ECE1783 Assignment 1 Report

https://github.com/Cheekibreaki/Multimedia

Jiping Li - 1005983269
Zoey Ze - 1005824468
Jingqi Zhu - 1005798004

Exercise 1: Chroma up-sampling, YUV pixel manipulation, YUV-RGB CSC

This exercise provided hands-on experience with fundamental video processing concepts, including color space conversions, chroma subsampling, and the effects of noise on image quality.

We need to read and upscale YUV 4:2:0 video data to YUV 4:4:4 format. Then apply color space conversion formulas to transform the upsampled YUV data into RGB format, which is saved as PNG files for visual comparison.

This process involved implementing a straightforward chroma upsampling method where each chroma sample was duplicated to correspond with four co-located luma samples. Finally, we looked at the impact of adding random noise to various color components, gaining insights into how different types of distortion affect perceived image quality.

Exercise 2: Basic block-based operations on video content and quality assessment metrics

Exercise 2 focused on applying basic block-based operations to video content and assessing quality metrics.

We extracted the Y-component (luminance) from 4:2:0 video sequences and saved them as Y-only files. These files were then processed by splitting each frame into blocks of various sizes (2x2, 8x8, and 64x64), applying padding where necessary to ensure divisibility.

We then calculated the average value for each block, replacing the original block with this average to create block-averaged versions of the Y-only files. We also plotted graphs to visualize the relationship between block size and objective quality measures across the tested sequences.

Exercise 3: Basic Motion Estimation/Compensation

3.1: Architecture Overview

The architecture provides simple video compression by dividing frames into blocks. Instead of storing entire frames, the architecture only stores the differences between predicted and actual blocks, known as residual. The predicted movements are stored motion vectors by finding similar blocks in previous frames. The architecture further reduces data by simplifying residuals through rounding.

Using these predictions and simplified differences, the decoder can reconstruct the video using reversed operations.

