Video Encoding Techniques

Aaron Leyesa
Christopher Mendez
Matthew Mendez
Mukund Manikarnike
Pawan Mahalle

Multimedia Information Systems
CSE 408/598
Fall 2015

school of computing, informatics, & decision systems engineering

ASU IRA A. FULTON SCHOOLS OF engineering

Contents

Tables and Figures

Abstract

This project aims at implementing all the video processing tasks that are regularly followed. Given, a video the common video processing tasks involve, predictive coding, quantization, compression. Given a compressed file, the original is obtained by following the reverse of the above mentioned processes. Coding and quantization are carried out because better compression can be achieved if the histogram of the data that is being carried out becomes peakier. The implementation reads the videos as separate signals, applies predictive coding mechanisms, quantization, compression and carries out the reverse of these processes on the finally produced output on a selected small region of the input video. This project illustrates how video processing techniques in short.

Introduction

This project aims at implementing a solution for the following tasks.

1. Temporal Predictive coding
    - This requires the region in the video mentioned to be read as a signal and encoded with a few temporal predictive coding functions which will be detailed in later sections and output the total absolute prediction error in a text file.
2. Spatial Predictive coding
    - This task is similar to task 1 except that the absolute prediction error would be output using Spatial Predictive coding functions.
3. Error Quantization Schemes
    - This requires 2 quantization schemes to be implemented on the outputs from either Task 1 or 2 and output the results into a text file
4. Encoding Quantized values
    - This requires the quantized values obtained from Task 3 to be encoded using different encoding schemes and output into a binary file.
5. A viewer that can decode and display the encoded values
    - This viewer will display the outputs that Task 4 generated.

Assumptions

All of these tasks will be carried out assuming that

1. The input is a video file and a 10 pixel by 10 pixel region.
2. All tasks are going to be carried out on the Y channel of the video

Requirements Specification

This section introduces each of the requirements that are required to be implemented as part of the project.

General Requirements

[REQ_1] The solution shall implement a program that takes a video file and 10 pixel by 10 pixel region as input.

Task I – Temporal Predictive Coding Requirements

[REQ_2] **TPC - 1 -** The solution shall implement a program that performs temporal predictive coding using original values in the specified region for all temporal coding techniques.

[REQ_3] **TPC - 2 -** The solution shall implement a program that shall perform temporal predictive coding using the function

$$s_i[t] = s_i[t-1].$$

[REQ_4] **TPC - 3** - The solution shall implement a program that shall perform temporal predictive coding using the function

$$s_i[t] = \frac{s_i[t-1]+s_i[t-2]}{2}.$$

[REQ_5] **TPC - 4** - The solution shall implement a program that shall perform temporal predictive coding using the function

$$s_i[t] = \alpha_1 \times s_i[t-1] + \alpha_2 \times s_i[t-2]$$

$$\alpha_1 \times s_i[t-2] + \alpha_2 \times s_i[t-3] = s_i[t-1]$$

$$\alpha_1 \times s_i[t-3] + \alpha_2 \times s_i[t-4] = s_i[t-2] \text{ where } \alpha_1 \text{ and } \alpha_2$$

$$\text{are two non-negative values such that } \alpha_1 + \alpha_2 = 1.0$$

In case there are not sufficient observations to compute α_1 and α_2, use $\alpha_1 = \alpha_2 = 0.5$ as default.

[REQ_6] The solution shall implement a program that shall save the outputs of all Temporal coding results in a text file whose name shall take the format
X_Y. tpc where X represents the name of the video file provided as input and Y represents the TPC function chosen ranging from 1 to 4 as stated from [REQ_2] through [REQ_5]

Task II – Spatial Predictive Coding Requirements

[REQ_7] The solution shall implement a program that shall treat each frame in the video as separate 2D signals for all spatial predictive coding techniques.

[REQ_8] **SPC – 1 –** The solution shall implement a program that shall perform spatial predictive coding using original values specified in the region.

