ELEC 483

Hybrid Video Encoder

Department of Electrical and Computer Engineering

Alexander Stewart and Kirsten Dohmeier

Contents

ABSTRACT

A hybrid video encoder and decoder allowing for variable DCT, quantization, motion estimation and GOP structure parameters was implemented. These parameters were varied and the corresponding results of the encoded and decoded videos were observed in the form of PSNR of the decoded video, execution time, and a crude compression ratio comparing the number of non-zero values in the encoded video to the number of non-zero values in the original and decoded videos. It was found that decreasing the number of DCT coefficients in the encoded video decreased PSNR and decreased the compression ratio, with fewer than 25 coefficients reducing PSNR below 30dB. It was found that by increasing the scaling factor of the quantization matrix that PSNR quickly dropped off, but the compression ratio decreased more slowly above a scaling factor of 1. Changing the number of P or B frames used had limited effect on PSNR. Increasing the number of B frames more noticeably decreased the compression ratio, but the presence of P and B frames drastically increased encoding time. Increasing the motion estimation search window had limited effects on PSNR, but dramatically increased execution time for only a small gain in compression ratio. Decreasing macroblock size used for motion estimation drastically increased computation time, produced a minor decrease in PSNR and a compression ratio which needs consideration of the motion vectors for analysis.

INTRODUCTION

Since the invention of the compact disk, methods for compressing digital data have been developed and steadily grown in complexity. Digital data compression provides a means of representing a signal in an encoded form that can be stored using fewer bits and correspondingly decoded to reproduce the initial signal or a reasonable approximation. Many video encoders today have a similar underlying structure, the block diagram for which is shown in Figure 1. For this project, a simple hybrid video encoder and corresponding decoder were implemented using MATLAB. Parameters of the individual blocks and group of pictures (GOP) structure were varied to analyse the effects of these parameters on the quality of the decoded video, computation efficiency, and a simplified compression ratio.

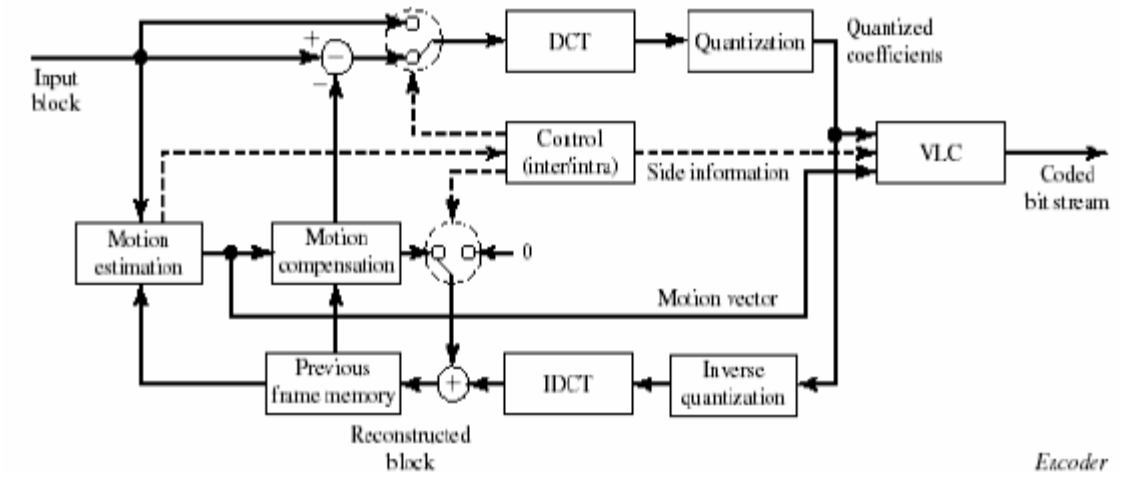

Fig. 1. Hybrid Encoder [1]

THEORY

As stated in the introduction, hybrid encoders have a similar structure whose block diagram is shown in Figure 1. This encoding structure has a number of variable features. Hybrid encoders allow variation in DCT and quantization block parameters, motion estimation parameters, and the GOP structure of the encoded frames. These were the parameters which were varied during the analysis of the implemented encoder and decoder.

At the level of individual DCT blocks, parameters of the DCT and quantization were made variable. Firstly, the scaling factor for quantization was user-definable. A larger scaling factor resulted in a greater number of zero elements in the DC transformed signal and thus greater compression. Secondly, the number of DCT coefficients kept was variable using the zig-zag scanning pattern. For motion estimation, the block size and search window size of the motion estimation macroblocks were both variable. In the GOP structure, the encoding order of Intra-coded pictures (I), Predicted pictures (P), and Bi-predictive pictures (B) could be varied. This was implemented such that the chosen number of P-frames between I-frames was constant throughout the video sequence, as was the number of B-frames between I/P-frames. In some standards, such as MPEG, this is not the case.

