Steven Karl
CSCI576 - Multimedia Systems Design
Parag Halvadaar
10/31/2022

Assignment 3

**Note - All figures are screenshots from the script I wrote to calculate/illustrate this problem. The script is also submitted in the submission zip and can be run to prove this is my work.

Question 1: DCT Coding *(20 points)*

In this question you will try to understand the working of DCT in the context of JPEG. Below is an 8x8 luminance block of pixel values and its corresponding DCT coefficients.

188	180	155	149	179	116	86	96
168	179	168	174	180	111	86	95
150	166	175	189	165	101	88	97
163	165	179	184	135	90	91	96
170	180	178	144	102	87	91	98
175	174	141	104	85	83	88	96
153	134	105	82	83	87	92	96
117	104	86	80	86	90	92	103

1. Using the 2D DCT formula, compute the 64 DCT values. Assume that you quantize your DCT coefficients using the luminance quantization table K1 on page 143 of the uploaded ITU-T JPEG standard. What does your table look like after quantization? *(5 points)*

Solution: Shown in Figure 3 below

```
******************************
Source Image Samples
******************************
  188.00   180.00   155.00   149.00   179.00   116.00    86.00    96.00
  168.00   179.00   168.00   174.00   180.00   111.00    86.00    95.00
  150.00   166.00   175.00   189.00   165.00   101.00    88.00    97.00
  163.00   165.00   179.00   184.00   135.00    90.00    91.00    96.00
  170.00   180.00   178.00   144.00   102.00    87.00    91.00    98.00
  175.00   174.00   141.00   104.00    85.00    83.00    88.00    96.00
  153.00   134.00   105.00    82.00    83.00    87.00    92.00    96.00
  117.00   104.00    86.00    80.00    86.00    90.00    92.00   103.00
******************************
Done
******************************
```

Figure 1: Source Image Samples

```
******************************
Forward DCT Coefficients
******************************
 1016.25   216.02    -6.85   -27.14    29.21   -20.71   -11.30     8.00
  136.19    52.65   -93.46    -7.29    34.01   -18.82   -11.26    10.64
  -45.89   -49.22    13.91    53.76    11.11   -24.67    -0.15     8.41
    8.88    38.07    47.96    15.62   -17.87   -10.86     4.17     3.69
   -1.28    -5.91    -1.24    -4.72     0.75     6.60     4.84     0.24
   -4.43    -1.20     3.35     8.12     7.01     6.12    -0.15     1.18
   -2.97    -2.12     0.85    -1.46     0.00    -3.35    -0.93    -1.21
   -0.78    -3.39    -0.58    -1.80    -4.20    -1.26     2.32     1.63
******************************
Done
******************************
```

Figure 2: Table after DCT

```
******************************
Beginning Quantization
******************************
64.00 20.00 -1.00 -2.00  1.00 -1.00  0.00  0.00
11.00  4.00 -7.00  0.00  1.00  0.00  0.00  0.00
-3.00 -4.00  1.00  2.00  0.00  0.00  0.00  0.00
 1.00  2.00  2.00  1.00  0.00  0.00  0.00  0.00
 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
******************************
Done with Quantization
******************************
```

Figure 3: Table after Quantization

2. In the JPEG pipeline, the quantized DCT values are then further scanned in a zigzag order. Ignoring your DC value, show the resulting zigzag scan AC values. *(2 points)*.

Solution:

The image below shows the zig zag scan of the values after quantization. This shows all values. The first value of 64 is the DC value so this can be ignored for grading purposes. I also wrote them out underneath if they are hard to read in the image.

```
******************************
Zig Zag Notation
******************************
[64, 20, 11, -3, 4, -1, -2, -7, -4, 1, 0, 2, 1, 0, 1, -1, 1, 2, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
******************************
Done
******************************
```

20, 11, -3, 4, -1, -2, -7, -4, 1, 0, 2, 1, 0, 1, -1, 1, 2, 2, 0, 0, 0, 0, 0, 1, 0, … (All zeros after this)

3. For this zigzag AC sequence, write down the intermediary notation *(5 points)*

Solution:

DC Intermediary Notation:

RLE: < Size > < Amplitude> < 7 > < 64 >

AC Intermediary Notation:

RLE: < Run Length, Size > < Amplitude>

```
******************************
Intermidary AC Notation
******************************
< 0, 5 >, < 20 >
< 0, 4 >, < 11 >
< 0, 2 >, < -3 >
< 0, 3 >, < 4 >
< 0, 1 >, < -1 >
< 0, 2 >, < -2 >
< 0, 3 >, < -7 >
< 0, 3 >, < -4 >
< 0, 1 >, < 1 >
< 1, 2 >, < 2 >
< 0, 1 >, < 1 >
< 1, 1 >, < 1 >
< 0, 1 >, < -1 >
< 0, 1 >, < 1 >
< 0, 2 >, < 2 >
< 0, 2 >, < 2 >
< 5, 1 >, < 1 >
******************************
Done
******************************
```

Figure: This shows the AC intermediary notation

4. For these are luminance values, write down the resulting JPEG bit stream. You will need to consult standard luminance code tables on page 150 of the ITU-T JPEG standard. *(6 points)*

Solution:

DC:

!Ignored!

AC's:

11010 10100

1011 1011

01 00

100 100

00 0

01 01

100 000

100 011

00 1

11011 10

00 1

1100 1

00 0

00 1

01 10

01 10

1111010 1

1010 (EOB)

5. What compression ratio do you get for this luminance block? *(2 points)*

Solution:

Original = 256 bits

New = 90 bits (Including EOB; Excluding DC Coef Bits)

Ratio 90:256 = 0.351 compression