[REQ_9] **SPC – 2 –** The solution shall implement a program that shall perform spatial predictive coding using Predictor A

[REQ_10] **SPC – 3 –** The solution shall implement a program that shall perform spatial predictive coding using Predictor B

[REQ_11] **SPC – 4 –** The solution shall implement a program that shall perform spatial predictive coding using Predictor C

[REQ_12] **SPC – 5 –** The solution shall implement a program that shall perform spatial predictive coding using the function

$\alpha_a \times A + \alpha_b \times B + \alpha_c \times C$, where α_1, α_2, and α_3 are three non-negative values such that $\alpha_1 + \alpha_2 + \alpha_3 = 1.0$

In case there are not sufficient observations to compute α_1, α_2, and α_3 use $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$ as default.

[REQ_13] The solution shall implement a program that shall save the outputs of all Spatial coding results in a text file whose name shall take the format

X_Y. spc where X represents the name of the video file provided as input and Y represents the SPC function chosen ranging from 1 to 5 as stated from [REQ_8] through [REQ_12]

Task III – Quantization Requirements

[REQ_14] The solution shall take as input, either the temporally encoded or the spatially encoded outputs as input.

[REQ_15] **PQ – 1** – The solution shall implement a program that shall output the result with no quantization

[REQ_16] **PQ – 2** – The solution shall implement a program that shall output the result with quantization of the error into 2^m uniform bins for a given m ≥ 1

[REQ_17] The solution shall implement a program that shall save the outputs of Quantization in a text file whose name shall take the format
X_Y_Z.tpq or X_Y_Z.spq
Where X represents the name of the video file provided as input, Y represents the TPC or SPC function chosen as detailed in the previous section and Z takes the value 0 for Quantization option 1 and m for Quantization option 2.

Task IV – Encoding Requirements

[REQ_18] The solution shall take as input, the output generated from Quantization.

[REQ_19] **ENC – 1** – The solution shall implement a program that outputs the quantized input as it is without any compression.

[REQ_20] **ENC – 2** – The solution shall implement a program that shall perform variable length encoding with Shannon-Fano coding.

[REQ_21] **ENC – 3 –** The solution shall implement a program that shall perform Dictionary encoding with LZW coding.

- **a.** In this case, the solution shall assume that the dictionary bit length is given as an input.

[REQ_22] **ENC – 4 -** The solution shall implement a program that shall perform Arithmetic coding.

[REQ_23] The solution shall implement a program that shall save the outputs of the encoding in a binary file whose name shall take the format
X_Y_Z_C.tpv or X_Y_Z_C.spv where C is the compression option.

[REQ_24] The solution shall implement a program that shall output

a. The total amount of distortion(signal to noise ratio) between the video and the encoded video
b. The size of the encoded video

Task V – Decoder and Viewer Requirements

[REQ_25] The solution shall implement a program that shall take the outputs of encoding as input
[REQ_26] The solution shall implement a program that shall decode the input read and display it.

Design Choices

This section describes the entire list of design choices that were made for this project.

[DSN_CH_1] **Figure 1** shows the overall architecture of the entire project.

