CSE166 – Image Processing – Homework #1

Instructor: Prof. Serge Belongie

`https://sites.google.com/a/eng.ucsd.edu/cse-166-fall-2013/`

Due (in class) 12:30pm Oct. 8, 2013.

Reading

- Skim GW (Gonzalez & Woods *Digital Image Processing*, 3rd Edition) Ch. 1 and 2.
- GW Sec. 2.6, 3.4, 3.5.0–3.5.1, 3.6.0–3.6.1
- Review Material, "A Brief Overview of Linear Systems," up to the first occurrence of the word "Fourier."
- Note on Linear Shift Invariant Systems (see website)

General Homework Guidelines

- Use the Cover Sheet provided.
- Please attach all code that you use. Attach code at end of submission.
- In general try to keep your answers concise. Use as many words as you need and no more. Also work on your presentation skills. This means organize your plots and displays. Always use titles and add captions when appropriate. *Points will be awarded for clarity and presentation.*

Written exercises

1. GW Problem 3.15.
2. GW Problem 3.17.
3. GW Problem 3.21.
4. Determine whether each of the following systems is (i) linear and (ii) spatially invariant. *Justify your answers.*
   (a) $g(x) = e^{f(x)}$
   (b) $g(x) = f(x)f(x-1)$
   (c) $g(x) = f(x) - f(x-1)$
   (d) $g(x) = f(x-2) - 2f(x-17)$
   (e) $g(x) = [\sin(6x)]f(x)$
   (f) $g(x) = \sum_{k=x-2}^{x+4} f(k)$
   (g) $g(x) = xf(x)$
   (h) $g(x) = f(2x)$

Programming exercises

1. Working with Plots and images in Matlab or Python.

   (a) Download Figure 1.14(c) from the textbook photo gallery. Load the image and convert it to double if necessary In Matlab, use `imread` to load the image, and convert the image from `uint8` to `double` by typing `I=double(I);`. In Python use pylab's (assumes you have run `from pylab import *`) `imread` to load the image. In python, the image is loaded in a floating point data type by default, no need to convert.

   (b) Read the help files for the following commands:

Matlab	Python
`plot`	`plot`
`fplot`	No known equivalent. Matlab's `fplot` is a shortcut to plot a function on a given range. The same can be accomplished in Python using `x=arange(0,40); y=sin(x/3); plot(x,y,'-')` .
`subplot`	`subplot`
`hold`	`hold`
`stem`	`stem`
`title`	`title`
`text`	`text`
`ginput`	`ginput`

`gtext`	No known equivalent. Matlab's `gtext` function will write the text string input parameter where the mouse is clicked. This simple function does the trick:
	`def gtext(str):`
	`  pt = ginput(1)`
	`  gtext(pt[0][0], pt[0][1], str)`

`getrect`	No known equivalent. Matlab's `getrect` function allows you to select a rectangle in the current figure, returning a vector of the form `[xmin ymin width height]`. A simple alternative is to use Python's `ginput(2)` to collect top left and bottom right points of the rectangle.
`image`	`imshow`
`imagesc`	`imshow` (Note: Matlab's `imagesc` scales the image data to the full range of the current colormap. Python appears to do this by default, see the documentation for `imshow` , in particular the `vmin`, `vmax`, and `norm` parameters.
`truesize`	No known equivalent.
`impixelinfo`	No known equivalent

`imcrop`	No known equivalent. Matlab's `imcrop` is an interactive cropping tool. You can crop an image in Python in a slightly less interactive way by prompting the user for two points (upper left and lower right corners) using `ginput`. For example
	`# load and display the image`
	`im = imread('/path/to/image')`
	`imshow(im, origin='lower')`
	`show()`
	
	`# prompt the user for two points`
	`pts = ginput(2)`
	
	`# crop the image using the two points`
	`# assuming the two points correspond to`
	`# opposing corners of the image`
	`crop=im[pts[0][1]:pts[1][1],`
	`  pts[0][0]:pts[1][0]]`
	The Python Imaging Library (PIL) also has a crop method, but it is not interactive, you would need to prompt the user for two points.

`colormap`	This is the `cmap` parameter to the `imshow` function.
`colorbar`	`colorbar`

You will be using these functions often so get to know them well.

(c) Make an m-file or Python script using the above commands to do the following for the image `I`. Display it as it appears in the figure in the book, using the proper aspect ratio. Interactively select a rectangle roughly enclosing the bottom half of the middle bottle. Crop the image using that rectangle. Use `subplot` to display the cropped image next to the original image in the same plot (you will need to redisplay the original image). Display the cropped image using a gray colormap and colorbar, with the maximum and minimum gray levels corresponding to the maximum and minimum brightnesses in the image. Next put the titles 'original image' and 'cropped image' on the respective images (this may require additional calls to `subplot`). Select a point on the original image that corresponds roughly to the center of the cropped image using `ginput`. Plot a large white '+' at the location of the point returned by `ginput`. In Matlab, use `hold` so the call to `plot` does not overwrite the original image. Finally add a silly caption to the image (using `text` or `gtext`).

