MPEG VIDEO CODEC

Electrical and Computer Engineering

San Diego State University

COMPE565 PROJECT # 4

Table of Contents:

TABLE OF FIGURES

1.0 Introduction

In the multimedia industry, the emerging compression standards are always a very intense area of research. The previous project focused on the implementation of two Motion Estimation Algorithms, namely; Exhaustive Search Algorithm and Conjugate Search Algorithm. This project deals with developing an MPEG encoder and a decoder. The MPEG compression methodology is considered as asymmetric—since the standardization body only specifies the requirements for the encoder. Therefore, the encoder needs to be algorithmic or adaptive whereas the decoder carries out fixed actions. This is considered advantageous in applications such as broadcasting where, the numbers of expensive complex encoders are small but the numbers of simple inexpensive decoders are large. The main criterion in video coding is given by compression standards. The more efficient the compression; the smaller the data rate and hence less is the cost. The basic principle behind encoding is the prediction of future frames with the help of the current frame and so on. Only the error image and the motion vectors are sent to the decoder where it is decoded with the help of the original reference frame. In order to ensure that the prediction at the encoder is accurate a decoder is implanted within the encoder. The motion compensation algorithm used to predict the future frames is the conjugate search

Algorithm:

The report is organized as follows:

2.0 Procedure for designing the Encoder

2.0 PROCEDURE FOR DESIGNING THE ENCODER

2.1 AIM

The aim of this project is to design an MPEG encoder and a decoder. Details of the procedure for designing the encoder are outlined below.

2.2 STEPS UNDERTAKEN

2.2.1 Read the sequence.

The test video sequence for this project is the "Coastguard_qcif". The file is read into Matlab as done in the previous assignments. First five frames of this video are considered in this project. The GOP is given to be 10.Hence the first frame will be an I frame and the next four frames will be P frames. The size of the given video sequence is 176 x 144.

2.2.2. Block Diagram and the procedure followed

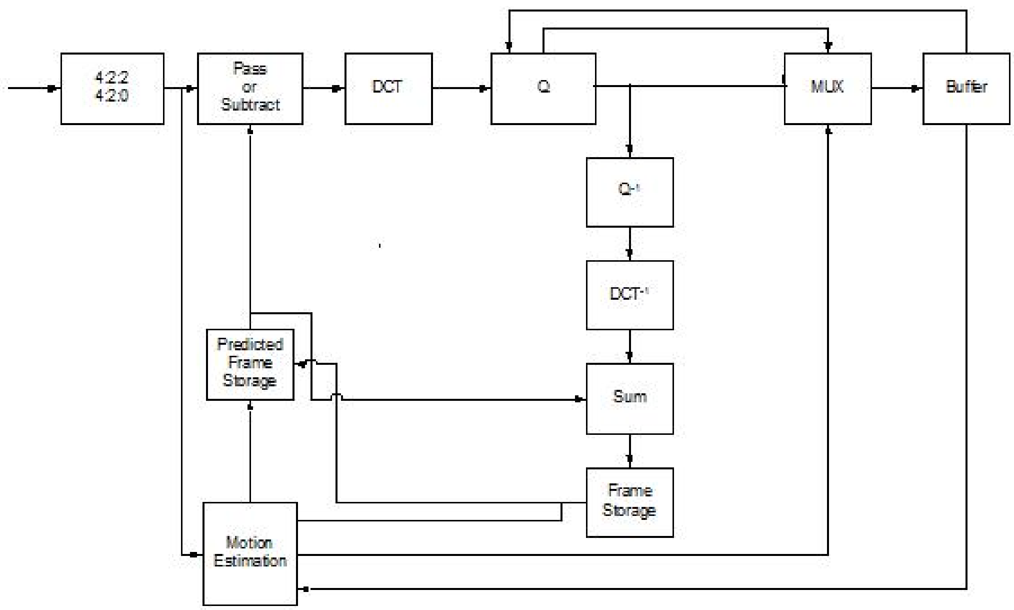

Figure 1: Block diagram of CODEC

The block diagram for a video codec is shown in Figure 1. Following steps outline the procedure:

- First the Luma and Chroma components are subsampled.
- Depending on whether the frame is an I frame or a P frames it is either passed on to the next stage without prediction or with prediction.
- Prediction for the P frames is done by implementing motion estimation algorithms
- DCT quantization and zigzag scan is performed on these frames.
- If it is a predicted frame, it passes through the decoder embedded in the encoder

where in it undergoes dezigzagging, inverse quantization, and inverse DCT.