ANALYSIS

For each combination of the input parameters, the execution time of both the encoder and decoder, the PSNR of the decoded video, and the ratio of nonzero elements between the original and decoded video (crude compression ratio) were used as evaluation parameters for the encoder and decoder.

DCT Coefficients

The analyse the effect of the number of DCT coefficients retained in the encoded video, the number of coefficients kept was varied. To make this measure independent of GOP structure, this analysis was performed using only I frames. As a result, the output image was not affected by block size or search window size. The main evaluation criteria for this experiment was the PSNR and compression ratio.

Quantization Scaling Factor

To analyse the effect of the degree of quantization of the DCT of the video, the scaling factor for quantization matrix was varied. As with the DCT coefficient experiment, this analysis was performed using only I frames. The main evaluation criteria for this experiment was the PSNR and compression ratio.

GOP Structure

To analyse the effect of using different GOP structures, the video sample was encoded using all combinations from 0 P and 0 B frames to 4 P and 4 B frames. The block size, window size, scaling factor and number of DCT coefficients were all kept constant in this analysis. As the encoded error should compensate for loss of image quality, the compression ratio and execution time were used as the main evaluation criteria.

Window Size

To analyse the effect of varying window size during motion estimation, the window size was varied while maintaining a constant GOP structure. The block size, quantization scaling factor and number of DCT coefficients were all kept constant. Compression ratio and execution time were used as the main evaluation criteria.

Macroblock Size

To analyse the effect of varying macroblock size during motion estimation, the macroblock size was varied as the video was encoded and decoded. The number of DCT coefficients, quantization scaling factor, GOP structure and search window size were all held constant. Compression ratio and execution time were used as the main evaluation criteria.

IMPLEMENTATION

A 50-frame animated video sequence of a collapsing star was used for analysis. Individual function .m files were created to perform the DCT, inverse DCT, quantization, inverse quantization, and motion estimation blocks in Figure 1. An encoder function accepted parameters for the number of P frames, B frames, block size, window size, scaling factor, number of DCT coefficients, as well as the video to be encoded. This function called the aforementioned blocks in the order necessary to reflect the input number of P and B frames. The motion compensation was implemented as a part of the motion estimation block. If the encoder function determined that the current frame to encode is a P or B frame, it called the motion estimation block (implemented using EBMA), which outputted both the motion vectors and the predicted frame. For B frames, this process was repeated with the second target frame and the two predicted frames were averaged to determine the final predicted image. The encoder function computed the error matrix from the predicted frame and the actual frame. The motion vectors were computed from the target I or P frames once they had been DC transformed, quantized, and the inverse applied, as this was the frame that the output frame would be determined from in the decoder.

In some hybrid encoder implementations a loop filter is used to reduce blocking errors created during motion estimation. The edges of prediction blocks often create sudden jumps at the edges of macroblocks. A loop filter is a low-pass filter which reduces these errors at the edges of macro blocks. This was not implemented in our hybrid encoder for two reasons. First, a loop filter would introduce an additional degree of freedom during our experiments, which would naturally have a very large range of values. Secondly, a loop filter could reduce or remove blocking errors caused by varying another parameter. This would increase uncertainty when observing the effects of specific encoder variables. Because of these factors, a loop filter is not included in our implementation.

Additional blocks from those shown in Figure 1. At the input to the encoder in the figure, blocks were created to convert the frames from RGB to YCbCr and subsample the chrominance components. Correspondingly, blocks were created to upsample and convert to RGB in the decoder. 4:2:0 chrominance subsampling was used to maintain symmetric resolution in the horizontal and vertical directions. The initial implementation performed the operations of the hybrid encoder using only the luminance component. In order to increase accuracy, the motion estimation was redesigned to include chrominance information to minimize the mean absolute difference (MAD) during block matching. However, to save on encoding time during multiple executions, the chrominance components were not encoded further than being subsampled.

