

Assignment A5: Geometry

CS 4640
Fall 2019

Assigned: 10 October 2019

Due: 24 October 2019

1. Use image chars45.jpg to study shape analysis using character templates for the characters:

A, B, C, D, E, F, G, H, I, K, L, M, N, O, R, S, T, U, V, Y
a, c, d, e, f, g, h, i, k, l, m, n, o, p, r, s, t, u, v, y
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Use a character from the image as a template, T , for each character above. Suppose the template is 25×15 , then for an unknown character window, C , of size $M \times N$, determine a fitness measure based on the formula:

$$\frac{|T \ominus C|}{MN}$$

Write a character classification function, *CS4640_char_template*, which returns the measure. Use the function to classify all characters in the image. Describe performance results per character and overall. State which connected components were used as character templates.

```
function m = CS4640_char_template(T,C)
% CS4640_char_template - measure similarity of template and character
% On input:
%     T (25x15 binary array): character template
%     C (MxN binary array): input test character
% On output:
%     m (float): distance measure between template and test character
```

```
% Call:
%     m = CS4640_char_template(T_A,cc);
% Author:
%     <Your name>
%     UU
%     Fall 2019
```

2. Develop a character classification scheme based on the Procrustes method. Use the same templates as in 1, and find a way to get a robust starting point. Then use `bwboundarytrace` to get the boundary points. Coerce them to be the same length as the template (use 75 pixels). Then use the `D` output from Matlab *procrustes* function as a distance measure. Report performance per character and overall.

3. Develop a Matlab function, *CS4640_persp_t*, to find the perspective transform between two point sets. Demonstrate it on the licences plate example from the text. Use point distance squared for corresponding points as the measure of performance and give results on a set of known transformations (e.g., specify a set of points, specify a perspective transform, transform the points, use the function to find the transform and compare both the transforms and the transformed points).

```
function T = CS4640_persp_t(pts1,pts2)
% CS4640_persp_t - find perspective transform between pts1 and pts2
% On input:
%     pts1 (Nx2 array): first set of 2D points
%     pts2 (Nx2 array): second set of 2D points
% On output:
%     T (3x3 array): perspective transform matrix
% Call:
%     T = CS4640_persp_t(pts1,pts2);
% Author:
%     <Your name>
%     UU
%     Fall 2019
```