Georgia Institute of Technology
George W. Woodruff School of Mechanical Engineering
ME6406 Machine Vision (Fall 2017)

All programs should be written using MATLAB. MATLAB and its toolboxes are available in ME Computer Lab (MRDC2105). To save the image from the ME6406 website page, point the mouse cursor on the image and click on the mouse's right button.   Choose <Save image as>.

Assignment #2: Due **Thursday September 28, 2017**. (Solutions should include m-files, results, and an explanation of your results in a document. All m-files must be submitted electronically in a zipped file **through T-square**. A confirmation email will be sent on receipt of submission.

1. Hough Transform

   a. Show that $\begin{bmatrix} x_o & y_o \end{bmatrix}^T = \upsilon \begin{bmatrix} g_x & g_y \end{bmatrix}^T$
   where $\upsilon = \left(xg_x + yg_y \right) / \left(g_x^2 + g_y^2 \right)$.

   b. With the parameters (x_o, y_o), write a MATLAB program to perform Hough transform (foot of normal parameterization). Determine the equations that characterize the straight edges in 'HW2_cuttter.jpg'. Find the intersections of these edges.

   c. Use Hough Transform to find one of the circles.

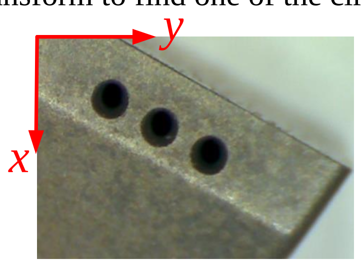

Fig. 1. HW2_cutter.jpg

2. Feature Points Detection

   a. In Matlab, write a program to extract the outer boundary of the object in 'HW2.jpg'. Plot the boundary over the image. Hint: Use Matlab function bwboundaries.m to trace region boundaries

   b. With the results obtained in a. Use ***rho-theta*** method discussed in class, implement using MATLAB, to locate all the corners of the object in 'HW2.jpg. Plot the $\rho\theta$ signature for locating the corners. Show the coordinates of the corners with respect to the coordinate system as shown. Compare them with the corners detected in Problem 1.

   c. Repeat (b) with ***curvature*** method, which is to locate the 'peaks' of the curvature along the boundary. Plot the graph that you used to locate the corners. Show the coordinates of the corners with respect to the coordinate system as shown.

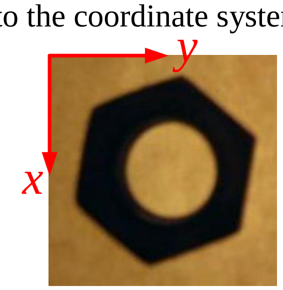

Fig.2 HW2.jpg

3. <u>Template Matching</u>

a. Forward transformation: Write a MATLAB function to perform the transformation for three template points with the parameters as shown in Table 1. Show and plot three points before and after transformation. The transformed points are accurate to three decimal places.

Table 1.

	Template			Parameters			
	1	2	3				
X	0	6	2	k	1.2	θ	30°
Y	0	3	7	x_d	8	y_d	5

b. Pseudo inverse method: Wirte a MATLAB function to perform the pseudo inverse method on the template and transformation points in part (a) to find parameters k, θ, x_d, y_d. Check these values with Table 1.

c. Polygon matching: Write a MATLAB program to perform template matching on the template and target as shown in Table 2 and Fig. 3 to identify the match points, and determine the transformation parameters; namely scale k, orientation θ, and displacement (x_d, y_d).

Table 2.

	Template				
	1	2	3	4	5
X	2	5	5	4	1
Y	1	2	4	5	3
	Target				
	a	b	c		
x	9.697	8.566	6.869		
y	6.96	8.091	5.263		

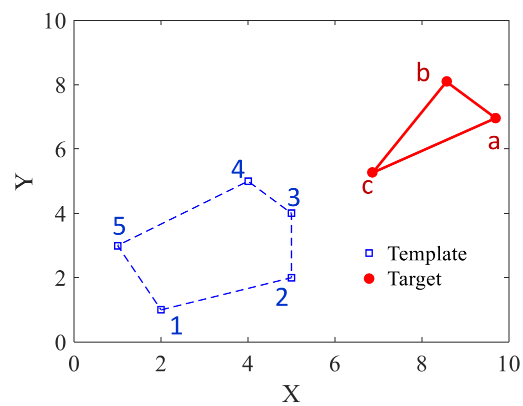

Fig. 3

Note 1: Given n points, there are $\binom{n}{3} = \frac{n!}{(n-3)!3!}$ unique triangles. Use Matlab function nchoosek( ) to compute all possible permutations of vertices to form unique triangles.

Note 2: As there are many triangles to be matched, your program must automate the matching process to avoid tedious manual matching.