UNIVERSITY OF TEXAS AT ARLINGTON
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

6367 **COMPUTER VISION** **SPRING 2019**

ASSIGNMENT 2 (100 POINTS)
ASSIGNED: 2/5/2019 DUE: 2/19/2019

This assignment constitutes 10% of the course grade. You must work on it individually and are required to submit a PDF report along with the MATLAB scripts described below.

Problem 1 (40 points)

The objective of this problem is to learn to how to implement basic image feature extraction operations.

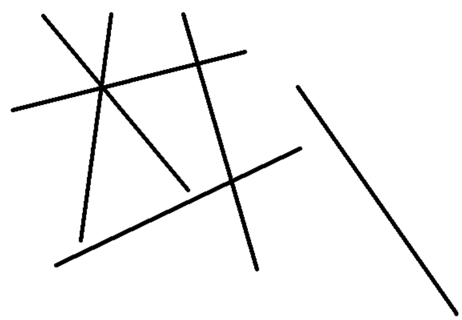

Figure 1: A set of lines.

Write a MATLAB script that implements the Hough transform. It must compute the parameters for all the lines present in the accompanying image "line.jpg" (shown in Figure 1). You will need to do edge detection before you apply the Hough transform and may use the MATLAB command `edge` with the method of your choice (Canny, Sobel, Roberts). However, you cannot use the MATLAB command `hough`.

Submission Instructions: *Submit a MATLAB script, hough.m, that performs all the operations stated above, along with any other files necessary to run the script. Using the line parameters computed from your Hough transform, create a new image by overlaying the detected lines onto the lines in the original image. You are required to include this image by embedding it in the PDF report (do not submit the image separately).*

Problem 2 (30 points)

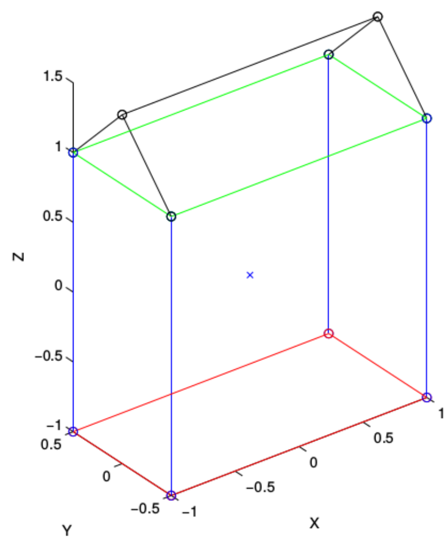

Figure 2: A wireframe house.

A simple wireframe house is shown in Figure 2. In this problem, you have to determine the image of the house as seen by placing the camera at various locations. The camera has scaling factors $\alpha = \beta = 200$ units, the image center is at $(50, 50)$, and it has zero skew. The coordinates of vertices of the house in the world frame are given as,

$$
{}^{W}P_i = \begin{bmatrix} -1 & -0.5 & -1 \\ -1 & 0.5 & -1 \\ 1 & 0.5 & -1 \\ 1 & -0.5 & -1 \\ -1 & -0.5 & 1 \\ -1 & 0.5 & 1 \\ 1 & 0.5 & 1 \\ 1 & -0.5 & 1 \\ -1 & 0 & 1.5 \\ 1 & 0 & 1.5 \end{bmatrix}
$$

(a) Write a MATLAB function `P_C = project_points(P_W, R, t)` that takes as input an $N \times 3$ vector of points with coordinates in the world frame and returns as output an $N \times 2$ vector of coordinates of points in the camera frame. `R` and `t` are the 3×3 rotation and 3×1 translation matrices from the camera-centric to world frame.

(b) Write a MATLAB script problem_2.m that uses `project_points` to determine the projection of each vertex of the house in the image, when the camera is placed at the following positions: (i) [10, 10, 0], (ii) [-10, 10, 0], (iii) [0, 0, 10], (iv) [10, 0, 0], and (v) [10, 10, 10]. In each case, the camera axis directly passes through the origin of the world coordinate frame. Display the generated images in a separate window for each camera location. Plot the lines joining the vertices of the house, as shown in Figure 2 for each of the images. You may find it useful to have separate colors for separate lines, and maintain the color scheme across images. For simplicity you may ignore occlusions, which may occur with a real camera, and simply display all the lines and vertices.

Submission Instructions: *Submit a MATLAB function project_points.m and a script problem_2.m that performs all the operations stated above, along with any other files necessary to run the script. Please take care to generate all the figures in new windows. You are also required to include the images by embedding them in the PDF report (do not submit the images separately). Clearly mark the camera location for each image in the report.*

Problem 3 (10 points)

Show that the determinant of a rotation matrix is ± 1.

Problem 4 (10 points)

(a) Let R_1 and R_2 be two rotation matrices on the plane. Prove or disprove the following: $R_1 R_2 = R_2 R_1$.

(b) Let R_1 and R_2 be two rotation matrices in 3D space. Prove or disprove the following: $R_1 R_2 = R_2 R_1$.

Problem 5 (10 points)

Write a MATLAB function `[A, theta] = get_axisangle(R)` that takes a rotation matrix `R` as input and returns the rotation axis `A` and angle `theta`.

Submission Instructions: *For Problems 3 and 4, submit the typed solutions in the PDF report. For Problem 5, submit a MATLAB function get_axisangle.m that performs the necessary operations.*