EXAMPLES

(For examples, please see attached disk)

DISCUSSION

DCT Coefficients

The number of DCT coefficients kept in the encoding process was varied from 1 to 64. The other parameters were held as follows:

- Quantization scaling factor: 1
- Search window size: 8
- Block size: 8
- GOP structure: I-frames (no P or B-frames)

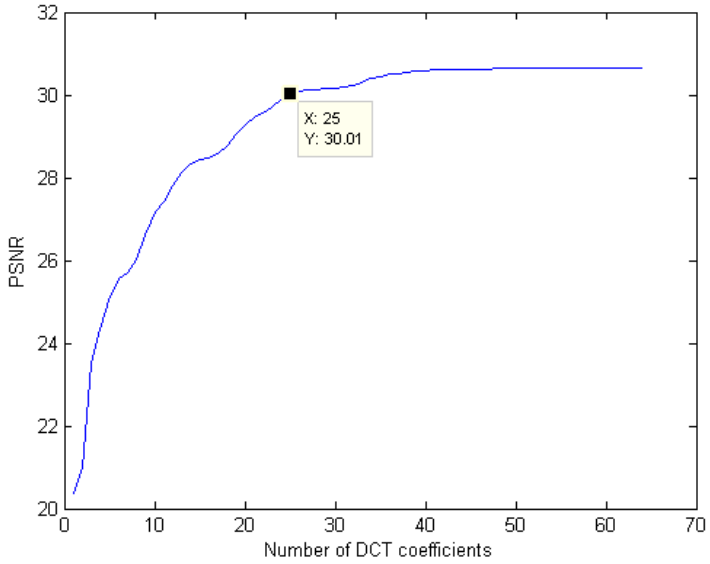

Fig. 2: PSNR for varying number of DCT coefficients

As expected, the PSNR increased rapidly with an increasing number of DCT coefficients. The PSNR approached a PSNR of just over 30 when all DCT coefficients were used. Once a PSNR of 30 was reached, there was very little increase in the PSNR when including more DCT coefficients. As shown in Figure 2, the PSNR is below 30 for fewer than 25 DCT coefficients.

A crude estimate of the compression ratio was determined by the ratio of non-zero elements in the encoded luminance component and the non-zero elements in the original frames. For comparison, this was repeated for the decoded video.

CR1 = nnz(encoded I-frames)/nnz(original frames)
CR2 = nnz(encoded I-frames)/nnz(decoded frames)

Figure 3 shows a similar trend in the compression ratio with respect to the number of DCT coefficients as that found for the PSNR. As the number of coefficients used increases, so does the compression ratio. This again begins to plateau around 25 coefficients. These trends clearly demonstrate the trade-off between compression vs. quality of the decoded video.

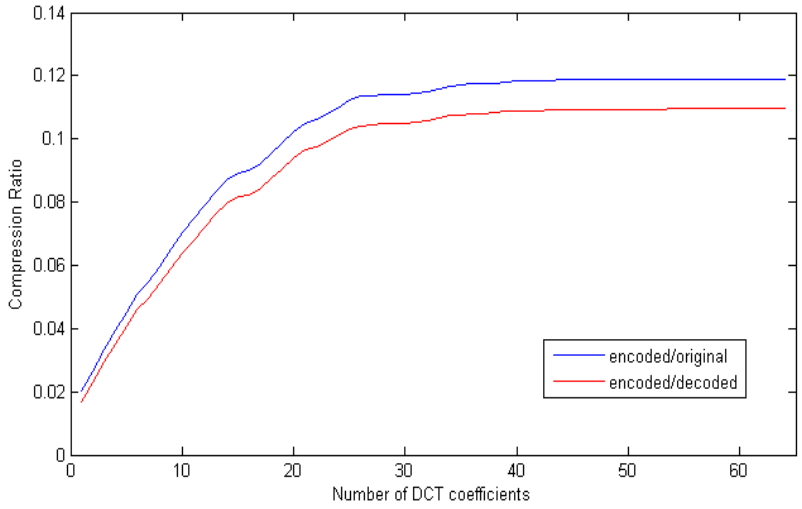

Fig. 3: Compression Ratio for varying number of DCT coefficients

Quantization

The standard quantization matrix was used with a scaling factor ranging from 2^{-4} to 2^{4} with the exponent incremented in steps of 0.25. The other parameters were held as follows:

- DCT coefficients: 64
- Search window size: 8
- Block size: 8
- GOP structure: I-frames (no P or B-frames)

