# Assignment A5: Geometry and Mathematical Morphology

***CS 6640***
***Fall 2020***

**Assigned:** 13 October 2020

**Due:** 29 October 2020

1. Consider the map F34086A1.TIF. Use the right columns of the map from 9500 to end for the following character recognition problem. Create 10 character templates by taking isolated characters, and making a 20x20 template window. These should be binary arrays with a border of zeros around the character. Next write a character recognition function *CS6640_char_rec* as described by the header which performs as follows:

- classify every connected component
- extract the connected component in a rectangular window, C, with zeros on the border
- resize C to be 20x20
- for theta = 0:359,
  - rotate C by theta to get Cr
  - resize Cr to 20x20
  - for every template, T
    - measure distance from template (sum of $T \sim= Cr$)
  - assign connected component to nearest template, if below threshold

Report performance on those extracted columns from the image. Note that the output of the function has two channels: the first has each pixel labeled either 0, if its connected component does not match any template closely enough, or with the index of the closest within threshold template. The second channel has at every pixel the orientation of the best fit template for the pixel's connected component.

```
function imo = CS6640_char_rec(im,templates)
% CS6640_char_rec - character recognition from templates
% On input:
%     im (MxN array): binary image
%     templates (vector struct): templates
%      (k).window (trows x tcols array): image of character
%      (k).char (string): character
% On output:
%     imo (MxNx2 array): character classifications and orientations
%        channel 1: recognized characters (by template index)
%        channel 2: orientation in degrees of character
% Call:
%     imo = Cs6640_char_rec(im,templates);
% Author:
%     T. Henderson
%     UU
%     Fall 2020
%
```

2. Develop an image registration function, *CS6640_register* as shown in the header. To register:

- Get tform using the polynomial option as demonstrated in Example 7.4 in the text.
- Extract a and b (polynomial coefficients) from tform.
- Range r,c over M and N
  - Get [u v] coordinates
  - Change from u,v to row rw (using v), col cw (using u) (this is in warped image space)
  - If inside MwxNw bounds, then interpolate a value for the output image at r,c using values from the warped image.

Demonstrate your function by (1) registering franklin-tn.jpg to franklin.TIF (the base image), and (2) cell rotated 75 degrees with cell (the base image). For warped images, use cpselect to pick corresponding points in the basis and warped image (choose at least 12, and save structure with all points). Give the corresponding points in the report for these, and show the registration results.

```
function [imo,a,b] = CS6640_register(pts,ptsw,im,imw)
% CS6640_register - register two images
% On input:
%     pts (nx2 array): basePoints data from cpselect
%     ptsw (nx2 array): inputPoints data from cpselect
%     im (MxN array): base image
%     imw (MwxNw array): warped image
% On output:
%     imo (MxN array): warped image mapped to base image frame
%     a (10x1 vector): cubic polynomial coefficients for x
%     b (10x1 vector): cubic polynomial coefficients for y
% Call:
%     [imo,a,b] = CS6640_register(cpstruct1.basePoints,...
%          cpstruct1.inputPoints,cell,cellr);
% Author:
%     <Your Name>
%     UU
%     Fall 2020
%
```

3. Explore Matlab's *bwmorph* to improve semantic segmentation. E.g., Try erode and dilate to eliminate small components; look at skeletons (say, in roads), branch and end points, etc. Provide clear analysis of benefits of your approach, or lack thereof. Apply this to the map1 or bottle images.