CEG4316
Lab Assignment #1
Plotting & Spatial Filtering of Images

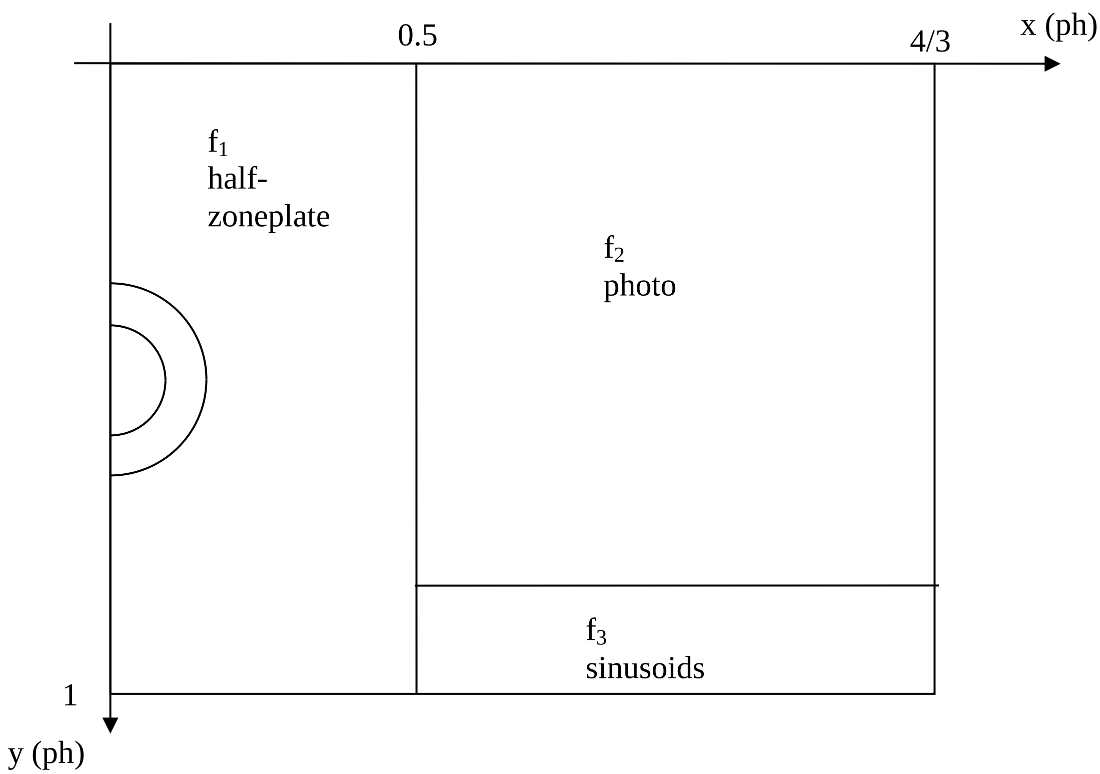
Due date: Your report is due by ***midnight October 15, 2013,*** submitted electronically on Blackboard Learn. Results will be demonstrated to the TA during the lab periods on September 26, October 3 and October 10 as you complete different parts. Attendance is mandatory and will be recorded.

Objectives:

1. Read, write and display images in MATLAB
2. Compute and display the frequency response of 2D filters
3. Filter images with various FIR filters and observe the effects.

Procedure:

A. Create and display a 4:3 image (4/3 ph by 1 ph) consisting of 1024 by 768 samples as follows.



You should create three images A, B, C and mosaic them as Z= [A [B; C]].Use `meshgrid` to generate A and C, of course making the appropriate adjustments to size, spacing, etc. Note that u_1, v_1, u_2, v_2 have units of c/ph and r is in ph.

$$f_1(x,y) = (0.5 + 0.5\cos(\pi(x^2 + (y - .5)^2)/r^2)), \quad r = 0.05, \quad 0 \le x < 0.5, \quad 0 \ge y < 1$$

$$f_2(x,y) = \text{photo}(x,y) \ \ \text{(640 by 640 image)}$$

$$f_3(x,y) = (0.5 + 0.25\cos(2\pi(56x + 9y)) + 0.25\cos(2\pi(30x - 180y))), \quad 0 \le x > 0.8\overline{3}, \quad 0 \le y < 0.1\overline{6}$$

$$f = f_1 + \mathcal{T}_{(.5,0)} f_2 + \mathcal{T}_{(.5,0.8\bar{3}).} f_3$$

where $\mathcal{T}$ is the shift operator. Save the test image as a TIFF file, and make sure you can display the saved image. Experiment with other values of r, u_1, v_1, u_2, v_2. (Make them variables in your m-file.)

Compute the following point-wise non-linear transformation of the image **Z** in two ways. Time each using tic, toc and compare. Also display both Z1 and Z2 and compare.