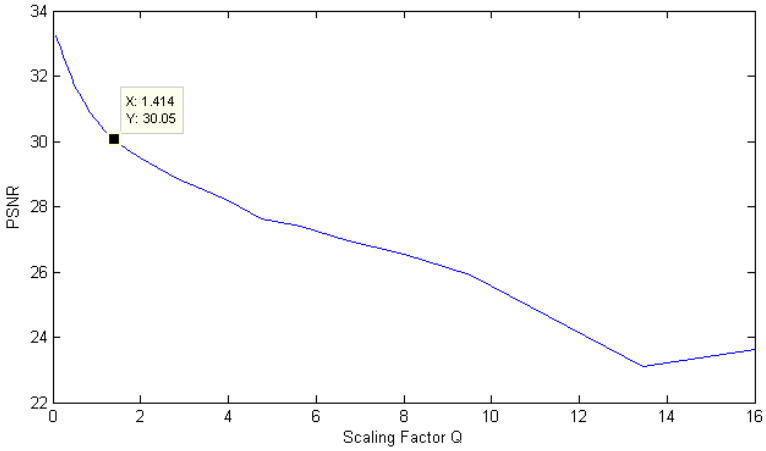

Fig. 4: PSNR for varying scaling factor

As the scaling factor increases, the PSNR decreases. This is expected, as when the scaling factor increases, the quantization performed by rounding each coefficient to its closest integer results in a greater loss of data, and a greater number of zero DCT coefficients. A PSNR above 30 was found for a scaling factor of or below $2^{0.5}$.

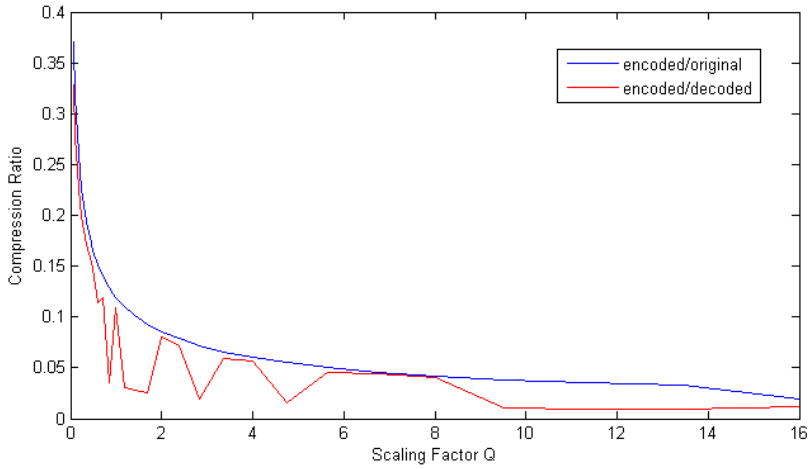

Fig. 5: Compression Ratio for varying scaling factor

The compression ratio again follows the same trend as the PSNR for increasing scaling factor, highlighting the inverse relationship between compression and quality of the decoded video.

GOP Structure

The GOP structure is described by numbering the P and B frames. The number assigned to P frames gave the number of P-frames between two consecutive I frames. The number assigned to B frames gave the number of B-frames between two consecutive I/P frames. The GOP structure was varied to include all combinations of P and B frames ranging from 0 to 4. The other parameters were held as follows:

- Scaling Factor: 1
- DCT coefficients: 25
- Search window size: 8
- Block size: 8

The greatest PSNR was achieved using 0 P-frames and 3 B-frames. In this case, the predicted frames achieved from motion estimation are more accurate than a case including P-frames. Two motion vector matrices determined by the preceding and following I-frames describe each B-frame. Likely, as the bi-directional motion estimation is so effective, the error matrix holds little information, and the information lost from quantization and only storing 25 DCT coefficients holds less significance in reconstructing the frame.

There is a decrease in PSNR when a fourth B-frame is used. It is expected that for a large number of B-frames, the motion estimation will lose accuracy as the number of frames between the target and anchor frames will increase, resulting in greater motion between target and anchor.

Furthermore, to account for a GOP structure that does not entirely describe the 50 frames in the video sequence used, any B-frames without a P/I-frame following it were rather encoded as I-frames. For example, a video sequence of 12 frames encoded with 1 P and 3 B-frames would be encoded as follows: IBBBPBBBIIII. The case with 3 B-frames would result in two consecutive I-frames at the end of the 50-frame sequence. The case with 4 B-frames would result in five consecutive I-frames at the end of the sequence lowering the PSNR achieved in using B-frames.

The addition of P-frames resulted in a lowered PSNR. As the motion estimation for P-frames is uni-directional and not as accurate, more information is held in the error matrices. This in turn results in a greater loss of information during quantization/removal of DCT coefficients. With the addition of P-frames, the motion estimation of B-frames also becomes less accurate as the motion estimation is determined from target P-frames.

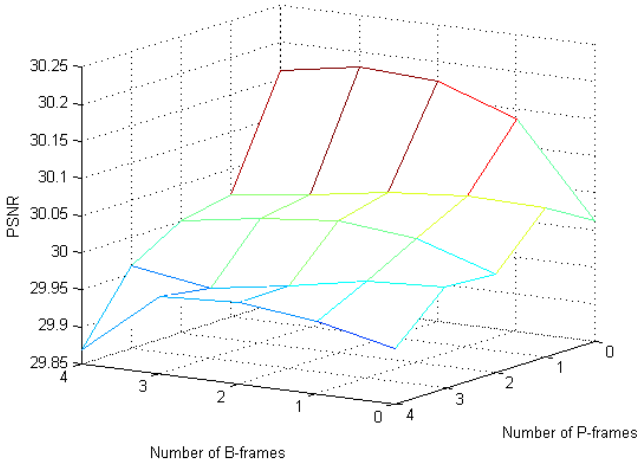

Fig. 6: PSNR for varying GOP structure

A crude estimate of the compression ratio was determined by the ratio of non-zero elements in the encoded luminance component (excluding the motion vector matrices) and the non-zero elements in the original frames. For comparison, this was repeated for the decoded video.

