CIE 425 - Information Theory and Coding

Final Project - H264 Video Compression

Fall 2022

Team Members:

Youssef Islam Elshabrawy 201-900-667

Amr Ahmed Elmasry 201-901-202

Ahmed Mohamed Ibrahem 201-902-227

1. Video Encoder

This function converts any supported MATLAB video into H.264 binary stream of bits.

```
clc; clear;
% Encode
[code, dict,Quant, Ls, fps, L, W] = MPEG_Encode("xylophone.mp4", 1);
code
```

```
code = 1×22938278
     0     0     0     0     0     1     1     1     1     0     1     1     0 ···
```

Function Parameters

The function Takes **3 input arguments:**

1. **Path/name** of the file to be converted (or a matlab prebuilt video like “xylophone.mp4”.)
2. **Quantization Level**: 1 for low, and 2 for high quantization.
3. [Optional] Number of frames to be encoded.

It returns **7 outputs:**

- The **binary encoded** frames and motion vectors.

```
code
```

```
code = 1×22938278
     0     0     0     0     0     1     1     1     1     0     1     1     0 ···
```

- The **huffman dictionary** (cood-book) for this code.

```
dict(1:5,:)
```

```
ans = 5×2 cell
```

	1	2
1	-175	1×21 double
2	-174	1×22 double

	1	2
3	-167	1×23 double
4	-165	1×23 double
5	-164	1×21 double

1. The **quantization level** (again)

```
Quant
```

```
Quant = 1
```

- The **Length of Streams**: It is a very important design parameter and it is an array holding the length of the frame followed by the length of its motion vector. Example:

```
Ls
```

```
Ls = 1×282
      154216           4      159892        2103      161531        2103 ···
```

Column 1 represents the **length of frame 1**, **column 2** is the **length of Motion Vector 1** (which is in this case 4 bits that will be ignored as frame 1 is an I-frame). Columns 3, and 4 represent the length of frame 2 and motion vector 2 respectively, and so on.

- The **Framerate** of the video which is extracted from the video object to be used in the decoder.

```
fps
```

```
fps = 30
```

- The **Length** of the frame

```
L
```

```
L = 240
```

- The **Width** of the frame

```
W
```

```
W = 320
```

Compression Ratio

For the Low Quantization:

141 is the number of frames in video, and 8 is log2(256)

```
OriginalSize = L * W * 8 * 141;
CR = OriginalSize/length(code)
```

```
CR = 3.7767
```

For the High Quantization:

```
[code_high, dict_high, Quant_high, Ls_high, fps2, L2, W2] = MPEG_Encode("xylophone.mp4", 2);
CR_high = OriginalSize/length(code_high)
```

```
CR_high = 11.3999
```

2. Video Decoder

This function reconstructs code from the encoder function into a grayscale version of the input video.

```
% Decode
matrix = MPEG_Decode(code, dict, Quant,Ls, fps,L, W,"myOutput_H")
```

```
matrix =
matrix(:,:,1) =

    54    53    52    52    52    51    48    46    43    42    42    41    39    38    38    37    38    38    38
    54    53    52    52    52    51    48    46    43    43    42    41    40    39    38    37    38    38    38
    53    52    52    52    51    50    48    46    43    43    42    41    40    39    38    38    38    38    38
    53    52    51    51    51    50    48    45    43    43    42    41    40    40    39    39    38    38    38
    53    52    51    51    51    50    47    45    43    43    42    42    41    40    40    39    38    38    38
    52    51    51    50    50    49    47    45    43    43    42    42    41    41    40    40    40    40    40
    52    51    50    50    50    49    46    44    43    43    42    42    42    41    41    41    41    41    41
    52    51    50    50    50    49    46    44    43    43    43    42    42    41    41    41    42    42    42
    50    50    50    50    50    48    47    46    44    44    44    43    43    42    42    42    42    42    42
    50    50    50    50    50    48    47    46    44    44    44    43    43    43    42    42    42    42    42
    50    50    50    50    50    48    47    46    44    44    44    44    43    43    43    42    42    42    42
    50    50    50    50    50    48    47    46    45    45    44    44    44    43    43    43    43    43    43
    50    50    50    50    50    48    47    46    45    45    45    44    44    44    43    43    43    43    43
    50    50    50    50    50    48    47    46    46    45    45    45    44    44    44    44    44    44    44
    50    50    50    50    50    48    47    46    46    46    45    45    45    44    44    44    44    44    44
    50    50    50    50    50    48    47    46    46    46    46    45    45    44    44    44    44    44    44
    52    50    48    47    47    47    47    46    47    47    47    46    46    45    45    45    43    43    43
    52    50    48    47    47    47    47    47    47    47    46    46    46    45    45    45    43    43    43
    53    51    48    46    46    47    48    48    47    46    46    46    45    45    45    45    43    43    43
    54    51    48    46    46    48    48    49    46    46    46    45    45    45    44    44    43    43    43
    55    52    48    46    47    48    49    49    46    46    45    45    45    44    44    44    43    43    43
    55    52    49    47    48    49    49    49    45    45    45    45    44    44    44    43    43    43    43
    54    52    49    48    49    49    49    49    45    45    45    44    44    44    43    43    43    43    43
    54    52    50    49    49    49    49    48    45    45    45    44    44    43    43    43    43    43    43
    52    52    52    51    51    50    50    50    46    46    45    44    44    45    46    46    47    47    46
    52    52    52    51    51    51    50    50    48    47    47    46    46    47    47    48    49    49    48
    52    52    52    52    51    51    51    50    50    49    49    49    49    49    49    50    51    51    50
    53    53    52    52    52    51    51    51    50    50    50    50    50    50    50    50    51    51    51
    53    53    53    52    52    52    51    51    50    50    50    50    50    50    50    50    51    50    50
    54    53    53    53    52    52    52    52    49    49    50    50    50    50    49    49    50    50    50
    54    54    53    53    53    52    52    52    49    50    50    51    51    50    50    49    50    50    50
     ⋮
```

