

Assignment A2: Spatial Filtering

CS 4640
Spring 2018

Assigned: 23 January 2018

Due: 8 February 2018

For this problem, handin Matlab .m files for the functions described by the headers below. Note that one of these is a driver which creates inputs for each function and runs the function on those inputs to obtain the output.

None of the functions should write to the interpreter, draw, etc.

```
function [mag,ori] = CS4640_edge(im)
% CS4640_edge - compute edge magnitudes and orientations
% On input:
%     im (MxN array): input image
% On output:
%     mag (MxN array): edge magnitudes
%     ori (Mxn array): edge orientations (gradient direction)
% Call:
%     [mag,ori] = CS4640_edge(im);
% Author:
%     <Your name>
%     UU
%     Spring 2018
%
```

```
function H_im = CS4630_Hessian(im)
% CS4640_Hessian - compute the eigenvalues of the Hessian at each
%                  pixel
% On input:
%     im (MxN array): graylevel image
```

```

% On output:
%     H_im (MxNx2 array): 2-channels with eigenvalues of Hessian
% Call:
%     H = CS4640_Hessian(im);
% Author:
%     <Your name>
%     UU
%     Spring 2018
%

```

```

function imo = CS4640_median(im,s)
% CS4640_median - compute median value at each pixel
% On input:
%     im (MxN array): input image
%     s (int): median window side length (must be odd)
% On output:
%     imo (MxN array): median filtered image
% Call:
%     imo = CS4640_median(im);
% Author:
%     <Your name>
%     UU
%     Spring 2018
%

```

```

function im =
CS4640_create_im_gen(f_name,M,N,Q,x_min,x_max,y_min,y_max,...
    args)
% CS4640_create_im_gen - create an image from a function with
% arguments
% On input:
%     f_name (string): name of function
%     M (int): number of rows in image
%     N (int): number of cols in image
%     Q (int): number of quantization levels in image
%     x_min (float): min x value for planar patch
%     x_max (float): max x value for planar patch
%     y_min (float): min y value for planar patch
%     y_max (float): max y value for planar patch

```

```

%      args (1xq vector): arguments for function (in addition to x,y)
% On output:
%      im (MxN array): image
% Call:
%      im = CS4640_creat_im_gen('CS4640_Log',100,100,64,-4,4,-4,4,2);
% Author:
%      <Your name>
%      UU
%      Spring 2018
%

```

```

function imo = CS4640_zc(im)
% CS4640_zc - zero crossings
% On input:
%      im (MxN array): input image
% On output:
%      imo (MxN array): zero crossings in image
% Call:
%      imo = CS4640_zc(im);
% Author:
%      <Your name>
%      UU
%      Spring 2018
%

```

```

function h = CS4640_Log(x,y,sigma2)
% CS4640_Log - Laplacian of Gaussian function
% On input:
%      x (double): x value
%      y (double): y value
%      sigma2 (double): variance of Gaussian
% On output:
%      h (double): LoG value
% Call:
%      h = CS4640_create_im_gen(3,2,2);
% Author:
%      <Your name>
%      UU
%      Spring 2018
%

```

```

function [imo,imzc] = CS4640_LapG(im,s,sigma2)
% CS4640_LapG - Laplacian of Gaussian
% On input:
%     im (MxN array): input image
%     s (int): side length of LoG filter
%     sigma2 (double): variance of Gaussian filter
% On output:
%     imo (MxN array): LoG image
%     imzc (MxN array): zero-crossings
% Call:
%     [imo,imzc] = CS4640_LaG(Tlgs,7,3);
% Author:
%     <Your name>
%     UU
%     Spring 2018
%

```

```

function CS4640_A2_driver
% CS4640_A2_driver - driver for A2 functions
% On input:
%     N/A
% On output:
%     N/A
% Call:
%     CS4640_A2_driver
% Author:
%     <Your name>
%     UU
%     Spring 2018
%

```