ENGG 111 - Homework Assignment 2

Part 1 - Intensity Transformations

Following the discussion in section 3.2.4 in Gonzalez & Woods, write a function in Matlab that will implement a Piecewise Linear Mapping $s = T(r)$, based on two points in the transformation function diagram (see figure 3.10(a) page 116 for example, which is reproduced below). Write a function that will take two inputs and produce one output:

```
imOut = plm(imIn, [ r1 s1 r2 s2])
```

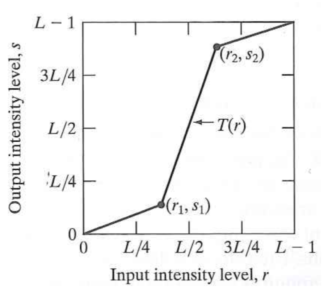
The 1st input is an image, the 2nd input is an array of values representing the two points (4 values) defining vertices in the PLM. The function will return an image imOut with its contrast stretched according to the linear mapping represents by the points (r1,s1) and (r2,s2).

Your piecewise linear mapping should implement a transformation that can be drawn with three line segments going from: (0,0) → (r1,s1) → (r2,s2) → (1.0,1.0). Use the "pout.tif" figure to test it out. Be sure to be consistent in how the program operates with either integer (uint8) and double image types.

Write a script that calls your function. This script should create two panels, one that shows the PLM (should look like the graph below) and one that contains the image after intensity the transformation represented by the PLM has been applied to the input image.

Use the ginput Matlab function inside a loop to collect the two points necessary to define the PLM. You can make it so that upon starting your PLM is straight from (0,0) to (1,1) with the two points at 1/3 and 2/3 of the diagonal.

Make the PLM vertex points visible, plot the PLM lines in black and the vertices in red.



Part 2 - Spatial Filtering

The median filter is a special case in a more general class of filters called "order-statistics filters." In this method of filtering, the output value is selected to be the nth item in the ordered (sorted by intensity values in the case of images) data points or pixels. This type of filters is much less sensitive to outliers in a data set and is very effective for noise removal in images.

One can expand this idea and create a "min" and a "max" filter which for a given kernel size will pick the min or max filter for that sub image and assign it to the output value.

Using the Matlab command imOut=ordfilt2(imIn,n,kernel); explore the following:

a) Compute the min filter by setting the arguments properly in ordfilt2()

b) Compute the max filter

c) Compute the difference (max - min) which is also known as the range

repeat this process for several kernel sizes (3, 5, 11, 31). Write a script that will perform this entire exercise and display the results. Use the image liftingbody.png (aka flying bathtub) as a test image and convert it to type double to make things simpler. Make sure to adjust your image after filtering. For each of these kernel sizes, create a 2x2 figure with the original image, the min-filtered version, the max-filtered version and the difference image. Make sure to properly scale your images. Explain the results. What is happening at the edges? What about the rest of the image?

Looking at the result with the 3x3 kernel. What function do you think this could approximate? We will see a better way of accomplishing that differently.

Part 3 - Image Enhancement

Referring to the last section in Notes 05, the subsection titled "Unsharp Mask Revisited," your task will be to implement a Matlab function such that it takes 2 inputs and produces one output:

```
imOut = enhance(imIn, V);
```

The function enhance will implement the unsharp mask operation on the image imIn to produce the output image imOut. It will in effect implement equation 0.28.

In the vector V you will send the following parameters to the function:

$$V = [\ k_1\ k_2\ r_o]$$

where k_1 and k_2 are the parameters of equation 0.28, and r_0 is the filter parameter. Use your favorite filter from the filters I presented. If it requires an additional parameter, then include it in V.

In you submitted assignment, demonstrate the effect of your filter for a few choices of parameters. For your test image, use the "WallPaper.jpg" image provided which is a low contrast color image. Write a script that will load the image, change it to grayscale , enhance it (with the parameters of your choice), and display an output image with the top half being unchanged and the bottom half being your enhanced image. I expect to see an image that has the appearance of much better defined edges.

Bonus points: how would you change your function to handle color images? Show an example.