Function Parameters

The function Takes 8 **input arguments**:

1. **Encoded Video** bits to be decoded.
2. **Huffman Dictionary** used in encoding
3. **Quantization Variable** (1 = low, 2 = high)
4. The **Length of Streams** specified in the encoder
5. The **frame rate** to be used in writing the output video
6. **Length** of original video
7. **Width** of original video
8. [optional] **Output file name** to write video in

It returns 1 **output**: Which is the 3D matrix of the frames

```
matrix
```

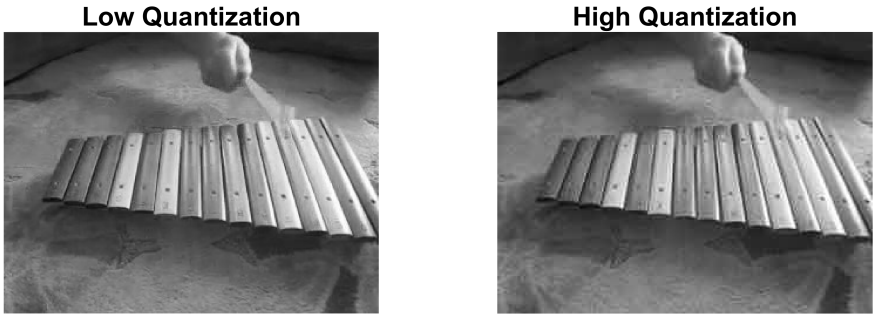
```
matrix =
matrix(:,:,1) =

    54    53    52    52    52    51    48    46    43    42    42    41    39    38    38    37    38    38    38
    54    53    52    52    52    51    48    46    43    43    42    41    40    39    38    37    38    38    38
    53    52    52    52    51    50    48    46    43    43    42    41    40    39    38    38    38    38    38
    53    52    51    51    51    50    48    45    43    43    42    41    40    40    39    39    38    38    38
    53    52    51    51    51    50    47    45    43    43    42    42    41    40    40    39    38    38    38
    52    51    51    50    50    49    47    45    43    43    42    42    41    41    40    40    40    40    40
    52    51    50    50    50    49    46    44    43    43    42    42    42    41    41    41    41    41    41
    52    51    50    50    50    49    46    44    43    43    43    42    42    41    41    41    42    42    42
    50    50    50    50    50    48    47    46    44    44    44    43    43    42    42    42    42    42    42
    50    50    50    50    50    48    47    46    44    44    44    43    43    43    42    42    42    42    42
    50    50    50    50    50    48    47    46    44    44    44    44    43    43    43    42    42    42    42
    50    50    50    50    50    48    47    46    45    45    44    44    44    43    43    43    43    43    43
    50    50    50    50    50    48    47    46    45    45    45    44    44    44    43    43    43    43    43
    50    50    50    50    50    48    47    46    46    45    45    45    44    44    44    44    44    44    44
    50    50    50    50    50    48    47    46    46    46    45    45    45    44    44    44    44    44    44
    50    50    50    50    50    48    47    46    46    46    46    45    45    44    44    44    44    44    44
    52    50    48    47    47    47    47    46    47    47    47    46    46    45    45    45    43    43    43
    52    50    48    47    47    47    47    47    47    47    46    46    46    45    45    45    43    43    43
    53    51    48    46    46    47    48    48    47    46    46    46    45    45    45    45    43    43    43
    54    51    48    46    46    48    48    49    46    46    46    45    45    45    44    44    43    43    43
    55    52    48    46    47    48    49    49    46    46    45    45    45    44    44    44    43    43    43
    55    52    49    47    48    49    49    49    45    45    45    45    44    44    44    43    43    43    43
    54    52    49    48    49    49    49    49    45    45    45    44    44    44    43    43    43    43    43
    54    52    50    49    49    49    49    48    45    45    45    44    44    43    43    43    43    43    43
    52    52    52    51    51    50    50    50    46    46    45    44    44    45    46    46    47    47    46
    52    52    52    51    51    51    50    50    48    47    47    46    46    47    47    48    49    49    48
    52    52    52    52    51    51    51    50    50    49    49    49    49    49    49    50    51    51    50
    53    53    52    52    52    51    51    51    50    50    50    50    50    50    50    50    51    51    51
    53    53    53    52    52    52    51    51    50    50    50    50    50    50    50    50    51    50    50
    54    53    53    53    52    52    52    52    49    49    50    50    50    50    49    49    50    50    50
    54    54    53    53    53    52    52    52    49    50    50    51    51    50    50    49    50    50    50
     ⋮
```

Results

```
matrix_high = MPEG_Decode(code_high, dict_high, Quant_high,Ls_high, fps2,L2, W2,"myOutput_High"
highFrame = uint8(matrix_high(:,:,141));
lowFrame = uint8(matrix(:,:,141));
figure
```

```
subplot(121); imshow(lowFrame);title('Low Quantization');xlabel(sprintf('CR = %.2f', CR))
subplot(122); imshow(highFrame);title('High Quantization');xlabel(sprintf('CR = %.2f', CR_high)
```



We tried the functions on a larger video with multiple frame changes to test our compression, and the results were as we expected!

Videos Link!

cr7_out.mp3 is compressed with compression ratio:

```
>> originalSize = L * W * 8 * sz

originalSize =

   1.3148e+09

>> compressedSize = length(readyForTransmittion)

compressedSize =

     0

>> compressedSize = length(readyForTransmittion)

compressedSize =

   245856103

>> CR = originalSize/compressedSize

CR =

    5.3479
```