Digital Video Coding Schemes

Fundamentals of Telecommunication

Javier Salazar & Andrew Bouasry
Electrical Engineering
University of Texas at Arlington
Arlington, Texas

***Abstract*—This paper describes digital video coding techniques for the project in Fundamentals of Telecommunications. For the project, Pulse Code Modulation (PCM), Differential PCM (DPCM), and Discrete Cosine Transform (DCT) methods are developed and compared against each other given compression ratio, signal to noise ratio, average reduced power and subjective testing.**

I. Introduction

For digital video coding techniques, there are a variety of methods available to compress the information. These methods take advantage of some kind of redundancy either in the spatial domain, frequency domain, or some other domain. Sparse representation means redundancy so the goal is to create a model that has little information but enough to reconstruct the video without high degradation. Some methods take advantage of spatial compression (within the frame space) and/or temporal compression where adjacent frames are considered in the compression scheme. There are two categories for every compression scheme which include lossy and lossless compression methods. In lossless methods, there is no information loss and the reconstructed signal will match the original signal. In lossy, it is impossible to perfectly reconstruct the signal so there will be some degradation. In this paper, three methods of video coding schemes are presented. First, there will be a discussion of pulse code information that will digitize a discrete signal, then the conversation will shift to differential PCM where the "differences" are transmitted as opposed to pixel values. This lossy method can take advantage of both spatial and temporal redundancy to compress the video. Finally, a DCT method is discussed that closely resembles the JPEG algorithm on the video sequence. This lossy method can also take advantage of spatial and temporal redundancy like DPCM.

II. Theory

Pulse Code Modulation

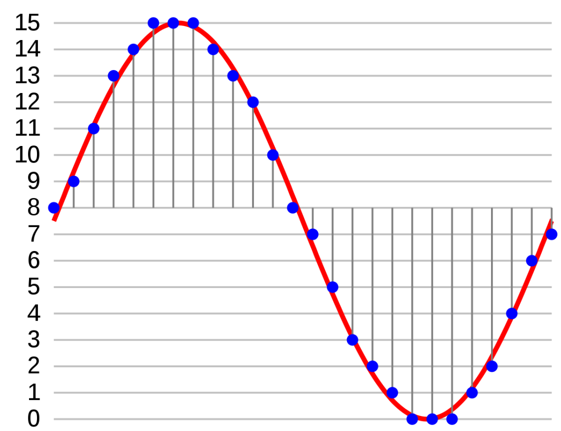

Figure 1. Sampling and quantization of analog signal.. Image courtesy: Aquegg~commonswiki

PCM is a method used to digitally represent analog signals by sampling time and quantizing amplitude as shown in Figure 1. Since video sequences are already digital, sampling is not performed in this project. The main concern is quantization so that an 8-bit signal per pixel can be represented with less bits to conserve bandwidth and transmit information efficiently. There are two ways to perform quantization with PCM and that is uniform quantization and variable quantization. In linear PCM, each range of the 8-bit domain will be mapped evenly to the new lesser bit domain so a uniform process occurs. Variable quantization is different in that levels vary depending on the amplitude such as the u-law algorithm used in North America. The discussion of uniform and variable quantization is illustrated by Figure 2. (a) that demonstrates a uniform mapping while (b) shows more levels for small amplitude values and more quantization for higher values.

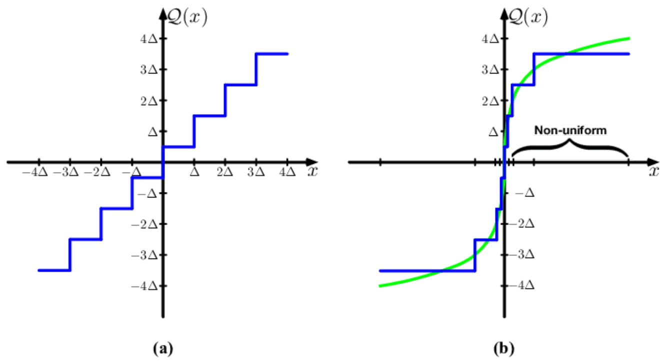

Figure 2. Uniform and variable quantization, respectively. Image courtesy: Biao Sun

The quantization step size can be determined by the following equation: $Q = floor(\frac{X_{max}-X_{min}}{L})$ where L is the total levels that can also be represented by 2^n where n is the number of bits and the X values correspond to the max/min signal values. Given this Q, we can find the index of the quantized value by $Q(X) = round(\frac{X-X_{min}}{Q})$ where X is the signal value and the quantization value is $Q(x) \times Q + \frac{Q}{2} + X_{min}$.

