

For information about the CS department's computer labs and/or to setup a new UNIX account, go to <http://www.cs.utexas.edu/facilities/computing/>.

## I. Using Matlab [75 points]

The goal of this problem set is to become familiar with basic Matlab commands, practice manipulating vectors and matrices, and try out basic image display and plotting functions. If you are unsure what a Matlab function does, check the reference manual (at the command line, type 'help' and then the command name).

1. Read over the provided Matlab introduction code and its comments. Open an interactive session in Matlab and test the commands by typing them at the prompt. (Skip this step if you are already familiar with Matlab.)
2. Describe (in words where appropriate) the result of each of the following Matlab commands. Use the `help` command as needed, but try to determine the output without entering the commands into Matlab. Do **not** submit a screenshot of the result of typing these commands.

a. `>> rand(4)`

b. `>> a = [1:100];`  
`>> b = a([1:3:end]);`

c. `>> f = [1001:2000];`  
`>> g = find(f > 1850);`  
`>> h = f(g);`

d. `>> x = 18.*ones(1,10);`  
`>> y = sum(x);`

e. `>> a = [1:100];`  
`>> b = a([end:-1:1]);`

3. Given a 100 x 100 matrix  $A$  representing a grayscale image, write a few lines of code to do each of the following. Try to avoid using loops.
  - a. Plot all the intensities in  $A$ , sorted in increasing value.
  - b. Display a histogram of  $A$ 's intensities with 20 bins.
  - c. Create and display a new binary image the same size as  $A$ , which is white wherever the intensity in  $A$  is greater than a threshold  $\tau$ , and black everywhere else.
  - d. Display the top left quadrant of  $A$ .

- e. Generate a new image (matrix), which is the same as  $A$ , but with  $A$ 's mean intensity value subtracted from each pixel. Set any negative values to 0.
- f. Use `randperm` to obtain a list of 5 random indices to select 5 rows at random from  $A$ , then assign the resulting 5 rows to a new vector called  $x$ .
- g. Let  $y$  be the vector:  $y = [1 \ 2 \ 3 \ 4 \ 5 \ 6]'$ . Use the `reshape` command to form a new matrix  $z$  that looks like this:

$$z = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}.$$

- h. Use the `min` and `find` functions to set  $x$  to the minimum value that occurs in  $A$ , and set  $r$  to the row it occurs in and  $c$  to the column it occurs in.
- i. Let  $v$  be the vector:  $v = [1 \ 8 \ 8 \ 2 \ 1 \ 3 \ 9 \ 8]$ . Use the `unique` function to reset  $v$  to  $[1 \ 2 \ 3 \ 8 \ 9]$ .

## II. Programming problem: measuring image noise [25 points]

This exercise is adapted from Trucco & Verri, Chapter 2.

Noise in the imaging sensor will affect the pixel intensity values we obtain. Even a sequence of images of the same static scene taken from a stationary camera will exhibit some variation in the pixel values at the same position in the image raster. Image acquisition noise is one reason it is dangerous to rely solely on pixel-level measurements to evaluate an image.

Write a Matlab program to estimate an imaging system's acquisition noise based on a sample of frames captured from a static scene. For a given image position, treat the pixel measurements over time as noisy instantiations of a random variable. For a given sequence, measure the mean brightness (grayscale intensity value) of each pixel and its standard deviation (noise).

Test your program on the provided (six) images of a static scene.

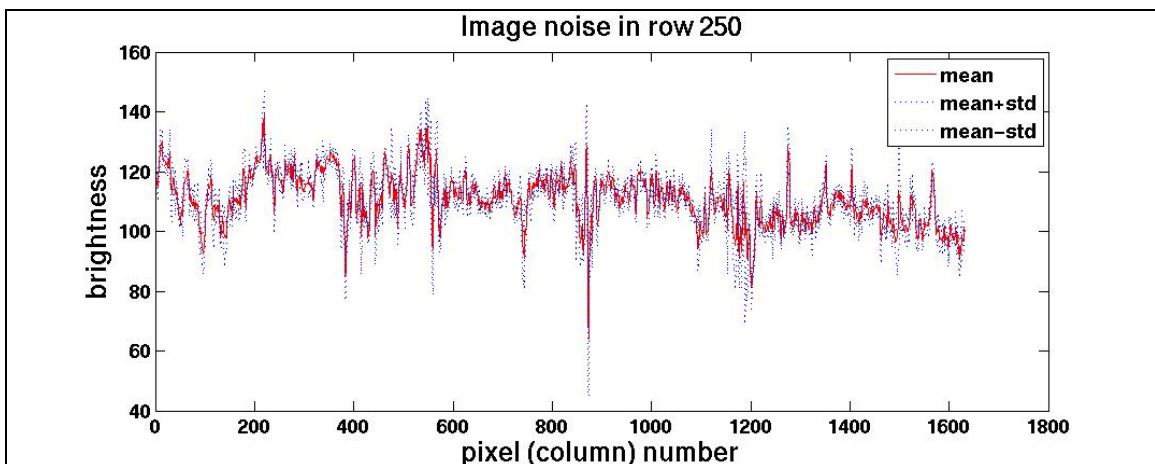
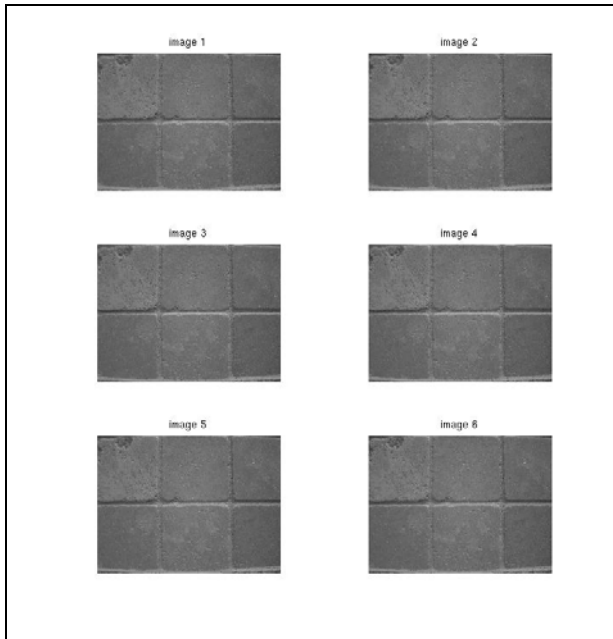
- o Use `'subplot'` to display all of the images in one figure.
- o Estimate the mean and standard deviation of the brightness at each pixel across the sequence.
- o Plot a graph showing the mean brightness and noise level (mean  $\pm$  standard deviation) for all the pixels in a given scanline (row) in the image.
- o Compute and print the average per-pixel noise.

This code fragment can be used to loop through all the image files in one directory (here assuming your images are in a sub-directory named `images`):

```
filelist = dir('images/*.pgm');
for i=1:length(filelist)
    imname = ['images/' filelist(i).name];
    nextim = imread(imname);
    . . .
end
```

Other useful Matlab functions for displaying plots: `title`, `legend`, `xlabel`, `ylabel`.

See example figure outputs below.



### Submission instructions: what to hand in

Create a single file with

- Your responses and code snippets for Part I questions
- Documented code for Part II
- Screenshot output for Part II

Name your file with your username, and submit it electronically via the turnin program.