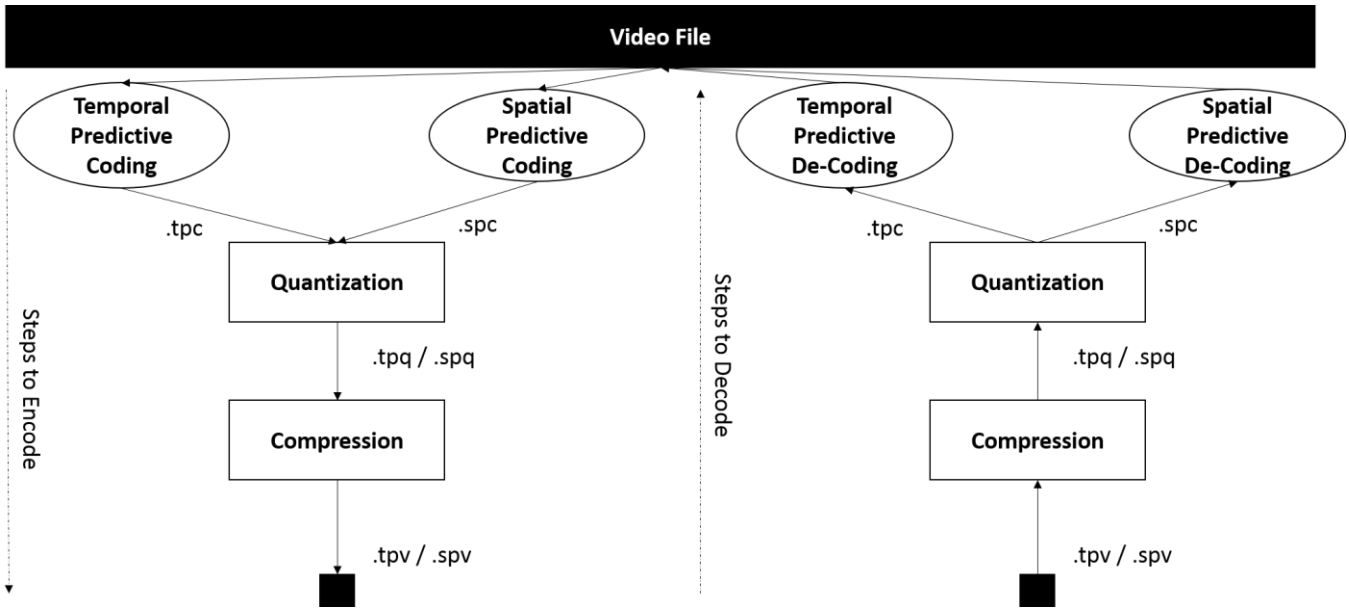

Figure 1 Architecture Overview

[DSN_CH_2] The spatial and temporal encoding functions take the video file as input and output the predictive encoded outputs in a text file with an extension .tpc or .spc depending on whether it is temporal or spatial encoding.

[DSN_CH_3] The predictive coding mechanisms treat the region of interest in the video as a matrix of integers varying with time or frame number and apply the predictive coding mechanism desired.

[DSN_CH_4] The quantization function takes the predictive coded outputs and applies quantization by classifying the predicted outputs into preset bins, the bin size varying from 2 to 256 depending on the input chosen. This ensures that the input having a certain pixel value get quantized to only a few bins as defined by the bin size.

[DSN_CH_5] The compression function takes the quantized output and creates an encoded output using any of the compression algorithms mentioned in the problem. This ensures that the number of bits allocated to each symbol reduces and hence the video is compressed.

[DSN_CH_6] The compressed output is read as part of task 5, decompressed, decoded and rendered as a video. The point to note here is that once quantized, the process cannot be reversed and hence, the output obtained from here isn't going to be same as the input provided.

[DSN_CH_7] At each stage, the interfaces have been defined such that the outputs are written to certain files with agreed upon formats and the inputs are read from the output files from the previous stages. This was done to ensure that mismatches in data structures used don't cause issues at a later stage.

Implementation

This section describes all the implementation strategies that have been used arrive at solutions to the problems mentioned and also the difficulties faced in arriving at solutions. Following are the few implementation stories which were the highlights of this project.

[IMP_ST_1] Following were the tools or libraries used in the implementation of this project

1. Python
2. Third Party Libraries
    i. OpenCV
        1. In order to perform video processing tasks
    ii. Numpy
        1. In order to perform mathematically intensive computations

[IMP_ST_2] Temporal and Spatial coding, read the video and output files with the corresponding predictive coded pixel values as outputs. Each of them store a certain reference value at the start depending on the kind of predictive coding strategy that is chosen in order to enable easy decoding.

[IMP_ST_3] In temporal coding, the main difficulty that we faced was

1. Coming up with a good format for storage of the coded output because this coding requires a relation of pixels across frames to be represented in the output.
    i. This difficulty was overcome by flattening out the entire set of pixels in each frame. This ensured that we could easily correlate pixels of neighboring frames.

