Lab 3: Intra- and Inter-Frame Coding

Rami Saad
Section 4
Toronto, Canada
rami.saad@ryerson.ca

***Abstract* - In this lab the effects of Intra- and Inter-Frame coding was observed. More specifically the effects of using 2D block based Discrete Fourier Transforms (DCT) for intra-frame coding and Differential pulse-code modulation (DPCM) for Inter frame coding. When simply using the DCT and IDCT transforms it resulted in a lossless image, but when quantized some jpeg artifacts were observed. Only with mode 2 did DPCM perform better than just using the frame difference.**

DCT; IDCT; DPCM; Quantization; Frames;

I. Introduction

The purpose of this lab was to understand the principles of Intra- and Inter-Frame coding. The effects of using 2D block based Discrete Fourier Transforms (DCT) for intra-frame coding and Differential pulse-code modulation (DPCM) for Inter frame coding. DCT was applied to an image followed by Quantization and then the IDCT or Inverse Discrete Fourier Transform, The Mean squared Error, Signal to noise ratio and the Peak Signal to Noise Ratio were all collected. DPCM modes 1-4 were applied to an sequence of frames from a video and the entropies and histograms were collect and compared to the frame difference.

The lab was primarily conducted using Matlab. Matlab has a suite of built in function such as dct2(), idct2() and round() which make converting to and from spatial to the frequency domain simple. Other in built functions such as imcomplement(), VideoReader() and rbg2gray were also used when handling and operating on image and video data.

II. Theory

In part 1 of this lab, the DCT transform was used to encode an images. The DCT transform converts the image from the spatial domain to the frequency domain. This transform has many advantage when compared to the Discrete Fourier Transform. First, the DCT coefficients are purely real, the transform is near-optimal for energy compaction meaning a larger proportion of signal energy is contained in fewer coefficients, and the computation is more efficient due to faster algorithms [3]. DCT is used in many image compression standards including JPEG, MPEG-1, MPEG-2, and MPEG-4. The DCT (1) and IDCT (2) are provided below.

$$F(u,v) = \frac{2}{\sqrt{MN}} C(u)C(v)[\sum_{x=0}^{M-1}\sum_{x=0}^{N-1} cos\frac{(2x+1)u\pi}{2M} cos\frac{(2y+1)u\pi}{2N} f(x,y)] \quad (1)$$

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} \frac{2}{\sqrt{MN}} C(u)C(v) cos\frac{(2x+1)u\pi}{2M} cos\frac{(2y+1)u\pi}{2N} F(u,v) \quad (2)$$

$$\text{where } C(u), C(v) = \frac{1}{\sqrt{2}} \text{ for u,v} = 0$$
$$C(u), C(v) = 0 \qquad \text{otherwise}$$

The main advantage of using the DCT transform is that file sizes can be reduced with quantization. First the DCT data is zigzagged similar to Figure 1, then quantized using the 8x8 quantization table with quality level 50 in Figure 2. There are different tables that can be used such as those with quality levels of 10 and 90, which would either reduce or increase image quality respectively [4].

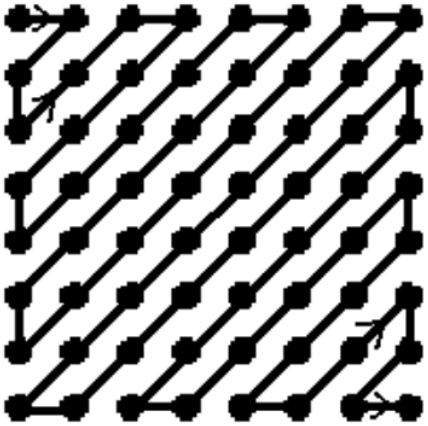


Figure 1: 2D ZigZag Conversion [3]

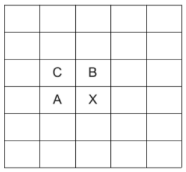
$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

Figure 2: Quantization Table, quality level 50 [2]

After the data has been quantized all the low frequency (most important) information will be located in the top left corner of the 2D array, while the high frequency data will located in the bottom right corner and will mostly reduced to zero. If the data was zigzagged there will be a long large amount of consecutive zeros which can be noted and then removed reducing the amount of information needed. This is a lossy process, but the more important low frequency data will be kept intact.