3.2: Visualization of Predicted Frame and Residual Value

3.2.1: varying i with fixed r=4 and n=3

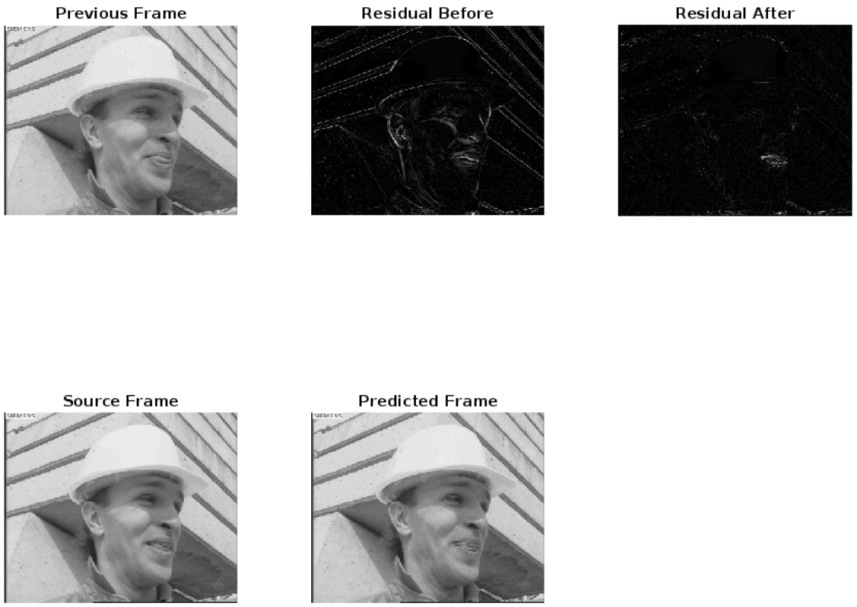


Figure 1: Frame 10 comparison base on i=8, r=4, n=3

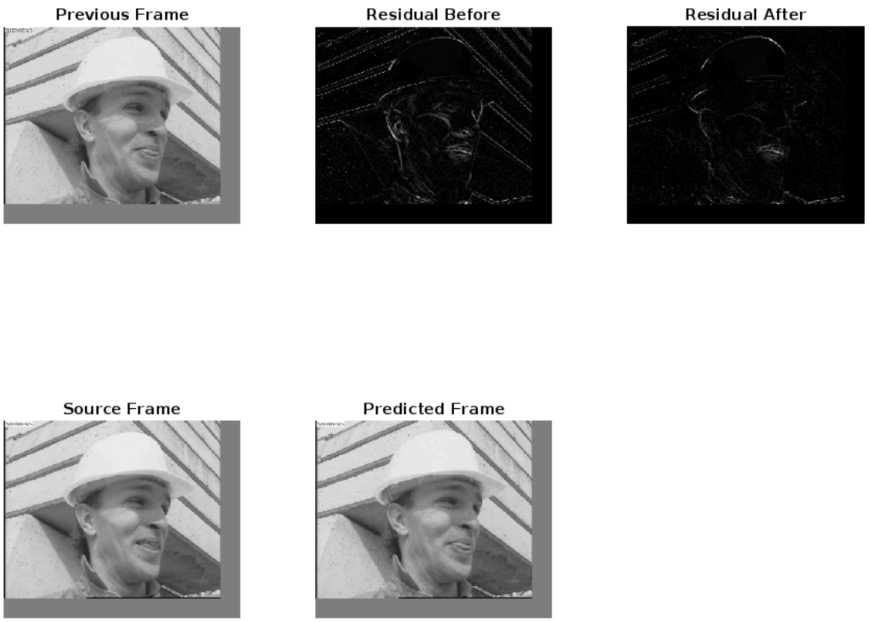


Figure 2: Frame 10 comparison base on i=64, r=4, n=3

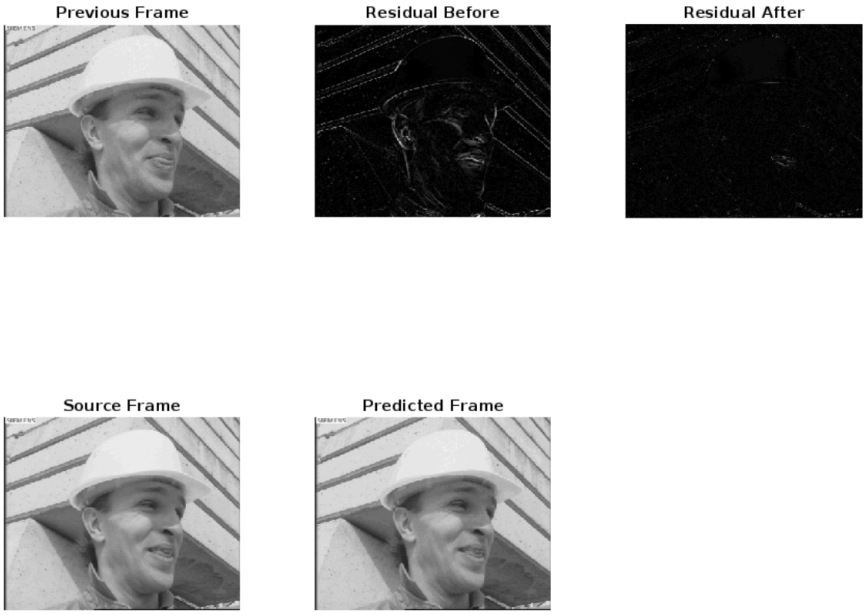


Figure 3: Frame 10 comparison base on i=2, r=4, n=3

3.2.2: varying varying r with fixed i=8 and n=3

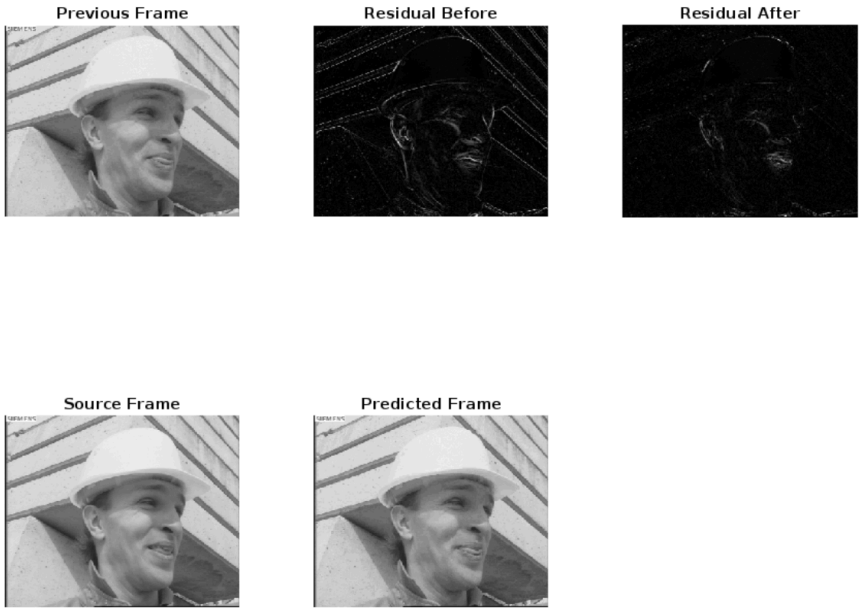


Figure 4: Frame 10 comparison base on r=1, i=8, n=3

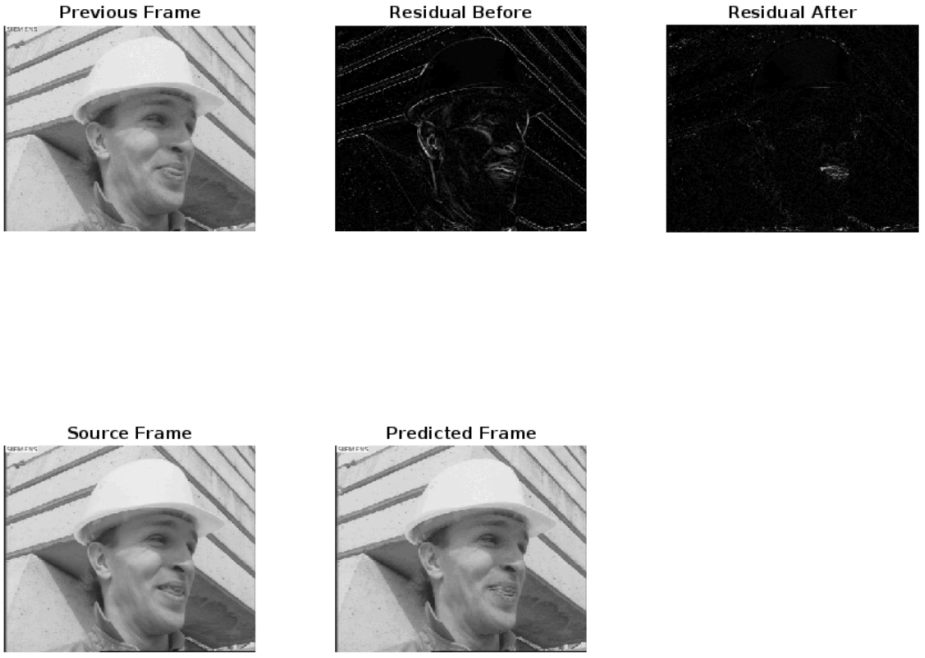


Figure 5: Frame 10 comparison base on r=4, i=8, n=3

