ECE1783H Assignment 1 Report

Shengxiang Ji - 1002451232

Liangjing Xie - 1009684619

Part 1. Basic Motion Estimation/Compensation

1.1 Plots

1.1.1 per-frame PSNR graph

We plotted the per-frame PSNR graphs for different encoder configurations for the Foreman CIF sequence, as shown in Fig. 1, 2 and 3.

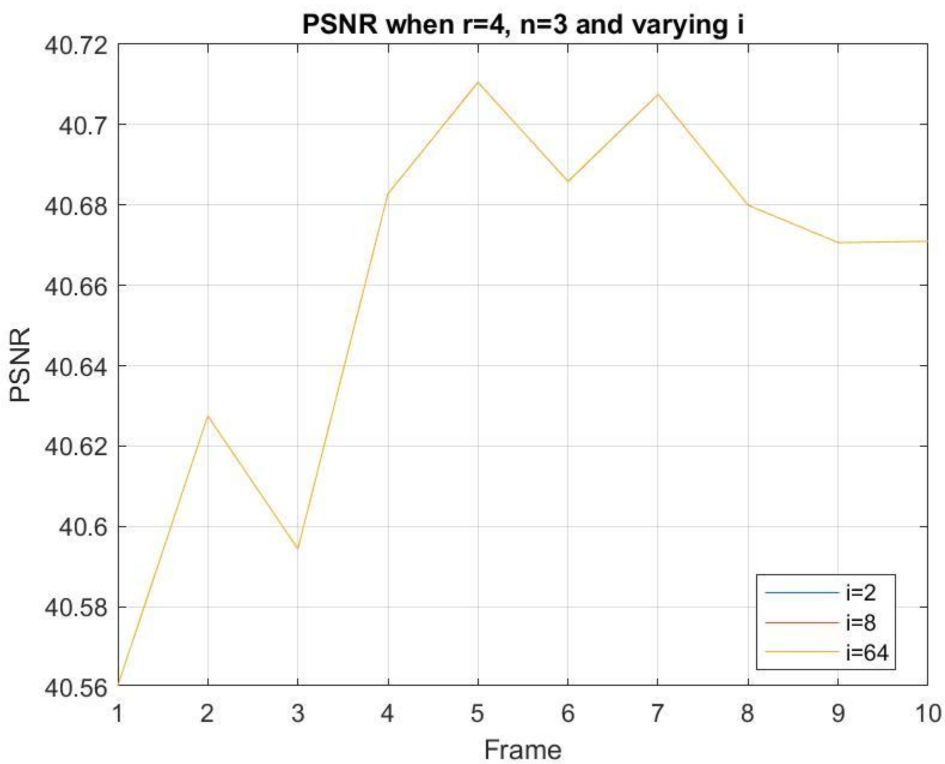


Fig. 1 Per-frame PSNR graph when varying i on Foreman CIF

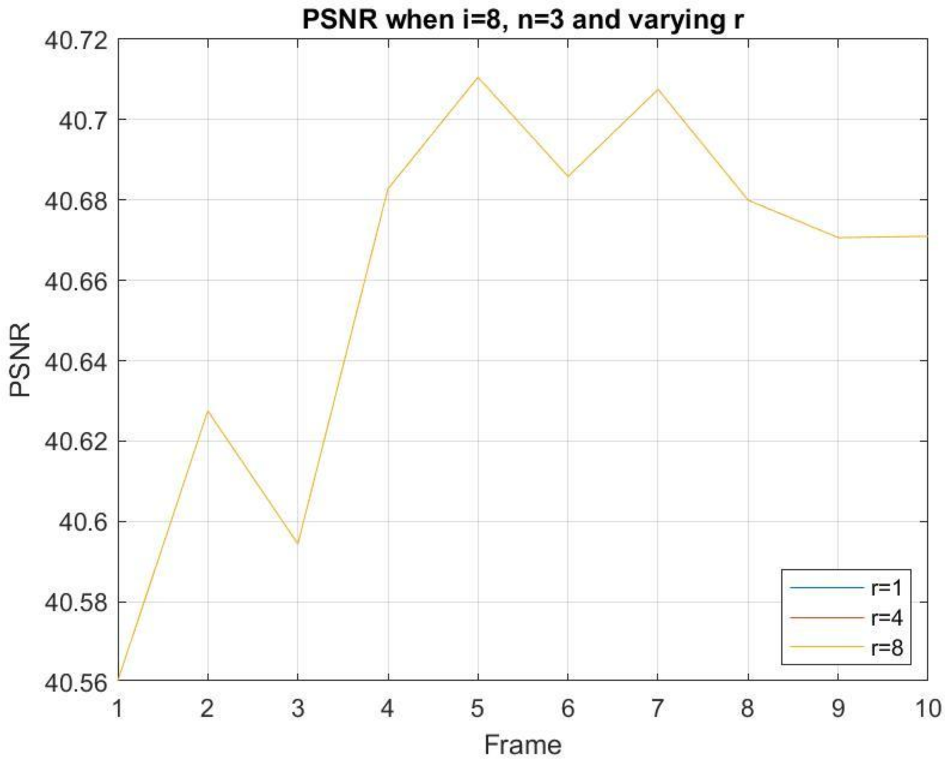


Fig. 2 Per-frame PSNR graph when varying r on Foreman CIF

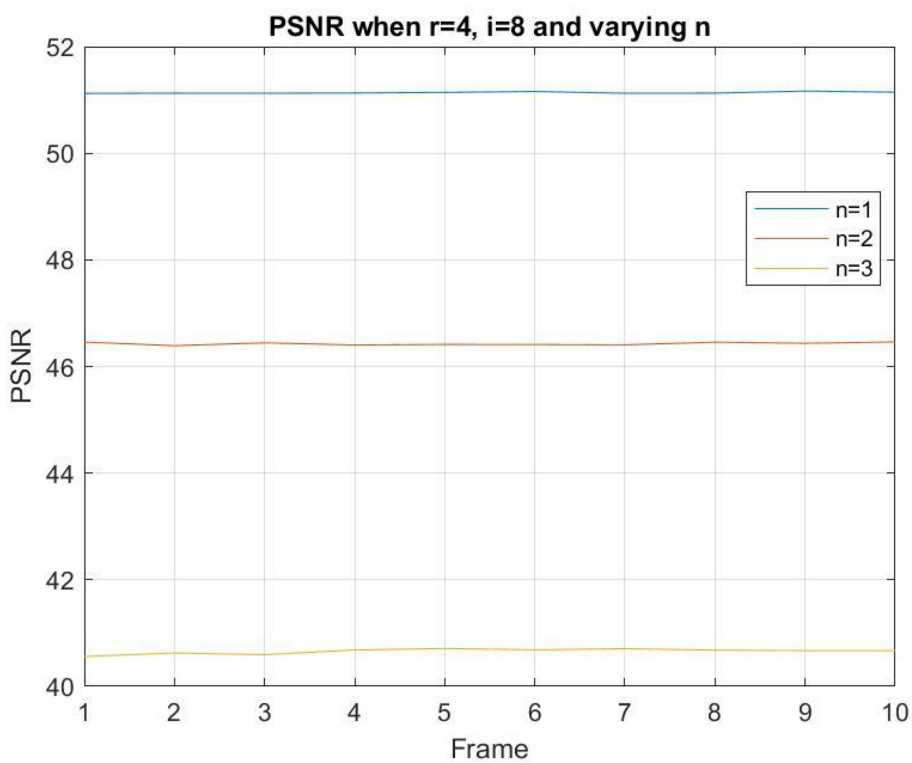


Fig. 3 Per-frame PSNR graph when varying n on Foreman CIF

We also plotted the same graphs for different encoder configurations for the akiyo QCIF sequence, which is 176x144, as shown in Fig. 4, 5 and 6.

