

Name: _____

CSE 634

Computer Vision for HCI

AU'11

Homework Assignment #2

Due: Tuesday 10/11

- 1) **Write a Matlab function to compute and display the 2D Gaussian derivative masks G_x and G_y for a given sigma (see class notes). Note: each mask is a square 2D matrix. [3 pts]**

```
[Gx, Gy] = gaussDeriv2D(sigma);
```

- 2) **Compute and display the gradient magnitude of an image (search the web for an interesting image; convert to grayscale if necessary; make sure to send image to grader with code). [2 pts]**

```
gxIm = Imfilter(Im, Gx, 'replicate');  
gyIm = Imfilter(Im, Gy, 'replicate');  
magIm = sqrt(gxIm.^2 + gyIm.^2);  
imagesc(magIm);
```

- 3) **Threshold and display the magnitude image with different threshold levels. [2 pts]**

```
tIm = magIm > T;  
imagesc(tIm);
```

- 4) **Compare the above results with the Sobel masks. [2 pts]**

```
Fx=-fspecial('sobel');  
fxIm = imfilter(Im,Fx);  
Fy=-fspecial('sobel');  
fyIm = imfilter(Im,Fy);  
magIm = sqrt(fxIm.^2 + fyIm.^2);  
tIm = magIm > T;  
imagesc(tIm);
```

[next page]

- 5) Run the matlab canny edge detector, `edge(I, 'canny')`, on your image and display the default results. [2]
- 6) Using the grayscale images (`walk.bmp`, `bg000.bmp`) provided on the WWW site, perform background subtraction 1 using simple image subtraction to identify the object. Experiment with thresholds. [2 pts]
- 7) Using the grayscale images (`walk.bmp`, `bg[000-029].bmp`) provided on the WWW site, perform background subtraction 2 using statistical distances. Experiment with thresholds. [4 pts]
- 8) Dilate the binary image resulting from step 7) using [1 pt]


```
d_bslm = bwmorph(bslm, 'dilate');
```
- 9) Perform a connected components algorithm, and keep only the largest region in L (save/display as an image). [1 pt]


```
[L, num] = bwlabel(d_bslm, 8);
```
- 10) Turn in all code, printouts of images, and discussion of results. Make a `HW2.m` script to do the above tasks and call needed functions. [2 pts]

Make sure to email your code (and selected image) to the grader.

Matlab help:

A useful way to load several images:

```
for i=1:N
    filename = sprintf('myimage_%03d.bmp', i);
    Im(:,:,i) = double(imread(filename));
end
```

The `%03d` gives a zero-padded number (e.g., 001), and the image storage is now a 3-D cube. To get image 3 from the cube, use `myIm = Im(:,:,3)`.

With such a cube, Matlab operations such as 'mean' can be told to work along certain dimensions of the cube.