[IMP_ST_4] In temporal and spatial coding, the common difficulties that we faced were

1. Solving for the alpha coefficients for predictors 4 and 5 respectively given the constraint that their summation adds up to 1.
    i. This difficulty was overcome by fixing the alphas to the default value

[IMP_ST_5] Quantization was a straight-forward process where given a certain coded output, we knew that it would fall in the range of -255 to 255 and hence, quantization was reduced to classifying these 512 values into a certain set of bins based on the chosen value of ‘m’. Given a certain value of ‘m’, these 512 values were classified into 2^m bins. This was very similar to creating the color map in phase 1.

[IMP_ST_6] The compression task was to implement Shannon Fano encoding, LZW encoding and Arithmetic Coding mechanisms in order to achieve compression.

1. Shannon Fano encoding algorithm was implemented using the standard algorithm published in the textbook[1]
2. LZW encoding was implemented using the standard algorithm published in the textbook[1], but wasn’t working as expected.
3. The other encoding mechanisms weren’t implemented.

[IMP_ST_7] The main difficulty in the compression task was to write the compressed output into a binary file and decoded it by reading from the binary file. An alteration of this was done to write into a text file and decode it by reading from the file.

[IMP_ST_8] The reverse process of all the above mentioned tasks was carried out in a straightforward manner by calling the decompress and decode functions of each of the above mentioned steps. The main difficulty in this task was to actually write the region of interest into a video file. The

problem we faced here was that the Open CV library wouldn't allow a very small region such as a 10 by 10 pixels to be written to the video. The alternatives, we used in this case were

1. Display a video with 100 by 100 pixels
2. Display images with 10 by 10 pixels in a loop. This solution doesn't give the true effect of a video. However, the first approach takes a lot of time for the algorithms to run.

Results

This section describes the results that were obtained from this project

[RES_1] The following few outputs give an illustration of how the outputs look at each stage in the process when no encoding, quantization or compression is done. The expectation from this should be that all intermediate outputs should be the same and that is clearly visible in the outputs shown below. On close observation, one would find that all the outputs below have the same pixel values because there's no coding, quantization or compression applied.

Original set of Pixels

2.original | 2_1.tpc | 2_1_1.tpq | 2_1_1_1.tpv | 2_1_1_1.tpv.decompressed | 2_1_1_1.tpv.decompressed.decoded

1	158	187	189	165	151	137	122	116	122	139	127	158	165	145	146	131	121
2	160	148	142	124	118	120	124	146	172	175	152	145	138	123	124	124	121
3	123	123	123	122	124	124	144	152	149	146	124	131	134	123	120	123	141
4	123	123	129	148	151	146	146	150	149	145	122	120	128	143	143	139	142
5	128	142	141	139	142	146	159	164	165	172	156	163	157	145	143	146	164
6	141	141	143	150	159	164	165	172	172	163	155	144	143	150	164	180	174
7	142	143	152	164	170	171	174	170	158	143	145	146	153	169	178	170	166
8	146	158	167	171	172	173	165	150	139	139	141	152	167	164	160	158	155
9	165	173	177	178	173	165	151	141	141	143	155	165	163	160	158	155	143
10	174	180	181	184	172	162	146	139	141	144	159	165	159	158	157	152	139

Not Predictive Coding

2.original | 2_1.tpc | 2_1_1.tpq | 2_1_1_1.tpv | 2_1_1_1.tpv.decompressed | 2_1_1_1.tpv.decompressed.decoded

1	158	187	189	165	151	137	122	116	122	139	127	158	165	145	146	131	121
2	160	148	142	124	118	120	124	146	172	175	152	145	138	123	124	124	121
3	123	123	123	122	124	124	144	152	149	146	124	131	134	123	120	123	141
4	123	123	129	148	151	146	146	150	149	145	122	120	128	143	143	139	142
5	128	142	141	139	142	146	159	164	165	172	156	163	157	145	143	146	164
6	141	141	143	150	159	164	165	172	172	163	155	144	143	150	164	180	174
7	142	143	152	164	170	171	174	170	158	143	145	146	153	169	178	170	166
8	146	158	167	171	172	173	165	150	139	139	141	152	167	164	160	158	155
9	165	173	177	178	173	165	151	141	141	143	155	165	163	160	158	155	143
10	174	180	181	184	172	162	146	139	141	144	159	165	159	158	157	152	139