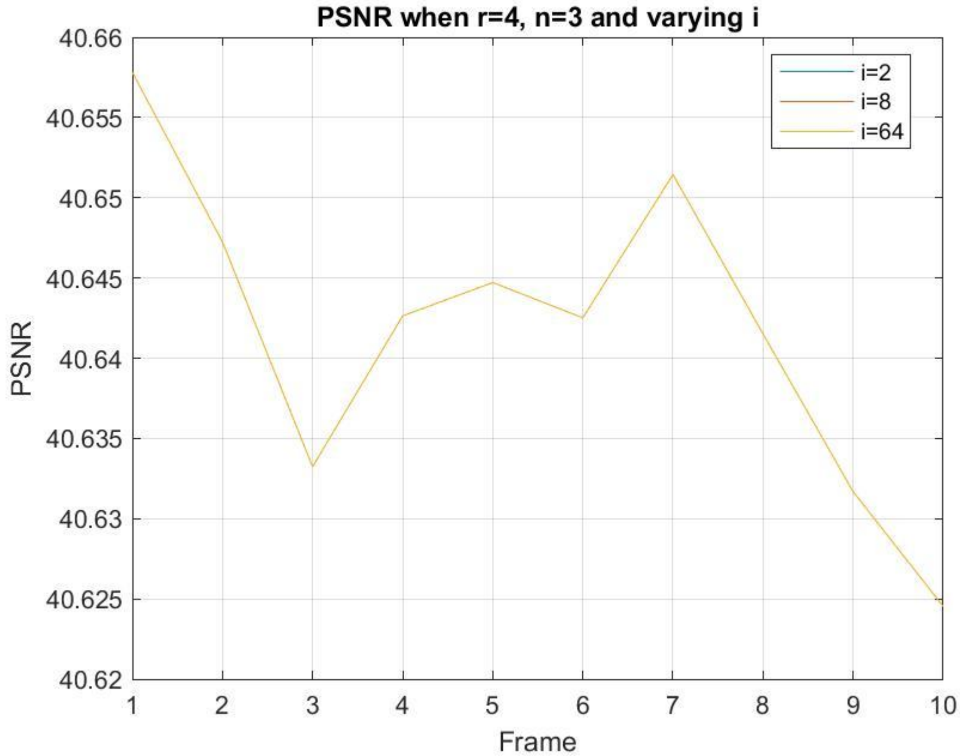


Fig. 4 Per-frame PSNR graph when varying i on Akiyo QCIF

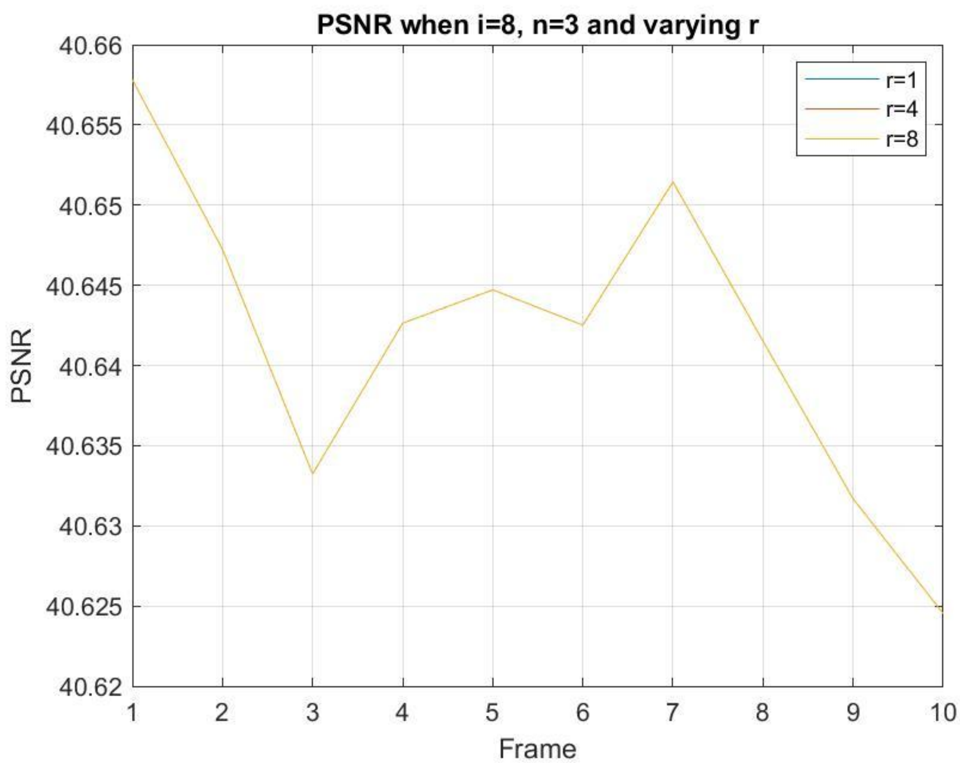


Fig. 5 Per-frame PSNR graph when varying r on Akiyo QCIF

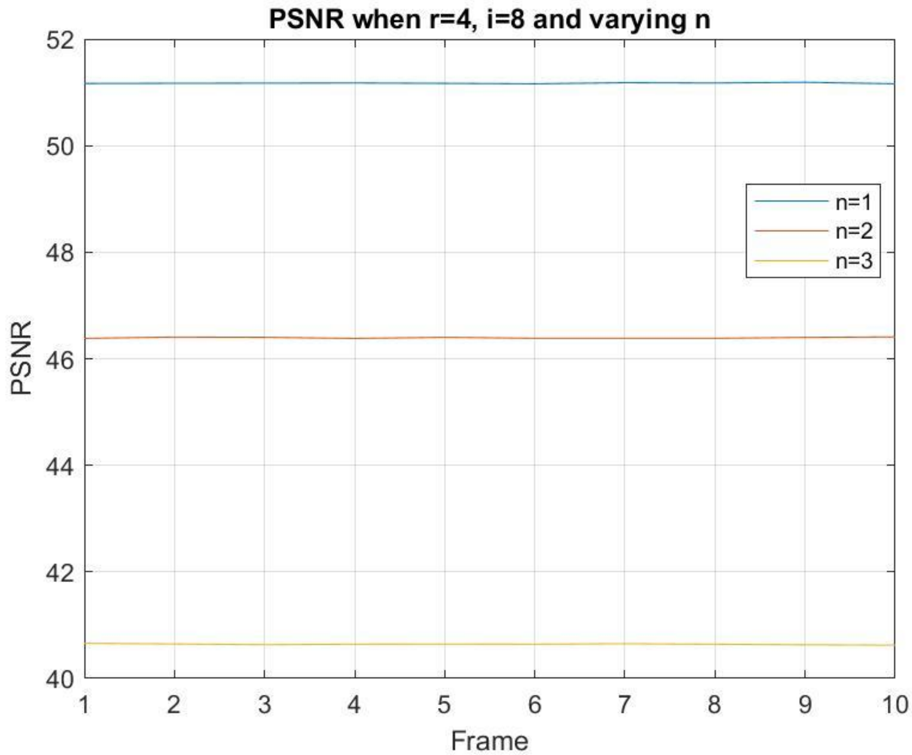


Fig. 6 Per-frame PSNR graph when varying n on Akiyo QCIF

1.1.2 per-frame average MAE graph

We plotted the per-frame average MAE graphs calculated during the MV selection process for different encoder configurations for the Foreman CIF sequence, as shown in Fig. 7, 8 and 9.

