<u>Introduction to Visual Information Processing CS455</u>

<u>Assignment 3</u>

1) Author's Information

Ali Arda Eker

aeker1@binghamton.edu

2) Purpose

Implementing DCT and IDCT algorithms. DCT is used to compress the given image and IDCT is used to retrieve the image back from the compressed one. Different parameters are trained in compression and results are discussed. Color models are also implemented and used to segment the given image.

3) Methods & Results

A) Theory Part

1) Middle column has RGB values (122, 255, 122)

2) **a)** Since the hue values are represented in the interval from 0 to 360, we can compress it into range of 0 to 255 because we have an 8-bit image. Thus we calculate the angle by using RGB values and get 170 for the bottom left region and 123 for the bottom middle region. The central part is by definition 0 because it is white. Background is also 0.

b) The center is 0 because in the original 3 color image it is white which has 255 on its 3 channels (R, G, B) and using the formula $x = 1 - \frac{3}{(\mathrm{R+G+B})} \min(R, G, B)$ which gives saturation component we can easily find $1 - \frac{3}{785} 255$ equals 0.

c) Intensity component is found by the formula $\frac{(R+G+B)}{3}$ and since the yellow, cyan and magenta has same (R + G + B) value that is 510, their corresponding regions is 170 as the yellow part is. Similarly red, green and blue has same (R + G + B) value that is 255 thus their regions is 85 as same as red. Central region is 255 because white has (R + G + B) value 765 and background is 0 because it has 0 on 3 of its channels.

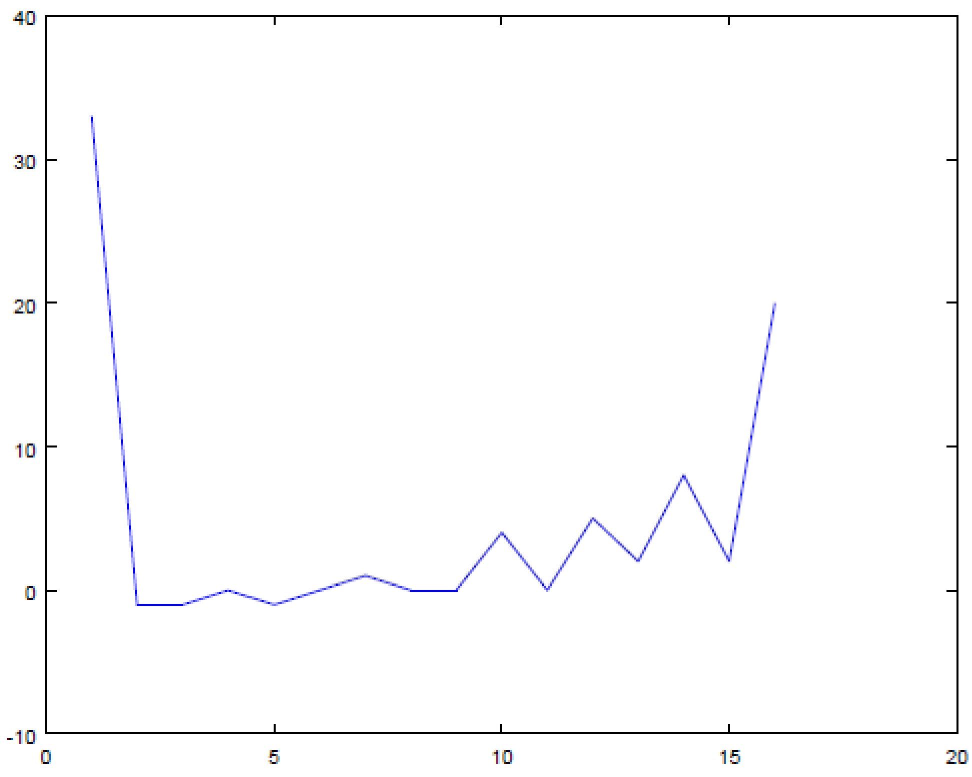
3) **a)** Green HSI component is (120, 1, 85) so if we interpret this as a RGB image it is seen as purple but if we interpret this as a HSI image it is again seen as green. Similarly for the red part it is (0, 1, 85) and if we interpret this as a RGB image it is seen as dark blue but if we interpret this as a HSI image it is again seen as red. Also for blue part HSI is (240, 1, 85) and it is seen as pink but if we interpret this as a HSI image it is again seen as blue.

b) Since the saturation value is 1 in all components of the HSI image, smoothing operation doesn't have an effect on it if we ignore the borders.

