CS143 Introduction to Computer Vision
Homework assignment 1.

Due: Problem 1 & 2 September 23 before Class

Assignment 1 is worth 15% of your total grade.
It is graded out of a total of 100 (plus 15 possible extra points, which must be done for graduate credit).

Goals.

- Introduce you to Matlab and the manipulation of images and vectors in Matlab.
- Introduce convolution and filtering.
- Introduce image pyramids.
- Allow you to experiment with image blending
- Experiment with image feature detection.

Files you need.

The images that you should work with are in the directory

`/course/cs143/asgn/asgn1/.`

Additionally, we provided four Matlab scripts `problem[1-4].m` in the directory `/course/cs143/asgn/asgn1/`, which you should start from. Their main purpose is to make it easier for us to grade your homework. You are therefore asked to copy those files and modify them, so that they perform the appropriate task and display your results. You are of course free to write your own scripts and/or functions and only add calls to these to the scripts we provided.

What to hand in.

You will hand in your Matlab code (and our support code needed to run your program), which is supposed to show all your results, and any text or explanations

that are required. Make an effort to submit neat commented code so we can attempt giving you partial credit where necessaary. If the result is wrong and we cannot follow your line of reasoning, we will not be able to award you any points. Please hand in the modified `problem[1-4].m` scripts as well as any other Matlab scripts or functions that your solution requires. Make sure that all your results are displayed properly! **Pay attention to the efficiency of your program. You will lose points if your program is very slow.**

You will also hand in any text or explanations electronically. They should either be in plain text format, in Postscript, or in PDF.

To handin the assignment run the commands below from the folder that contains your solutions. The entire contents of that folder will be zipped and sent to us.

For assignment 1, problems 1 and 2: /course/cs143/bin/handin asgn1_p1_p2
For assignment 1, all problems: /course/cs143/bin/handin asgn1all

General Remarks.

You may use built-in Matlab routines to perform convolution and image resizing. See the Collaboration Policy on the website:

`http://www.cs.brown.edu/courses/cs143/collaboration.html`

Problem 1. Image pyramids.

(20pts) For the "Mona Lisa" image, build a 5 level Gaussian pyramid and display it in a format *similar* to the one in the figure.

Also implement and display a Laplacian (difference of Gaussian (DoG)) pyramid.

Problem 2. Image derivatives.

(20pts) In this part of the assignment you are to compute the first partial derivatives of the Barbara image in the x and y directions. Do this at every scale in the Gaussian pyramid and show the results. Also state which masks you used to implement your derivative filters.

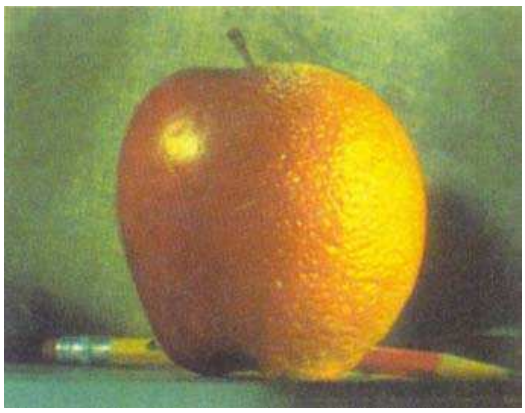
Using these derivatives, compute the gradient magnitude at each scale; ie $|\nabla I| = \sqrt{I_x^2 + I_y^2}$.

Choose a threshold to detect edge locations using the gradient magnitude. Show the detected "edge" locations as black points on a white image.

Write a paragraph explaining how you chose the threshold and what problems this approach might have for edge detection.

(5 extra pts) Combine information across scales in your detection of edges above. Explain what you did, how it worked, and why.

Problem 3. Image Blending.



For this problem you will be reconstructing the blended image shown above. This technique is introduced as *image blending* and is a way of splicing to images together in such a way that the transition between them is smooth and natural looking.

This problem uses color images. You can think of color images is being three separate images, one for each color channel. We have taken care of loading, splitting the color channels, and displaying the results for you, so

you only need to work with the individual color components, which work the same way as the grayscale images you worked with previously.

See section 3.4.4 (pp. 165) of your textbook for details on the algorithm.

(5 extra pts) Find two other images and blend them. This requires that the images are alligned, the same size, and that you create a mask between them.

Problem 4. Image derivatives and image features.