In part 2, lossless Inter-Frame encoding was explored. The most simple form of Inter-Frame encoding is to take the difference between two sequential frames from a video. The first frame is considered the reference frame and referred to as the I frame. The second frame is predicted from the reference frame and is called the P frame. P frames will typically result in lower entropies because it is expected that there will not be much difference between frames.

DPCM, Differential pulse-code modulation, more specifically modes one through four were implemented on different frames of a video. DPCM is a form of predictive coding which means there will be a transmission of difference between the current pixel and a previous pixel(s) of an image. This can be observed in the figure 3 diagram. DPCM was not used on pixels of the same image like it would be in image compression, for this lab the difference was calculated using pixels of the previous frame when compared to the current frame.



Prediction Index	Prediction
0	No prediction
1	A
2	B
3	C
4	A + B – C
5	A + ((B – C)/2)
6	B + ((A – C)/2)
7	(A + B)/2

Figure 3: DPCM modes for encoding pixels

III. Methodology

Intra-Frame Coding (DCT)

Subsection (1)

1. A function named dct_function was created which takes in one input argument named img_data and returns 1 output arguments dct_table.
2. This function uses the built in Matlab function dct2() to calculate the discrete cosine transform of an MxM vector of data.
3. The result is stored in the dct_table output variable.

Subsection (2)

1. The dct_function was used on a sample 4x4, 8x8, and 32x32 areas of the image 'cameraman.tiff'.
2. Images were collected using the result of the function imcomplement(log(abs(F(u,v)))).
3. The built in function imcomplement() was used so that larger values would have a darker color.

Subsection (3)

1. For each of the 4x4, 8x8, and 32x32 DCT data the Inverse Discrete Cosine Transform was applied to each of them using the built in Matlab function idct2().
2. The results were then saved to different images.

Subsection (4)

1. For each of the 4x4, 8x8, and 32x32 DCT data from subsection (2) certain high and low frequency data were removed before applying the IDCT transform.
2. The results were then saved to different images.

Subsection (5)

1. A function named DCT_Lossless was created which takes 1 input argument img_data_in and has two output arguments DCT and img_data_out which are DCT of the image and the output image respectively.
2. The function first reads the size of the image and stores it in two variables h and w.
3. Then the two output vectors are initialized with zeros using the size of the input image.
4. The DCT transform is applied to 8x8 blocks of the input image and is stored in the DCT vector. Since the image is of size 256x256 no padding is necessary.
5. The IDCT transform is applied to each of the 8x8 blocks of the DCT vector and the result is stored in the img_data_out output vector.
6. The result were saved to an image.

Subsection (6)

1. A function named DCT_Lossy was created which takes 1 input argument img_data_in and has two output arguments DCT and img_data_out which are DCT of the image and the output image respectively.
2. The function first reads the size of the image and stores it in two variables h and w.
3. Then four vectors, quant_data, unquant_data, DCT, and img_data_out are initialized with zeros using the size of the input image.
4. The DCT transform is applied to 8x8 blocks of the input image and is store in the DCT vector. Since the image is of size 256x256 no padding is necessary.
5. Each of the 8x8 DCT tables are divided by the quality 50 quantization table and then rounded to the nearest whole number, the result of this operation is store in the quant_data vector.
6. The quant_data vector is then multiplied by the quantization table again and the result is stored in the un_quant_data vector.
7. The IDCT transform is applied to each of the 8x8 blocks of the un_quant_data vector and the result is stored in the img_data_out output vector.
8. The result were saved to an image.

