CEG4316
Computer Assignment #3
Color Image Manipulation and Still Image Compression

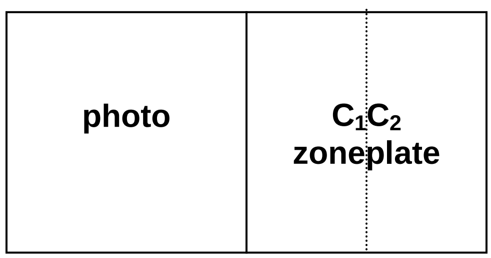
Due date: Your report is due by December 3, 2013 on Blackboard Learn. Results will be demonstrated during the lab periods on November 14, 21, 28 in STE 2060.

Objectives:

1. Subsample and interpolate the chrominance component of images.
2. Experiment with some aspects of DCT-based image compression.

Procedure:

1. You will find a 1024 wide by 512 high R'G'B' color image `testpattern_lab3_q2.tif` on the course web page. It is arranged as follows:



This R'G'B' image is assumed to be in device space, i.e., Rec. 709 gamma-corrected RGB. The C_1C_2 zoneplate is defined in the luma-chrominance space induced by the Bayer color filter array as follows. First, define

$$z(x,y) = \cos\big(512\pi((x-1.5)^2 + (y-.5)^2)\big)\text{rect}(x-1.5, y-.5)$$

The C_1 zoneplate is given by

$$\begin{aligned} f_L(x,y) &= 0.5 \\ f_{C1}(x,y) &= 0.25z(x,y), \quad 1 \le x \le 1.5 \\ f_{C2}(x,y) &= 0 \end{aligned}$$

The C_2 zoneplate is given by

$$\begin{aligned} f_L(x,y) &= 0.5 \\ f_{C1}(x,y) &= 0, \qquad\qquad 1.5 \le x \le 2.0 \\ f_{C2}(x,y) &= 0.25z(x,y) \end{aligned}$$

Of course, the C_1 and C_2 zoneplates have been converted to R'G'B'. All components are assumed to be sampled on a rectangular lattice with equal horizontal and vertical spacing $X = \frac{1}{512}$ ph.

From the 1024×512 color image described above, generate three 1024×512 images consisting of the L component, the C_1 component (plus 0.5) and the C_2 component (plus 0.5), where

$$\begin{bmatrix} f_L \\ f_{C1} \\ f_{C2} \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ -0.25 & 0.5 & -0.25 \\ -0.25 & 0.0 & 0.25 \end{bmatrix} \begin{bmatrix} f_{R'} \\ f_{G'} \\ f_{B'} \end{bmatrix}.$$

Visualize these components as if they were grayscale images. Using the function `imresize` with bicubic interpolation, subsample ***both*** the C_1 and C_2 components in the horizontal and vertical directions by 2, 4, 8 and 16 respectively (the same subsampling factor for both directions). Then, with a second use of `imresize` with bicubic interpolation, interpolate them back to the original size and view them as grayscale images. Finally, in each case, convert back to R'G'B' using the original L component and compare with the original color image. How much chrominance subsampling do you think is acceptable? Base your decision on the *image* part of the test pattern. However, take note and comment on what happens to the chrominance zoneplate part.

2. Write a MATLAB program `q2_ask.m` to do the following: Transform the grayscale image `'alfred_Y.tif'` with the 8 by 8 block DCT, using the MATLAB function `blkproc` or `blockproc`. Prompt the user for p, the percentage of coefficients to set to zero. Then set the last p% of the coefficients to zero along the zigzag scan and convert back to the spatial domain. Compute the PSNR between the resulting image and the original image (pg.11-3 of course notes – remember $f_{\max} - f_{\min} = 1.0$). Write a second program `q2_curve.m` to generate a graph that depicts the PSNR on the ordinate versus p on the abscissa, for values of p between about 5% and 98%. Choose appropriate values of p to get a reasonably smooth curve. Comment on the relationship between p, the PSNR and the visual quality you observe. What percentage of coefficients must be set to zero to get a PSNR of 30 dB? 40 dB? For what percentage of coefficients set to zero do you start to just notice the difference between the original and the reconstructed images? A routine `zigzag.m` is provided for your convenience if you want to use it.

3. Write a MATLAB program `q3_constant.m` to do the following: Transform the image `'alfred_Y.tif'` with the 8 by 8 block DCT. Prompt the user for a quantizer stepsize Δ. (Values of Δ should be well within the range [0.01, 0.2].) Uniformly quantize all the transform coefficients, using the *same* step size Δ for all coefficients. Compute the percentage of coefficients that are zero in the transformed image (the MATLAB function `nnz` may be helpful). Transform back to the spatial domain and visually compare the reconstructed image with the original image. Compute the PSNR between the resulting image and the original image. Experimentally, find a step size that gives a barely noticeable difference with the original image, and find a step size that gives a very noticeable distortion.

Write a second program q3_jpeg.m to do the same thing using the JPEG quantization table (pg. 11-9 of course notes, but divide values by 255 – the MATLAB statement is provided on the course web page), prompting the user for the parameter *qf*. Find a value of the parameter *qf* that gives a barely noticeable difference with the original, and one that gives a very noticeable difference. Again, compute the percentage of coefficients that are zero in the transformed image and the PSNR in each case.

4. As in question 2, construct curves of PSNR vs Δ and PSNR vs *qf* for the two approaches of question 3 on the image `'alfred_Y.tif'`. Try to cover about the same range of PSNR values. What values of Δ and *qf* result in a PSNR of about 30 dB in the two cases? 40 dB? Compare the three reconstructed images with ~35 dB PSNR, from question 3 and this question, and comment on their relative subjective quality according to your opinion. They could be all equivalent or one may be better than the others. Compare the percentage of non-zero coefficients for the three images.