Differential Pulse Code Modulation

Differential PCM is a signal encoder that uses PCM as a base but adds information based on the prediction of future samples given past samples. In essence, the pixel values are not sent but the "differences" among them. It is the difference signal from the video data and the predictor model that is quantized and sent through the transmission path. In doing this, short-term redundancy (pixel window positively correlated) is eliminated, thus a compression ratio of ~2 can be achieved assuming a standard 4-bit quantizer of 8-bit PCM data. In Figure 3 below, the block diagram is shown for the creation of the DPCM signal.

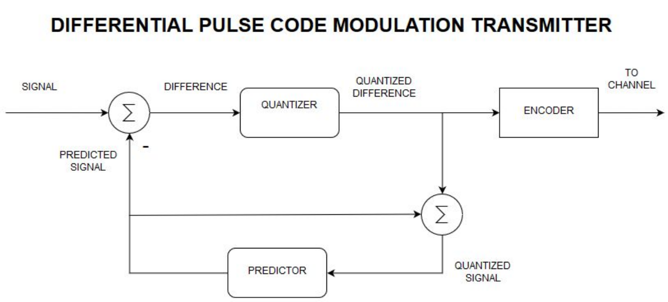

Figure 3. DPCM transmitter block diagram.

Given an input signal $X[n]$, the goal of DPCM is to build a predictor model $X_{hat}[n]$ that most closely resembles the signal $X[n]$. Intuitively, higher order predictor models works better than low order up to a limit. Since adjacent pixels are highly correlated, this fact can be utilized to build the predictor model. As the distance to pixels grow, the correlation, naturally, will decrease so there is a trade-off between computational complexity and predictor quality. For the project, an order 3 predictor model is used which means all adjacent pixels are used within a given frame. To construct $X[n]$ we must intelligently scan the sequences of images to minimize possible error. The method first used is a line scanning method that scans row by row with one rule, move one pixel at a time between rows and frames. Figure 4 illustrates this concept. However, due to technicalities with the following technique, the scanning method is simplified to a row by row method from left to right for each row.

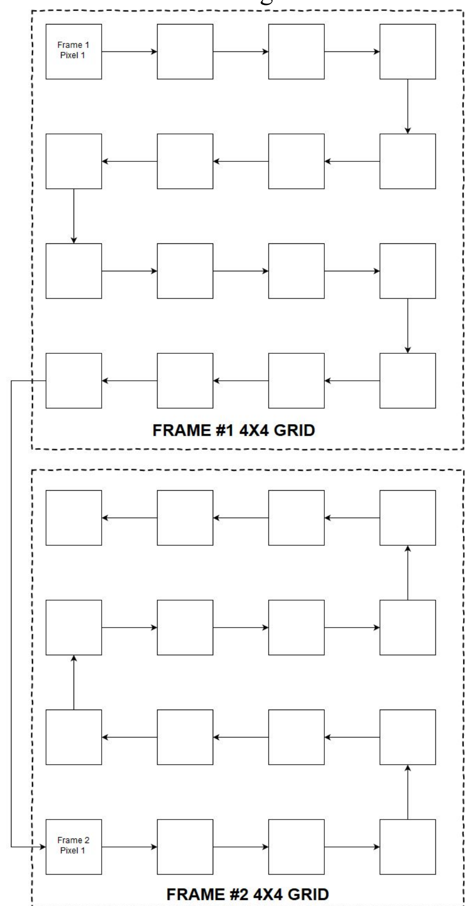

Figure 4. Line scan method.

Given the line scanned signal $X[n]$, we can generate the predictors easily since they are adjacent pixels. Let

$$Y_1 = X[n-1]$$
$$Y_2 = X[n-width]$$
$$Y_3 = X[n-width-1]$$

to represent nearby pixels where width is the horizontal dimension of the video. Thus, for the predictor model, we can use a weighted average representation

$$X_{hat} = aY_1 + bY_2 + cY_3$$

The task of finding the predictor coefficients is the result of minimizing the expected value between the signal $X[n]$ and the model $X_{hat}[n]$ assuming we treat these signals as random signals which is a fair assumption given the physical phenomena involved in pictures. Using probabilistic concepts, the problem can be posed as the following:

$$E[(X - X_{hat})^2] = E[(X - aY_1 - bY_2 - cY_3)^2] = 0$$

With some minimization calculus, taking derivatives with respect to a, b, and c will yield equations where the solutions of a, b, and c have the minimum cost. This can be written as

$$\begin{aligned} aE[Y_1^2] + bE[Y_1Y_2] + cE[Y_1Y_3] &= E[XY_1] \\ aE[Y_1Y_2] + bE[Y_2^2] + cE[Y_2Y_3] &= E[XY_2] \\ aE[Y_1Y_3] + bE[Y_2Y_3] + cE[Y_3^2] &= E[XY_3] \end{aligned}$$