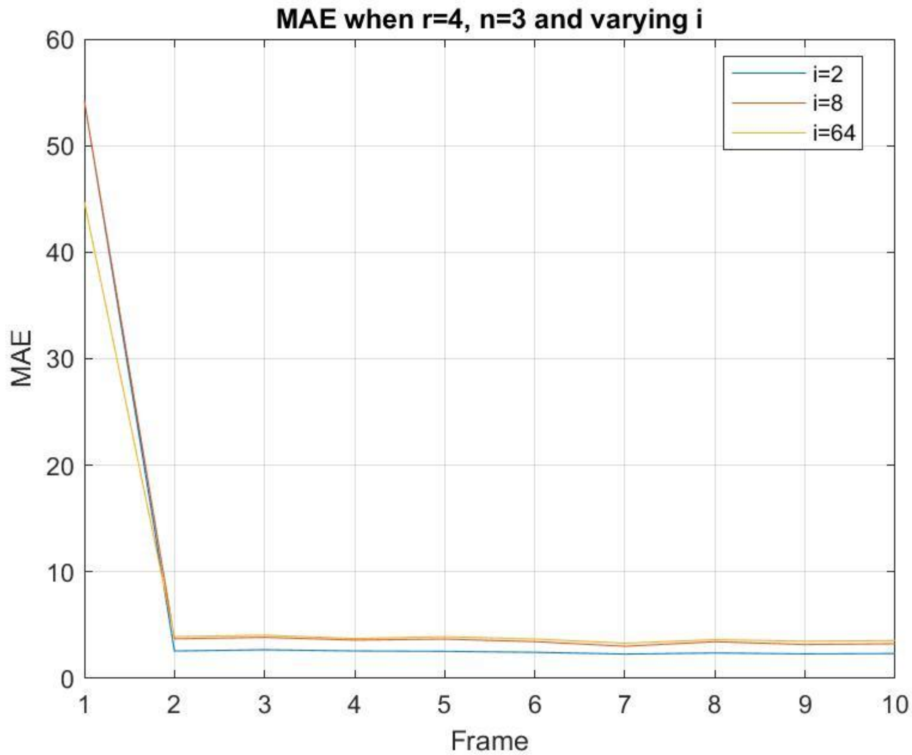


Fig. 7 Per-frame Average MAE graph when varying i on Foreman CIF

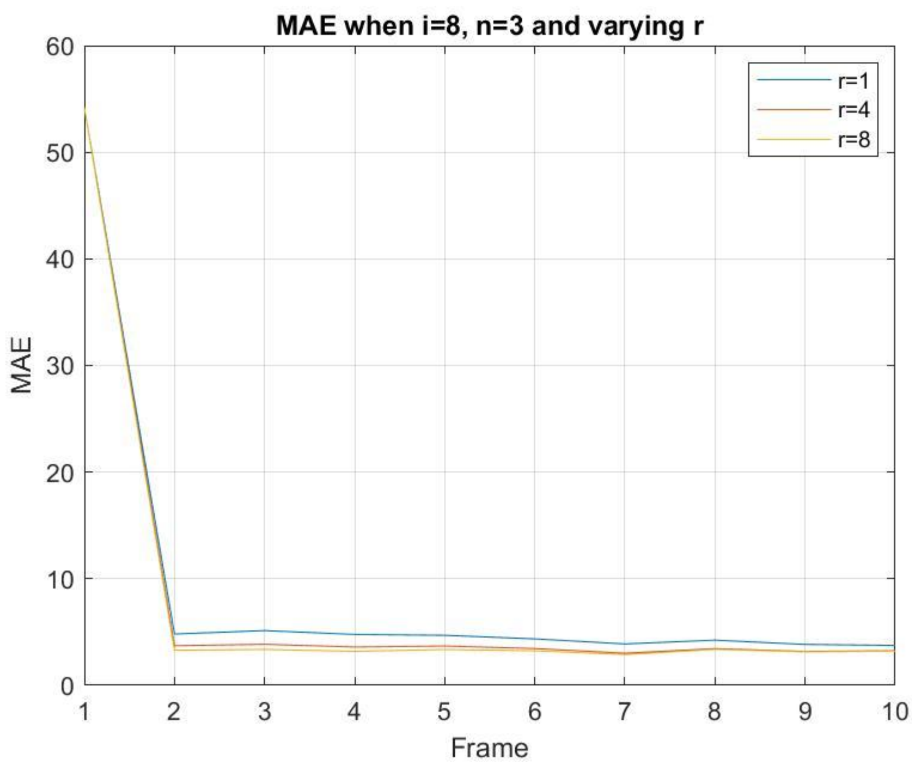


Fig. 8 Per-frame Average MAE graph when varying r on Foreman CIF

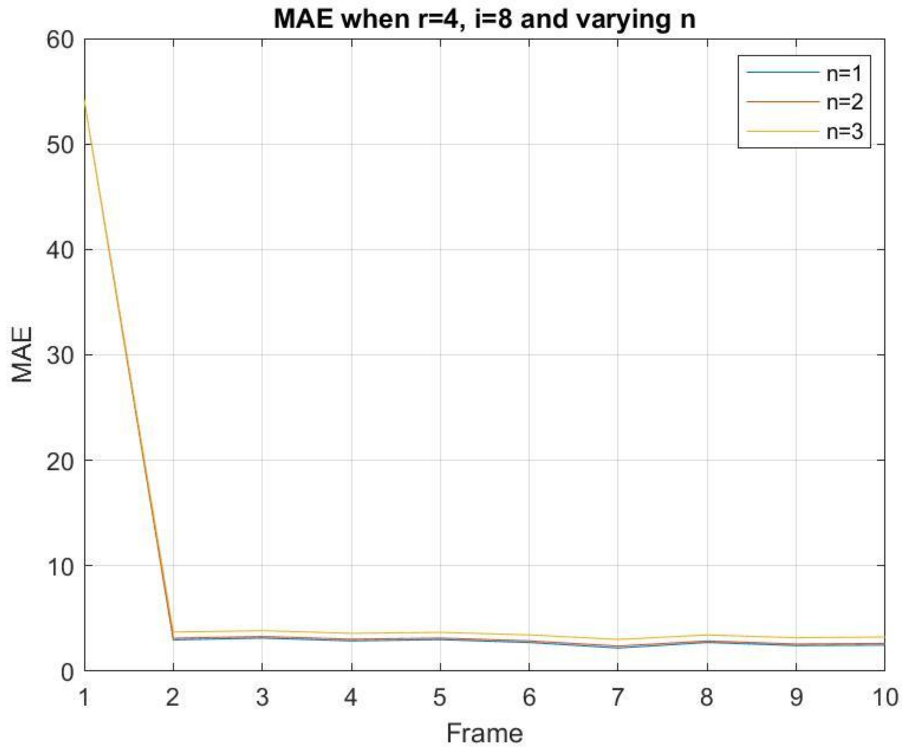


Fig. 9 Per-frame Average MAE graph when varying n on Foreman CIF

We also plotted the same graphs for different encoder configurations for the akiyo QCIF sequence, which is 176x144, as shown in Fig. 10, 11 and 12.