CR = nnz(encoded I-frames + P/B-frame encoded error matrices)/nnz(original frames)

CR2 = nnz(encoded I-frames + P/B-frame encoded error matrices)/nnz(decoded frames)

Unlike in the previous analysis where only I-frames were used, the compression is not inversely proportional to the quality of the decoded video. The case with 0 P-frames and 3 B-frames once again yielded the best result. As the motion vectors were not considered in the compression ratio, it's expected that a GOP structure with the greatest number of B-frames would result in the most compression. Because of the way the GOP structures ending in a B-frame were handled, using 3 B-frames resulted in the 50-frame sequence being encoded with 14 I-frames and 36 B-frames. The same number of I and B-frames resulted when using 4 B-frames; however, the error matrices using 4 B-frames contain more information as there are more B-frames between consecutive I-frames.

As discussed earlier, there is not a distinct trade-off between compression and PSNR in changing the GOP structure. The trade-off is rather in computational efficiency. Figure 9 shows a increase in the time necessary for encoding with the addition of P-frames, and an even greater increase with the addition of B-frames. The decoding time is less significant, as it is drastically smaller than the encoding time, and has a relatively low variance. Decoding times are shown in Figure 10, and follow a

similar pattern as the increasing encoding time. However, when both P and B frames were used, decoding time decreased. Upon analysis, it was observed that the coding scheme for recreating predicted B frames from P frames was more efficiently coded than recreating B frames from I frames. In the former case, the target frames for a group of B frames were decoded, and then the group of B frames were decoded. In the latter case, target frames were decoded for each B frame in turn. As a result, the more P frames present, the more efficient B frame decoding became.