While this is correct, the observation can be made that all Y predictors are shifted versions of X so this can be further simplified by the substitution

$$E[Y_1^2] = E[Y_2^2] = E[Y_3^2] = E[X^2]$$

Linear algebra is utilized for this well-conditioned matrix to solve for the predictor coefficients from the linear equations. The method used is Cramer's rule of determinant ratios. Once the coefficients are known from a given video sequence, X_{hat} can be reconstructed and the difference signal can be produced and denoted as $e[n] = X[n] - X_{hat}[n]$. Using the uniform quantization technique mentioned in the section above, the difference signal can be quantized such that $e[n] = e_q[n] + q[n]$ where $e_q[n]$ represents the quantized signal and $q[n]$ represents the quantization noise introduced. The quantized signal can then be put through encoder techniques and sent to the channel. Further discussion on encoder techniques is included in the DCT section below.

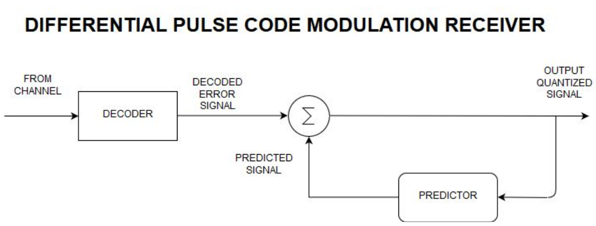

Figure 5. DPCM receiver block diagram.

In the figure above, the DPCM receiver is shown. After the signal is decoded, it will go through the predictor model given that the predictor coefficients are sent. The predictor signal X_{hat} is reconstructed from the difference samples and the difference is added, at the same time, with the predictor signal to get $X_q[n]$ which is the quantized signal X.

However, this DPCM method does not take full advantage of spatial redundancy of the video. While there is strong correlation between adjacent pixels, pixels of previous frames should be considered since frames more or less stay the same between adjacent frames. Given this new viewpoint, a new predictor model X_{hat} is proposed of order 6. This interframe model consists of the three previous predictors plus the new following predictors:

$$Y_4 = X[n - frameWidth \times frameHeight]$$

$$Y_5 = X[n - frameWidth \times frameHeight - 1]$$

$$Y_6 = X[n - frameWidth \times frameHeight - frameWidth]$$

In essence, these predictors are best summarized by the picture shown in Figure 6 below.

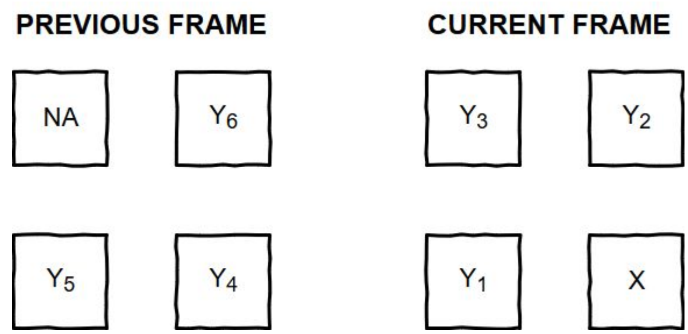

Figure 6. DPCM Interframe predictor model.

Given this model, the same procedure is done to generate solutions for $X_{hat} = aY_1 + bY_2 + cY_3 + dY_4 + eY_5 + fY_6$.

$$\begin{aligned} aE[Y_1^2] + bE[Y_1Y_2] + cE[Y_1Y_3] + dE[Y_1Y_4] + eE[Y_1Y_5] + fE[Y_1Y_6] &= E[XY_1] \\ aE[Y_1Y_2] + bE[Y_2^2] + cE[Y_2Y_3] + dE[Y_2Y_4] + eE[Y_2Y_5] + fE[Y_2Y_6] &= E[XY_2] \\ aE[Y_1Y_3] + bE[Y_3Y_2] + cE[Y_3^2] + dE[Y_3Y_4] + eE[Y_3Y_5] + fE[Y_3Y_6] &= E[XY_3] \\ aE[Y_1Y_4] + bE[Y_4Y_2] + cE[Y_4Y_3] + dE[Y_4^2] + eE[Y_4Y_5] + fE[Y_4Y_6] &= E[XY_4] \\ aE[Y_1Y_5] + bE[Y_5Y_2] + cE[Y_5Y_3] + dE[Y_5Y_4] + eE[Y_5^2] + fE[Y_5Y_6] &= E[XY_5] \\ aE[Y_1Y_6] + bE[Y_6Y_2] + cE[Y_6Y_3] + dE[Y_6Y_4] + eE[Y_6Y_5] + fE[Y_6^2] &= E[XY_6] \end{aligned}$$