No Quantization

2.original | 2_1.tpc | 2_1_1.tpq | 2_1_1_1.tpv | 2_1_1_1.tpv.decompressed | 2_1_1_1.tpv.decompressed.decoded

1	158	187	189	165	151	137	122	116	122	139	127	158	165	145	146	131	121
2	160	148	142	124	118	120	124	146	172	175	152	145	138	123	124	124	121
3	123	123	123	122	124	124	144	152	149	146	124	131	134	123	120	123	141
4	123	123	129	148	151	146	146	150	149	145	122	120	128	143	143	139	142
5	128	142	141	139	142	146	159	164	165	172	156	163	157	145	143	146	164
6	141	141	143	150	159	164	165	172	172	163	155	144	143	150	164	180	174
7	142	143	152	164	170	171	174	170	158	143	145	146	153	169	178	170	166
8	146	158	167	171	172	173	165	150	139	139	141	152	167	164	160	158	155
9	165	173	177	178	173	165	151	141	141	143	155	165	163	160	158	155	143
10	174	180	181	184	172	162	146	139	141	144	159	165	159	158	157	152	139

No Compression

2.original | 2_1.tpc | 2_1_1.tpq | 2_1_1_1.tpv | 2_1_1_1.tpv.decompressed | 2_1_1_1.tpv.decompressed.decoded

1	158	187	189	165	151	137	122	116	122	139	127	158	165	145	146	131	121
2	160	148	142	124	118	120	124	146	172	175	152	145	138	123	124	124	121
3	123	123	123	122	124	124	144	152	149	146	124	131	134	123	120	123	141
4	123	123	129	148	151	146	146	150	149	145	122	120	128	143	143	139	142
5	128	142	141	139	142	146	159	164	165	172	156	163	157	145	143	146	164
6	141	141	143	150	159	164	165	172	172	163	155	144	143	150	164	180	174
7	142	143	152	164	170	171	174	170	158	143	145	146	153	169	178	170	166
8	146	158	167	171	172	173	165	150	139	139	141	152	167	164	160	158	155
9	165	173	177	178	173	165	151	141	141	143	155	165	163	160	158	155	143
10	174	180	181	184	172	162	146	139	141	144	159	165	159	158	157	152	139

Decompressed

2.original | 2_1.tpc | 2_1_1.tpq | 2_1_1_1.tpv | 2_1_1_1.tpv.decompressed | 2_1_1_1.tpv.decompressed.decoded

1	158	187	189	165	151	137	122	116	122	139	127	158	165	145	146	131	121
2	160	148	142	124	118	120	124	146	172	175	152	145	138	123	124	124	121
3	123	123	123	122	124	124	144	152	149	146	124	131	134	123	120	123	141
4	123	123	129	148	151	146	146	150	149	145	122	120	128	143	143	139	142
5	128	142	141	139	142	146	159	164	165	172	156	163	157	145	143	146	164
6	141	141	143	150	159	164	165	172	172	163	155	144	143	150	164	180	174
7	142	143	152	164	170	171	174	170	158	143	145	146	153	169	178	170	166
8	146	158	167	171	172	173	165	150	139	139	141	152	167	164	160	158	155
9	165	173	177	178	173	165	151	141	141	143	155	165	163	160	158	155	143
10	174	180	181	184	172	162	146	139	141	144	159	165	159	158	157	152	139

Decoded

2.original | 2_1.tpc | 2_1_1.tpq | 2_1_1_1.tpv | 2_1_1_1.tpv.decompressed | 2_1_1_1.tpv.decompressed.decoded