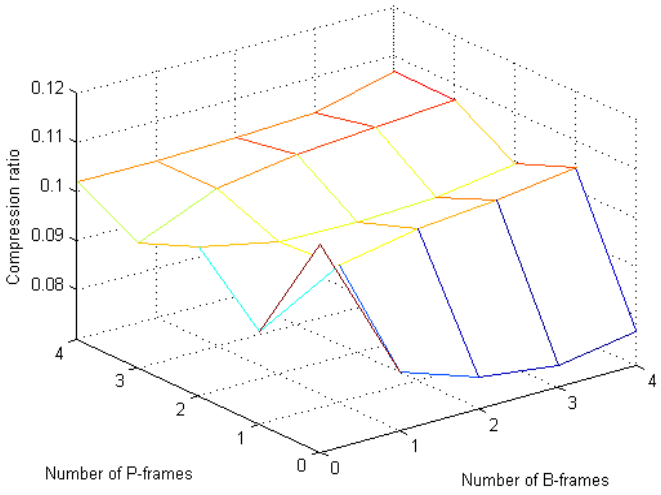

Fig. 7: Compression ratio for varying GOP structure

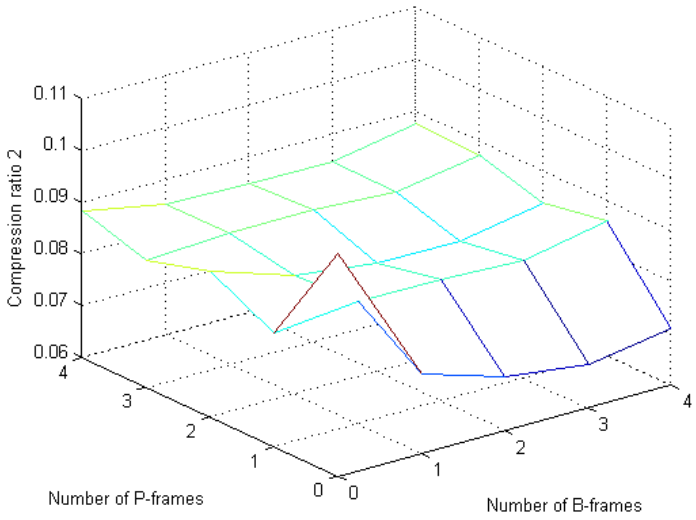

Fig. 8: Compression ratio 2 for varying GOP structure

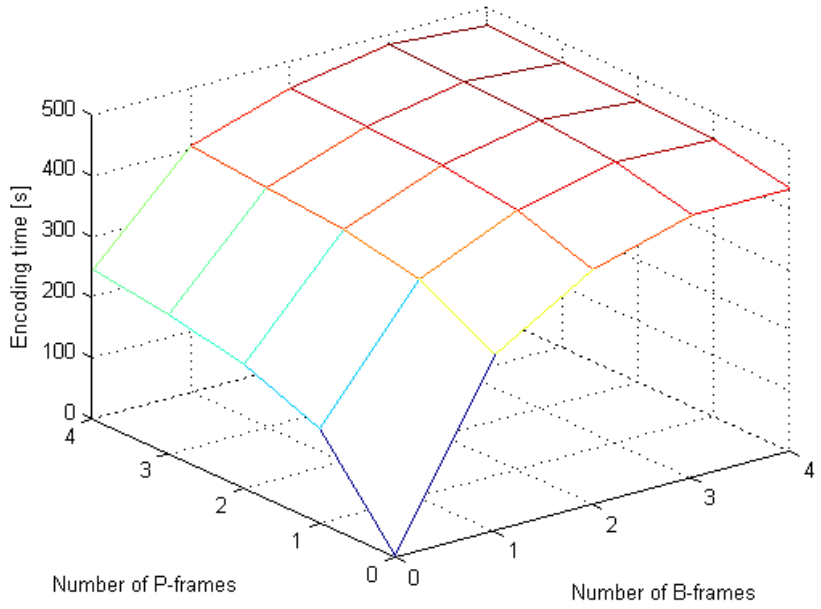

Fig 9: Encoding time for varying GOP structure

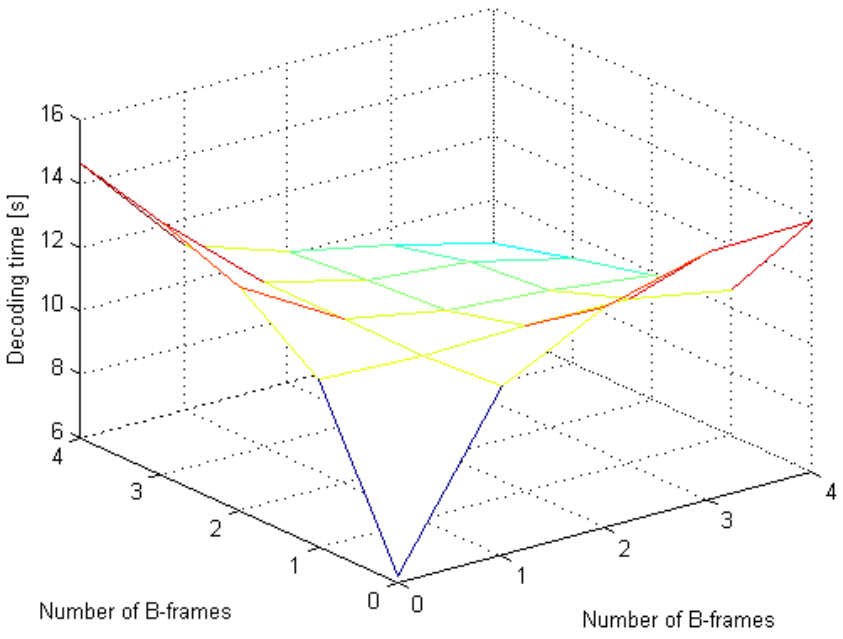

Fig 10: Decoding time for varying GOP structure

Search Window Size

During motion estimation, the search window size determines the range from the current location in the anchor frame to search for the best matching macroblock in the target frame. In this analysis, search window sizes from 0-20 were explored in steps of 4. The following parameters were used:

- DCT coefficients: 25
- Quantization scaling coefficient: 1
- Block size: 8
- GOP structure: 2 P frames, 2 B frames (sequence IBBPBBPBBI)

25 DCT coefficients were used to make the experiment more sensitive to PSNR while maintaining a baseline PSNR above 30 as determined in the previous sections. The GOP structure was selected because, as determined previously, this structure had a relatively short execution time, good compression ratio, and made use of both P and B frames.

PSNR remained relatively constant around 30, likely varying due to minor factors in the DCT or quantization blocks. The encoding time, shown in Figure 11, increased exponentially with increasing search window size. The compression ratio, shown in Figure 12, was found to be a minimum for a search window size of 4, indicating amounts of motion between anchor and target frames of approximately this range. The slightly increasing compression ratio for larger search windows may indicate a transition from a small number of larger errors to a larger number of small errors when the search window was increased. This would result in more non-zero values, while the small difference magnitude of the macroblock may indicate the best match for the anchor block.