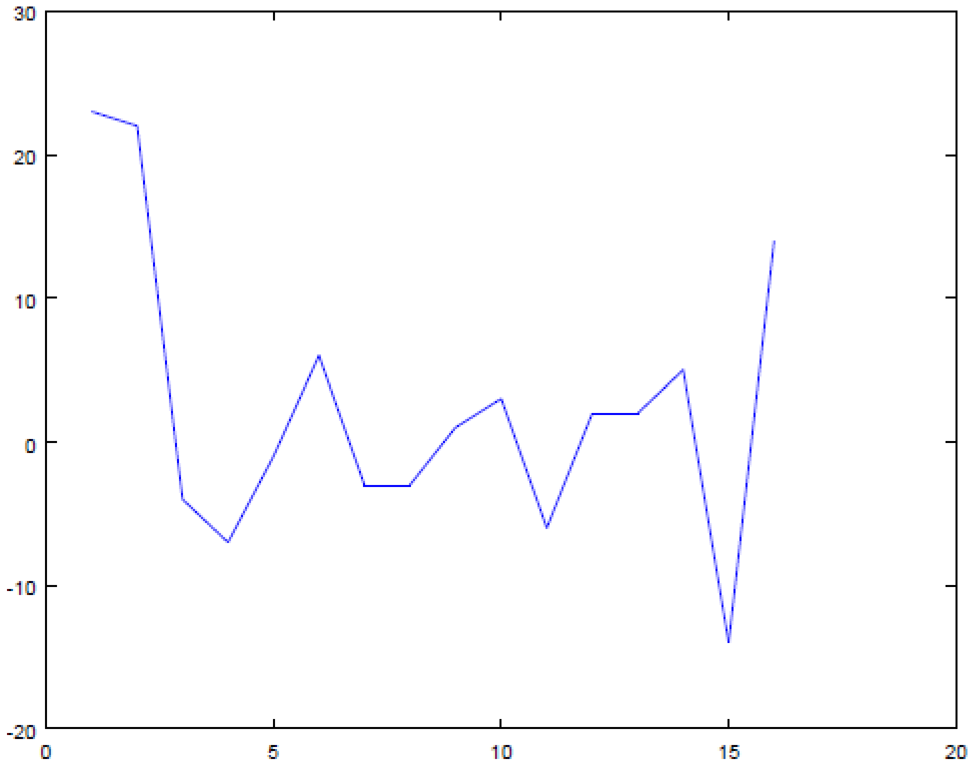
c) Hue values are 120 in green region, 0 in red region and 240 in blue region so if we smooth the whole image the region between red and green part closes to the 60 and the part between green and blue closes to the 180. Also the central part becomes around 120. This mean central part looks like green, regions between red and green look like yellowish and between green and blue look like blue.

B) DCT Based Image Compression

1) **a)** Each row separately converted to the frequency domain by using 1-d discrete cosine function.



b) 16 bit long single vector converted to the frequency domain by using 1-d discrete cosine function