Likewise, we have the following relationship that simplifies the system further:

$$E[Y_1^2] = E[Y_2^2] = E[Y_3^2] = E[X^2] = E[Y_4^2] = E[Y_5^2] = E[Y_6^2]$$

The advantage of this interframe DPCM is the higher quality predictor of higher order. While this seems more computationally expensive at the price of higher quality, it is important to note that the computation cost is mostly based on the transmission section of the compression scheme as the predictor coefficients are determined. On the receiver side, there is a marginally higher cost of computation due to only adding three more terms, that is, complexity 6x as opposed to 3x where x is the total pixel samples in the video sequence.

Discrete Cosine Transform Based Method

Much like DPCM, a DCT based compression method takes advantage of spatial redundancy but the main difference is the frequency analysis performed on the spatial components of the video sequence. For any signal that is linear and time invariant, the Fourier basis is a natural basis space to represent that signal. Images and videos fall under this domain since we can use shifted deltas with different coefficients to represent a certain sample in time. Given this relationship, we can apply

Fourier analysis. Moreover, images and videos contain real data only. These are not complex signals so there is no need to apply general Fast Fourier Transform methods since the real data will be mapped to the complex world. Hence, Discrete Cosine Transform can be used to map the real data to real data as this is a better approach. Naively, we may consider to apply DCT on the entire video sequence. It does not take much thought to consider the number of DCT coefficients needed to accurately represent a big signal. Clearly, applying DCT on a whole signal full of information will result in a DCT signal full of information. Naturally, this thinking leads to a sliding window or block based DCT compression method to capture a "goldilocks" amount of information. Consider an intraframe DCT based compression scheme. This means blocks are considered on an image to image basis. Historically, 8x8 blocks have been used in the JPEG standard due to subjective testing and computational cost (DCT complexity N^2). This is the block size used in the compression project. Consider a larger block size say 32x32, given an image of the sky with some clouds, there is a higher probability of the block containing both the cloud and the sky. This is not a good deal since there are edges in the block so there are high frequency components that will be quantized, thus leading to low image quality. Consider the inverse situation of a small block size say 4x4. There is very little information contained in this block. Furthermore, there is no point in frequency analysis since quantizing the frequency components will still lead to the same amount of information contained in the block itself. The block must be large enough so that some high order coefficients become negligible. The block size must remain on orders of 2 due to fast DCT techniques utilized that are radix-2 algorithms so 8x8 remains as a good compromise.

After the block size is selected, forward DCT is performed to get the frequency components of the block. The DCT2 equation used is the following:
$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} cos(\frac{\pi(2m+1)p}{2M}) cos(\frac{\pi(2n+1)q}{2N})$ where p and q are between 0 and M-1/N-1 and where $\alpha_p = \frac{1}{\sqrt{M}}\ p = 0\ o.w. \sqrt{\frac{2}{M}}\ for\ 1 \le p \le M-1$ and likewise for α_q. Once fast DCT is utilized on a specific block, the normalization matrix shown below in Figure 7 is used as a variable quantizer to the DCT coefficients. In Lathi's communication textbook, the author uses the L2 norm of the matrix to normalize the coefficients uniformly. This is a simple and straightforward approach to quantize the frequency information but makes the unfair assumption that all frequencies are perceived the same. In the quantization matrix shown below, we see that each frequency is quantized differently. The low frequency coefficients at the top left are quantized less than the high frequency coefficients at the bottom right. The quantization matrix used is the standard (Q = 50) matrix used in the JPEG algorithm that was derived through subjective testing on how human eyes interpret different frequency distortions.

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

Figure 7. JPEG normalization matrix (Quality = 50). Image Courtesy: FelixH~commonswiki.