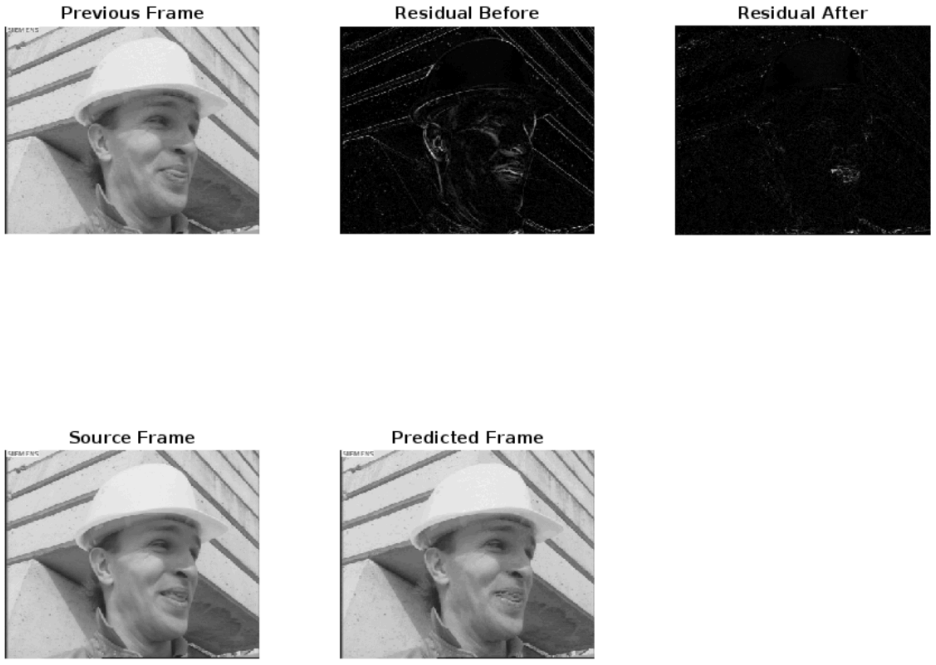


Figure 6: Frame 10 comparison base on r=8, i=8, n=3

3.2.3: varying n with fixed i=8 and r=4

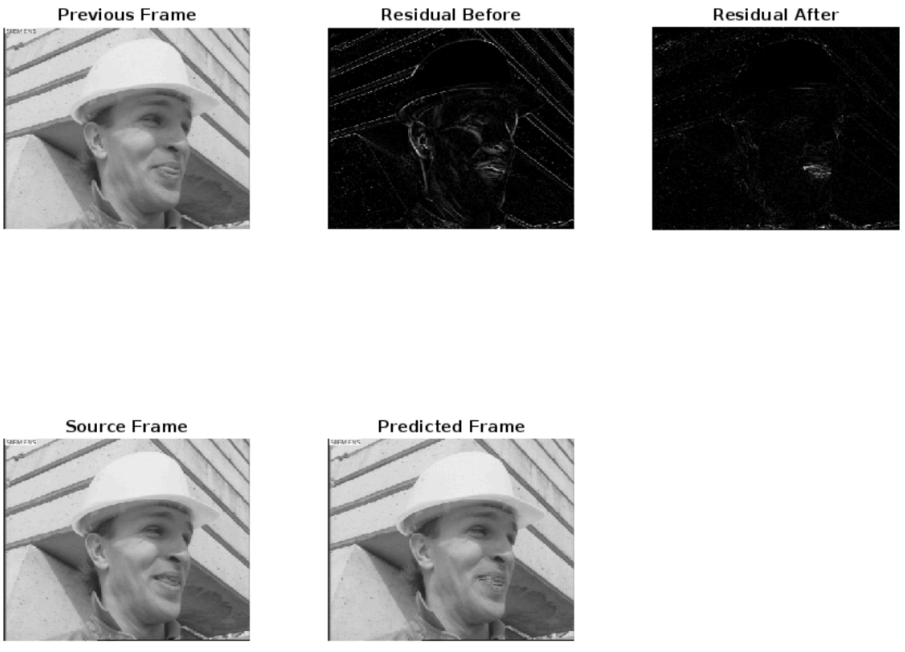


Figure 7: Frame 10 comparison base on n=1, i=8, r=4

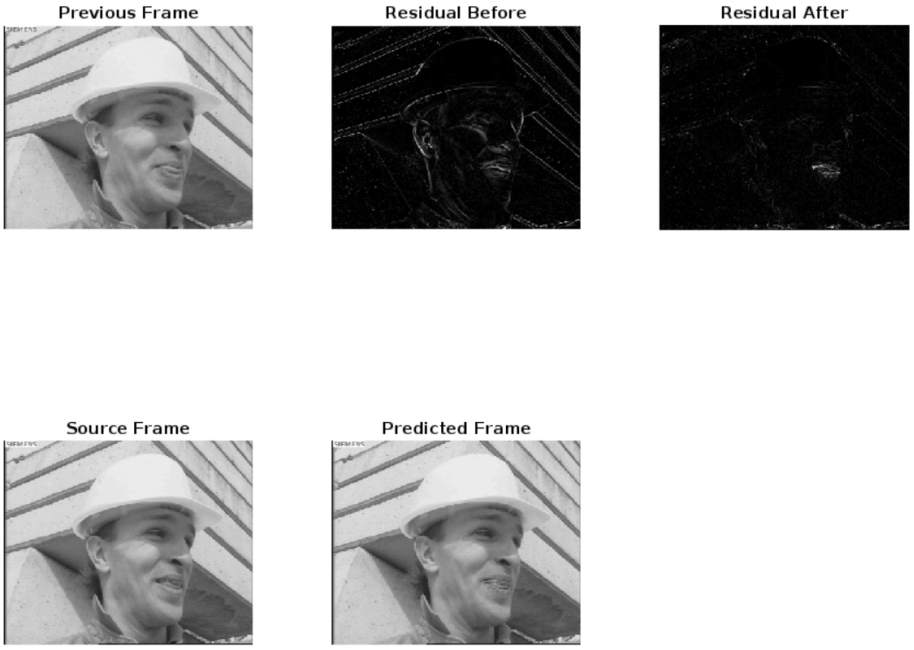


Figure 8: Frame 10 comparison base on n=2, i=8, r=4

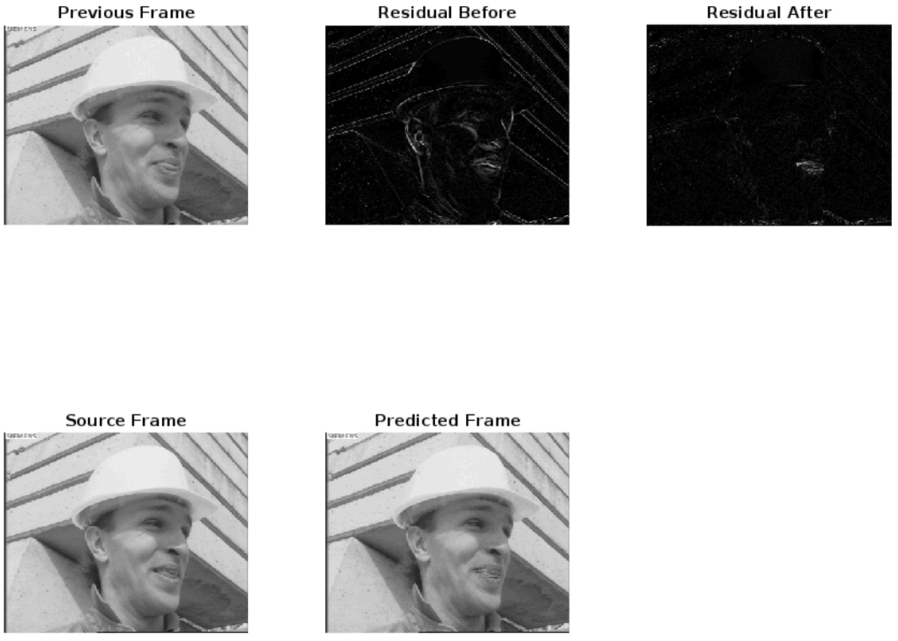


Figure 9: Frame 10 comparison base on n=3, i=8, r=4

3.3: Per-frame PSNR and MAE graphs of Foreman CIF

3.3.1: Varying i with fixed r=4 and n=3

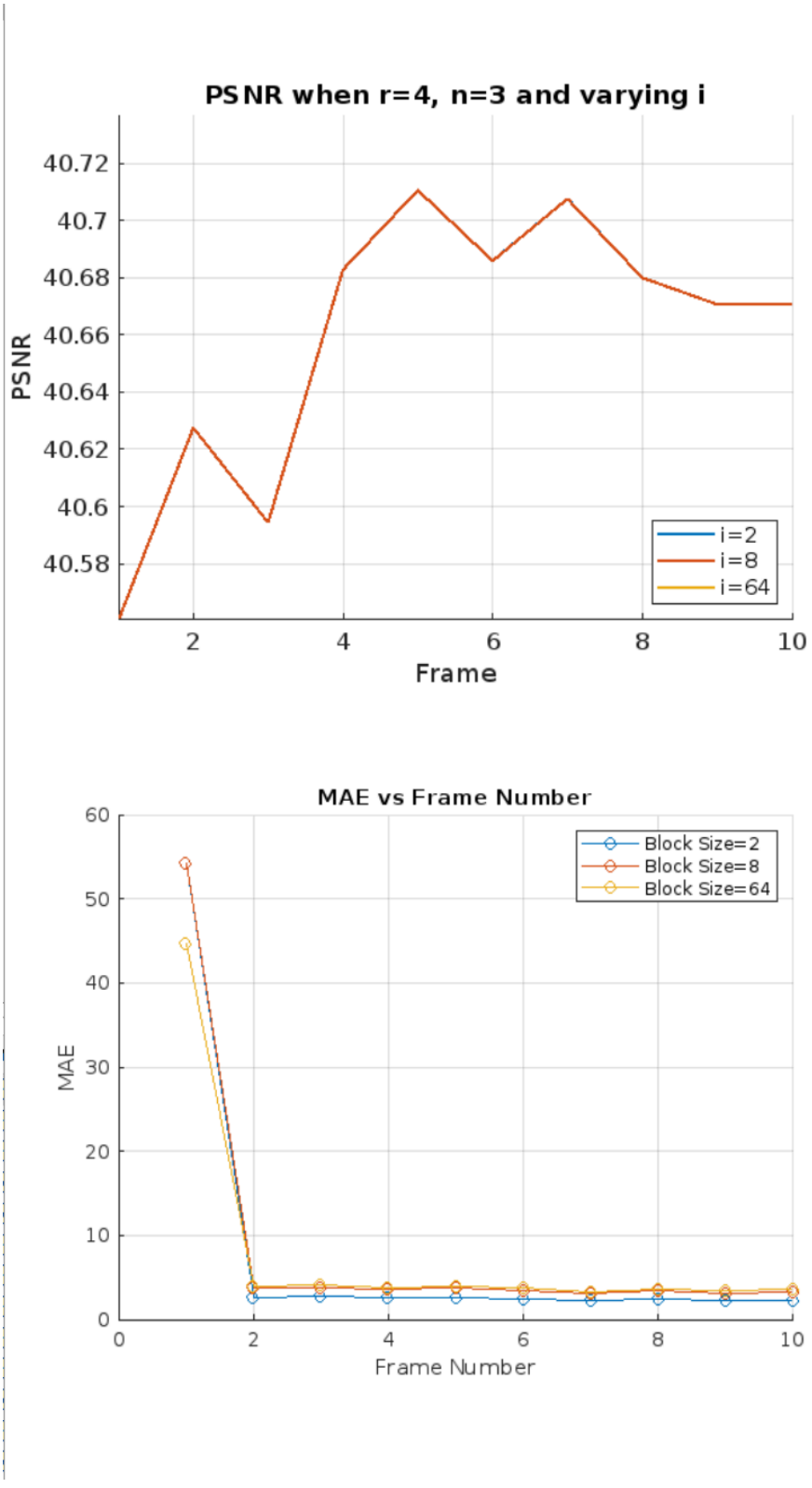
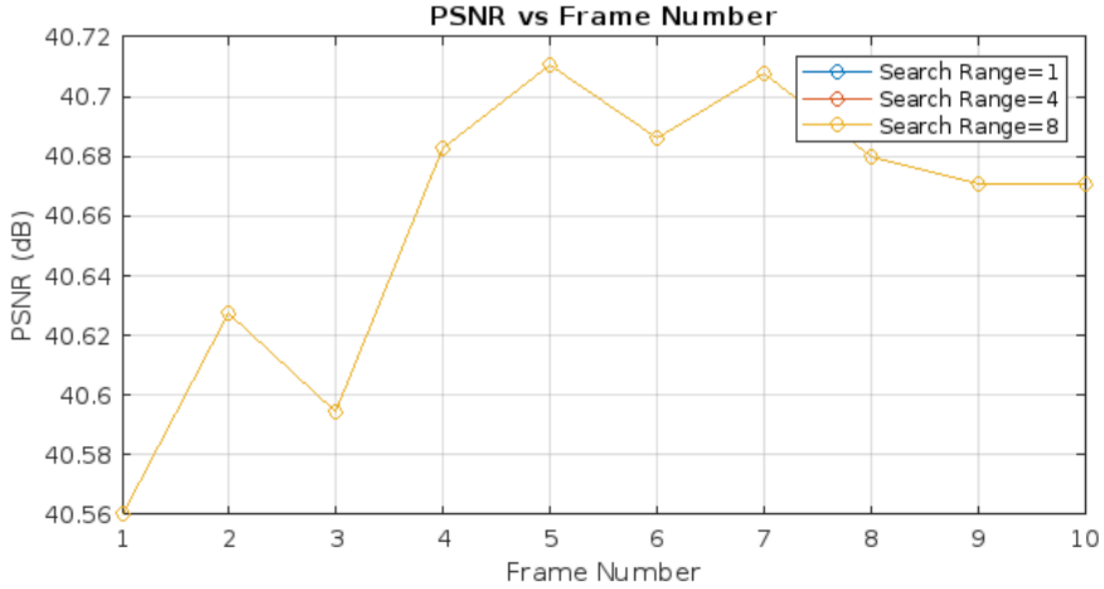