1	158	187	189	165	151	137	122	116	122	139	127	158	165	145	146	131	121
2	160	148	142	124	118	120	124	146	172	175	152	145	138	123	124	124	121
3	123	123	123	122	124	124	144	152	149	146	124	131	134	123	120	123	141
4	123	123	129	148	151	146	146	150	149	145	122	120	128	143	143	139	142
5	128	142	141	139	142	146	159	164	165	172	156	163	157	145	143	146	164
6	141	141	143	150	159	164	165	172	172	163	155	144	143	150	164	180	174
7	142	143	152	164	170	171	174	170	158	143	145	146	153	169	178	170	166
8	146	158	167	171	172	173	165	150	139	139	141	152	167	164	160	158	155
9	165	173	177	178	173	165	151	141	141	143	155	165	163	160	158	155	143
10	174	180	181	184	172	162	146	139	141	144	159	165	159	158	157	152	139

[RES_2] The following few outputs give an illustration of how the outputs look at each stage in the process when encoding, quantization and compression are done. Since quantization is done here, the outputs and inputs wouldn't match. However, we could expect the quantized output and the decompressed output to be the same. The only way to measure the performance is by measuring how distorted the output video is. The following few outputs have been generated using temporal coding.

Original Set of Pixels

2.original | 2_1.tpc | 2_1_1.tpq | 2_1_1_1.tpv | 2_1_1_1.tpv.decompressed | 2_1_1_1.tpv.decompressed.decoded

1	158	187	189	165	151	137	122	116	122	139	127	158	165	145	146	131	121
2	160	148	142	124	118	120	124	146	172	175	152	145	138	123	124	124	121
3	123	123	123	122	124	124	144	152	149	146	124	131	134	123	120	123	141
4	123	123	129	148	151	146	146	150	149	145	122	120	128	143	143	139	142
5	128	142	141	139	142	146	159	164	165	172	156	163	157	145	143	146	164
6	141	141	143	150	159	164	165	172	172	163	155	144	143	150	164	180	174
7	142	143	152	164	170	171	174	170	158	143	145	146	153	169	178	170	166
8	146	158	167	171	172	173	165	150	139	139	141	152	167	164	160	158	155
9	165	173	177	178	173	165	151	141	141	143	155	165	163	160	158	155	143
10	174	180	181	184	172	162	146	139	141	144	159	165	159	158	157	152	139

Temporally Coded using Predictor 1

2_2.tpc | 2_2_2_2.tpv.decompressed | 2_2_2_2.tpv.decompressed.decoded | 2_2_2_2.tpv | 2_2_2.tpq

1	158	187	189	165	151	137	122	116	122	139	127	158	165	145	146
2	2	-39	-47	-41	-33	-17	2	30	50	36	25	-13	-27	-22	-22
3	-37	-25	-19	-2	6	4	20	6	-23	-29	-28	-14	-4	0	-4
4	0	0	6	26	27	22	2	-2	0	-1	-2	-11	-6	20	23
5	5	19	12	-9	-9	0	13	14	16	27	34	43	29	2	0
6	13	-1	2	11	17	18	6	8	7	-9	-1	-19	-14	5	21
7	1	2	9	14	11	7	9	-2	-14	-20	-10	2	10	19	14
8	4	15	15	7	2	2	-9	-20	-19	-4	-4	6	14	-5	-18
9	19	15	10	7	1	-8	-14	-9	2	4	14	13	-4	-4	-2
10	9	7	4	6	-1	-3	-5	-2	0	1	4	0	-4	-2	-1

Quantized output

2_2.tpc | 2_2_2.tpq | 2_2_2_2.tpv.decompressed | 2_2_2_2.tpv.decompressed.decoded | 2_2_2_2.tpv

1	144	176	176	176	144	144	112	112	112	144	112	144	176	144	144
2	16	-48	-48	-48	-48	-16	16	16	48	48	16	-16	-16	-16	-16
3	-48	-16	-16	-16	16	16	16	16	-16	-16	-16	-16	-16	16	-16
4	16	16	16	16	16	16	16	-16	16	-16	-16	-16	-16	16	16
5	16	16	16	-16	-16	16	16	16	16	16	48	48	16	16	16
6	16	-16	16	16	16	16	16	16	16	-16	-16	-16	-16	16	16
7	16	16	16	16	16	16	16	-16	-16	-16	-16	16	16	16	16
8	16	16	16	16	16	16	-16	-16	-16	-16	-16	16	16	-16	-16
9	16	16	16	16	16	-16	-16	-16	16	16	16	16	-16	-16	-16
10	16	16	16	16	-16	-16	-16	-16	16	16	16	16	-16	-16	-16

