-10 of the walking video. I then send every frame into a function that performs DCT and quantization. In this part of the code I extract the top corner 8x8 block and quantized both the luminescence and chrominance parts by using the luminescence and chrominance matrices. Then the DC DCT coefficients can be obtained and the AC DCT coefficients can be obtained through the zig zag operation.

1. Decoder

In order to get the inverse, I multiplied the luminescent quantization matrix and chrominance quantization matrix to the blocks to reverse the effects of when I divided them during encoding. I then have the frame go through linear interpolation to upsample the cb and cr components of the frame. Then I combined this cb, cr, and y to get the reconstructed frames. The Y component is gotten from the motion estimation part of the code that will be explained next. I then converted these reconstructed frames back into rgb and compared it to the original rgb frames.

1. Motion Estimation

I use a function to perform motion estimation and predict frames. I do this by traversing through every macroblock. I have a series of if statements that check the sides of the frame to ensure that the current macroblock is within the frame. These if statements help determine if the current macroblock is within the search window. I then declare a reference block to be the top left corner of that search window and set a variable min to a very big number. I also record the position of the current block I am at . I then have a set of for loops that depend on the search window where I loop through every block there. Within these for loops, I determine the minimum value of SAD and once I find the block that has that value, I record the current motion vectors. This is because the macroblock with the minimum SAD is the best match. These motion vectors will then be stored. To get the difference P frame, I took the current frame being looked at and subtracting it by the best matched frame.