Figure 10: Per frame PSNR and MAE of r=4,n=3, and i being 2,8 and 64

3.3.2: Varying r with fixed i=8 and n=3



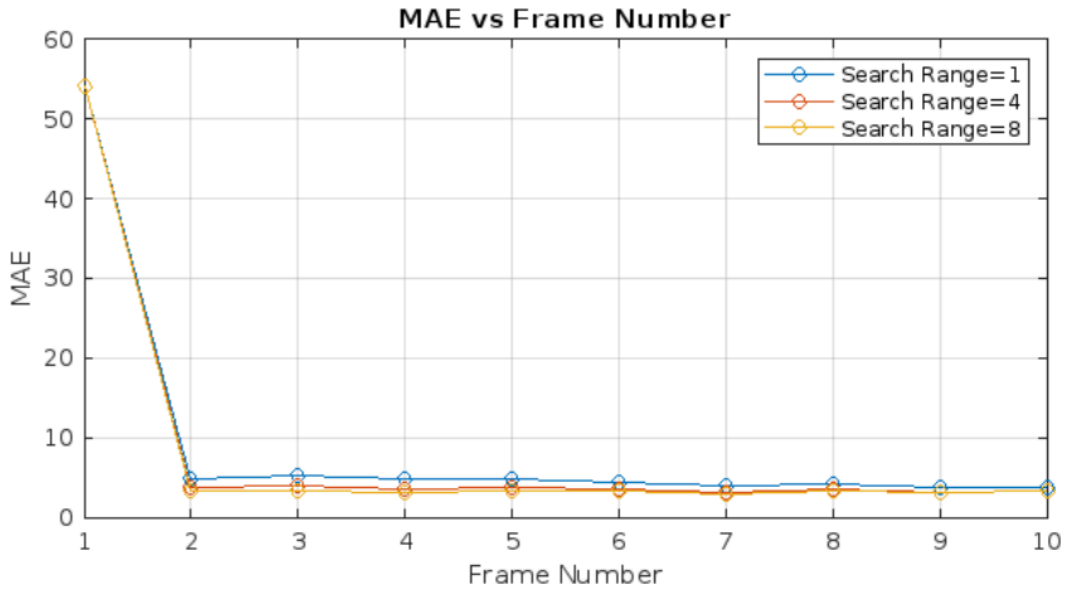


Figure 11: Per frame PSNR and MAE of i=8,n=3, and r being 1,4 and 8

3.3.3: Varying n with fixed i=8 and r=4

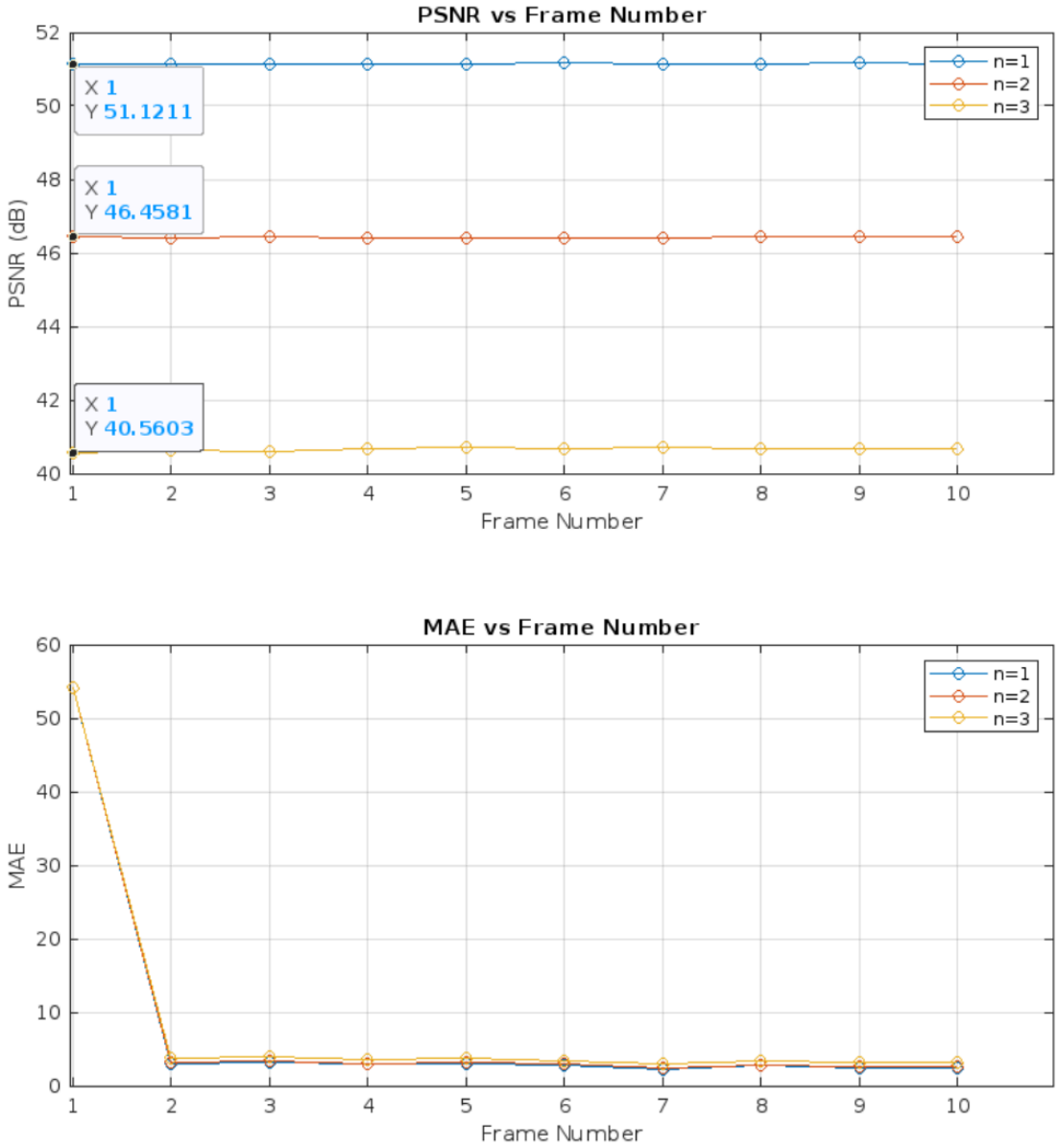


Figure 12: Per frame PSNR and MAE of r=4,i=8, and n being 1,2 and 3

3.3.1 Question: *Given the per-frame PSNR graph measured between original and reconstructed frames and the per-frame average MAE graph calculated during the MV selection process in the deliverables, which one will show clear variation with i and/or r? and why? Explain the results you are seeing.*

According to the per-frame PSNR and MAE graphs generated above, PSNR shows the clearest variation with n (residual approximation parameter). This is because n directly affects the quality of reconstruction by controlling the quantization of residuals (2^n rounding). Larger n values result in coarser quantization, leading to lower PSNR values.

