Georgia Institute of Technology
George W. Woodruff School of Mechanical Engineering
ME6406 Machine Vision (Fall 2023)
Assignment #4: (ME6406Q Due **hard-deadline on Tuesday Dec. 5th 2023, 23:59pm EDT**)

## 1. Pose Estimation [1]

Figure 1(a) shows a calibration block with one circular pocket on its top face, where the *object coordinate system* $X_oY_oZ_o$ is defined at one of its lower corners, and the axes are aligned with the three orthogonal edges of the block (Fig 1b). An image (similar to Fig. 1b) of the component is captured by a camera at an unknown position/orientation; see Lecture slides for the definitions of the camera specification listed below:

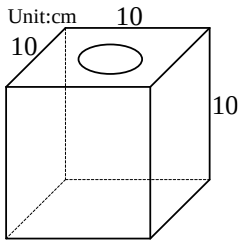


Fig 1(a)

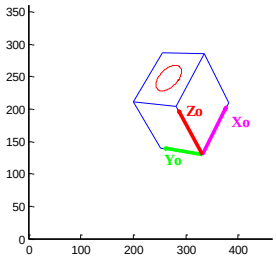


Fig 1(b)

| $f$ [cm] | $dx'$ [cm] | $dy'$ [cm] | $C_x$ [pixel] | $C_y$ [pixel] | $N_{cx}$ | $N_{cy}$ | $N_{cz}$ |
|---|---|---|---|---|---|---|---|
| 0.79 | $6\times10^{-4}$ | $6\times10^{-4}$ | 376 | 240 | –0.4226 | –0.6943 | –0.5826 |

a. Use Hough Transform to identify the 9 straight edges in '*block.png*'. You may use MATLAB built-in HT algorithm. Find the 9 normal vectors $\mathbf{n}=(af,bf,c)^T/|(af,bf,c)^T|$ as defined in [1].

   Note: The line equation $au+bv+c=0$ in [1] is defined on the 2D image plane, where $(u, v)$ are the physical coordinates measured from the center of the image plane; hence, when you obtain the coefficients associated with pixel coordinates, be sure to perform appropriate conversions so that the '$a$', '$b$' and '$c$' have consistent (physical) units.

b. Estimate the object pose using the circle/ellipse correspondence, the line correspondence (Results of Part 1a), and the orthogonal constraint. Use the normal vector $[N_{cx}\ N_{cy}\ N_{cz}]^T$ in the camera coordinates given above for the circle/ellipse correspondence. Determine the values of [**R**] and **T**.

   *Suggested functions: hough.m, houghpeaks.m, houghlines.m. Nonimal value of $T_z$ is 120 cm*

## 2. Artifical Neural Network (ANN) based on back-propagation (BP) learning

Design a fully connected artificial neural network (ANN) to recognize ⌜ and ⌟ on a binary 4x4 square grid. Use Fig. 2(a) to train and Fig. 2(b) to test the ANN by back-propagation learning:

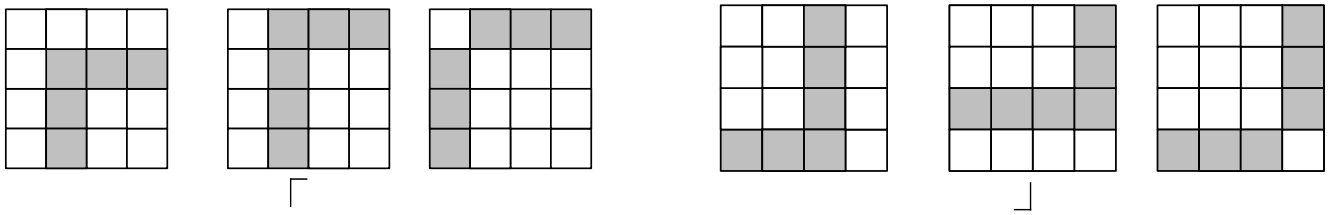

Figure 2(a) Training data

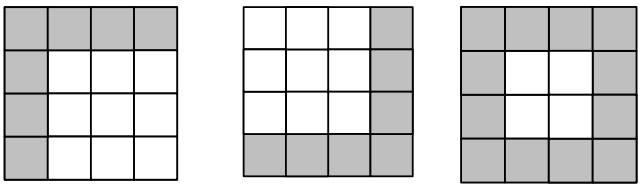


Figure 2(b) Test data

Perform the following steps to demonstrate your understanding:

a. Clearly define your neural network structure (the number of inputs, outputs, and hidden layers/nodes). Provide a schematic of your chosen ANN structure.

