CMPT 365 Written Assignment 3

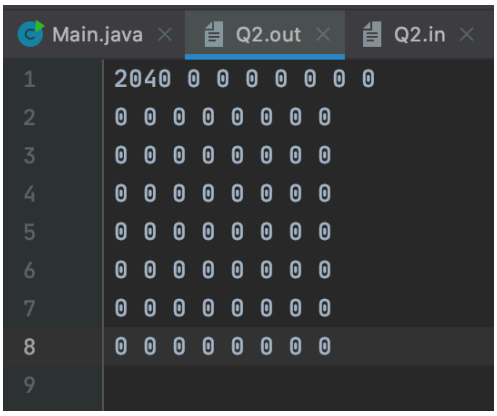
Junchen Li

301385486

2021/4/18

Textbook question

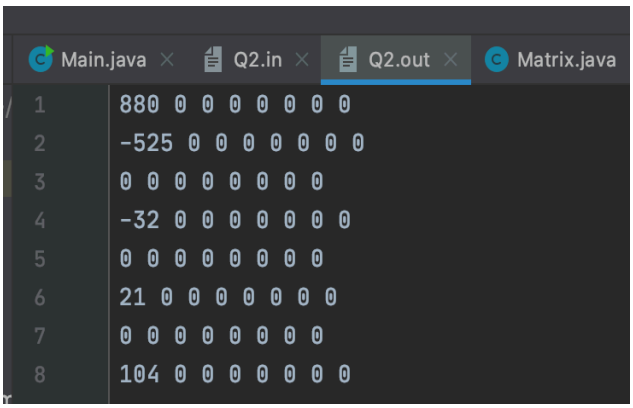
1. (P 297 Q4) For each block size N by N. There only has one DC in N by N. So for this question, in the 8 by 8 block the fraction that original pixels would use is $\frac{1}{8*8} = \frac{1}{64}$.
2. 5(a) If we want to find the largest DCT coefficient value, the first idea is we let every pixel be the max range value. The range is 0…255 so I make a pure white image which every pixel is 255. Then I put this matrix into our programming assignment 2. The follow screenshot is the result that I got after running.



The largest coefficient is 2040 and all other DCT coefficient value is 0.

3. (p336 Q4)

i) We could also put the matrix into the code and running it. The follow plot is the running result.



We could see the value of F(0,0) is 880. Also we can using the average-intensity to do. We sum all entry $(20 * 16) + (80 * 16) + (140 * 16) + (200 * 16) = 7040$ The matrix size is 8*8. The F(0,0) = $8 * \frac{7040}{8*8} = 880$.

ii) Because we can consider the absolute value, the largest value is |-525| = 525. It is the F(1,0). The F(1,0) is negative because of the phase is 180 degree apart.

b)

We first shrink the matrix by reducing simply averages each 2 by 2 block into a single pixel value. Think like $\begin{matrix}20 & 20\\ 20 & 20\end{matrix} = 20 * \frac{4}{4} = 20$. So process the original matrix, we got the $X_2 = \begin{matrix}20 & 20 & 20 & 20\\ 80 & 80 & 80 & 80\\ 140 & 140 & 140 & 140\\ 200 & 200 & 200 & 200\end{matrix}$. We do the same process again we could get other reduction matrix. $X_4 = \begin{matrix}50 & 50\\ 170 & 170\end{matrix}$ (e.g.(20+20+80+80)/4 = 50). For expansion the single pixel value for four times. The X_2 will change to the original matrix from the textbook. Expansion(X_2) =

$$\begin{matrix}20 & 20 & 20 & 20 & 20 & 20 & 20 & 20\\ 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20\\ 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80\\ 80 & 80 & 80 & 80 & 80 & 80 & 80 & 80\\ 140 & 140 & 140 & 140 & 140 & 140 & 140 & 140\\ 140 & 140 & 140 & 140 & 140 & 140 & 140 & 140\\ 200 & 200 & 200 & 200 & 200 & 200 & 200 & 200\\ 200 & 200 & 200 & 200 & 200 & 200 & 200 & 200\end{matrix}$$

The difference between the original image and compressed image is just find difference value for each entry. The difference of X_2 all the entry change to the zero. And for X_4:

Difference(X_2) = $\begin{matrix}-30 & -30 & -30 & -30\\ 30 & 30 & 30 & 30\\ -30 & -30 & -30 & -30\\ 30 & 30 & 30 & 30\end{matrix}$ (e.g. 20-50= –30, 200 – 170 = 30…)

The codewords generated by subtract for each second line.

$X_4 = \begin{matrix}50 & 0\\ 120 & 0\end{matrix}$ $X_2 = \begin{matrix}-30 & 0 & 0 & 0\\ 60 & 0 & 0 & 0\\ -60 & 0 & 0 & 0\\ 60 & 0 & 0 & 0\end{matrix}$ The X_2 still have all entry are zero.

4. (P364 Q8 a)

The mode of H.263 is derived from H.261, so some property could be same as the H.261. We know the QCIF (The Quarter of Common Intermediate Format) is the quarter of CIF(352*288). The QCIF has the picture format is 176*144 which means there are 176*144 pixels for luminance channel (Y). The key difference between two

standards is the macroblock size N is 8 can not affect the complexity in H.263. We assume the frame rate is f, then if we use the brute-force (sequential search) method, the complexity of motion estimation as follow:

$$(2p+1)^2 * 3\cancel{N^2} * \frac{C*R}{\cancel{N*N}} * f$$

$$= (2p+1)^2 * 2CRf$$

So the complexity of sequential search method is O($p^2 * CRf$)`1

(b) For the 2D logarithmic method if the frame rate still is f:

$$\left(8 * \left(\lceil \log_2 p \rceil + 1\right) + 1\right) * N^2 * 3 * \frac{C*R}{N*N} * f$$

So the complexity of 2D logarithmic method is O (log p*CR*f)

5. (P530 Q9)

Construct a corresponding table

Band	8	9
Level (db)	60	25

(a) if the 9th band at 40 dB, from the table we know at 9th band have 25 dB. So it will not ignore and this MP3 only send 40-25 = 15 dB.

(b) If the original signal at 20 dB, then it will send 3 bits only. Compared with the 7 bits, it will save 4 bits.

(c) It will not send any bits.

EX.1

(0,45) (3.28) (7.12) (0,11) (12,7) (0,0). The first one value 45 is the DC component value and the (0,0) is the end of block value, so the input matrix as follow:

45	0	0	0	0	0	0	0
0	28	0	11	0	7	0	0
0	0	12	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	0	0	0	0	0	0

If we simply use the regular left-to-right and top-to-bottom scan, the corresponding output as follow: (0,45) (8,28) (1,11) (1,7) (4,12) (0,0).

EX.2

Answer: The most time-consuming part in the video compression is Motion Estimation(ME). The reason is we need to a lot of comparisons in the search window and for each comparison step we need to three operations. Usually, a normal video at least has some minutes long even a movie has a few hours. The amount of work will skyrocket. Also switching between different scenes, some light and Angle factors can make motion estimation more difficult. So the most time consuming part in video compression is motion estimation.