(i) Z1 = direct computation (you figure out how)

(ii)
```
for i = 1:768
     for j = 1:1024
         if(Z(i,j) < 0.0031308)
             Z2(i,j) = 22.92*Z(i,j);
         else
             Z2(i,j) =1.0555*Z(i,j)^0.41666 - 0.055;
         end
     end
end
```

B. For each of the 2D finite-impulse-response (FIR) filters in the following list, compute and display the *frequency response* using the MATLAB function `freqz2`. Generate both perspective (mesh) plots and contour plots, where contours are spaced by 0.1 (unless you think some other spacing is more informative). Label the frequency axes in cycles per picture height (c/ph) with the v–axis pointing down. Assume a square lattice with X = 1/768 ph. ***Only include in your report those plots and images that are explicitly requested, and answer any specific questions.*** However, in your report, give a *qualitative* description of ***each*** frequency response and comment on any distinctive features. Be prepared to show *any* frequency response in the lab demos. Filter the test image Z1 of part A with each filter given below. Describe in your report the changes you see in the filtered image and how these relate to the form of the filter's frequency response. Use the 2D convolution function `conv2` in MATLAB to convolve the image with each of the filter unit-sample responses, to create a new 1024 by 768 image, using the option `'same'`. Note that some filters remove the DC component and you have to add an offset (say 0.5) to be able to view the image correctly.

---

1. Moving average filter: $h[n_1X, n_2X] = \frac{1}{49}, -3 \leq n_1, n_2 \leq 3$; 0 otherwise.
2. Separable filters: Let
   h_1 = [1 10 45 120 210 252 210 120 45 10 1]/1024 and
   h_2 = [-3 0 25 0 -150 256 -150 0 25 0 -3]/512
   be two 1D filters defined for $-5 \leq n \leq 5$ with the center point corresponding to n = 0.
   Consider the four separable filters:
   $h_{11}[n_1X, n_2X] = h_1[n_1X]h_1[n_2X]$ $\quad h_{12}[n_1X, n_2X] = h_1[n_1X]h_2[n_2X]$,
   $h_{21}[n_1X, n_2X] = h_2[n_1X]h_1[n_2X]$ $\quad h_{22}[n_1X, n_2X] = h_2[n_1X]h_2[n_2X]$.
   In each case, generate the two-dimensional unit sample response and then ignore separability

in the filtering with `conv2` (but see 5. below). Include the contour and mesh plots of $h_{11}[n_1X, n_2X]$ and $h_{12}[n_1X, n_2X]$, and the images processed with these two filters, in your report.

3. Gaussian filters: $h[n_1X, n_2X] = c_i \exp(-(n_1^2 + n_2^2)/2r_i^2)$, $-9 \le n_1, n_2 \le 9$, for the values r_1 = 1.0 and r_2 = 3.2. For each value of r_i, determine the value of c_i that gives a DC gain of $H(0, 0) = 1.0$. Include the contour and mesh plots of the second Gaussian filter with r_2 = 3.2 in your report.

4. Gabor notch filter:
$$h[n_1X, n_2X] = \delta[n_1, n_2] - 2c_2 \exp(-(n_1^2 + n_2^2)/2r_2^2)\cos(2\pi(u_2 n_1 + v_2 n_2)/768)$$
$-9 \le n_1, n_2 \le 9$ where u_2 = 30 c/ph and v_2 = -180 c/ph, and r_2 and c_2 are as in item 3. Also, filter the corrupted image `PhotoPlusNoise_2013` from the course web page and explain what happens. Include a contour plot of the filter response in your report, as well as the original and filtered `PhotoPlusNoise` images.

5. For the first Gaussian filter of question 3 (r_1 = 1.0), express it in separable form. Compared to the direct implementation of question 3, see if it is any faster to first filter with the horizontal filter and then the vertical filter, by using `conv2` twice. Also try the separable mode of `conv2`. Use MATLAB's timing functions `tic` and `toc` to check the computation time. Report all the times and comment if the image outputs are the same in all cases.

For the lab demos, you will be asked to demonstrate the above eight filters by showing the frequency response plots, as well as the original and filtered image. Show these to the TA as soon as you are satisfied with the result.

Your report should be submitted electronically on virtual campus using the method described on the course web page. It should be prepared using a good text processing system and submitted in PDF or MSWord format. Pay attention to the quality of any images pasted in the report, avoiding resizing if possible.

For more information, read sections in the MATLAB Image Processing Toolbox User's Guide on image formats, reading and writing images and on image filtering (Chapter 2, 3, 8) or equivalent info from the `html` version. http://www.mathworks.com/help/toolbox/images/index.html
Some of the MATLAB functions that you *may* require or that *may* be useful: `imread, imwrite, imshow, imfinfo, im2double, uint8, ones, freqz2, conv2, tic, toc, input.`