ENGG 111 - Homework Assignment 5

Part 1 - Programming Mutual Information

Program the Mutual Information computation in a Matlab. You can use the matlab code I showed in class as a basis for your program. The big difference here will be: your code should accept grayscale (double) images valued 0-1. It should send an error message if other types of images are supplied or if the range of values is outside 0-1.

Instead of assuming 256 possible pixel values, as was done in my example Matlab code, you will create 100 bins equally spaced and assign all values to these ranges when computing the joint and marginal distributions. The joint distribution will be 100x100 while the marginals will be 1x100.

Your function, given two input images should return the mutual information and the joint distribution normalized:

```
[MI h] = micomp (imIn1, imIn2);
```

Where MI is a 1x1 scalar representing the mutual information and h is the 100x100 joint distribution.

Test your code with the example MRI slices provided: mr1.png and mr2.png.

Write a script that will call your Matlab code. You should create a 3x2 panel of images with: a) mr1 overlapping itself with no shift and no rotation, b) the joint distribution of this pair of images, write the MI in the title of that panel, c) the same pair of images in the next row, but this time with mr1.png overlapped with itself after rotating it by 5° and d) the same pair of images again with mr1 overlapped with itself after shifting by [10 10] and rotating by 15°. In each panel (left column) the title should indicate the translation (r and c) values as well as the angles.

You will know if your code works correctly if the 1st joint distribution has the characteristic appearance resulting from an exact overlap, as we have seen in class.

To make things more interesting, time your code using the "tic" and "toc" Matlab calls. Run micomp with mr1.png and mr2.png through a loop of 50 times then average. Best time will get extra points.

Part 2 - Radial Distortion

In this exercise, I would like to introduce you to the use of the software package "ImageMagick" which is freely downloadable from the internet (lookup ImageMagick on google). You have the choice of either installing it on your machine (works with Mac & Windows) or doing this part of the exercise on one of the networked Linux machines at Thayer.

In class I showed a simple way to perform radial distortion correction. A more sophisticated approach is implemented in the program ImageMagick.

First, some excellent reading on using "convert" to correct distortion at: http://www.imagemagick.org/Usage/distorts/

Second, be sure to read the section on Barrel Distortion in that page.

Using the command in the form:

convert -distort barrel "A B C" imIn.jpg imOut.jpg

where A B C are numerical values. Start by trying the example in the pages cited. Experiment with values for A B C (note that I leave D out, as per the explanations in the referenced pages). find the combination of A B C works best to straighten out the grid in the test image provided (Grid2.jpg).

If you would like you can use matlab to make system calls to the operating system using one of two methods. The first consists of giving matlab a command preceded by an exclamation point. It will be executed by the operating system:

!unixCommand argument1 argument2

where unixCommand can be any unix (or Windows) command that works on your machine. Another method is to create a string to be executed and then use the "system('commandString')" call in Matlab.

Show a few example of barrel and pincushion distorted replicas of the original image, and give the best combination of ABC that produces the best image.

I suggest you present a 3 rows by 2 column figure with subplots showing (Upper Left) highly distorted pincushion, (Upper Right) moderately distorted pincushion, (Middle Left) original image, (Middle Right) best image (with ABC values in title), (Lower Left) moderately distorted pincushion, (Lower Right) highly distorted pincushion.

In your report, show the commands that produced the figures shown.