Compressed output using Shannon Fano Algorithm

2_2.tpc | 2_2_2.tpq | 2_2_2_2.tpv | 2_2_2_2.tpv.decompressed | 2_2_2_2.tpv.decompressed.decoded

```
1  11101,176|111100,-80|1101,-48|10,-16|0,16|1100,48|11100,80|11111,112|111101,144
2  11110111101111011110111110111110111111111111111111111101111111111011110111101111
3  01101110111011101100011001100010101010100000110001010100010001100011011101000000
4  11011010100000101010101001010000101101100000110011001100011011000000001010101000
5  00000001001010101000001000000001010101011000000010101010010000101010100010101010
6  00010100000011001100000001100000001010000010001010100000101010101010000000100000
7  01000000001010101000110001010101010100000101010101010000101010100010100000010100
8  00000001010101000000101010101010000010101000101000010100000101000000100001000010
9  00000010101010100010101010101010101001010101010101010001010100000000010101010100
10 00000101010000010101010101000101010101000000010101010100000000000010000000000001000
```

Decompressed Output using Shannon Fano Algorithm (Same as Quantized Output)

2_2.tpc | 2_2_2.tpq | 2_2_2_2.tpv | 2_2_2_2.tpv.decompressed | 2_2_2_2.tpv.decompressed.decoded

1	144	176	176	176	144	144	112	112	112	144	112	144	176	144	144
2	16	-48	-48	-48	-48	-16	16	16	48	48	16	-16	-16	-16	-16
3	-48	-16	-16	-16	16	16	16	16	-16	-16	-16	-16	-16	16	-16
4	16	16	16	16	16	16	16	-16	16	-16	-16	-16	-16	16	16
5	16	16	16	-16	-16	16	16	16	16	16	48	48	16	16	16
6	16	-16	16	16	16	16	16	16	16	-16	-16	-16	-16	16	16
7	16	16	16	16	16	16	16	-16	-16	-16	-16	16	16	16	16
8	16	16	16	16	16	16	-16	-16	-16	-16	-16	16	16	-16	-16
9	16	16	16	16	16	-16	-16	-16	16	16	16	16	-16	-16	-16
10	16	16	16	16	-16	-16	-16	-16	16	16	16	16	-16	-16	-16

Decoded Output

2_2.tpc | 2_2_2.tpq | 2_2_2_2.tpv | 2_2_2_2.tpv.decompressed | 2_2_2_2.tpv.decompressed.decoded

1	144	176	176	176	144	144	112	112	112	144	112	144	176	144	144
2	160	128	128	128	96	128	128	128	160	192	128	128	160	128	128
3	104	136	136	136	136	152	136	136	120	152	104	120	152	152	120
4	148	148	148	148	132	156	148	116	156	156	100	108	140	156	140
5	142	158	158	126	118	170	158	142	154	170	150	162	162	170	146
6	161	137	169	153	141	179	169	145	171	147	109	119	135	179	159
7	167	163	179	155	145	190	179	127	146	142	113	156	164	190	168
8	180	166	190	170	159	200	158	120	142	128	95	153	165	168	147
9	189	180	200	178	168	179	152	107	160	151	120	170	148	163	141
10	200	189	211	190	147	173	139	97	167	155	123	177	140	149	128

[RES_3] The following few outputs give us a picture of the outputs in the same way that the previous 2 results provide except that the following outputs were generated using Spatial coding.