The local spatial correlation captures something about the local image structure. It can be expressed as

$$c(x, y, \Delta x, \Delta y) = \frac{1}{|W(x, y)|} \sum_{x_k, y_k \in W(x,y)} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$$

where $I(x, y)$ is an image, $W(x, y)$ is a neighborhood of pixel locations centered on (x, y) and $(\Delta x, \Delta y)$ represents some spatial shift in the image.

(10pts) Derive an approximation for the spatial correlation function as

$$c(x, y, \Delta x, \Delta y) \approx [\Delta x \ \Delta y] A [\Delta x \ \Delta y]^T.$$

Your task is to figure out what the matrix A should be.

Start by approximating $I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y)$ by its first order Taylor approximation. Recall that the Taylor series expansion of a function is

$$f(x_0 + \Delta x) = f(x_0) + \frac{d}{dx} f(x_0) \Delta x + \frac{d^2}{dx^2} f(x_0) \frac{\Delta x}{2!} + \cdots$$

You only need the first order approximation. Once you have this approximation, constructing A is a matter of rearranging things into a matrix form. Show your work!

Hint: A will be a 2×2 matrix constructed from the first partial image derivatives. For those who are interested, it is a positive semi-definite matrix.

(15pts) Compute this matrix at every pixel in the CIT image. Take W to be a 5×5 window.

The matrix A is sometimes called the "structure tensor"; it captures information about the local variation in the image gradient. We can understand something about the local image structure by examining the eigenvectors and eigenvalues of this matrix.

In particular, the eigenvectors provide a coordinate system that is aligned with the principal axis of the image variation; that is, the eigenvectors tell us about the local orientation of image features.

The eigenvalues (which are always non-negative) tell us something about what *kinds* of image structures are present. If both eigenvalues are small, then the local neighborhood is roughly homogeneous. If one eigenvalue is large and the other is small, then there is an edge present in a particular orientation. If they are both large, then there is a more complex structure such as a corner or other junction.

The eigenvalues can be computed using Matlab's `eig` function

```
[V,D] = eig(A);
```

This returns a diagonal matrix of eigenvalues along the diagonal of D. The eigenvectors are stored column-wise in V.

Using a sparse set of pixel locations (e.g. every 4th pixel, or other value you decide is appropriate), draw an ellipse oriented along the principal axis of the image variation at that point (as given by the eigenvectors of the structure tensor) with semiaxis proportional to the eigenvalues of the structure tensor at that point. You should obtain an image similar to the example we give on page 6 of this handout.

Hint: We provide a matlab function for drawing ellipses:

```
ellipse(ra,rb,ang,x0,y0,C);
```

The ellipse drawn has axis ra and rb, is centered at the point $x0, y0$ and rotated by the angle ang. The color of the ellipse is C. Figure out how to compute the angle and the ratio between ra, rb and the eigen values (you may want to test it on simple image first).

Hint: You need to adjust the scale of your ellipses. Since their size is proportional to the eigenvalues, you can divide by the maximum value to give a normalized size, and then multiply by a fixed scale for the desired size of the ellipse.

(15pts) When both eigenvalues are large, we will think of this as an “interest point” or “feature”.

Find the strongest 20 interest points at each scale in your pyramid. Show the locations of the points on the finest scale image. For every interest point found, do nonmaximum suppression. Compute an image that represents the importance of each feature in terms of the eigenvalues. Using this ”salience” image, first find the most salient feature. Then mask out (set to zero) the pixels in a neighborhood around this. Repeat the search for the next most salient feature and so on. Otherwise you may have feature points clustered around the same position.

Draw a circle around each point, with the radius of the circle indicating the scale at which the point was found (larger circles for coarse scales).

Hint: The following Matlab command draws circles:

```
rectangle('Position', [x y d d], 'Curvature', [1 1]);
```

The circle drawn inscribes a square with the upper left corner at (x, y) and an edge length of d.

Hint: You may want to test your feature detection algorithm on a very simple image first, for example just synthetic image of a rectangle.

Problem 5. Hybrid Images (200-Level Credit).

(15pts) Take a look at the images at: http://cvcl.mit.edu/hybridimage.htm

A hybrid image is made by combining one image where there are only low spatial frequencies and one that contains only high frequencies. Using your Gaussian and Laplacian pyramids get a low frequency image from a coarse level of the Gaussian pyramid and high frequency (band pass) image from a fine level of the Laplacian pyramids. Linearly combine these.

For this problem, you should find two images you would like to combine, align them and combine them at different spatial frequencies to create something comparable to what is shown on the web page above.

??? give them two test images?

??? how many points?