Since the image lies between 0 and 255, the frequency coefficients are shifted away from 0 so the image is normalized from -128 to 127 in order to get frequency coefficients shifted towards zero. Each entry in the DCT block is divided by the corresponding quantization value according to the following equation:

$$Quantized\ Block = floor((block + NM/2)./NM)$$

where block is the DCT unnormalized block, NM is the quantization matrix multiplied by a scale factor to adjust quantization, and floor function that rounds to lower end of integers. Since the top left of the block contains the most significant information and the bottom right contains mostly zeros, we can perform a zig zag scanning method to convert all blocks into a one dimensional signal where the trailing end is sparse.

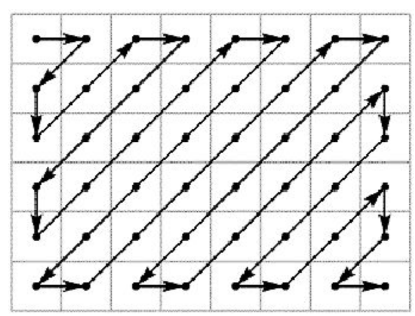

Figure 7. Zig-Zag scanning method. Image Courtesy: Laurent Duval.

Each block is scanned as shown in Figure 7 and interwoven in such a way that the first value from each zigzag signal is put into a bigger signal before moving on to next value in the signal. In the end, each one dimension zigzag signal that represents each frame is interwoven using the same idea to create one complete signal that contains the frequency components of all blocks in the video. The advantage of incorporating this idea is that the overall structure of the zigzag signal is preserved. In other words, the final video signal appears similar to the zigzag signal for one block. The most important information is sent first and becomes sparser towards the end of the signal. This signal is shortened by removing the trailing zeros from the zigzag scanned blocks. By having this sparse information at the end, we will use Huffman coding and run-length coding to compress the signal further for transmission.

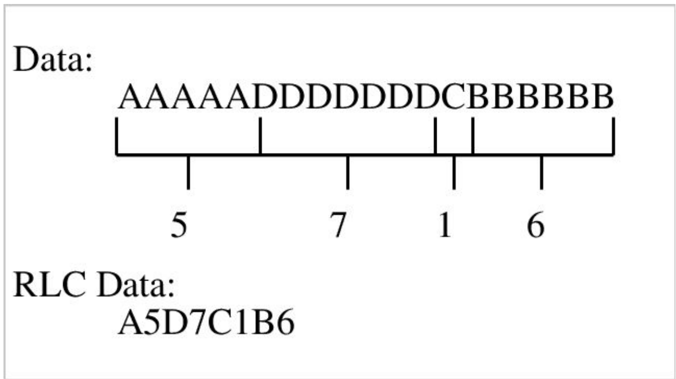

Figure 8. Run-length coding example. Image Courtesy: Jolon Faichney.

As shown in Figure 8, the one dimensional DCT signal is compressed to the idea known as Run-Length coding. Here, we take advantage of repeating information by making a substitution where we write the number of times that a symbol repeats before switching to another symbol. In this way, the signal is compressed without loss of information. However, this signal can be compressed further by analyzing the entropy of the signal itself. Data can be losslessly compressed by analyzing the frequency of different values and assigning shorter encodings to the most common values. For example, the most common value in a given image could be the value 55 and the value 0 may have low frequency. Here, it makes more sense to encode 55 into a 0 since this will require less bits to send in order for the receiver to reconstruct the image. From this method, we create the encoded source signal and send the codebook to the receiver so that the receiver knows how to decode based on the entropy of the signal.

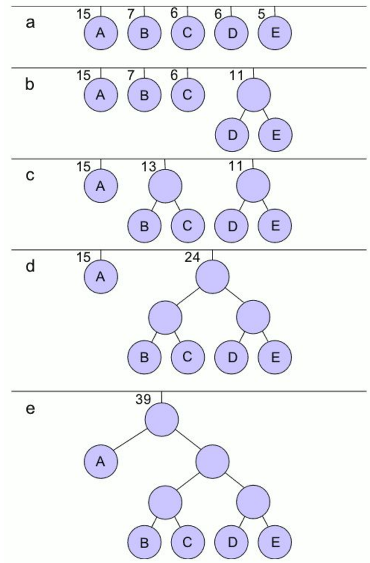

Figure 9. Huffman coding example. Image Courtesy: Andreas Roever.

As illustrated in Figure 9, we can see how a signal with 5 different values can be encoded. We notice that A is the highest frequency symbol and decreases from here. We can see from section a to section e how the encoding is performed. In the tree branch, it is clear that from the beginning, A requires the least branches to get to where as the other symbols require more bits in order to represent them. This encoding scheme is performed on the run-length coding signal to get the final transmitted signal. The entire process of the DCT transmission is shown in the figure below.