- This decoded frame is now used to predict further future frames.
- If it is an I frame, t is passed on directly to the decoder where it is decoded by performing dezigzag scan, inverse quantization, and inverse DCT.
- This decoded I frame is further used to decode the following frames by adding motion vectors to the error frame.

2.2.3. Encoder Design

At the encoder the first frame is encoded as an I frame. The following steps are performed:

1. The luminance and chrominance components are separated and stored in different matrices.
2. DCT and quantization is performed on the Y Cr and Cr Matrices. Functions are called to implement DCT and quantization.
3. The next step would be to perform zigzag scan on the quantized I frame. Again a function is implemented to perform the same and the DC and AC coefficients are separated. Since the spatial resolution of the given video sequence is 176X144, the Luminance component is being encoded we will have **396 DC coefficients** and **(396x63) AC coefficients**.
4. Steps 2 and 3 are repeated for every block until all the blocks in the luminance component are encoded.
5. Steps 2, 3 and 4 are repeated for the Chrominance components.
6. The first frame is now ready to be sent to the decoder.
7. Once the first frame is encoded, we proceed to encode the next four frames which are predicted from the already encoded I frame. For simplicity sake, this project assumes the remaining frames in a GOP to be P frames. Conjugate search

algorithm is used to predict the motion of the Macroblocks as this is faster than the exhaustive search algorithm. The reference to this discussion can be found in the previous project.

8. More specifics of the project are as follows:
9. Macro block from the current frame is accessed.
10. An appropriate search window is obtained. Functions are implemented to perform steps 7 and 8.
11. It should be noted that the reconstructed I frame should be available at the encoder i.e. There is a decoder built into the encoder. This reconstructed I frame is treated as the reference frame and the best match for the macroblock in consideration is found.
12. Once the best match is found, the error residue is calculated and stored in another matrix and the motion vectors are determined.
13. Steps 2 and 3 are repeated for the error block.
14. Steps 7 through 11 are repeated for all the macro blocks in the frame.
15. The encoded frame is now ready to be sent to the decoder.
16. Steps 7 through 12 are repeated for the subsequent frames.

2.2.4. Decoder Design

The decoder performs exactly the inverse operation as that of the encoder. The steps followed are as outlined below;

1. The luminance and chrominance components are separated and stored in different matrices.
2. A function is called which performs dezigzag scan of the encoded frame.
3. Dezigzag is followed by inverse quantization and inverse DCT operations .
4. Steps 2 and 3 are repeated for all the blocks.

5. Steps 2 through 5 are repeated for the Chrominance components.
6. The I frame is now restored and it can be used at the encoder to obtain subsequent P frames.
7. Mean square error and PSNR is calculated for the restored I frame.

   Once, the I frame is decoded, the decoder moves on to decode the other P frames.
8. Steps 2 through 4 are repeated for the subsequent frames.
9. The decoded error frame is now added to the decoded luminance component of the I frame.
10. Mean square error and PSNR is calculated for the restored P frame.
11. Steps 8 through 10 are repeated for the subsequent frames.
12. The final PSNR is obtained by taking the average of the five PSNR’s that were obtained for each frame.