b. Derive the weight update rule assuming a uni-polar sigmoid function for each processing element.

c. Write a Matlab program (NN_train.m) to train the two data sets in Figure 2(a). Plot the converge curve (mean squared error vs. number of epoch) to validate your ANN algorithm. Save the final trained weights in a file (NN_weights.mat) and include it with the submission of your solutions.

d. Write a Matlab program (NN_test.m) to test the three data in Figure 2(b) by reading the weights (NN_weights.mat). Your solutions should show the output values and results. Hint: The range of the activation element, sigma function, is 0~1. The output values (0, 1) correspond to the two patterns (⌜ and ⌟); the closer the value to two ends means closer to the corresponding patterns whereas an intermediate value represents none of both.

## 3. Color

**I. Artificial Color Contrast (ACC):** Consider two color samples [225, 88, 96], [149, 135, 134] respectively representing target and noise in Fig 3, which are to be spatially separated in color space using DoG as discussed in class so that classification can be more easily performed:

$$h_i(x,y) = G_{\sigma c} * f_j(x,y) - G_{\sigma s} * f_k(x,y)$$

where $f_j(x,y)$ with ($j$=1, 2, 3) corresponds to *RGB* component images respectively; and $f_k(x,y)$ are some linear combinations of *RGB* component images to be designed.

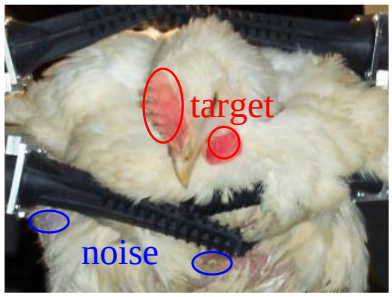

Fig 3. Chicken

a. Derive the following equations with $f_k(x,y)$ equal to $+(R-G)$ and $+(R+G-B)$:

$$h_1(x,y) = DoG * R + G_{\sigma s} * G \qquad h_a(x,y) = DoG * R + G_{\sigma s} * [B-G]$$
$$h_2(x,y) = DoG * G + G_{\sigma s} * [2G-R] \qquad h_b(x,y) = DoG * G + G_{\sigma s} * [B-R]$$
$$h_3(x,y) = DoG * B + G_{\sigma s} * [B-(R-G)] \qquad h_c(x,y) = DoG * B + G_{\sigma s} * [2B-(R+G)]$$

b. Perform the ACC transformation ($\sigma_c$=1, $\sigma_s$=10) on sample color patterns (100×100 each) with the following combinations: 1-2-3, 1-2-c, 1-b-3, 1-b-c, a-2-3, a-2-c, a-b-3, a-b-c.

**II. Color-based Image Segmentation:** Color is important information. Same objects commonly have their domain color. L-a-b color system is the color-opponent space with the L lightness dimension and a-b color-opponent dimensions. The color-based image segmentation can be performed by applying the clustering method on the points in a-b domain with the post process. Follow the steps to segment the target of the RGB image '*Chicken.jpg*'.

Step 1. Transfer pixels from RGB to Lab color system.

Step 2. Apply k-means clustering on data in a-b domain with cluster number ($k$=3).

Step 3. Erode the segment image to filter out small fragments.

**III. Principle component analysis (PCA):** Use the RGB image '*Chicken.jpg*' for the following.

a. Determine the covariance matrix of data.

b. Derive the components (eigenvectors) with eigenvalues arranged in a descending order.

c. Obtain the maximum and minimum values of three component matrices. Show these three matrices (images) with linear mapping from the minimum and maximum values to the range of (0-255).

**Reference:**

[1] Qiang Ji, Mauro Costa, Robert Haralick, and Linda Shapiro, "An Integrated Linear Technique for Pose Estimation from Different Features," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 13, No. 5, 1999

[2] Simpson, P. (1992). Foundations of neural networks, Chapter 1 in *Artificial Neural Networks*, Lau, C. & Sanchez-Sinencio, E. (Eds.), pp. 2-13, IEEE Press, New York, NY.

Other handouts posted in Canvas: 00_ANN2.pdf and 00_ANN03-Example.pdf