# Assignment A7: Segmentation II

***CS 4640***
***Spring 2018***

**Assigned:** 10 April 2018

**Due:** 26 April 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.

Some notes:

- Indent headers correctly (5 spaces indented lines)
- Do not exceed 72 characters per source line
- CS4640_A7_driver: should show that each function works

None of the functions should write to the interpreter, draw, etc., unless explicitly required by the function description.

```
function [x,y] = CS4640_ac_initial_box(r1,c1,r2,c2,gap)
% CS4640_ac_initial_box - initialize rectangular snake points
% On input:
%     r1 (int): upper left corner row
%     c1 (int): upper left corner col
%     r2 (int): lower right corner row
%     c2 (int): lower right corner col
%     gap (int): gap between snake pixels
```

```
% On output:
%     x (kx1 vector): x coordinates of snake points
%     y (kx1 vector): y coordinates of snake points
% Call:
%     [x0,y0] = CS4640_ac_initial_box(5,5,25,25);
% Author:
%     <Your name>
%     UU
%     Spring 2018
%


function [xs,ys] =
CS4640_ac_initial_circle(M,N,r0,c0,radius,del_theta)
% CS4640_ac_initial_circle - initialize circular snake points
% On input:
%     M (int): row size of image
%     N (int): col size of image
%     r0 (int): row center of circle
%     c0 (int): col center of circle
%     radius (float): radius of circle
%     del_theta (float): step in theta for circle
% On output:
%     x (kx1 vector): x coordinates of snake points
%     y (kx1 vector): y coordinates of snake points
% Call:
%     [x0,y0] = CS4640_ac_initial_circle(31,31,16,16,9,0.1);
% Author:
%     <Your name>
%     UU
%     Spring 2018
%


function [MO,im_ac,x,y] =
CS4640_ac(im,x0,y0,alpha,beta,gamma,max_iter)
% CS4640_ac - compute active contour
% On input:
%     im (MxN array): gray-level image
%     x0 (Kx1 vector): x (row) locations for curve
%     y0 (Kx1 vector): y (col) locations for curve
```

```
%     alpha (float): coefficient for D2 array
%     beta (float): coefficient for D4 array
%     gamma (float): coefficient for external force vector
%     max_iter (int): max number of iterations
% On output:
%     MO (movie): movie of snake motion
%     im_ac (MxN array): binary array with final curve points
%     x (float): final x values of snake
%     y (float): final y values of snake
% Call:
%     sq = zeros(31,31);
%     sq(11:20,11:20) = 100;
%     [sqc,xf,yf] = CS4640_ac(sq,x0,y0,1,1,0.2,5000);
% Author:
%     T. Henderson
%     UU
%     Spring 2018
%


function phi = CS4640_phi_circle(M,N,r0,c0,radius)
% CS4640_phi_circle - create circular phi function
% On input:
%     M (int): number of rows in array
%     N (int): number of cols in array
%     r0 (int): center of circular function
%     c0 (int): column of circular function
%     radius (int): radius of circle
% On output:
%     phi (MXN array): each pixel has signed distance to circle
%     boundary
% Call:
%      phi = CS4640_phi_circle(201,201,25,25,3);
% Author:
%     <Your name>
%     UU
%     Spring 2018
%


function [MO,phi,tr] = CS4640_level_set(im,max_iter,del_t,r0,c0)
```

```
% CS4640_level_set - level set of image
% On input:
%     im (MxN array): gray level or binary image
%     max_iter (int): maximum number of iterations
%     del_t (float): time step
%     r0 (int): row center of circular level set function
%     c0 (int): column center of circular level set function
% On output:
%     MO (movie): movie of level set propagation
%     phi (MxN array): final phi array
%     tr (qx1 vector): sum(sum(abs(phi_{n+1}(r,c) - phi_{n}(r,c))))
% Call:
%     [MO,phi,tr] = CS4640_level_set(im,300,0.1,25,25);
% Author:
%     <Your name>
%     UU
%     Spring 2018
%


function CS4640_A7_driver
% CS4640_A7_driver - driver for A7 functions
% On input:
%     N/A
% On output:
%     N/A
% Call:
%     CS4640_A7_driver
% Author:
%     <Your name>
%     UU
%     Spring 2018
%
```