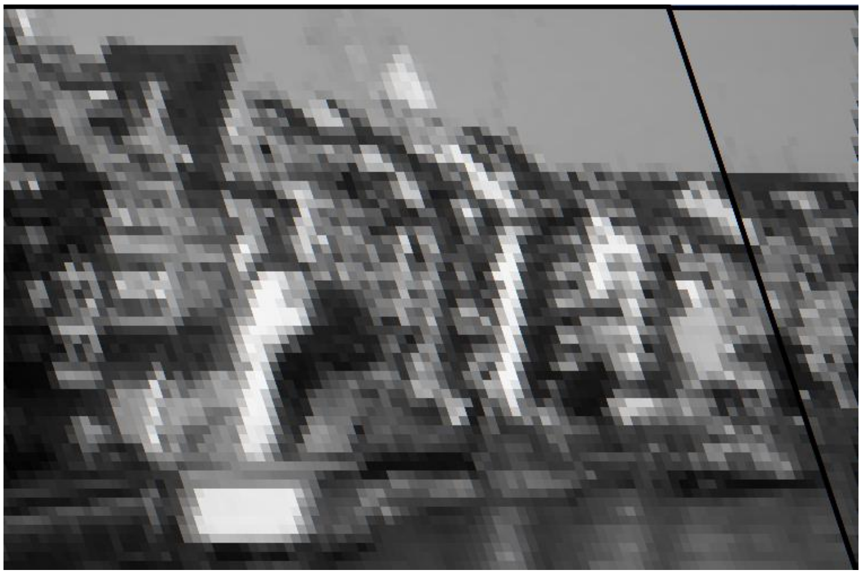
c) When we compress the image with block size 8 the information lost will be less. The values after first index are closer to the 0 so when we leave them out, our loss will be minor compared to the size 16 block. Block size 16 has a bigger absolute values after the first index that will make it more information leaking after we retrieved the original images back. But it saves us more space since it needs less byte to be kept. Thus If we have a low bandwidth I would suggest to use block size 16 but if the information is very important we should use 8 block size.

2) 2D Image Compression

First of all we find the intensity values for every pixel of the given image by dividing sum of its red, blue and green channels by 3 to obtain intensity image I. To convert the image I from spatial domain to frequency domain discrete cosine transformation with block size 8 by 8 is used. The DCT formula in the slides are implemented.

After generating frequency domain image F, we truncated it in 2 ways to have 2 different frequency images D1 and D2. D1 is constructed by only keeping the DC values (direct current – low frequency) of each block and D2 is found by keeping a 3 by 3 block in each block's upper left corner. Other values are set to 0 in both images.

Next step is converting images D1 and D2 back to the spatial domain from the frequency domain. IDCT formula in the slides are implemented and we extracted the both compressed images. The upper image is R1 which is found by applying IDCT to the D1 and the bottom one is R2 which is found using D2.

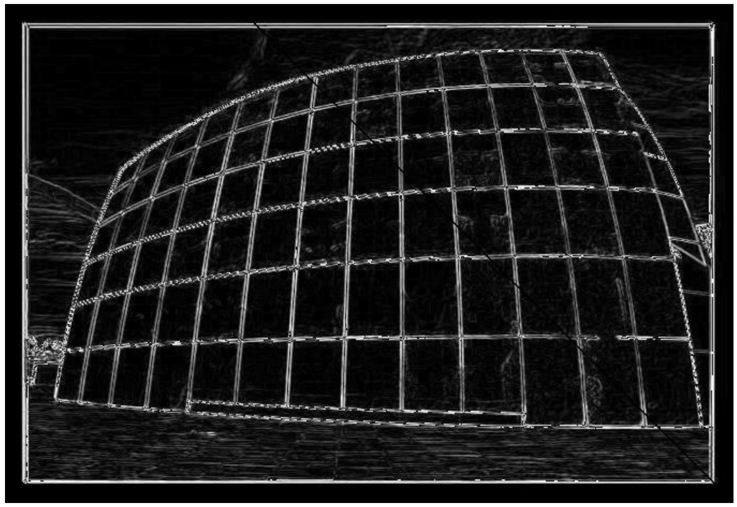


As it is seen the R1 is more blurred. Because it is constructed by only keeping DC values in each block and this means other values are lost during compression. Although it occupies less space and easier to transmit compared to R2, after compression more information is lost.

C) Color Based Image Segmentation

First step was generating HSI models for the given RGB images. For each pixel I found the intensity value and then assigned it to the 3 channels. This resulted a grey level image which will be used for segmentation.

Second step was emphasizing edges of the intensity images. I used sobel operators for this. I convolved the image with the sobel operator in x direction then in y direction separately. Then took the root of the sum of the squares of convolved images and edges are emphasized as shows below.



Third and final step was performing a thresholding operation on both edge images to get the segments. I set the threshold value to the 100 after I tried many other different values. The pixels which have values above 100 are set to 255 and others to 0. After the thresholding segmented images are seen as below.