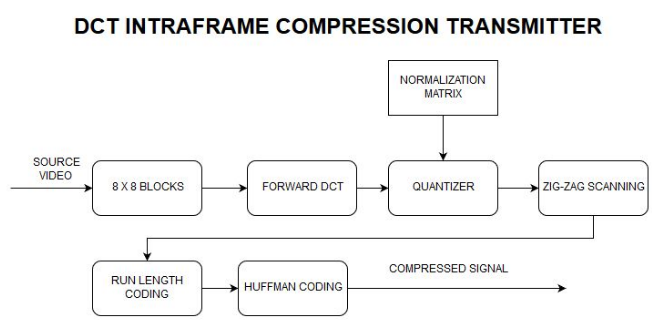

Figure 10. DCT transmitter block diagram.

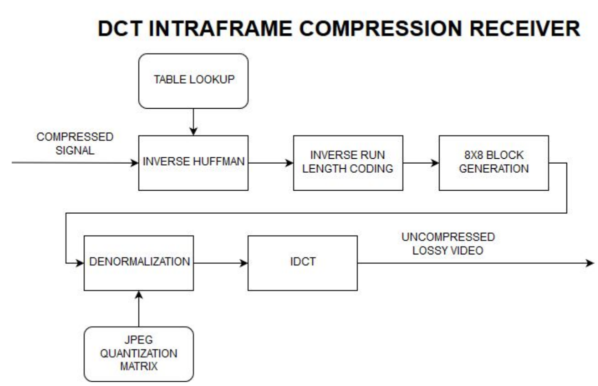

Figure 11. DCT receiver block diagram.

Illustrated in Figure 11, the receiver incorporates inverse huffman, inverse run length coding, creation of 8x8 blocks, denormalization using the same quantization matrix, and finally inverse DCT to get the new video. For the entire project, the flow chart shown below incorporates all of the functions used to generate results along with the respective flowcharts within the functions.

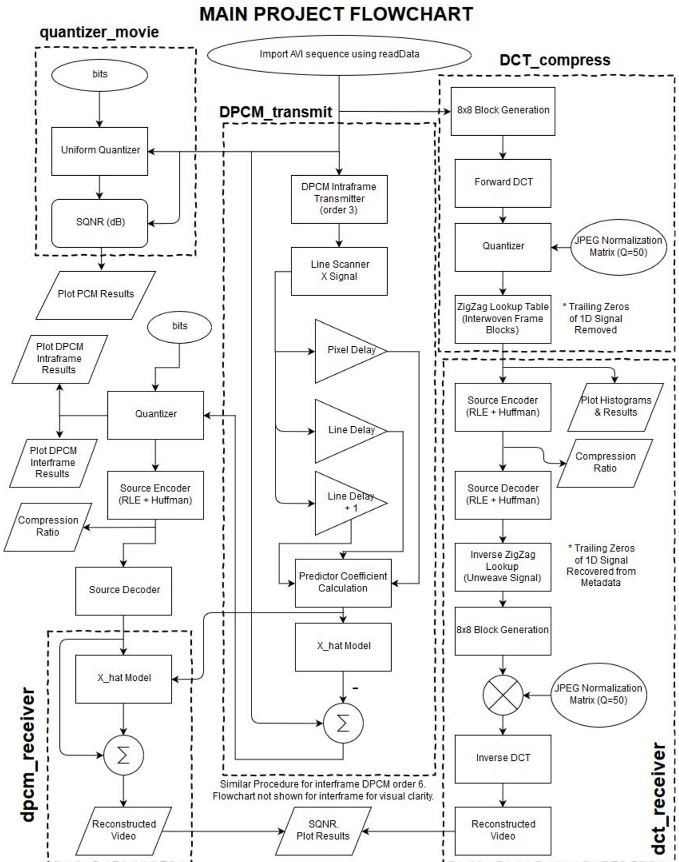

Figure 12. Main project flowchart.

III. Results

Figure 13. Video sequence first frame.

Figure 14. Pulse code modulated frame (4-Bit).

Figure 15A. Intraframe DPCM image with 4-bit quantizer.

Figure 15B. Interframe DPCM image with 4-bit quantizer.

Figure 16. Intraframe DPCM image with 3-bit quantizer..

Figure 17. Intraframe DPCM image with 5-bit quantizer.

Figure 18. DCT variable quantizer image (scale = 1).

Figure 19. DCT variable quantizer image (scale = 0.1)

Figure 20. DCT variable quantizer image (scale = 10).