Original Values

3_1.spc | 3_2.spc | 2_4_1_2.tpv.decompressed.decoded | new 1

1	173	173	174	173	172	170	169	169	173	176
2	173	173	174	173	173	172	170	170	173	175
3	172	172	174	174	173	172	170	170	173	174
4	173	173	174	174	174	173	172	170	171	172
5	174	174	173	174	175	175	174	172	170	170
6	174	174	174	174	178	176	176	174	171	170
7	173	173	174	175	179	180	179	176	171	170
8	174	173	174	175	180	181	181	178	172	171
9	174	173	174	175	181	182	182	179	173	172
10	174	174	175	175	181	182	182	179	173	172

Spatially Coded with Predictor 5

3_4.spc | 3_3.spc | 3_5.spc

1	173	173	174	173	172	170	169	169	173	176
2	173	0	1	0	1	1	0	1	3	1
3	172	0	1	1	43	0	-1	0	1	19
4	173	1	-3	0	0	0	1	-4	-1	-1
5	174	1	-1	-2	1	1	1	-2	0	0
6	174	0	-1	2	4	-9	0	0	0	1
7	173	0	2	1	-5	1	2	0	40	4
8	174	0	1	0	4	-1	-1	1	29	2
9	174	0	0	0	0	0	0	-3	2	0
10	174	1	1	-1	-1	-5	0	0	0	0

Compressed Output

3_5_1_2.spv | 3_5_1_2.spv.decompressed.decoded | 3_4.spc | 3_3.spc | 3_5.spc | 3_5_1.spq | 3_1.spc

```
1  00000,0|00001,1|0001,2|0011,3|010001,4|01010,5|011000,6|011001,7|011101,8|01111,9
2  10111001101110011011000110111001110001011011100011011010010110100101110011100011
3  10111001000000000100000000010000100000000001001100001
4  11000101000000000100001110110111000000010000000000110110101
5  10111001000010100000000000000000000000010100100100000100
6  10110001000010010000101000010000100001001010000000000
7  10110001000000010000010100011000010000000000000000001
8  10111001000000001000010101100001000100000110100100010001
9  10110001000000000100000010001001000010000001110010011000 1
10 10110001000000000000000000000000000000100000001000000
```

Decoded Output

3_5_1_2.spv | 3_5_1_2.spv.decompressed.decoded | 3_4.spc | 3_3.spc

1	173	173	174	173	172	170	169	169	173	176
2	173	173	174	173	173	172	170	170	173	175
3	172	172	174	174	173	172	170	170	173	174
4	173	173	174	174	174	173	172	170	171	172
5	174	174	173	174	175	175	174	172	170	170
6	174	174	174	174	178	176	176	174	171	170
7	173	173	174	175	179	180	179	176	171	170
8	174	173	174	175	180	181	181	178	172	171
9	174	173	174	175	181	182	182	179	173	172
10	174	174	175	175	181	182	182	179	173	172

Conclusion

This section brings to a conclusion, the efforts made in this project and also talks briefly about further work.

The project gave us a deep understanding of video processing strategies including predictive coding, quantization, compression and decompression strategies. The complexities involved in using the right algorithms and achieving right results in order to obtain same outputs on decoding the encoded outputs gave us an understanding of what intricacies need to be taken care of in implementing these algorithms.

Although the algorithms implemented, themselves were complex, sufficient efforts were put in formalizing file structure and how data needs to be written to ensure that different tasks are in sync with what needs to be done.

Further Work

Further efforts on this project would be categorized into basic and advanced set

Basic Set

The basic set of tasks are the ones that weren't implemented and need to be implemented for the project to be complete.

1. Fixing issues in LZW compression
2. Implementing Arithmetic Coding
3. Writing to and reading from binary files for the compression algorithms
4. Implementing a mechanism to render videos appropriately

Advanced Set

The advanced set of tasks would be optimization

1. Currently the implemented scripts aren't written in a modular fashion leading to duplication of work at several places which would need to be rectified.

Note – Reasons for the project being incomplete have been addressed in the confidential evaluation.

Bibliography

[1]	Ze-Nian Li, Mark S. Drew, Jiangchuan Liu, in *Fundamentals of Multimedia*