Must be emailed to `boyland@ufl.edu` as a pdf file by 5:00 PM on Wednesday, December 11. Please try to keep file size under 2MB. Each answer must include Matlab code (if used), any rendered images required and the answer to all questions.

Part A: Noise Filter

1. Get the files `proj3a.txt` and `mask1.txt` from the homework webpage and put the content of the first file into a Matlab script and the second make a Matlab function. Save the files and run the script.
2. It should produce three images. The original image, the noisy image, and the filtered image.
3. Questions:
   (a) If A and B are $M \times N$ matrices, their convolution using Matlab indexing $m = 1, \ldots, M$ and $n = 1, \ldots, N$ is

   $$(A * B)_{m,n} = \sum_{k=1}^{M} \sum_{\ell=1}^{N} A_{k,\ell} B_{m-k+1, n-\ell+1}.$$

   with indices cyclically reduced. Show with a hand calculation that the array B defined by the function `mask1` implements this local averaging

   $$(A * B)_{m,n} = \frac{1}{5}(A_{m,n} + A_{m-1,n} + A_{m,n-1} + A_{m+1,n} + A_{m,n+1}).$$

   (b) Examine the noisy and filtering images produced. Does the filtering decrease the speckling? Does the filtering improve or blur the image resolution?

   Note: you do not have to turn in any images or code for Part A, just the answers to the Questions.

Part B: Edge Detection

1. Get the file `proj3b` from the class homework website and use the code in it to load the circuit board image, rescale it, turn it into a single greyscale image `zg`, display that and a reverse image of that. Note that this image is a different size than the one in Part A.
2. If $f(x, y)$ represents the values of a greyscale image then points (x, y) where

   $$\frac{\partial f}{\partial y}(x, y)$$

   is large in magnitude indicate horizontal edges. The simplest discrete approximation of this is

   $$\frac{\partial f}{\partial y}(x, y) \approx \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}.$$

Since we are not concerned with scale, we seek a mask H that implements

$$(A * H)_{m,n} = A_{m+1,n} - A_{m,n}.$$

Note that in a matrix the first coordinate is vertical, so the y-direction is in the first coordinate.

(a) Write a Matlab function `mask2` that creates this mask H.

(b) Apply it to the array `zg` as done in Part A to create a filtered array `hg`.

(c) Since it is the magnitude of the filtered array that matters, display `abs(hg)`as a reverse image (the reverse makes the elements with largest magnitude black).

3. Repeat B2, but this time with a mask you have created to find vertical edges.

4. Now create a procedure that displays all edges, both horizontal and vertical and use it to repeat the steps of B2. Hint: what matters is the magnitude of the gradient of f, or

$$\sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}.$$