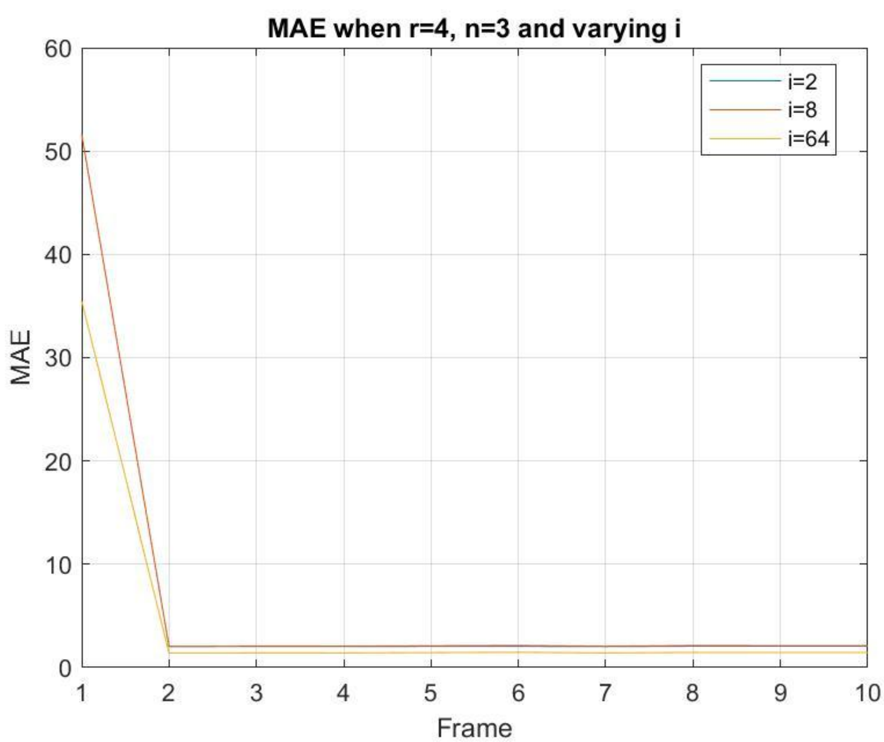


Fig. 10 Per-frame Average MAE graph when varying i on Akiyo QCIF

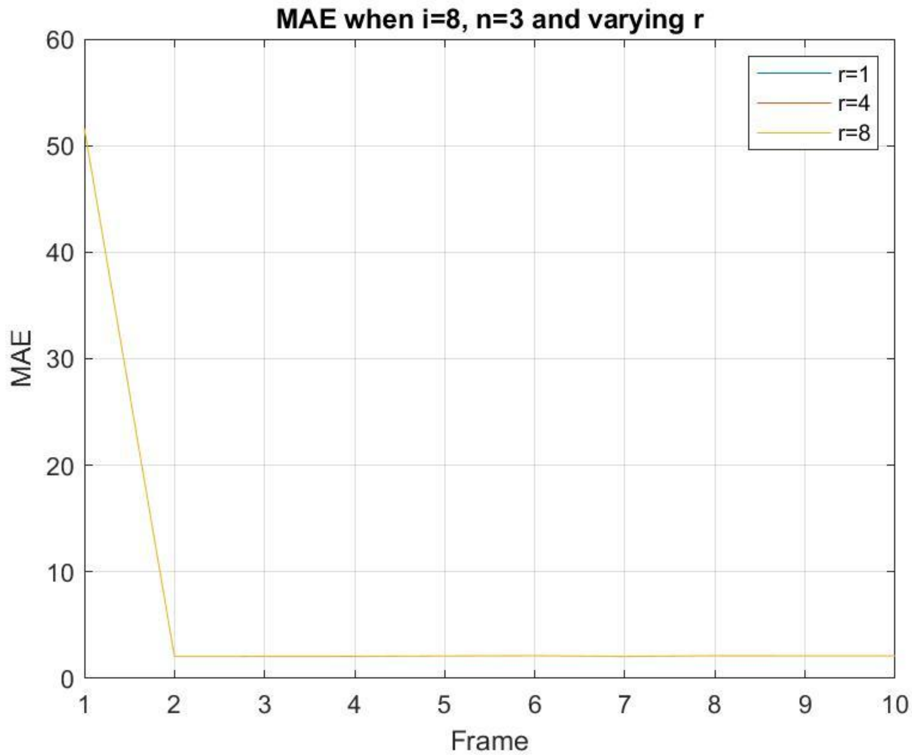


Fig. 11 Per-frame Average MAE graph when varying r on Akiyo QCIF

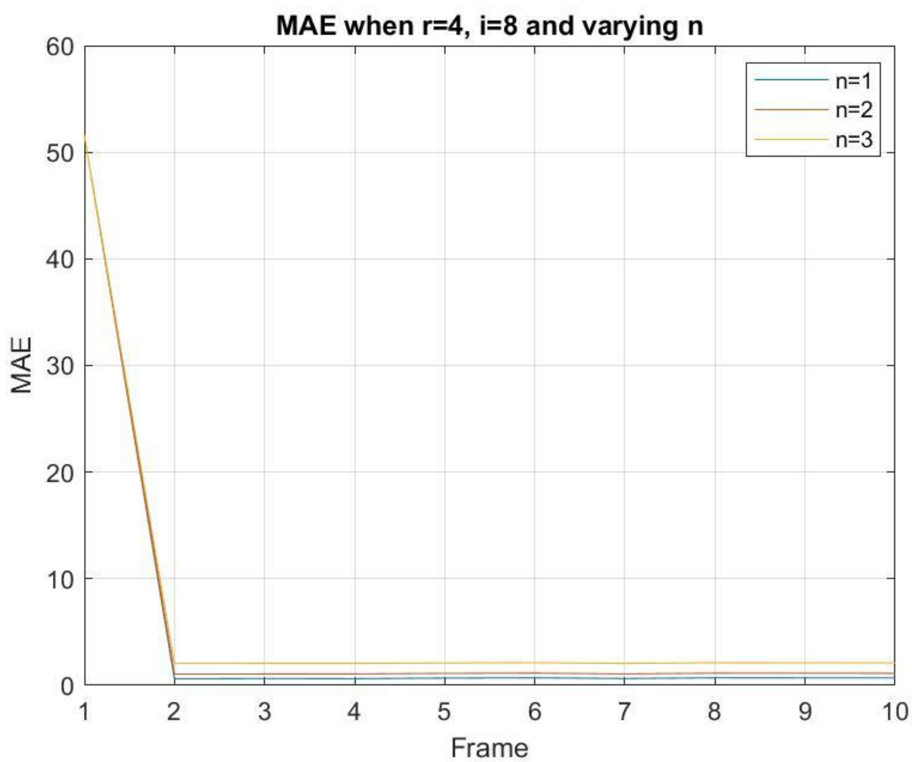


Fig. 12 Per-frame Average MAE graph when varying n on Akiyo QCIF

1.2 Frame Visualization

1.2.1 Case 1: i = 8

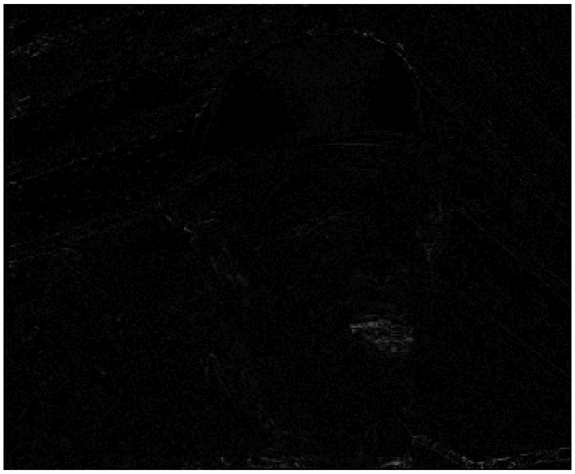
We plotted the residual before and after motion compensation, as well as the predicted frame before reconstruction, for frame 10 of the Foreman CIF sequence, with an encoder configuration of i=8, r=4 and n=3, as shown in Figure 13.

Reference (previous reconstructed) frame

Source frame (to encode)

Absolute value of residual with no motion comp.

Predicted frame after motion comp.



Absolute value of residual after motion comp.