The PSNR graph remains relatively stable across different values of i and r, this is because residuals are added back to the predicted frames, and the residual compensation mitigates the impact of prediction errors that i and r may cause.

The MAE graphs show noticeable variations with i and r, this is because MAE directly measures how well the motion estimation algorithm finds a match of blocks between frames. Larger block size (i) or limited search range (r) would reduce the chance of finding good matches.

3.4: Per-frame PSNR and MAE graphs of Akiyo QCIF

Akiyo QCIF sequence - 176*144:

3.4.1: Varying i with fixed r=4 and n=3

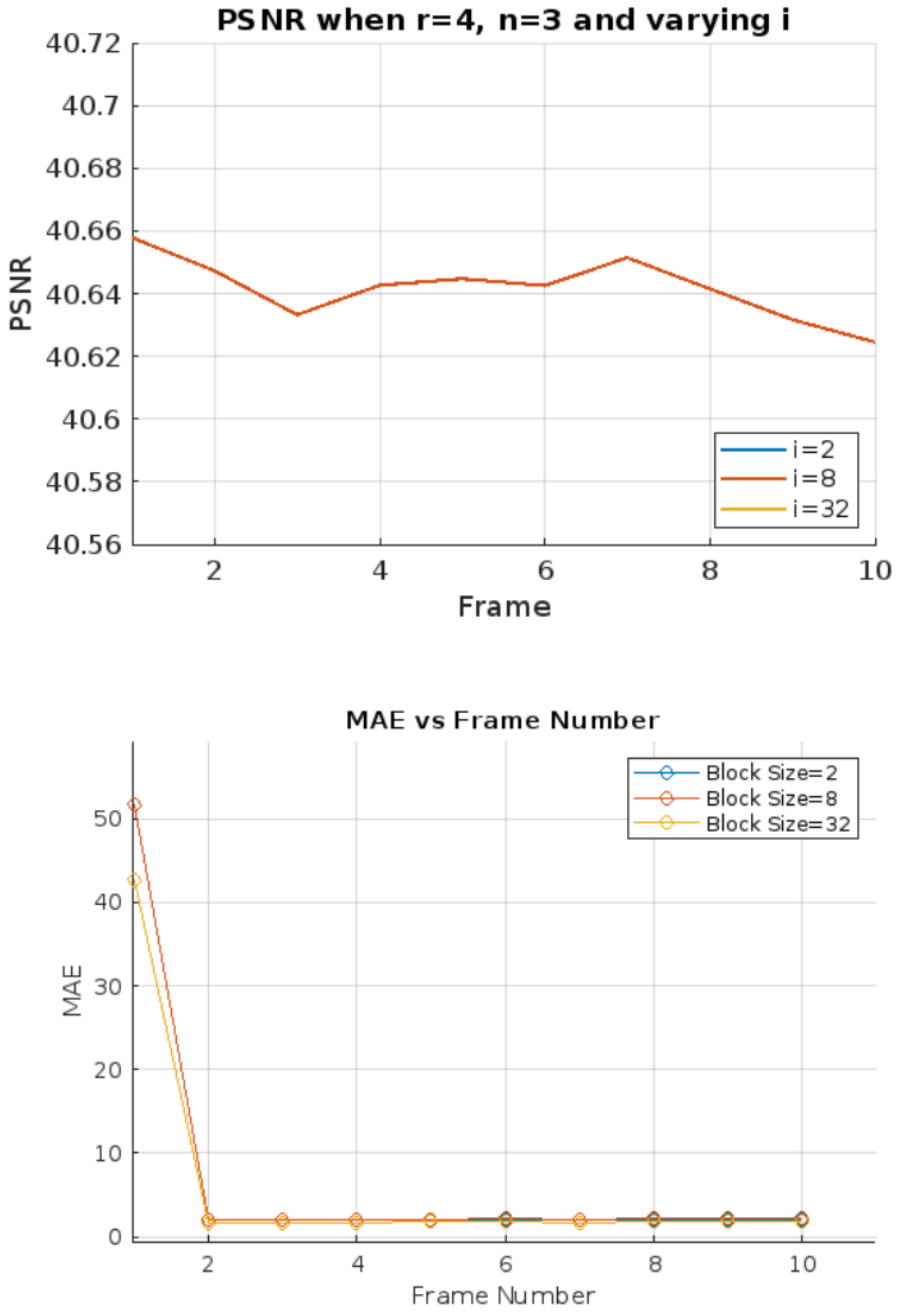


Figure 13:Per frame PSNR and MAE of r=4,n=3, and i being 2,8 and 32

3.4.2: Varying r with fixed i=8 and n=3

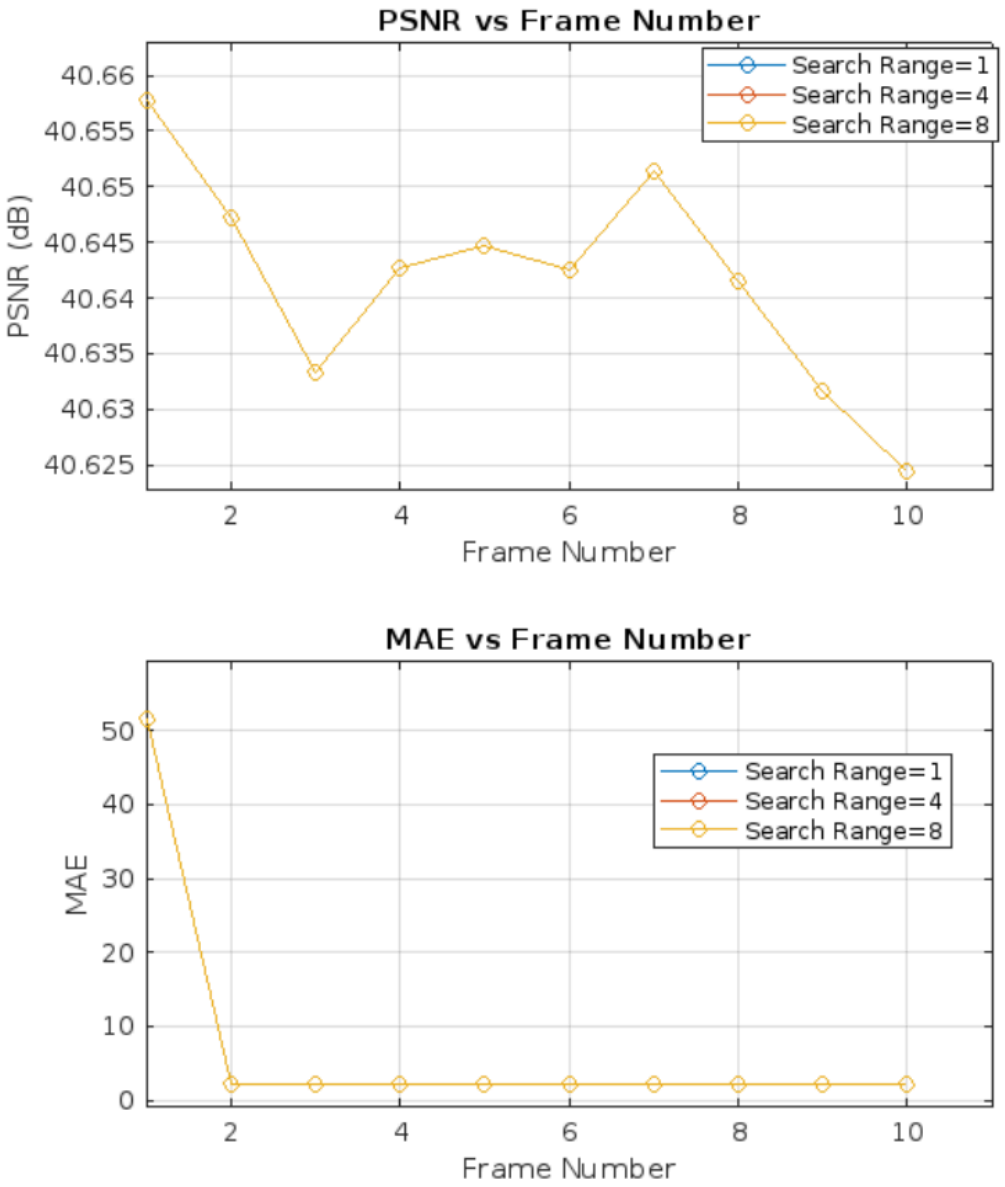


Figure 14:Per frame PSNR and MAE of i=8,n=3, and r being 1,4 and 8

3.4.3: Varying n with fixed i=8 and r=4

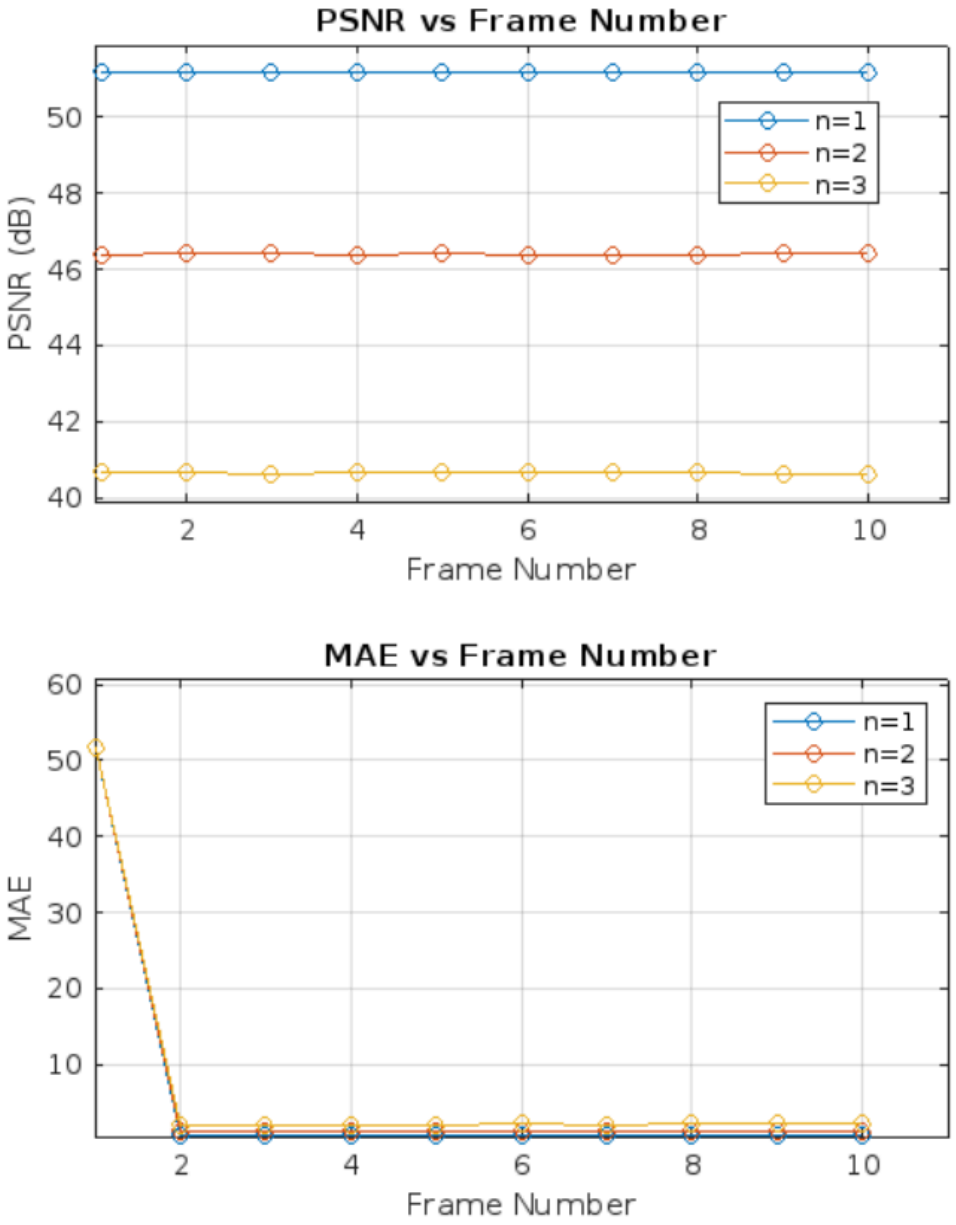


Figure 15:Per frame PSNR and MAE of r=4,i=8, and n being 1,2 and 3

3.5: Encoding Time and Residual Magnitude of Foreman CIF

Encoding Time at r=4, n=3:

Block Size (i)	Encoding Time (seconds)
2	170.12
8	8.11
64	0.72

Encoding Time at i=8, n=3:

Search Range (r)	Encoding Time (seconds)
1	4.93
4	8.16
8	15.93

Magnitude of residuals of r=4, n=3 and varying i:

	i=2	i=8	i=64
Frame 1	5495609	5495609	5495609
Frame 2	260848	377140	482288
Frame 3	272525	390763	498471
Frame 4	261290	364915	460144
Frame 5	257659	374411	481583
Frame 6	248761	350091	454845
Frame 7	231793	306291	406799
Frame 8	241903	349364	447739
Frame 9	233679	322703	429279
Frame 10	236877	329672	437143

Magnitude of residuals of i=8, n=3 and varying r:

	r=1	r=4	r=8

Frame 1	5495609	5495609	5495609