3.0 RESULTS

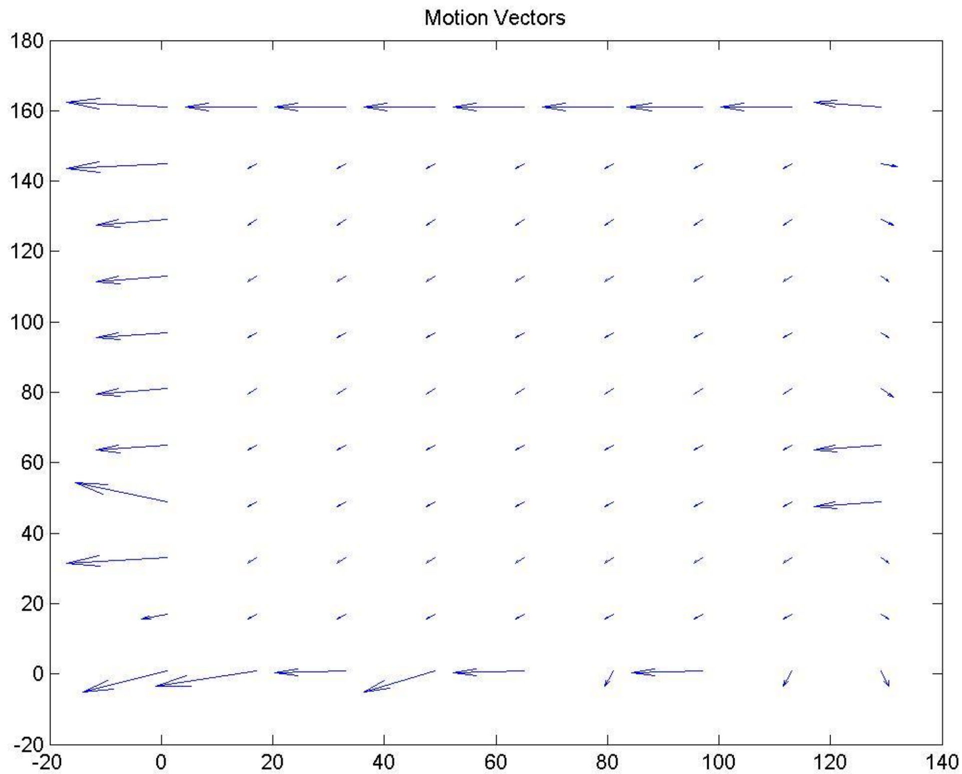

Figure 2: Motion vector for the second frame

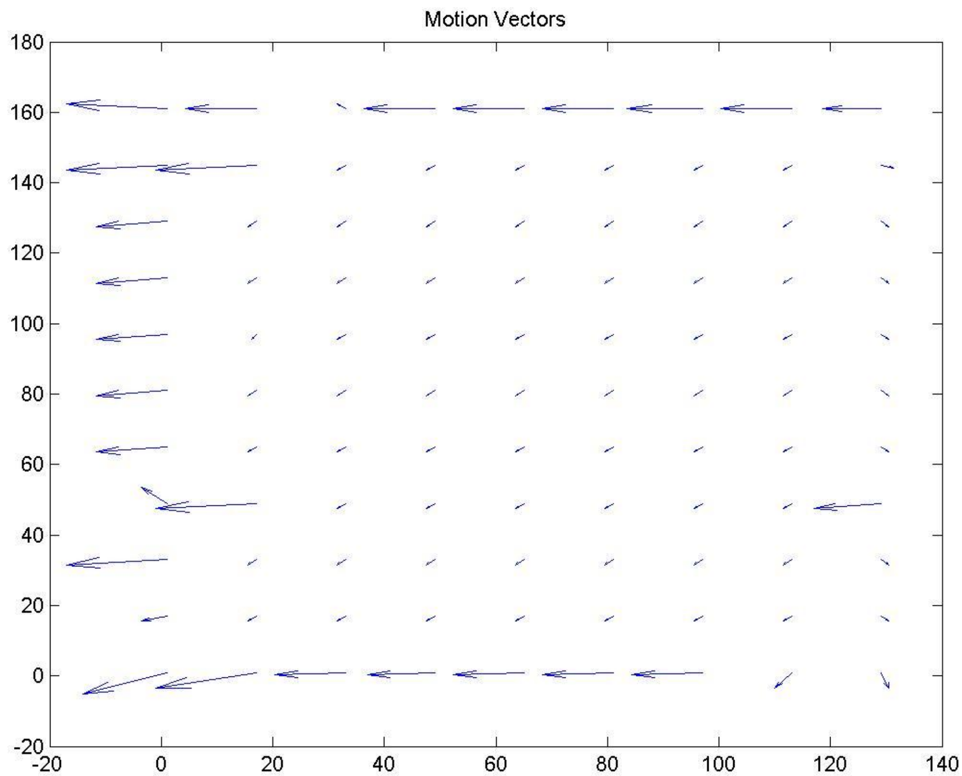

Figure 3: Motion vector of the third frame

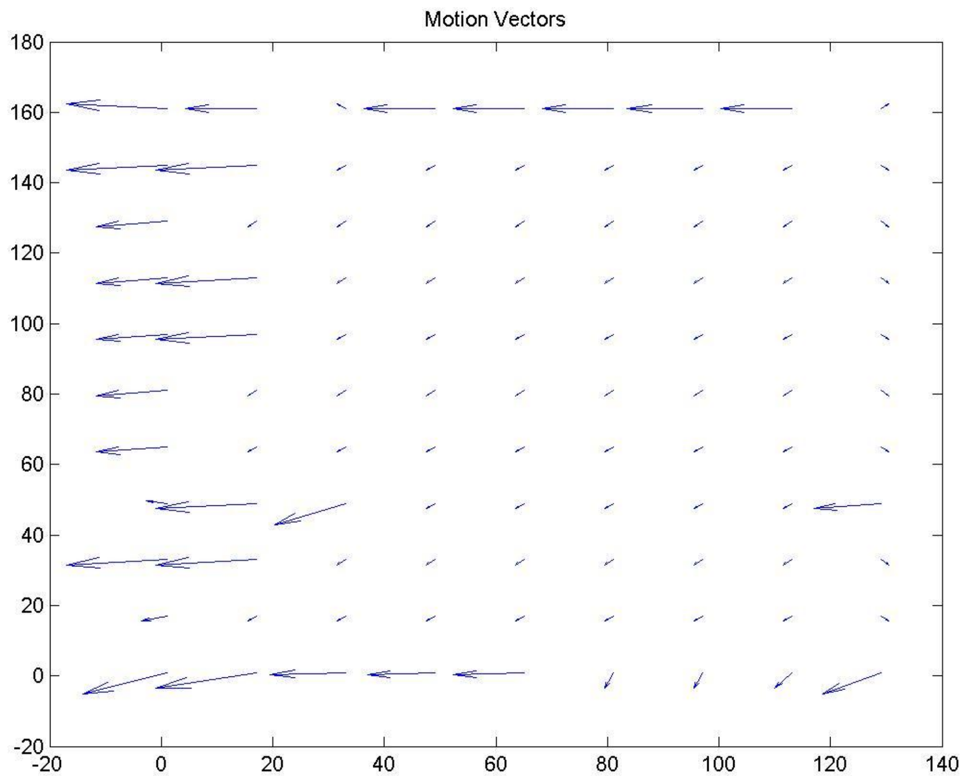

Figure 4: Motion vector for the fourth frame

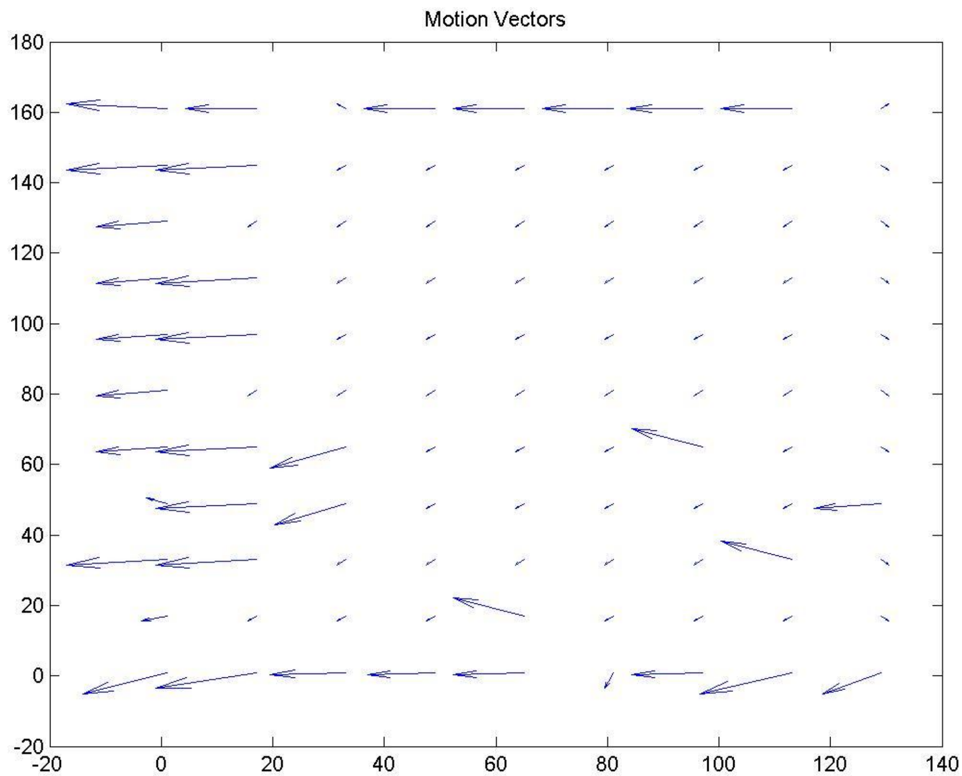