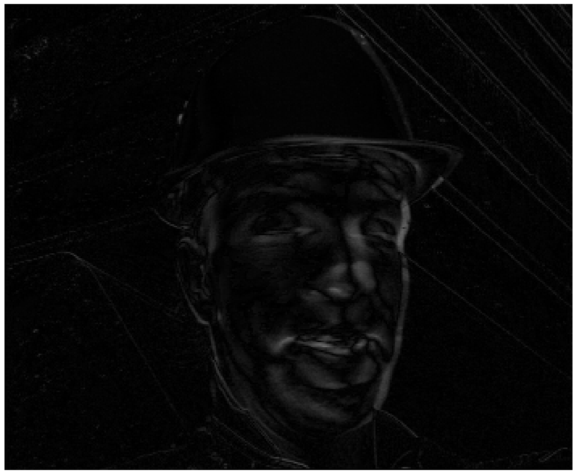
Fig. 13 Encoding process for frame 10 of Foreman CIF

1.2.2 Case 2: i = 64

We plotted the residual before and after motion compensation, as well as the predicted fram before reconstruction, for frame 3 of the Foreman CIF sequence, with a encoder configuration of i=64, r=4 and n=3, as shown in Figure 14.

Reference (previous reconstructed) frame

Source frame (to encode)



Absolute value of residual with no motion comp.

Predicted frame after motion comp.

Absolute value of residual after motion comp.

Fig. 14 Encoding process for frame 3 of Foreman CIF

1.3 Questions

1. 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.

As we can see from the graphs plotted in sections 1.1.1 and 1.1.2, the per-frame PSNR graphs do not show any clear variations with i and r, whereas the per-frame average MAE graphs show clearer variations with i and r. This is because the PSNR graphs are measured against the **reconstructed** frames, which takes the residual into account. In other words, the differences between the predicted frames and the current frames will be compensated by the residuals, therefore the reconstructed frame will not vary when i and r are changed. However, the per-frame average MAE during the MV selection process looks at the previous frame and the current frame. When the block size is larger, it's harder to find a good match in the previous frame, therefore the average MAE will be larger. When the search range is larger, it's easier to find a good match in the previous frame, therefore the average MAE will be smaller.

2. What is the effect of the i and r parameters on residual magnitude and encoding time? Provide a table that shows how encoding time varies, and another table for residual magnitude.

We ran the encode many times on the Foreman CIF sequence with different encoder configurations, and documented the magnitude of residuals and encoding time, as shown in the tables below.

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

	i=2	i=8	i=64
Frame 1	5532328	5532328	5532328
Frame 2	184784	356016	473024
Frame 3	200776	370432	487848
Frame 4	187168	341264	445696
Frame 5	183176	351568	469400
Frame 6	173192	326328	443376
Frame 7	150616	273688	384464
Frame 8	164928	326880	437032
Frame 9	154752	298304	417936
Frame 10	159480	304912	424400

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

	r=1	r=4	r=8
Frame 1	5532328	5532328	5532328
Frame 2	474856	356016	305032
Frame 3	507080	370432	316248
Frame 4	467712	341264	294232
Frame 5	460928	351568	314328
Frame 6	426048	326328	301352

Frame 7	370232	273688	257752
Frame 8	416096	326880	318352
Frame 9	372856	298304	292576
Frame 10	362104	304912	300112

Encoding time of r=4, n=3 and varying i

	i=2	i=8	i=64
time	42.3145	6.8654	4.4970

Encoding time of i=8, n=3 and varying r

	r=1	r=4	r=8
time	5.1072	6.9015	12.1584

From the above tables, we can discover that increasing i will generally increase the residual magnitude, but will decrease the encoding time. Increasing r will decrease the residual magnitude but will increase the encoding time.

1.4 File Output

Y-only reconstructed file of the first 10 frames of Foreman CIF :
The Y-only reconstructed file is in
.\ex3Output\ex3_foreman420_cif_i8_encoderReconstructionOutput\foreman420_cif10.yuv.
As can be seen in Fig. 15, the file is exactly 1,013,760 bytes.

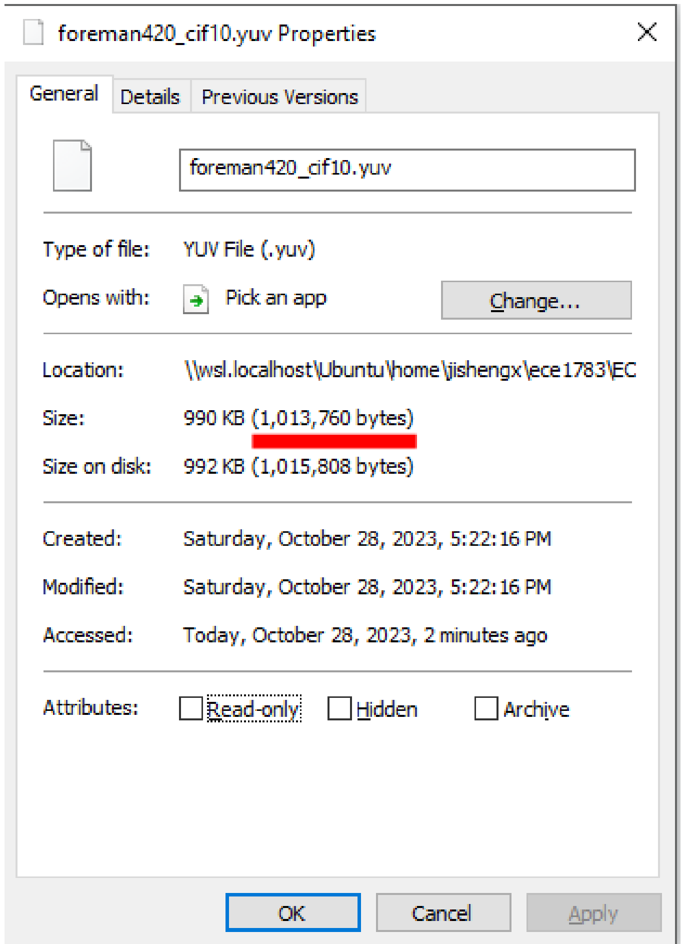


Fig. 15 Y-only dump for the first 10 frames of Foreman CIF

Text file with the found motion vectors:

The motion vectors of the first 10 frames of the Foreman CIF are in .\ex3Output\ex3_foreman420_cif_i8_MVOutput\MVTextFile.txt. As can be seen in Fig. 15, the file contains 15840 x-y pairs.

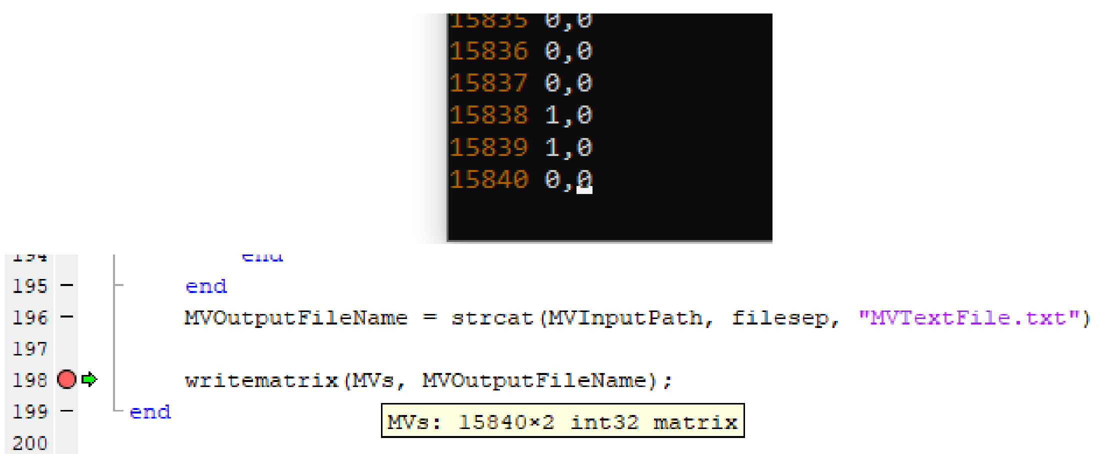


Fig. 15 Motion Vectors for the first 10 frames of Foreman CIF

Part 2. Realistic Encoder/Decoder

2.1 Plots

2.1.1 PSNR vs Bitcount

For a set of specified parameters i = 8 and a search range of 2, we plotted the relationship between PSNR and bitcount. In each plot, we included three curves corresponding to different values of IPeriod (as indicated by the labels). we got the different points of a curve by varying the QP=0:1:(log2(8)+7), as shown in Fig. 16.