Frame 2	488236	377140	333708
Frame 3	521065	390763	342065
Frame 4	484239	364915	323419
Frame 5	476291	374411	341477
Frame 6	442041	350091	328623
Frame 7	394313	306291	293379
Frame 8	430092	349364	342672
Frame 9	389473	322703	318897
Frame 10	379946	329672	325806

The tables above indicate that an increase in block size (i) leads to a greater residual magnitude and a decrease in encoding time. Conversely, an increase in search range (r) reduces the residual magnitude but causes an increase in encoding time.

3.6: File Output

The Y-only reconstructed file of the first 10 frames of Foreman CIF can be found at Assignment1/Exercise_3/Outputs/decoded_Y_foreman_i-8_r-4_n-3.yuv. As indicated in the screenshots, the file is 1,013,760 bytes in size.

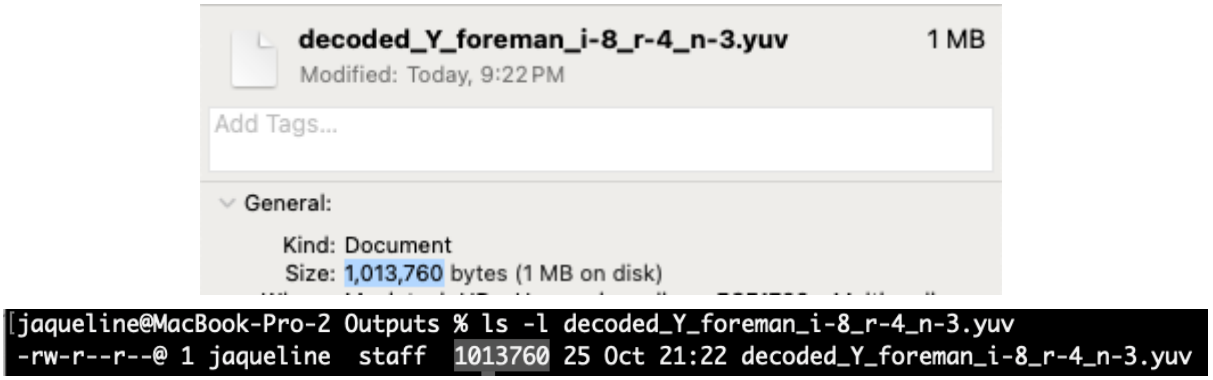


Figure 16: Y-only reconstructed file info

The motion vector txt file can be found at Assignment1/Exercise_3/Outputs/motion_vectors_i-8_r-4_n-3.txt
As it is shown in the figure, the motion vector contains 15840 (dy,dx) pairs.

```
15824   Block (217, 281): dy = 0, dx = 3
15825   Block (225, 281): dy = 0, dx = 0
15826   Block (233, 281): dy = 0, dx = 2
15827   Block (241, 281): dy = 0, dx = 4
15828   Block (249, 281): dy = 0, dx = 1
15829   Block (257, 281): dy = 0, dx = 0
15830   Block (265, 281): dy = 0, dx = -1
15831   Block (273, 281): dy = 0, dx = -1
15832   Block (281, 281): dy = 0, dx = -1
15833   Block (289, 281): dy = 0, dx = 0
15834   Block (297, 281): dy = 0, dx = 3
15835   Block (305, 281): dy = 0, dx = 0
15836   Block (313, 281): dy = 0, dx = 0
15837   Block (321, 281): dy = 0, dx = 0
15838   Block (329, 281): dy = 0, dx = 1
15839   Block (337, 281): dy = 0, dx = 1
15840   Block (345, 281): dy = 0, dx = 0
15841
```