Figure 5: Motion vector for the fifth frame

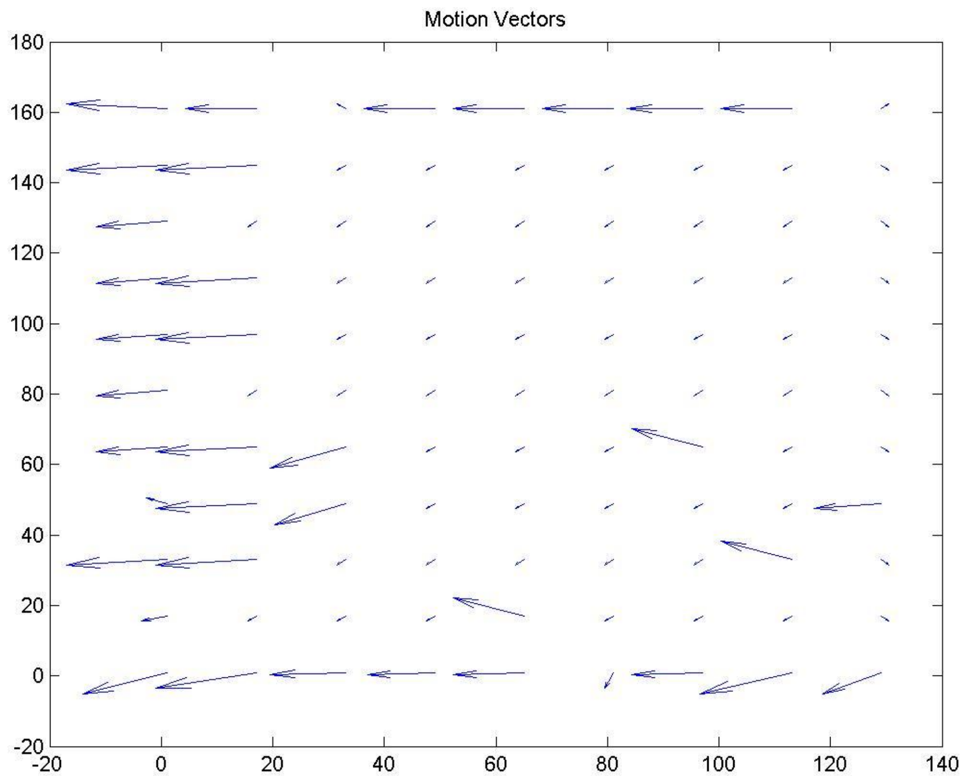

Figure 5: Motion vector for the sixth frame

Figure 6: Original Image

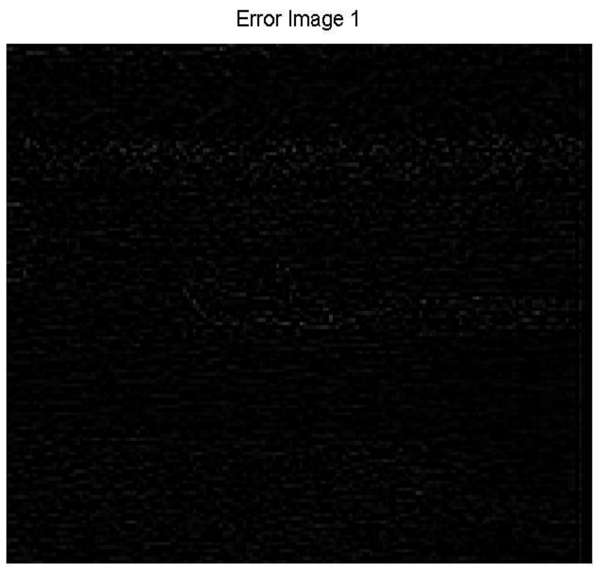

Figure 7: Error Image (1)

Figure 8: Reconstructed Image (1)