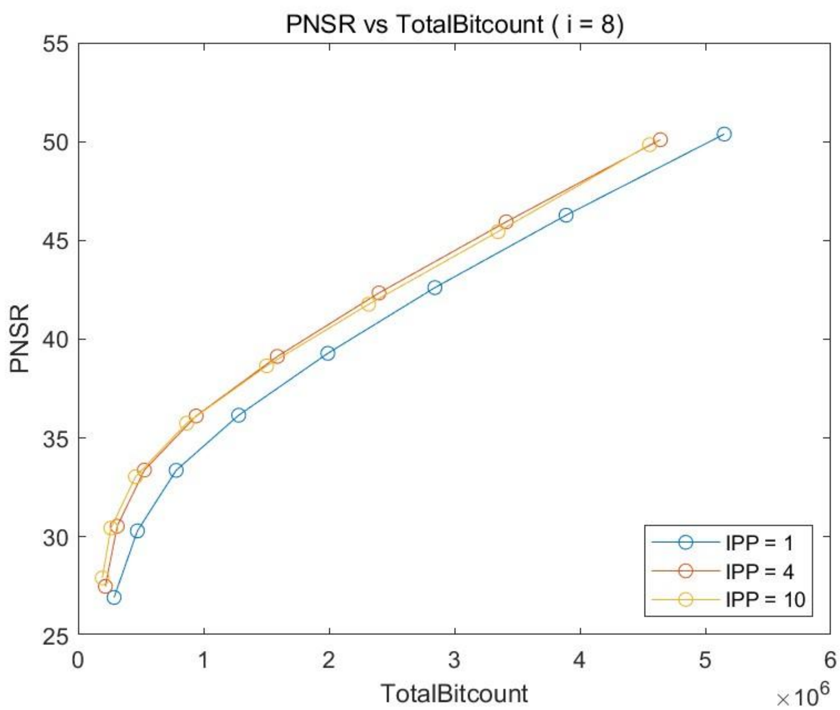


Fig. 16 R-D plot when i=8

Subsequently, with the value of i set to 16 and QP = 0:1:(log2(16) + 7), we repeated the aforementioned procedures and generated the plot in Fig 17.

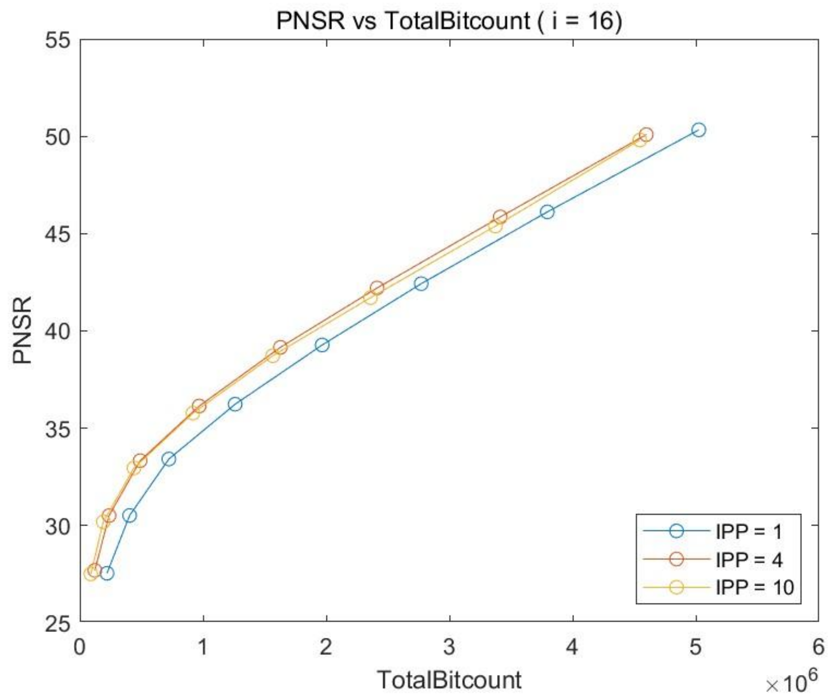


Fig. 17 R-D plot when i=16

2.1.2 Execution times

While conducting the experiments, we also recorded the time taken for each encoding and decoding action. we plotted the following graphs with time in seconds on the y-axis and the values of QP on the x-axis (the parameter values are indicated in the graphs), as shown in Fig. 18 through Fig. 21.

