

# CSCI 576 Assignment 3

**Instructor: Parag Havaladar**

**Assigned on 10/19/2020**

**Solutions due 11/02/2020 by 4 pm afternoon**

## **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:

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

- Using the 2D DCT formula, compute the 64 DCT values. Assume that you quantize your DCT coefficients uniformly with  $Q=100$ . What does your table look like after quantization. (5 points)
- 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 when  $Q=100$ , (2 points).
- For this zigzag AC sequence, write down the intermediary notation (5 points)
- Assuming these are luminance values, write down the resulting JPEG bit stream. For the bit stream you may consult standard JPEG VLC and VLI code tables. You will need to refer to the code tables from the ITU-T JPEG standard which is also uploaded with your assignment. (6 points)
- What compression ratio do you get for this luminance block ? (2 points)

Note – you only need to turn in written/printed answers. To compute your DCT, you may write a script in a language of your choice or use any conventional libraries to do so.

## **Programming on Block Based Motion Compensation (80 points)**

This programming assignment will help you gain an understanding of issues that relate to motion compensation. Given two consecutive image frames –  $\text{frame}_n$  and  $\text{frame}_{n+1}$ , write a function that creates and display two images

- a *predicted or reconstructed frame* for  $\text{frame}_{n+1}$  built using motion compensation techniques from  $\text{frame}_n$  and,
- error difference frame.

You will be using the full brute force method, by dividing  $\text{frame}_{n+1}$  into  $16 \times 16$  blocks and performing motion compensation within an input search area  $k$ . The details of the algorithm have been explained in the class lecture and also detailed in the textbook. Here is how your program will be invoked

*MyMotionPredictor.exe*  $\text{frame}_n.\text{rgb}$   $\text{frame}_{n+1}.\text{rgb}$   $k$

Input to your program will be three parameters - two consecutive rgb image frames of size  $640 \times 480$  (**first parameter is previous frame, second parameter is frame to predict, you have ample successive video frames from the assignment 2 dataset**) and search parameter  $k$ , which will define the search area to search into,  $k$  can have values from 1 to 32. Speed is important, but more important is the accuracy of your output.

Although your inputs are .rgb frames, you will need to compute motion vectors only on the Y channel – so your process will need to produce and display two gray level images.

### **Conversion of RGB to YUV**

Given R, G and B values the conversion from RGB to YUV is given by

$$\begin{array}{rcll} Y & = & 0.299 & 0.587 & 0.114 & R \\ U & & 0.596 & -0.274 & -0.322 & G \\ V & & 0.211 & -0.523 & 0.312 & B \end{array}$$

Remember that if RGB channels are represented by  $n$  bits each, then the YUV channels are also represented by the same number of bits. In this assignment you are asked to use just the Y value for doing motion prediction.

Example output is shown below



predicted Y channel for  $\text{frame}_{n+1}$



error difference for Y channel

### **What should you submit?**

- Your source code, and your project file or makefile. If you have questions, please confirm the submission procedure with the TAs. **Please do not submit any binaries or data sets.** We will compile your program and execute our tests accordingly.
- Along with the program, also submit an electronic document (word, pdf, pagemaker etc) for the

written part and any other extra credit explanations.