Things to turn in:

- Printout of your Matlab or Python script.
- Printout of program output.

2. Moving Averages.

(a) Convolution can be thought of as a 'moving average.' The values of the convolution kernel give the weights on each pixel used in the average. Consider the kernel $hbox =$

$[1, 1, 1]/3$. Create a vector `x` of length 100 representing a step edge with additive Gaussian noise (a step edge goes from a constant low value to a constant high value). Make the height of the edge 1.0 and the standard deviation of the noise 0.05. (Hint: (Matlab) add `0.05*randn(1,100)` to `x`, (Python) add `.05*randn(100)` to `x`.) Convolve `x` with `h_box` and make plots before and after.

(b) Given $y = x \star hbox$ ($\star$ is the convolution operator) , write down an expression for the value y_k for generic k (in terms of x_k). Here y_k represents the kth item in vector y (`y(k)` in Matlab, `y[k-1]` in python). What is the qualitative effect of convolution with `hbox`? This kernel is known as a 3-tap boxcar lowpass filter, or simply a box filter.

(c) Set `h_pasc` equal to the fifth row of Pascal's triangle and normalize it so that it sums to 1. Plot this kernel using `stem`. This is known as a 5-tap binomial lowpass filter. Perform the convolution with `x` and produce the plots as before. Compare this result (qualitatively) to the previous result using the boxcar. In particular, why might one want to use one filter vs. the other?

(d) Set `x2` equal to row 375 of Figure 3.35(a). Convolve this signal with each of the above kernels and plot the results.

Things to turn in:

- Written answer for part 2b and 2c.
- Plots: All plots should appear on the same page (use `subplot`) and should have appropriate titles and captions. Specifically, arrange the plots of `x, x*h_box, x*h_pasc` vertically, and likewise for `x2`. Also show the stem plots for `h_box` and `h_pasc` at the bottom of the page.

3. Moving Differences.

(a) When the convolution kernel has negative values, we can think of it as a 'moving difference.' This kind of kernel is useful for edge detection and texture analysis. Consider the kernel `[1 0 -1]/2`. This is known as the 'centered first difference.' The kernel `[1 -1]`, which is less commonly used, is known simply as a 'first difference.' As in Exercise 2, write down an expression for y_k for each of these kernels. Why do you suppose the centered version is more commonly used?

(b) Here are two ways of writing the derivative of a function $f'(x)$ as a limit:

$$f'(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon} \quad \text{or} \quad f'(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x-\epsilon)}{2\epsilon}$$

Explain how these definitions are related to the two first difference kernels. What does ϵ represent when $f(x)$ comes from an image?

(c) Apply the centered first difference kernel to row 238 of Figure 3.35(a) and plot the result, before and after. This time, when you plot the signals, crop the filtered signal by removing its first and last samples. This will make it the same length as the original signal and will also align it. Explain the result: what do the positions, heights, and signs of the spikes represent?

(d) Write down the expression for the centered second difference. Compare it to the limit definition of the second derivative. Explain how to obtain this kernel from the centered first difference kernel.

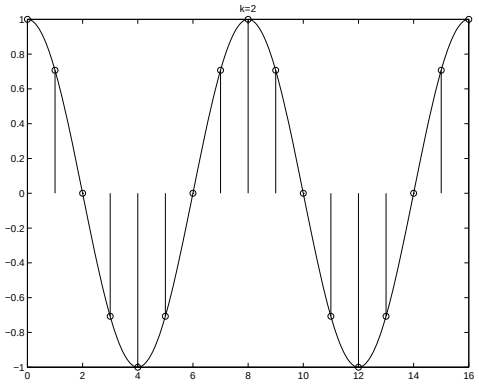
Things to turn in:

- Written answers for part 3a, 3b, 3d.

- Printout of plots and written answer for part 3c. Again, all plots should appear on the same page and should have appropriate titles and captions.

4. Sampling and Aliasing.

   (a) Consider the function $f(x) = \cos(\omega_o x)$ with $\omega_o = 2\pi k/N$. Let $N = 16$ and consider the interval $x \in [0, 16]$. Let $f(n)$ denote $f(x)$ sampled at the integers $0, 1, \ldots, 16$. In the same graph, plot the continous function $f(x)$ (using `fplot` in Matlab or the trick shown before in python) and $f(n)$ (using `stem`) for $k = 0, 1, \ldots, 16$. Again, you will need to use `hold` in creating the plots. Use `subplot` to arrange the plots in an array on a single sheet of paper. As an illustration, the following plot shows the case for $k = 2$.



   (b) Indicate the value of ω_o at which $f(x)$ hits the Nyquist frequency. Note that at this point we can write $f(n) = (-1)^n$.

   (c) Now look up the word "alias" in the dictionary and write down the definition. Explain why this term is used to describe $f(n)$ when ω_o exceeds the Nyquist frequency.

Things to turn in:

- Printout of script and plot for part 4a.
- Written answer for parts 4b, 4c.