ENGG 111 - Homework Assignment 4

Part 1 - Grayscale Erosion and Dilation

This homework set is intended to have you explore new morphological algorithms on grayscale images. This is discussed at length in section 9.6 of G & W, you should make sure to read it and understand it.

You should write your code without resorting to built-in Matlab functions, except for the most basic ones. More specifically, do not use imerode imdilate and similar operations provided in Matlab. Write your Matlab code to be very explicit about what you do. Also include lots of comments so we can understand what you are doing. Lack of comments may cost you points. Execution speed here is not the goal, you will not be graded on code efficiency, although if you were to vectorize as much as possible that would be fine.

Write a Matlab function such that:

```
imOut = erodeDilate(imIn, SE, flag);
```

will return an eroded or dilated version of the grayscale image imIn into imOut. Remember how erosion and dilation are defined for gray images, see the notes or G & W. The image types will be double (0.0 to 1.0) to keep things simple. The actual operation performed by the function will be erosion if the flag is 0 and dilation if the flag is 1.

For the purpose of this homework, the structuring element matching will be limited to the ones ("1") in the array, the zeros in the SE represent "don't care" pixels. Furthermore, for this exercise, the SE should be presented as a matrix, do NOT use the strel function to create SEs.

Test your SE on the following images:

```
im1=imread('rice.png');   % Matlab supplied test image

im2=imread('son1.gif');   % Image file supplied in a previous homework
```

Try different size SEs. I suggest using a circular (to the extent that you can approximate one) disc with radii varying from 3 to some size sufficiently large to remove all the included objects (i.e. grains of rice or letters). Show 4 panels each with a series of output images dilated or eroded versions of the test images with different size SEs.

I suggest you use the "montage" command to create panels with many images. I also bring to your attention the "text" command which you can use to label the individual images on a montage using a bright color, either yellow or red. Write a script that uses a loop to go through the various combinations, you should produce four figures with many frames in each.

Part 2 - The Top-Hat and Bottom-Hat Transformations

(I didn't make up this terminology)

The Top-Hat Transformation is defined for grayscale images as:

$$g(x, y) = f(x, y) - (f(x, y) \circ b);$$

where b is the SE, g(x,y) is the output image and f(x,y) is the input image. Note that since we are dealing with grayscale images here, we are operating on images, not regions. Remember that the open circle designates the "opening" operation, which itself is based on both erosion and dilation.

Similarly the Bottom-Hat Transformation is defined for grayscale images as:

$$g(x, y) = (f(x, y) \bullet b) - f(x, y)$$

Write a function that will use your MEX (or Matlab) function erodedilate to perform the top and bottom hat transformation (do NOT use Matlab's built in function):

```
imOut = topBottomHat(imIn,SE, flag);
```

the flag is used to pick top (1) or bottom (0) hat. Here again we use the same notation as in the notes, with the black circle indicating the "closing" operation.

Show four figures with montages of using your function on the two test images (similar to above).

While your homework results should be based on your own erodeDilate function, you are encouraged to compare your results with those of Matlab's builtin functions.

Part 3 - Using The Top-Hat Transformation for Shading Correction

Based on your findings in the first two parts, you should have an idea of what size SE will remove all the enclosures (grains of rice or letters) in the two test images and leave you with the Background only. Using these SEs, apply the top-hat transformation to the test images and threshold them after correcting their background.

For each test image, show the original image, the thresholded original image (with uncorrected background) and the thresholding obtained after correcting the background with the top hat transformation.

How does the thresholded images compare when using the top-hat transformation and without it.