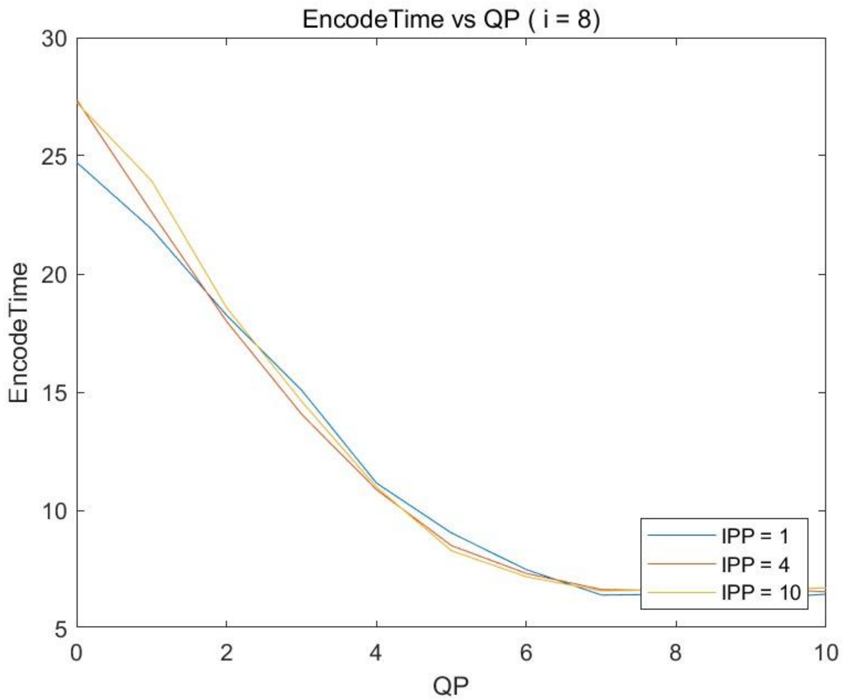


Fig. 18 Encoding times when i=8

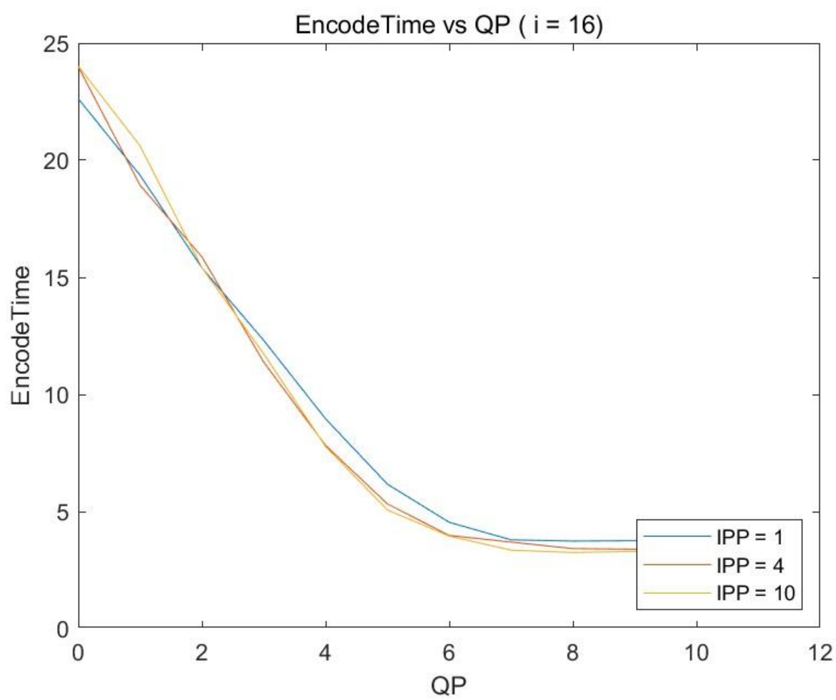


Fig. 19 Encoding times when i=16

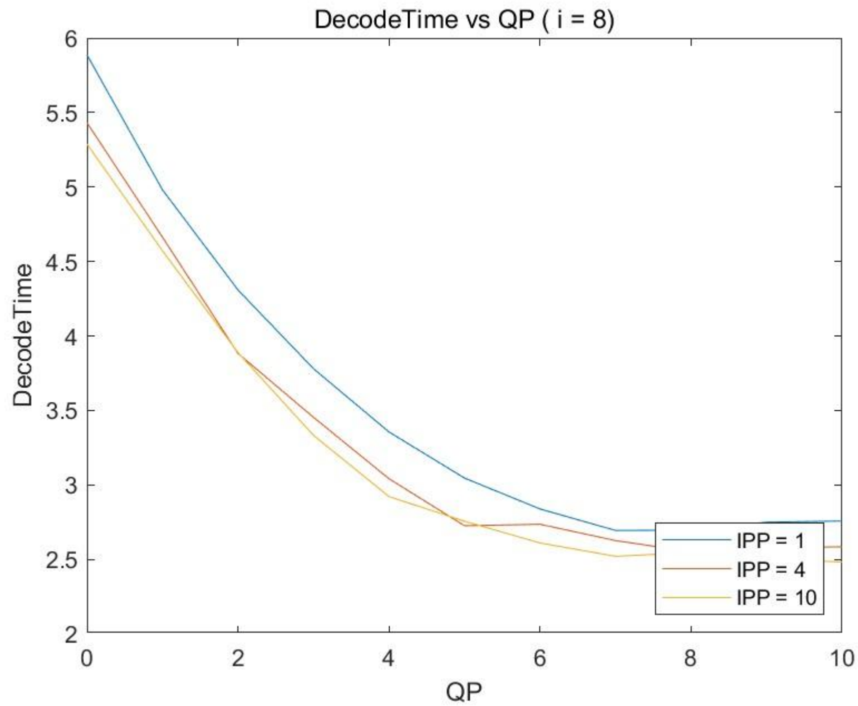


Fig. 20 Decoding times when i=8

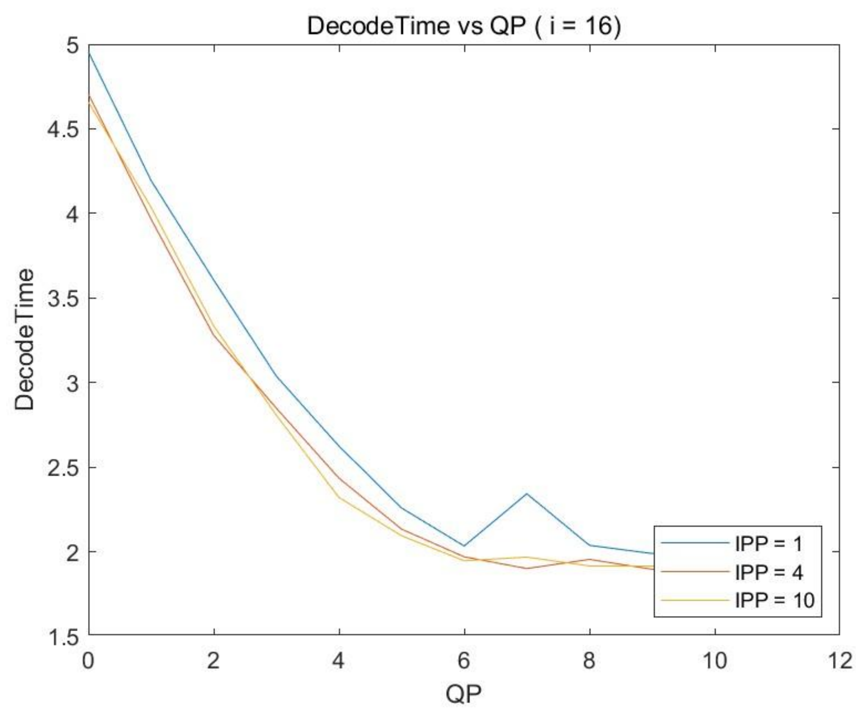


Fig. 21 Decoding times when i=16

From the above images, it can be seen that as the QP (Quantization Parameter) value increases, both encoding and decoding times decrease. This is because increasing the QP value reduces the total bit count. Additionally, for the same parameter values, encoding time is significantly greater than the time required for decoding, which is due to the fact that encoding involves more computational tasks. When the value of 'i' increases, both encoding and decoding times decrease. This is because increasing the value of 'i' leads to a reduction in the number of blocks within the same frame, further reducing the computational workload.

2.1.3 Bitcount vs FrameIndex

For (i=8, QP=3) and (i=16, QP=4) experiments, plot the bit-count (on y-axis) vs. frame index (on x-axis) curves. Show plots for three different values of I_Period (1, 4 and 10). These plots are shown in Fig. 22 and 23.