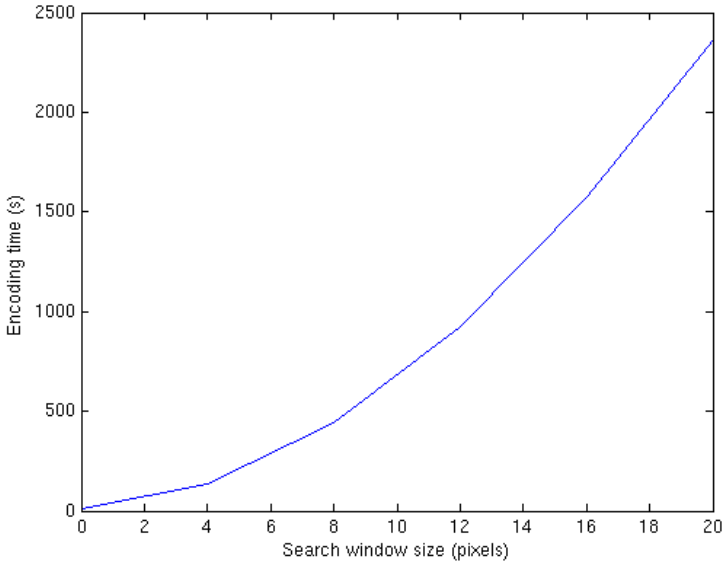

Fig 11: Encoding time for varying search window sizes

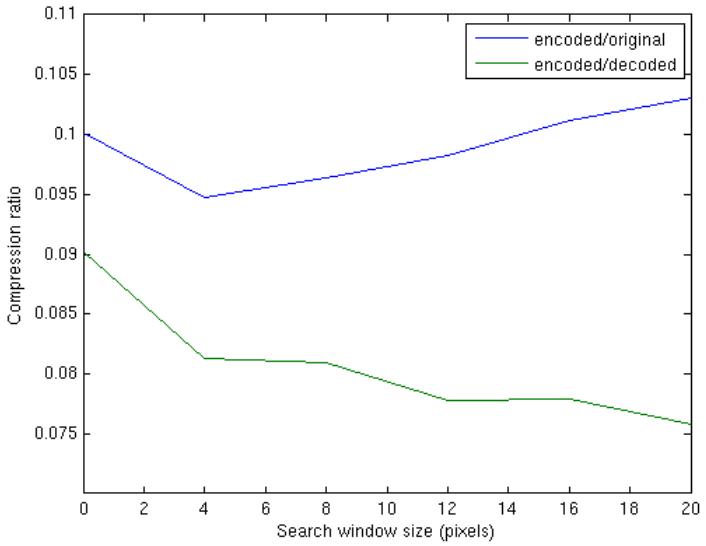

Fig 12: Compression ratio for varying search window sizes

Macroblock Size

During motion estimation, the macroblock size indicates the square blocks each frame is broken into when motion prediction is performed. The frame width and height must be integer multiples of this block size. To analyse the effects of manipulating macroblock size, the block size was increased by powers of 2 from 1 to 16. The other parameters were kept constant as follows:

- DCT coefficients: 25
- Quantization scaling coefficient: 1
- Search window size: 8
- GOP structure: 2 P frames, 2 B frames (sequence IBBPBBPBBI)

With increasing block size, a minor decrease in PSNR was observed. A consistent increase in compression ratio was observed as well as shown in Figure 13. However, it should be noted that this compression ratio was significantly worse when considering the number of motion vectors necessary to reconstruct the image, as illustrated in Figure 14. The encoding time varied significantly with macroblock size as shown in Figure 15.