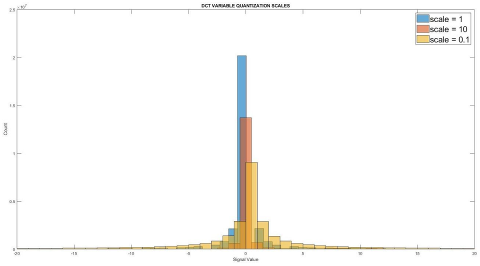

Figure 21. DCT Scale histogram comparison.

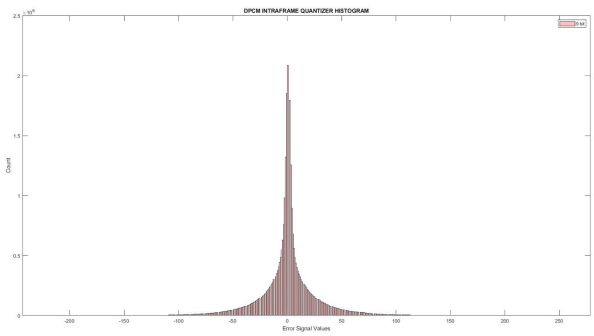

Figure 22. DPCM Intraframe error histogram without quantization.

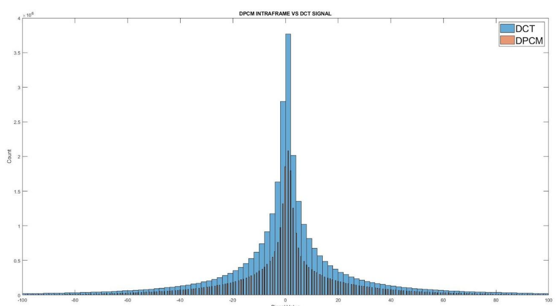

Figure 23. DPCM Intraframe vs DCT signal histogram (no quantization).

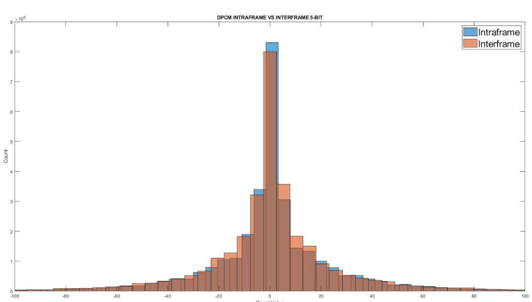

Figure 24. DPCM Interframe vs Intraframe 5-bit histogram.

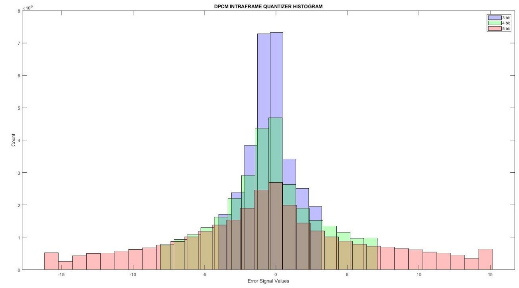

Figure 25. DPCM Intraframe error signal histograms for different quantizer cases.

TABLE I

Digital Video Schemes Results				
Technique	SNR	Avg. Power	Visual Test	Comp. Ratio
PCM 4bit	30.30	4.90	2	2
DPCM Intra 4bit	10.74	6.14	3	2
DPCM Inter 4bit	8.73	6.30	4	2
DCT Intra (Q=50)	24.30	10.24	1	4.15

IV. Discussion

From Figure 14, we can see how PCM will convert the 8-bit PCM image from Figure 13 to 4-bit PCM. Upon close inspection near the ball, we can see more defined contours as opposed to Figure 13. This approximation to the 8-bit image is seen analytically from the 4-bit PCM SNR that is 30.30 from Table 1. While PCM contains a high SNR, PCM by itself is not a good compression technique since it places uniform weight to the entire interval of the signal i.e. assumes a uniform distribution. Given a particular image or video sequence, due to physical phenomena, will tend to have more of a Gaussian distribution rather than uniform. Much like applying Mu-law for telephone systems to put greater weight to smaller sounds, a variable quantization scheme that takes advantage of the underlying image histogram distribution will work better than a uniform quantizer. Moreover, the average power is 4.90, performed second best through subjective testing, and since the signal changed from 8-bit to 4-bits, there is a 2:1 compression ratio.