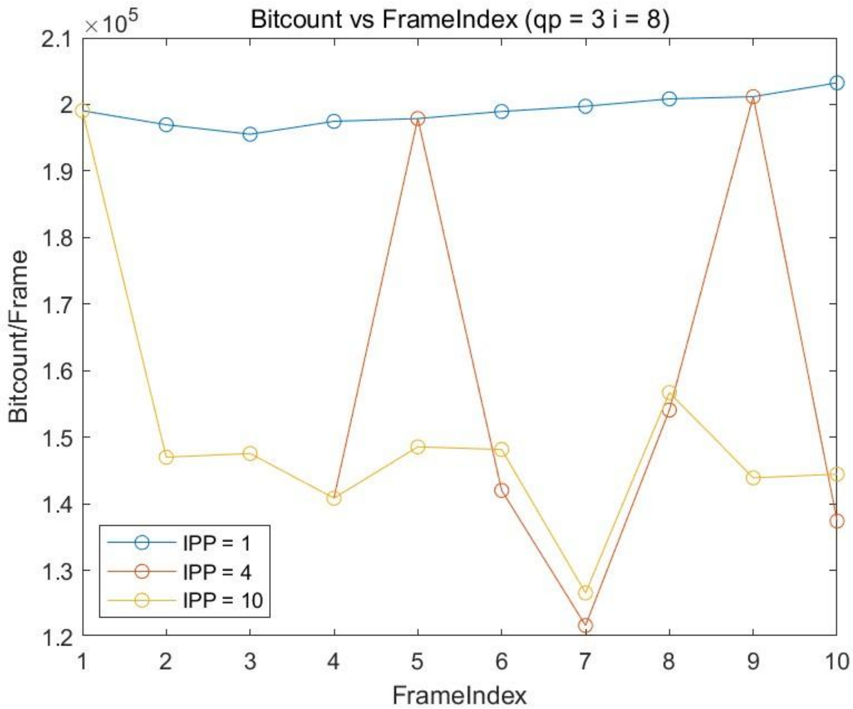


Fig. 22 Bitcount for different frames when qp=3 and i=8

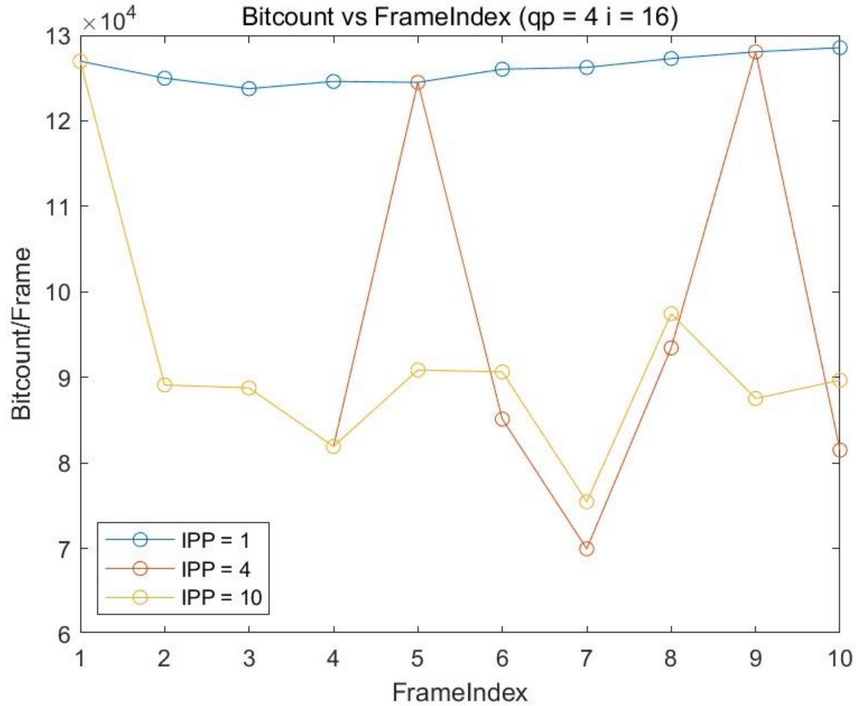


Fig. 23 Bitcount for different frames when qp=4 and i=16

2.2 Questions

1. Which kind of frames (Intra or Inter) typically consume more bitrate? Why? Is this true across the entire QP range?

Intra frames typically consume more bitrate compared to Inter frames. In the figures titled "Bitcount vs FrameIndex" (i.e. Fig. 22 and 23), we can clearly observe that Intra frames consume more bitrate than inter frames. When the value of I Period is set to 1, the bitcount for each frame is at its maximum. When the I Period is set to 4, there is a sharp increase in the bitcount value every four frames. When the I Period is set to 10, out of the ten frames analyzed, only the first frame is an I frame, and its bitcount value is the highest.

This pattern arises from the fact that Inter frames exploit redundancy between successive frames and only record changes between frames. In video files, adjacent frames often share a significant degree of similarity, enabling inter-frame prediction to achieve compression. Intra-frames, in contrast, compress each frame individually without considering the other frames in the sequence. This means that intra-frame compression requires more data to be stored for each frame.

In our experiments, it was observed that Intra frames have a higher bitcount compared to Inter frames, regardless of the specific value of the QP. In real-world scenarios, the exact bitcount can vary due to several factors, including the video content and the specific encoding settings. However, the greater bitcount of Intra frames in comparison to Inter frames is primarily determined by the prediction methods of these two frame types and the characteristics of the video file, rather than QP. Therefore, we can assert that Intra frames typically require a higher bitcount, irrespective of the selected QP value.

2. 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?

originalBitcount = width * height * nFrame * 8 = 352 * 288 * 10 * 8 = 8,110,080 (bits)

(1) IPeriod= 1
compressedBitcount = Bitcount(QTCCoeffs) + Bitcount(MDiffs)
comparisonRatio = originalBitcount / compressedBitcount = 4.0739
averagePSNR = $\sum_{1}^{10} PSNR(Frame_i)/10$ = 39.2890

(2) IPeriod= 4
compressedBitcount = Bitcount(QTCCoeffs) + Bitcount(MDiffs)
comparisonRatio = originalBitcount / compressedBitcount = 5.1049
averagePSNR = $\sum_{1}^{10} PSNR(Frame_i)/10$ = 39.1356

(3) IPeriod= 10
compressedBitcount = Bitcount(QTCCoeffs) + Bitcount(MDiffs)
comparisonRatio = originalBitcount / compressedBitcount = 5.3967
averagePSNR = $\sum_{1}^{10} PSNR(Frame_i)/10$ = 38.6708

Script

```
clc; clear; close all;
param_8_2_3_1 = struct( 'yuvInputFileName',
'foreman420_cif.yuv','yuvOutputFileName',
'DecoderOutput\DecoderOutput.yuv','nFrame', 10,'width', 352,'height',
288,'blockSize', 8,'r', 2,'QP', 3,'I_Period', 1 );
param_8_2_3_4 = struct( 'yuvInputFileName',
'foreman420_cif.yuv','yuvOutputFileName',
'DecoderOutput\DecoderOutput.yuv','nFrame', 10,'width', 352,'height',
288,'blockSize', 8,'r', 2,'QP', 3,'I_Period', 4 );
param_8_2_3_10 = struct( 'yuvInputFileName',
'foreman420_cif.yuv','yuvOutputFileName',
'DecoderOutput\DecoderOutput.yuv','nFrame', 10,'width', 352,'height',
288,'blockSize', 8,'r', 2,'QP', 3,'I_Period', 10 );
[comparisonRatio, averagePSNR] = answer_2_2_1(param_8_2_3_1)
[comparisonRatio, averagePSNR] = answer_2_2_1(param_8_2_3_4)
[comparisonRatio, averagePSNR] = answer_2_2_1(param_8_2_3_10)
function [comparisonRatio, averagePSNR] = answer_2_2_1(param)
   originalBitcount = param.width * param.height * param.nFrame * 8;
```

```
    comparisonBitcount = double(0);
    totalPSNR = double(0);
    ex4_encoder(param.yuvInputFileName, param.nFrame, param.width,
param.height, param.blockSize, param.r, param.QP, param.I_Period);
    load('QTCCoeffs.mat', 'QTCCoeffs');
    load('MDiffs.mat', 'MDiffs');
    ex4_decoder(param.nFrame, param.width, param.height, param.blockSize,
param.QP, param.I_Period, QTCCoeffs, MDiffs);
    YOutput = importYOnly(param.yuvOutputFileName, param.width,
param.height, param.nFrame);
    [YOriginal, U, V] = importYUV(param.yuvInputFileName, param.width,
param.height, param.nFrame);

    for i=1:param.nFrame
        comparisonBitcount = comparisonBitcount +
sum(strlength(QTCCoeffs(i,:)), "all") + sum(strlength(MDiffs(i,:)),
"all");
        totalPSNR = totalPSNR + psnr(YOutput(:, :, i), YOriginal(:,:,i));
    end
    comparisonRatio = originalBitcount / comparisonBitcount;
    averagePSNR = totalPSNR / param.nFrame;
end
```