

Assignment A6: Mathematical Morphology

CS 4640
Fall 2019

Assigned: 22 October 2019

Due: 7 November 2019

For this problem, handin a report of your work, as well as the Matlab .m files for the functions described by the headers below.

Some notes:

- Indent headers correctly (5 spaces indented lines)
- Do not exceed 72 characters per source line

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

1. Develop a Matlab function, *CS4640_circle*, which produces a digital approximation to a circle.

```
function imc = CS4640_circle(M,N,r0,c0,radius)
% CS4640_circle - creates digital circle image (may not all lie in
% image)
% On input:
%     M (int): number of rows of image
%     N (int): number of cols of image
%     r0 (int): row of center point of circle
%     c0 (int): col of center point of circle
%     radius (float): radius of circle
```

```

% On output:
%     imc (MxN array): image with circle centered at (r0,c0)
% Call:
%     cir1 = CS4640_circle(51,51,26,26,10);
% Author:
%     <Your name>
%     UU
%     Fall 2019
%

```

2. Develop an algorithm to produce the correct skeleton for digital circles. This should determine that a connected component is a circle if it is “close enough.” Explain your reasoning, implement the indicated Matlab function, *CS4640_skel_cir*, and demonstrate it on a reasonable set of examples.

```

function im_skel_cir = CS4640_skel_cir(im)
% CS4640_skel_cir - find skeleton of circles
% On input:
%     im (MxN array): binary image
% On output:
%     im_skel_cir (MxN array): 1's at circle centers
% Call:
%     cirs = CS4640_skel_cir(im);
% Author:
%     <Your name>
%     UU
%     Fall 2019
%

```

3. Develop an algorithm using mathematical morphology methods which takes in a binary image and produces a data structure (vector struct) which has the following fields:

- **bounding box:** min row, min col, max row, max col
- **bounding box height:** max row - min row + 1
- **bounding box width:** max col - min col + 1

- **number of pixels:** number of 1's in connected component
- **boundary pixels:** kx2 array of row, col values of boundary (trace around the boundary in counter-clockwise order)
- **Euler number:** Euler number for connected component
- **Long vertical stroke:** Boolean for this feature
- **skeleton:** mx2 array of skeleton pixels

Describe how mathematical morphology methods were used to extract better connected components (e.g., suppression of small components, reconstruction of remaining components, etc.). Demonstrate results on images 45.jpg and 56.jpg.

```
function cc_data = CS4640_cc_data(im)
% CS4640_cc_data - connected component features
% On input:
%     im (MxN array): binary image
% On output:
%     cc_data (struct vector): connected component feature values
%         (k).bounding_box (1x4 vector): [min_row,min_col,max_row,max_col]
%         .height (int): height of bounding box
%         .width (int): width of bounding box
%         .num_pixels (int): number of foreground pixels
%         .boundary (kx2 array): [rows,cols] of boundary (follow
%             neighbors counter-clockwise around boundary)
%         .Euler (int): Euler number
%         .v_stroke (Boolean): has long vertical stroke
%         .skeleton (mx2 array): [rows,cols] of skeleton pixels
% Call:
%     cc45 = CS4640_cc_data(im45);
% Author:
%     <Your name>
%     UU
%     Fall 2019
%
```