CEG4311
Computer Assignment #3
Color Image Manipulation and Still Image Compression

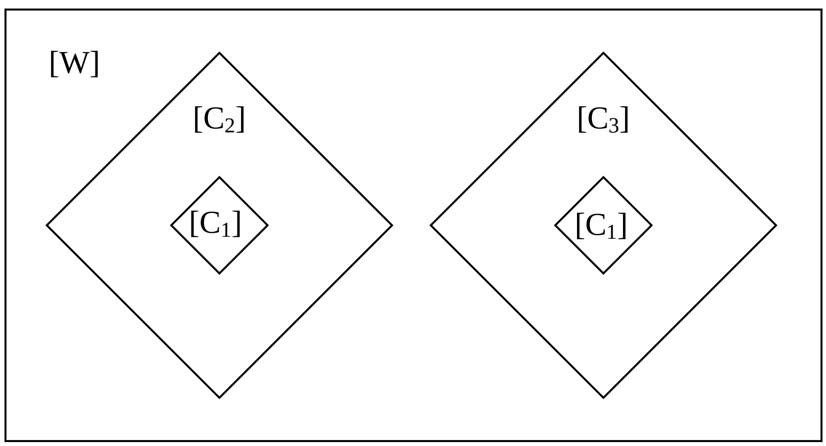
Due date: Your report is due by December 2, 2007 on virtual campus. Results will be demonstrated during the lab periods on November 7, 14, 21 and 28 in STE 2060.

Objectives:

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

Procedure:

1. Create and display a 512×256 color image as follows:



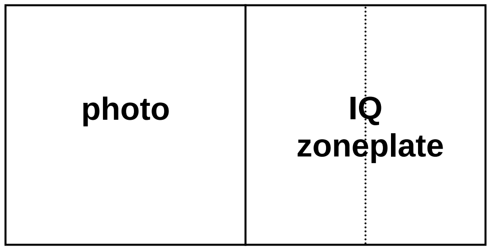
The *horizontal width* of the small diamonds is 0.3 ph and the *horizontal width* of the large diamonds is 0.85 ph. Here, [W] represents white and [C_1], [C_2], [C_3] are three colors that are specified by their Rec. 709 gamma-corrected RGB values as follows:

[W]: $W_{R'} = 1.0$, $W_{G'} = 1.0$, $W_{B'} = 1.0$; [C_1]: $C_{1R'} = 0.835$, $C_{1G'} = 0.816$, $C_{1B'} = 0.345$;
[C_2]: $C_{2R'} = 0.937$, $C_{2G'} = 0.882$, $C_{2B'} = 0.294$; [C_3]: $C_{3R'} = 0.714$, $C_{3G'} = 0.725$, $C_{3B'} = 0.557$.

The black lines above are for illustration only and are not part of the image. The resulting color image will be of the form A(1:256,1:512,1:3). (Hints: You can generate the three RGB color planes separately, and combine them using the `cat` function. You can initially generate the full constant rectangle, then overlay the diamonds for C_2 and C_3, and finally overlay the diamonds for C_1. You can use the approach shown on pg. 2-68 in the course notes to draw the diamond shaped regions.)

Describe any discrepancies between what you see and what you know you have generated, and comment on it. Modify your program to generate only the small diamonds on the white background.

2. You will find a 512×256 R'G'B' color image `testpattern_lab3_2007_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., gamma-corrected rec. 709 RGB. The IQ zoneplate is defined in luma-chrominance space as follows. First, define

$$z(x,y)=\cos(256\pi((x-1.5)^2+(y-.5)^2))\mathrm{rect}(x-1.5,y-.5)\ .$$

The I zoneplate is given by

$$\begin{aligned} f_{Y'}(x,y)&=0.5\\ f_I(x,y)&=0.25z(x,y), 1\le x\le 1.5\\ f_Q(x,y)&=0.0\end{aligned}$$

The P_R zoneplate is given by

$$\begin{aligned} f_{Y'}(x,y)&=0.5\\ f_I(x,y)&=0.0\\ f_Q(x,y)&=0.25z(x,y), 1.5\le x\le 2\end{aligned}$$

All are assumed to be sampled on a rectangular lattice with equal horizontal and vertical spacing $X=\frac{1}{256}\mathrm{ph}$. The NTSC luma-chrominance components are related to the R'G'B' values by

$$\begin{bmatrix} f_{Y'}\\ f_I\\ f_Q\end{bmatrix}=\begin{bmatrix} .299 & .587 & .114\\ .596 & -.275 & -.321\\ .212 & -.523 & .311\end{bmatrix}\begin{bmatrix} f_{R'}\\ f_{G'}\\ f_{B'}\end{bmatrix}$$

From the $512\times 256\times 3$ color image described above, generate three 512×256 images consisting of the *Y*' component, the *I* component (plus 0.5) and the *Q* component (plus 0.5). Visualize these components as if they were black-and-white images. Using the function `imresize` with bicubic interpolation, subsample both the *I* and *Q* 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. In each case, convert back to R'G'B' and compare with the original. How much chrominance subsampling do you think is acceptable? Base your decision on the *photo* part of the image. However, take note and comment on what happens to the zoneplate part.

3. Write a MATLAB program `q3_ask.m` to do the following: Transform the image `'alfred_Y.tif'` with the 8 by 8 block DCT. 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. 5-4 of course notes – remember $f_{max} - f_{min} = 1.0$). Write a second program `q3_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? 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. You can use `blkproc` for the block processing.

4. Write a MATLAB program `q4_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 q4_jpeg.m to do the same thing using the JPEG quantization table (pg. 5-8 of the 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.

5. As in question 3, construct curves of PSNR vs Δ and PSNR vs *qf* for the two approaches of question 4 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? Compare the three reconstructed images with ~30 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.

6. Bonus question (up to 15% extra marks). Using the MATLAB jpeg writing function with `imwrite`, make a curve of PSNR versus the `'Quality'` parameter (i.e., write and then read the image; see the documentation for `imwrite`). Can you determine a relationship between `'Quality'` and *qf*? Do this with `'alfred_Y.tif'` and see if it applies to other images.