Figure 17: Screenshot of motion vector txt file

Exercise 4: More Realistic Encoder/Decoder

4.1: Architecture Overview

The architecture implements a video encoding and decoding system with several key components aimed at compressing video data efficiently. The system employs transform coding, where each block of signed residuals is transformed using a 2D DCT (or two cascaded 1D DCTs). Quantization is applied to the DCT coefficients, using a matrix that depends on a quantization parameter (QP), with lower frequencies quantized more finely than higher frequencies.

In the prediction stage, I-frames are encoded using intra-prediction, selecting horizontal or vertical modes based on minimizing Mean Absolute Error (MAE). P-frames are encoded using motion estimation implemented in exercise 3, where motion vectors are differentially encoded relative to the previous block. Both motion vectors and intra-prediction modes are encoded using differential encoding and decoded at the receiver.

The entropy encoding step compresses the frame type markers, differential prediction information, and quantized transform coefficients by scanning them in diagonal order into 1d, followed by Exponential-Golomb coding and Run-Length Encoding (RLE) of zero and non-zero values. The decoder reverses these steps to reconstruct the video frames.

4.2: Rate-Distortion (R-D) Plots for Varying I_Periods and Block Sizes

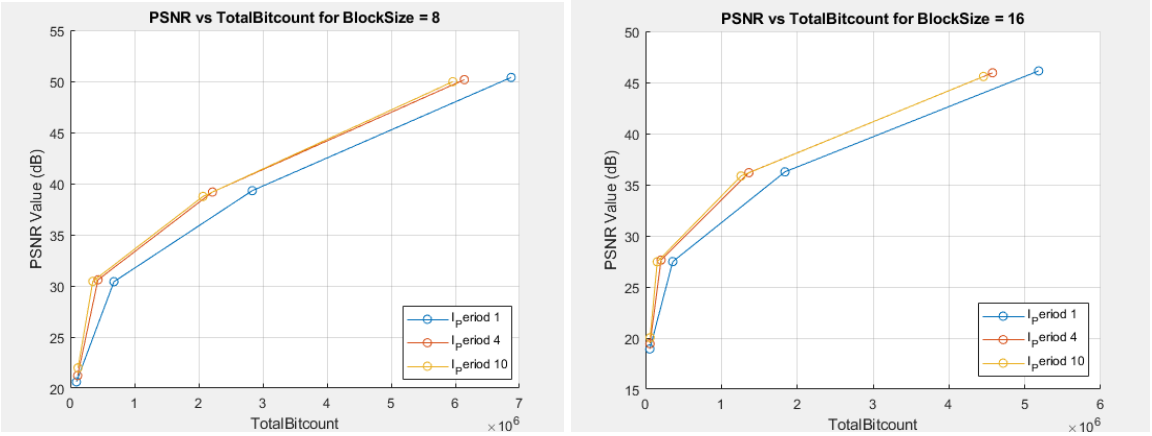


Figure 18,19: R-D plot when Block Size i = 8 and 16

The two plots illustrate the relationship between PSNR (Peak Signal-to-Noise Ratio) and Total Bitcount for different configurations of I-period and Block Size. The first plot, corresponding to a Block Size of 8, shows that as the Total Bitcount increases, the PSNR also improves for all three I-period values (1, 4, and 10). Among these, I-period 10 achieves the highest PSNR values with consistent growth, while I-periods 1 and 4 exhibit lower performance. The second plot, for a Block Size of 16, similarly shows an increase in PSNR with greater Total Bitcount, with I-period 10 consistently yielding the best results. These observations prove that smaller block sizes, higher Total Bitcount and Higher I-period values contribute to better PSNR performance.