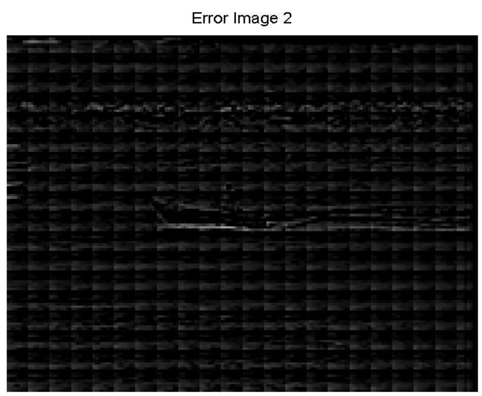

Figure 9 Error Image(2)

Figure 10: Reconstructed Image (2)

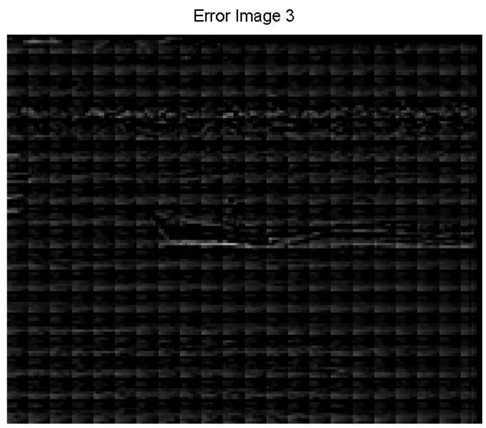

Figure 11 Error Image(3)

Figure 12: Reconstructed Image (3)

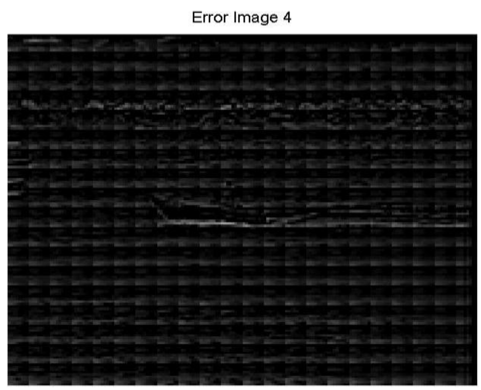

Figure 13 Error Image(4)

Figure 14: Reconstructed Image (4)

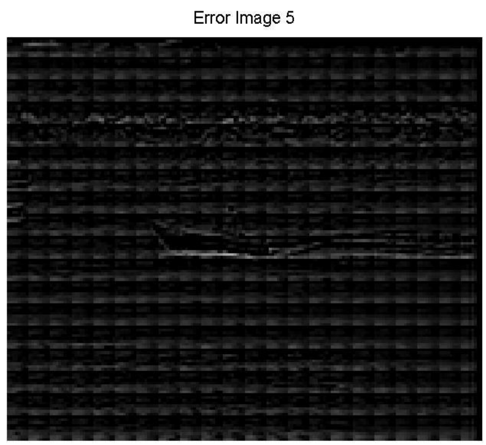

Figure 15 Error Image(5)

Figure 16: Reconstructed Image (5)

4.0 DISCUSSION

5.0 Conclusion

6.0 References

1] Ze-Nian Li and Mark.S.Drew, "Fundamentals of Multimedia," Princeton Hall, 2004.

2] Website: www.**wikipedia**.org

3] Multimedia Communication Systems