ECE 172a	Machine Vision	#10	Pamela Cosman	2/3/12

HOMEWORK 3: Due Friday Feb 10, 11am

(but I'll accept this without penalty up to Monday Feb 13, 11am)

In Matlab, read in the bellpeppers image with the command
`pep = imread('peppers512','tif');`
Look at it with imshow. A noisy version of this image is named noipep512.tif. Read it in to Matlab and call it noipep. It was created with the commands:

```
noi = rand(512);
noipep = pep .* (noi > 0.1) + 255 * (noi < 0.1);
```

1. Perform Sobel edge filtering with the 3×3 Sobel filters, and produce a grayscale edge map (which I will call pepsob) in which the orthogonal edge gradients have been combined with the square root of sum of squares method. Don't worry about how the boundary of the image is treated. Don't use the built-in function edge() in Matlab. Write your own routine, but you can use filter2 or imfilter to do the filtering. We will define the *true binary edge map* to be the grayscale map pepsob thresholded at 50:

   ```
   trueBEM = (pepsob > 50);
   ```

   Just to verify that your Sobel routine is doing at least approximately what mine is doing, if I do

   ```
   sum(sum(trueBEM))
   ```

   I get ans = 11368, meaning that 11368 pixels are getting declared edge pixels in my binary edge map.

   Look at the binary edge map produced by a variety of thresholds. What threshold would you say corresponds to an extremely conservative binary edge map, in which almost every 1-pixel corresponds to the proper edge of a bellpepper? What features tend to show up in the binary edge map over a large range of thresholds which are *not* edges of bellpeppers?

2. We would like to see how well various edge detectors, applied to the noisy version, can do at recreating the true binary edge map that was generated from the clean image. First of all, use your Sobel filter on the noisy image, and threshold it at 50. How does it look? Is there any threshold where the binary edge map looks pretty reasonable?

3. One attempt at getting better edge detection amidst the noise goes as follows:

   ```
   grayedge = pyr7(noipep);
   CBEMp = (grayedge > 60);
   ```

   where pyr7.m is on the web site. Explain what pyr7 is doing, and compare the resulting binary edge map to the one from the previous part.

4. A second attempt at getting better edge detection comes from doing the noise cleaning explicitly using median filtering, and then following this up with a simple edge detector:

   ```
   newim = medfilt(noipep,3);
   ```

   where medfilt.m is on the web site. Characterize what kind of median filtering medfilt is doing. Look at newim– how well is the median filtering doing? Write a routine which does edge detection using the simple separated pixel difference operators combined with `max`. Call the grayscale output of this `newrun`.

5. Write a routine plotroc.m to generate a Receiver Operating Characteristic (ROC) curve. Your call to plotroc should look like this:

   ```
   plotroc(trueBEM,grayedge,1,200,10)
   ```

   Here, trueBEM is the true binary edge map which serves as the standard of comparison. grayedge is the *grayscale* candidate edge map which results from some edge detection method. The 3 numbers are the starting, ending, and step size for the threshold. If the routine takes too long to run, you can reduce the range of the thresholding, or else make the step size larger.

   Actually what we really need is a version of plotroc which can plot two separate ROC curves on the same plot, to compare two edge detection methods. So your function call should be of this form:

   ```
   comproc(trueBEM,grayedge,newrun,1,200,10)
   ```

   Here grayedge is the grayscale edge map from the pyr7 method, and newrun is the grayscale edge map from the median filtering + separated pixel differences method.

   Explain what the plot tells you. Note that the curves should come out fairly smooth, with no sharp corners. If your curve doesn't look smooth, you might have forgotten to normalize your edge detection filter (i.e., the coefficient out front should normalize the sum of positive or negative mask coefficients to unit gain) or else you need to use a smaller step size in the ROC routine.

6. What various tweaks to the pyr7 method or the median filtering method do you think would improve the edge detector performance (as measured by the ROC plot)? List as many as you can think of. Try out at least two of your proposed improvements and make a new comparative ROC plot to see if they really help.

What should be turned in:

1. Problem 1
   (a) Code for Sobel filter function
   (b) Answer of sum of edge pixel of $trueBEM$
   (c) Under what threshold you say it is extremely conservative
   (d) What features tend to show up over large range of thresholds which are not edges of bellpeppers. Illustrate your finding by looking at the figures you provided.

(e) Three binary edge image figures. One with threshold 50, and other two with smaller and larger thresholds.

2. Problem 2

   (a) Can you find a threshold you think is reasonable to represent edges? Illustrate it by your figure.

   (b) Two binary image figures, one is noise image with Sobel filter thresholded at 50. Another one shows your trail with different threshold to find more reasonable binary edge map.

3. Problem 3

   (a) Explain what pyr7 is doing.

   (b) Compare result binary edge image with 2.

   (c) One figure shows binary edge image result with threshold 60.

4. Problem 4

   (a) Explain what kind of median filter medfilt is? How well is it doing?

   (b) Write a routine of simple separated pixel difference operators.

   (c) Figure shows edge image which uses max to combine result of medfilt + simple separated pixel differences.

5. Problem 5

   (a) Function $comproc(bin, edge1, edge2, th1, th2, stepthresh)$ which plots ROC curves to compare performance of edge1 and edge2 images.

   (b) Compare ROC curve of results produced by 3. and 4..

   (c) Figure shows ROC curves of edge image produced by 3. and 4..

6. Problem 6

   (a) Write down at least two approaches to improve performance of ROC curve.

   (b) Code for two approaches you implemented.

   (c) State whether your approaches do or do not improve the performance by looking at the figures and ROC curves.

   (d) Two figures of edge images for your approaches. Two figures of ROC curve produced by $comproc()$ to compare your approaches to the approach in 4..