Inter-Frame Coding (DPCM)

Subsection (1)

1. The built in Matlab video 'xylophone.mp4' was read into the Matlab workspace using the VideoReader() function.
2. The first 11 frames of the video was converted to black and white using the rgb2gray() function and then stored in the 3D (2 spatial and 1 temporal dimensions) vector named "frames".
3. The difference of the between the X_n and X_{n+1} frames was calculated and stored in the difference vector.
4. The results of the difference between frame 1 and 2 was stored as an image after the imcomplement() function was used to enhance the differences.

Subsection (2)

1. Using the MyEntropy() function created in lab 2, the entropy for each of the 10 differences was computed and the results were stored in a graph.

Subsection (3)

1. The compression ratio of the encoded frames assuming ideal entropy was calculated.

Subsection (4)

1. Modes 1 - 4 of the DPCM of Figure 3 was calculated A, B, and C values were retrieved from the previous frame.
2. The average entropies were calculated for each of the modes so that they can be compared with the frame difference method
3. Histograms of each of the DPCM images were created.

IV. Results

Intra-Frame Coding (DCT)

Figure 4: 4x4 input

Figure 5: 4x4 DCT

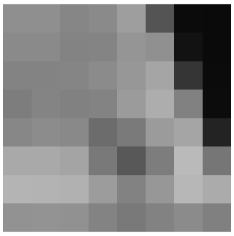


Figure 6: 8x8 input

Figure 7: 8x8 DCT

Figure 8: 32x32 input

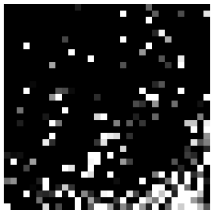


Figure 9: 32x32 DCT

Figure 10: Low Freq Gone

Figure 11: High Freq Gone

Figure 12: 8x8 DCT blocks of input image

Figure 13: Lossless DCT reconstruction

Figure 14: Lossy DCT reconstruction (with Quantization)

Table 1: Quantization Errors

	MSE	SNR	PSNR
Errors	43.5147	26.162	31.744

Figure 15: Sample Frame 1

Figure 16: Sample Frame 2

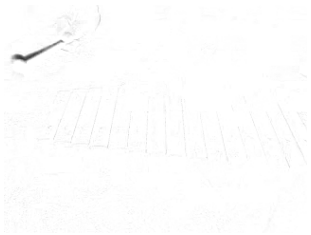


Figure 17: Frame Difference Frames 1 and 2

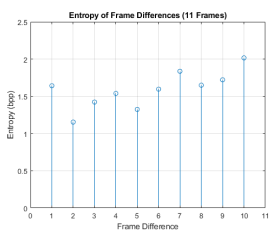


Figure 18: Plot of Entropies of Frames Differences (1-10)

Table 2: Average Entropy of Coding Methods

	Frame Diff.	DPCM 1	DPCM 2	DPCM 3	DPCM 4
Avg. Entropy	1.5901	5.0219	0.0587	3.4197	3.4456
Comp. Ratio	3.6818	1.5115	10.2481	2.0854	2.0727