4.3: Execution Time Plots for Varying I_Periods and Block Sizes

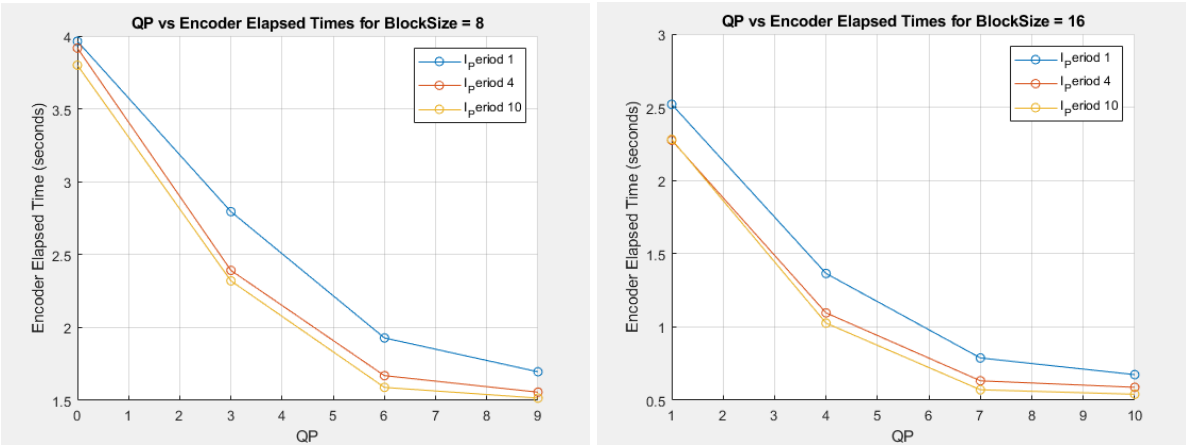


Figure 20,21: Per QP encoding times when Block Size i = 8 and 16

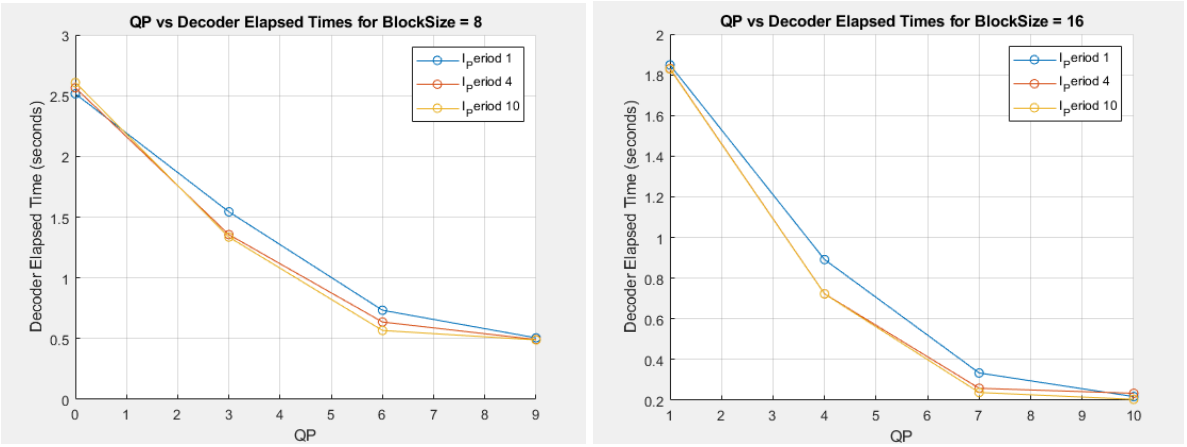
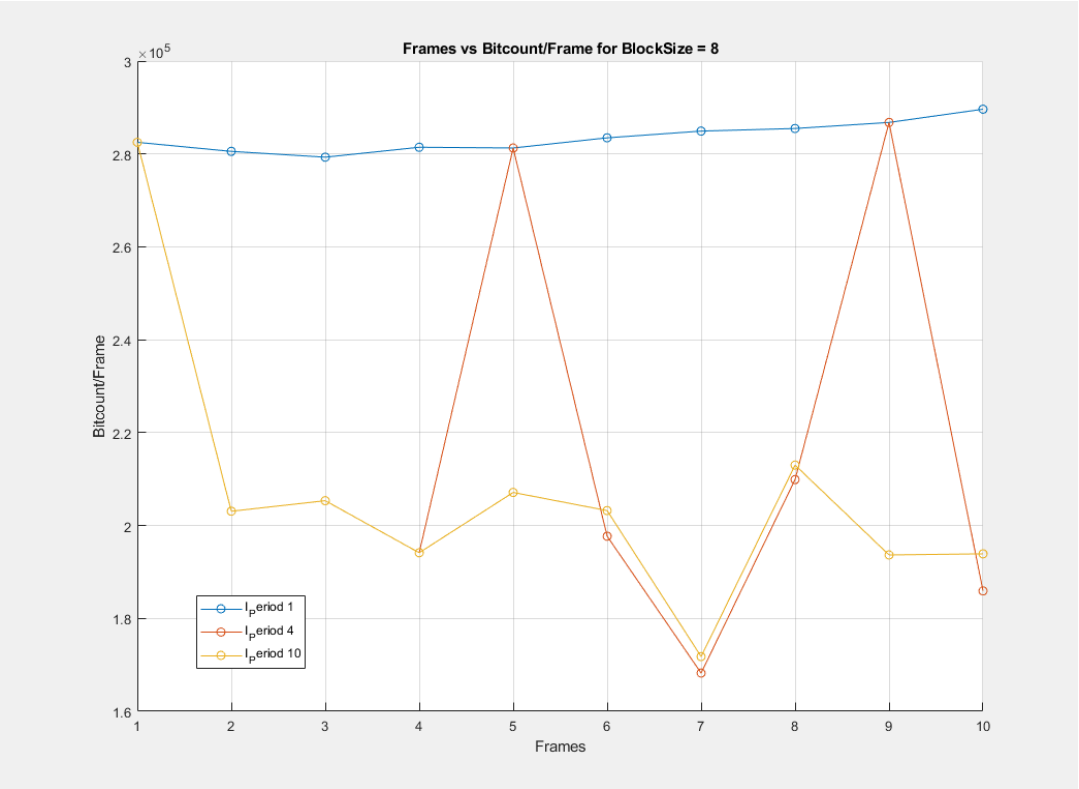


Figure 22,23: Per QP decoding times when Block Size i = 8 and 16

The provided figures illustrate the relationship between encoding/decoding times and the Quantization Parameter (QP) for two different block sizes (8 and 16) across varying I-frame periods (I_Period 1, 4, and 10). In both encoding and decoding, the elapsed time decreases as QP increases, due to the reduced complexity with higher quantization levels. Additionally, shorter I_Periods (such as I_Period 1) result in significantly longer processing times compared to larger I_Periods (like I_Period 10), as more frequent intra frames require more computation. For both block sizes, BlockSize 8 generally takes longer to process than BlockSize 16, reflecting the increased computational burden of smaller blocks, particularly at lower QP values.

4.4: Bit-Count Plots per Frame for Varying I_Periods



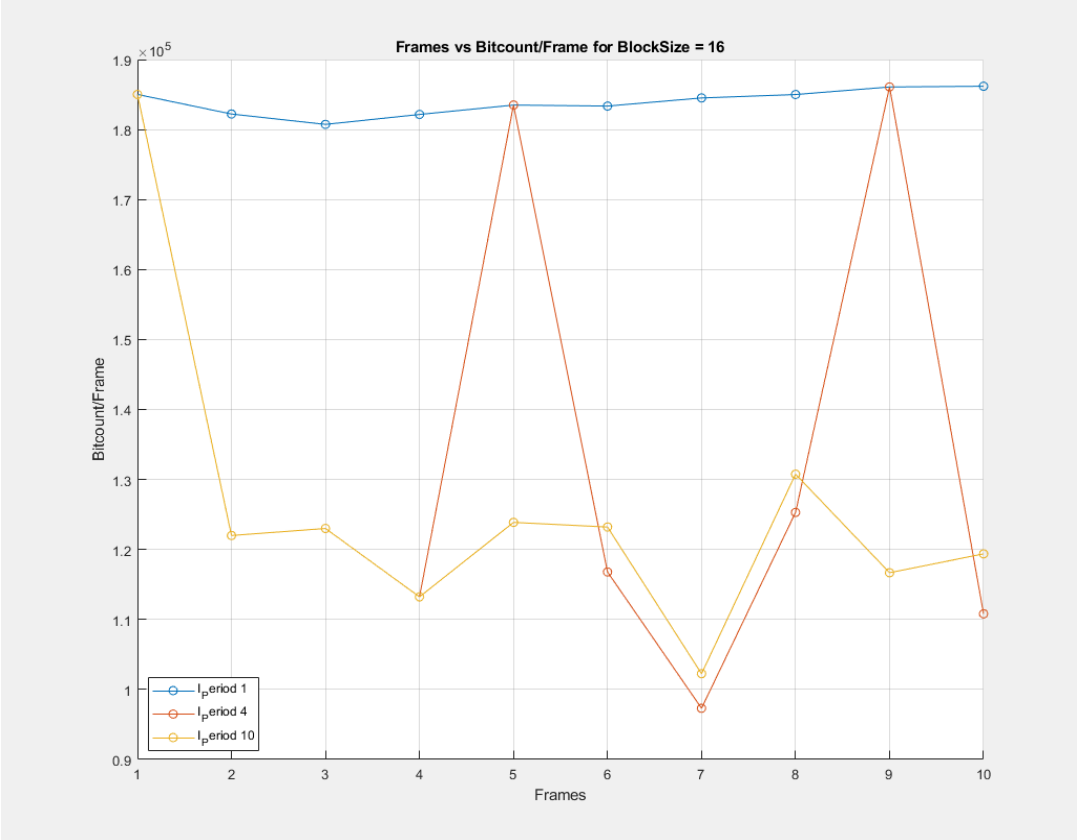


Figure 24,25: Per Frame Bit Count when Block Size i = 8 , QP = 3 and i = 16, QP = 4

4.4.1: Question: *Which kind of frames (Intra or Inter) typically consume more bitrate? Why? Is this true across the entire QP range?*

From the last 2 Figures, it is evident that intra frames typically consume more bitrates than inter frames for both Blocksize = 8 and Blocksize = 16 cases. For instance, when I_period equals to 4 and Block Size equals 16/8, at frame 1,5 and 9, Bit Counts are around 6000 bits higher than other frame bit counts. Overall, Whenever a frame is based on intra prediction, it requires approximately 50-100% more bitrate compared to a frame using inter prediction.

I-frames are encoded independently, meaning they contain all the data needed to reconstruct the frame without referencing other frames. As a result, they require more bits to store all the spatial information, leading to higher bitrates. In contrast, Inter frames leverage temporal redundancy by encoding only the differences (residuals) between the current frame and reference frames (previous frames), using motion estimation and compensation. This allows Inter frames to encode changes more efficiently, resulting in lower bitrates compared to I-frames. Because of this predictive nature, Inter frames are more efficient at compressing sequences with minor frame-to-frame differences.

4.4.2 Question: *For (i=8, search range = 2, and QP=3), what is the compression ratio of the 10 frames of Foreman CIF compared to the uncompressed Y component? What is the average PSNR?*

unCompressedY Bits = 352*288*8*10 = 8110080

- IPeriod = 1
    - Compressed Bits = 2955904
    - Compression Ratio:2.743
    - Average PSNR: 38.4729
- IPeriod = 4
    - Compressed Bits = 2105728
    - Compression Ratio:3.8514
    - Average PSNR: 38.4574
- IPeriod = 10
    - Compressed Bits = 1887384
    - Compression Ratio:4.29699
    - Average PSNR: 38.2119