Using the model outlined in the theory section, we can generate an error signal as shown in Figure 22. Looking at

Figure 15A, we can see a blurrier reconstructed image using DPCM due to a smaller bit quantizer. If a 5-bit quantizer is used with DPCM, then we get the results shown in Figure 17. The results look decent for 5-bit quantization but as we go to 4-bit and 3-bit DPCM quantization, then we notice slope overload at the edges of objects and overall blurriness of image quality. The 3-bit DPCM quantizer is shown in Figure 16. The SNR for the DPCM 4-bit quantizer is 10.74. The reason why the ratio may be so low is dealing with the fact that the first column and first row for each image in the video sequence is 0 as a result of the line scanning method chosen and the predictor model. Therefore, the SNR for 4-bit DPCM is higher than the recorded value. By observing Figure 15, we can see the black bars in the left and top border of the image with some gradual fading as the image gets closer to the center so this is what was discussed. Furthermore, the average power is 6.14, it performed third best on the subjective test and achieved a compression ratio of 2:1. For interframe coding as illustrated in Figure 15B, we got similar performance values as intra frame coding. The reason why the order 6 model performed the same as the order 3 models deals with the motion of the video. The calendar is moving up and down along the wall, the train and ball are moving, the camera is panning both horizontally and vertically while zooming out. This complex motion means that given a current frame, it is difficult to predict the next frame since there is a high amount of movement. In Figure 24, this dilemma is clearly shown by comparing the transmission signals between interframe and intraframe coding.

For the DCT scheme, the reconstructed image is shown in Figure 18 assuming a scale factor of 1 meaning the normalization matrix is the default Q = 50 used. In Figure 19 and 20, these DCT reconstructed images are generated using different scale factors to see how uniformly scaling the normalization matrix will affect the quality of the images with scaling 0.1 and 10, respectively. In figure 20, there is a visibly significant amount of distortion while Figure 18 and 19 have great reconstruction quality. In Figure 21, the histograms of the DCT coefficients are plotted for all three cases to illustrate how the range of the histogram will decrease as the scale factor increases thus indicating higher quantization. The SNR for the DCT method outlines is 24.30 which is below the 4-bit PCM method. Clearly, this shows a problem within the DCT algorithm outlined. Most likely, there is some issues with how the ZigZag one dimensional signal is being generated for the entire video sequence that is the interwoven signal of all blocks in the video. This is where the DCT algorithm outlined deviates from the well-known JPEG algorithm. Moreover, the DCT method performed first in the subjective test and achieved a compression ratio of 4.15 while having average power of 10.24. Normally, with JPEG compression ratio, we expect to see 8:1 or 10:1 compression. In an old 1080p video used for this project, we were able to achieve 8:1 compression ratio but were unable to get the same ratio for this particular, smaller video file. In Figure 23, we have the essentially unquantized (9-bit) DPCM error signal compared against the unquantized DCT coefficient signal. From visual inspection, DCT has more counts near 0 in comparison to DPCM. Therefore, without doing any kind of quantization, we can expect DCT to perform much better than DPCM since it has a better model.

For the different video compression techniques the average power of the error of each technique seemed to increase from 4-bit PCM to 4-bit intraframe DPCM to 4-bit interframe DPCM and lastly at DCT. Since the average power increases then the error between the original samples and the predicted values have a greater gap between them. This shows us that compression took place as opposed to the error being small and little to no compression occurring. Looking at table I, the average power is increases between PCM and DPCM intraframe but both still have a 2:1 compression ratio. There is then a slight increase in average power between the intraframe DPCM and interframe DPCM with interframe DPCM still having 2:1 compression ratio. The DCT average power is increased relatively more when compared to the other methods and has a compression ratio greater than 4. The analysis that the average power of the error increasing correlating to more compression is observed through the increasing powers and the compression ratio of DCT being greater than the others.

V. Conclusion

It was found that redundancy of pixel information within the spatial domain and temporal domain was useful in digital video compression through the use of various digital video compression techniques. The different video compression techniques (PCM, DPCM Intraframe, DPCM Interframe, and DCT) can approximate the original signal into a similar signal but of lesser quality. Comparisons were made between the different compression techniques with code simulated on quantized and non-quantized signals. From the quantized signals, the video was reconstructed with less information than what was originally sent between transmitters and their respective receivers. The different modulation and transform techniques proved to be able to compress a video signal for efficient data transmission and reconstruct a lossy version of the video.

VI. References

[1] B. P. Lathi, *Modern Digital and Analog Communication Systems.* (4th ed.) New York: Oxford University Press, 2009.

[2] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard.* New York: Van Nostrand Reinhold, 1993.

VII. APPENDIX

See subsequent pages that show enlarged figures from this paper.