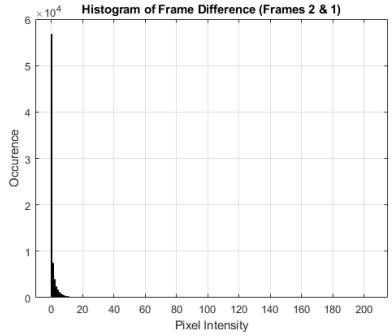


Figure 19: Frame Difference Histogram

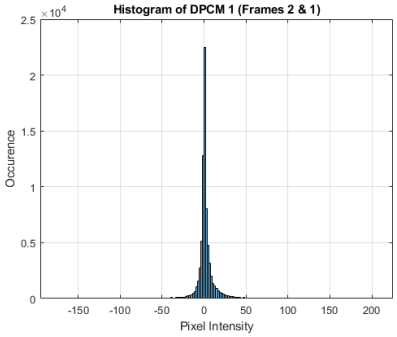


Figure 20: DPCM mode 1 Histogram

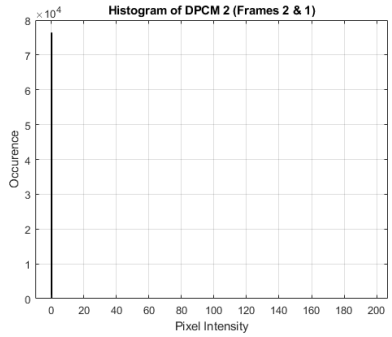


Figure 21: DPCM mode 2 Histogram

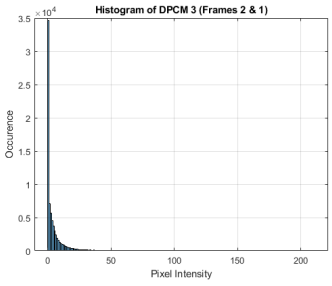


Figure 22: DPCM mode 3 Histogram

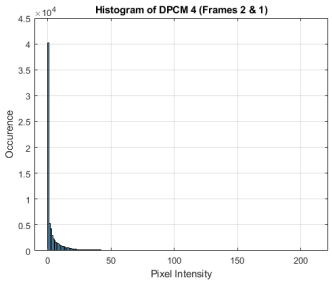


Figure 23: DPCM mode 4 Histogram

V. Discussion

In part one, The benefits of the Discrete Cosine Transform and Quantization were explored. In figures 9 it can be observed that the lower frequency data in the top left corner of the image is more prominent in the 32 x 32 DCT. Furthermore, in figures 10 and 11 when the Lower frequencies of the image is removed more information is lost, but when the higher frequencies are removed a lot of information was still retained. Therefore, the lower frequency information is more important to retain when trying to implement lossy compression.

When trying to quantize the data in between the DCT and IDCT steps, the high frequency information is removed. As explained in the process of dividing a 8x8 DCT block by the quantization table and rounding to the nearest whole number removes much of the higher frequency information. The data in the top left corner of the quantization table is smaller than the data in the bottom left corner, this results in more of the higher frequency data being reduced to zero compared to much of the low frequency data remaining relatively intact. These properties of DCT and Quantization can be observed in figure 14's clear quantized image and table 1's low mean squared error, high signal to noise ratio, and high peak signal to noise ratio.

In part two, 10 frame differences were calculated from 11 sample frames collected from a video. The average entropy of these frame differences was about 1.59 meaning that from the remain 10 frames (reference not included) only 1.59 bits per pixel are needed to encode the image when compared to the 8 bits per pixel needed originally. This result is somewhat expected considering that there is not much movement or change from frame to frame which can be observed in figure 17.

Most of the DPCM modes performed worse than the frame difference calculation. DPCM mode 2 was the exception, however, performing more than 2 times better than the frame difference method. The poor performance of the other modes and excellent performance of the DPCM 2 mode can be explained by average change between the frames. Most of the movement as seen in figure 17 appears to be vertical and the DPCM mode 2 encoding happens to perform best against this type of change while the others do not.

VI. Conclusion

In this lab, the benefits of DCT, Quantization and DPCM encoding were observed. The DCT transform excels at compacting most of the image information in the low frequency data. When combined with the quantization through the use of quantization tables highly accurate lossy reconstructions with small file sizes can be obtained. This is evident with figure 14's good reconstruction.

Using the different DPCM modes were beneficial when trying to encode video data. The DPCM mode 2 provided the best result when compared against all other methods because of the average vertical motion in the frames. Extrapolating further, using the lossy DCT method in part one on the I reference frames of a video in conjunction with using the most appropriate DPCM mode on the P frames could result in a highly compressed, high accurate video encoding format.

References

[1] "Lab 2: Lossless Compression," D2L, 2020.

[2] P. Ze-Nian Li and Mark S. Drew Fundamentals of Multimedia. Boston, MA: Course Technology Cengage Learning, 2014.

[3] "JPEG Compression/Decompression using SystemC." https://www.ee.ryerson.ca/ courses/ coe838/lectures/JPEG-SystemC.pdf. .

[4] K. Cabeen and P. Gent, "Image Compression and the Discrete Cosine Transform," https://www.math.cuhk.edu.hk/ lmlui/dct.pdf. .