**Results**

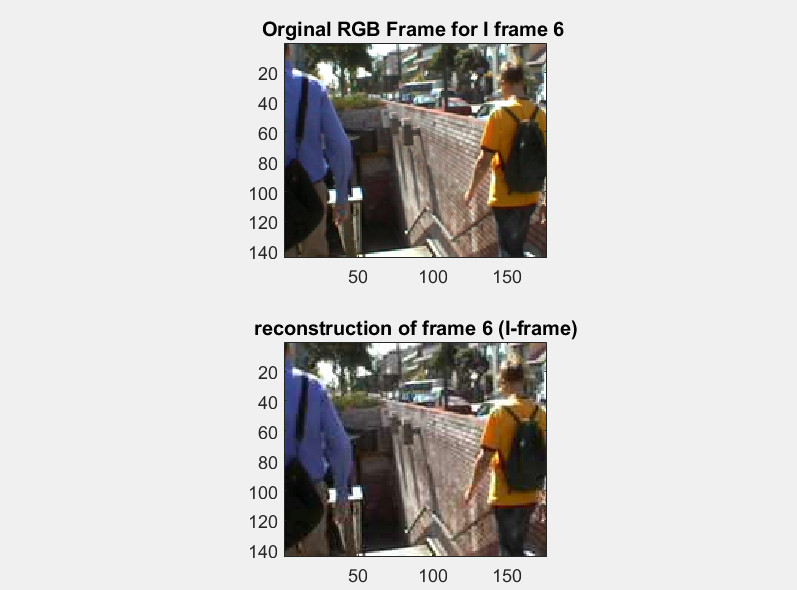
****

Figure 2: Original and Reconstructed frame 6, I frame

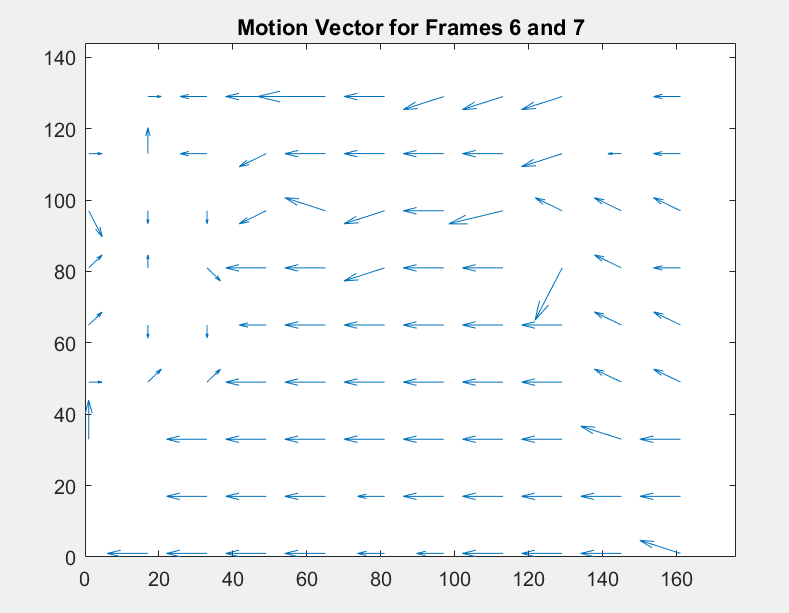


Figure 3: Motion Vectors form frame 6 and 7

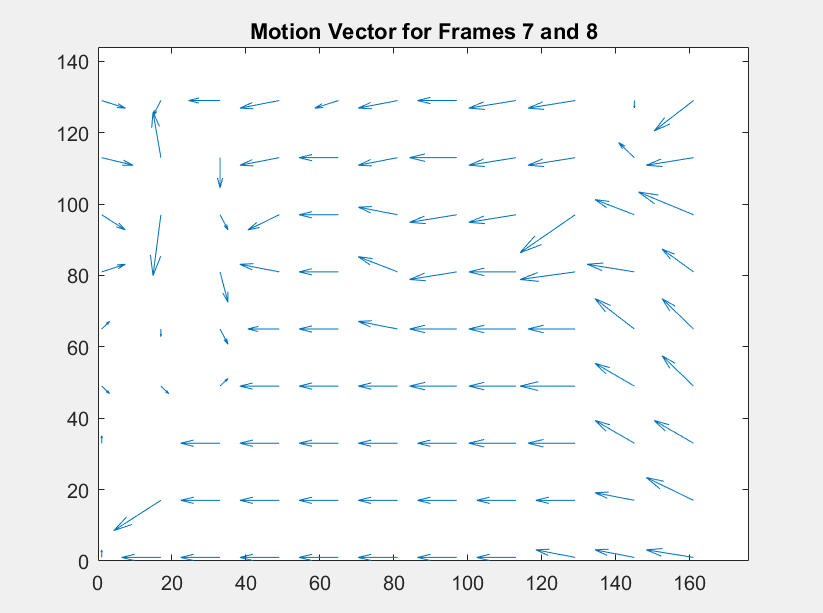


Figure 4: Motion Vector for Frames 7 and 8

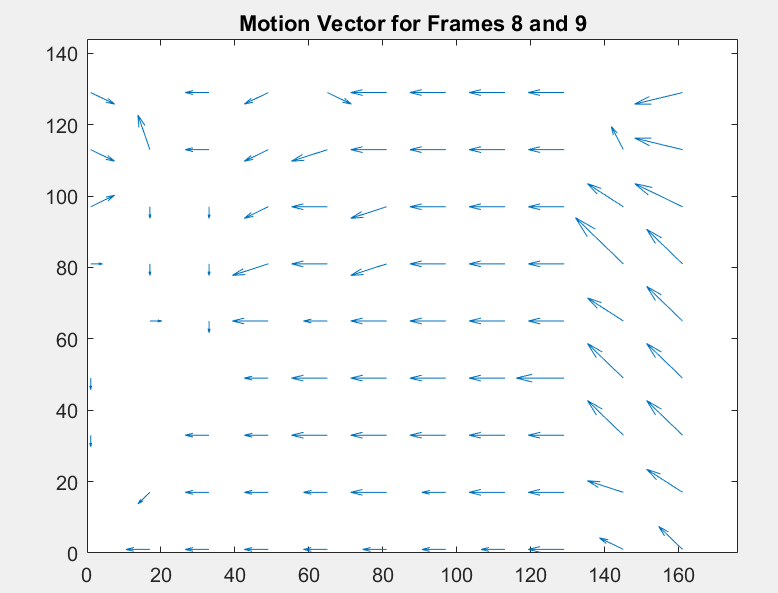


Figure 5: Motion Vector for Frames 8 and 9

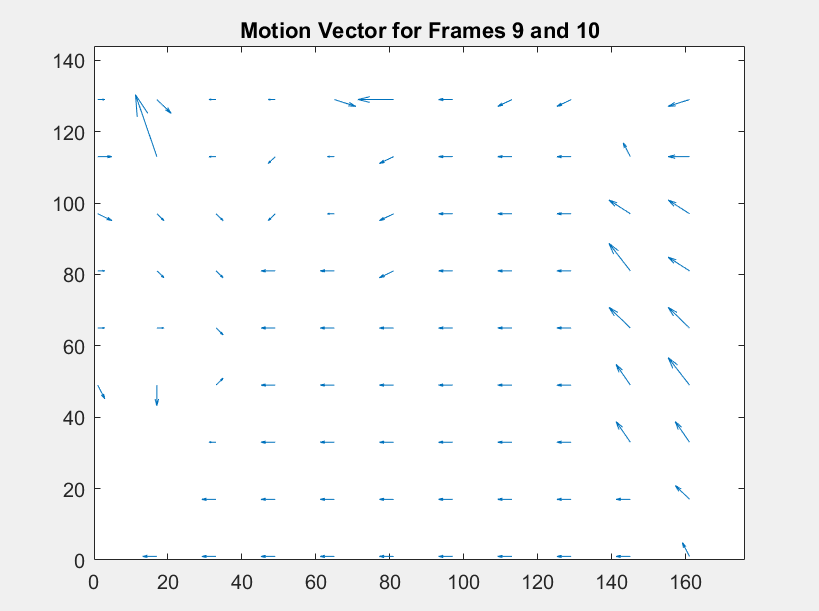


Figure 6: Motion Vector for Frames 9 and 10

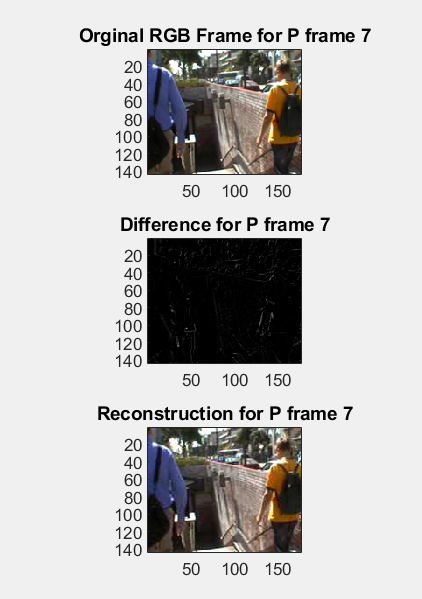


Figure 7: Original, Difference, and Reconstructed P Frame 7

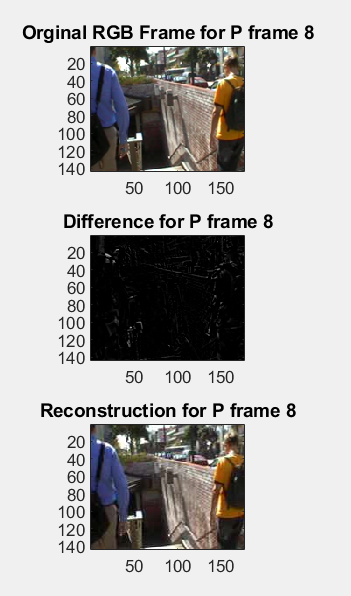


Figure 8: Original, Difference, and Reconstructed P frame 8

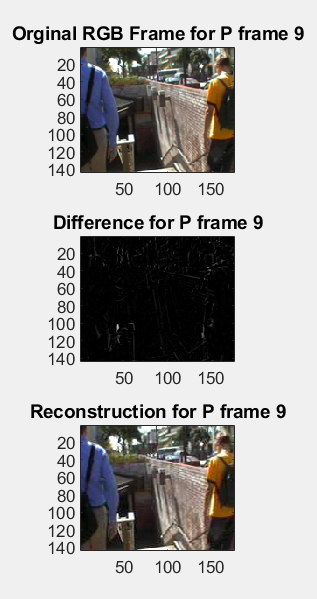


Figure 9: Original, Difference, and Reconstructed P frame 9

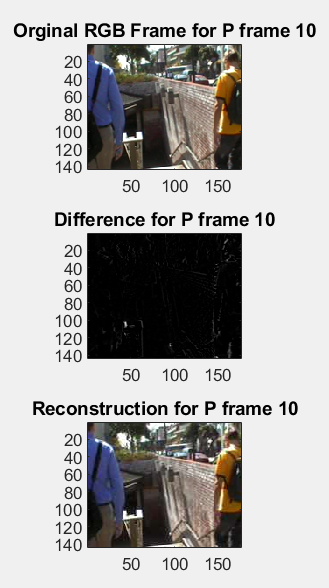


Figure 10: Original, Difference, and Reconstructed P frame 10

**Conclusion**

In this assignment, I learned how to implement everything I learned into one process. I implemented an encoder in this assignment to subsample, obtain DCT, and quantize frames 6 through 10 of the given video. I then implemented a decoder in this assignment to inverse quantize, inverse DCT, and upsample through linear interpolation for frames 6 through 10. I then implemented code to predict motion estimation frame to frame. Implementing these processes and connecting them together helped me gain an overview of the whole process of video coding.

**Reference**

[1] S. Kumar, “Video Compression Part1”, Compe 565, 2021

[2]S. Kumar, “Video Compression Part2”, Compe 565, 2021

[3] S. Kumar, “Lecture Topic #4: Image Compression”, Compe 565, 2021

[4]S. Kumar, “Lecture Topic #5: Video Compression”, Compe 565, 2021

[5] S. Kumar, “Lecture Topic #1: Multimedia Fundamentals Slides”, Compe 565, 2021

[6]S. Kumar, “Lecture Topic #2: Compression Fundamentals”, Compe 565, 2021