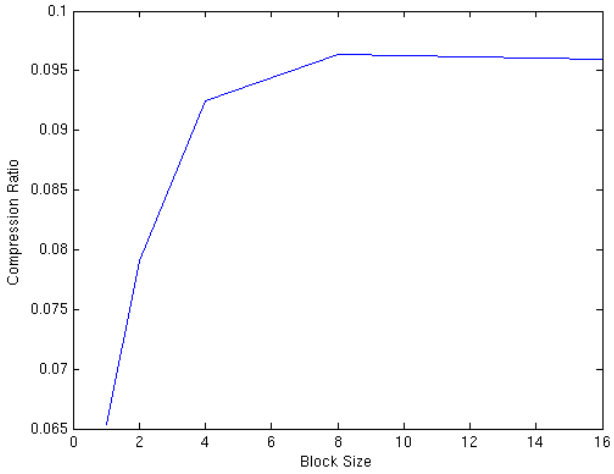

Fig 13: Compression ratio varying with macroblock size

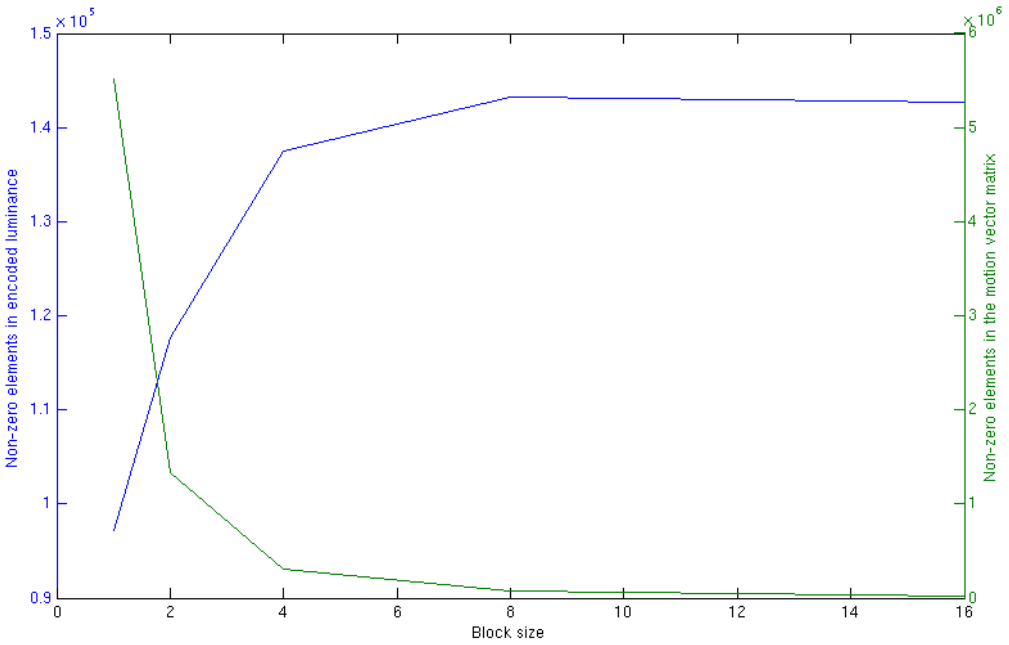

Fig 14: Non-zero encoded matrix components varying with block size

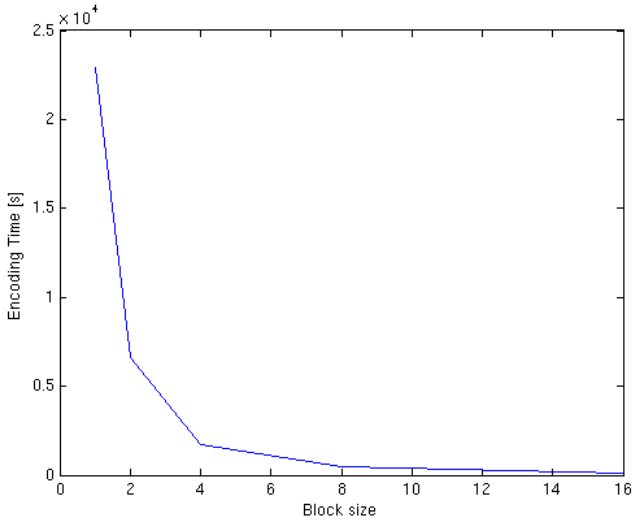

Fig 15: Encoding time with varying macroblock size

CONCLUSION

It was found that decreasing the number of DCT coefficients in the encoded video decreased PSNR and decreased the compression ratio, with fewer than 25 coefficients reducing the PSNR below 30dB. By increasing the scaling factor of the quantization matrix, PSNR quickly dropped off but the compression ratio decreased more slowly above a scaling factor of 1. Changing the number of P or B frames used had limited effect on PSNR. Increasing the number of B frames more noticeably decreased the compression ratio, but the presence of P and B frames drastically increased encoding time. Increasing the motion estimation search window had limited effects on PSNR, but dramatically increased execution time for only a small gain in compression ratio. Decreasing macroblock size used for motion estimation drastically increased computation time, produced a minor decrease in PSNR and a compression ratio which needs consideration of the motion vectors to consider. The weaknesses and strengths of the encoder and decoder have been analysed, with the major weaknesses of the system being the large encoding times. However, with some modification, particularly, the minimization of necessary inverse quantizations and inverse DCTs, this encoding time could be at least partially reduced.

REFERENCES

[1] Y. Wang, J. Ostermann and Y.-Q. Zhang. *Video Processing and Communications*. Upper Saddle River, N.J.:Prentice-Hall, 2002.