UNIVERSITY OF TEXAS AT ARLINGTON
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

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

ASSIGNMENT 1 (100 POINTS)
ASSIGNED: 1/22/2019 DUE: 2/5/2019

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 (50 points)

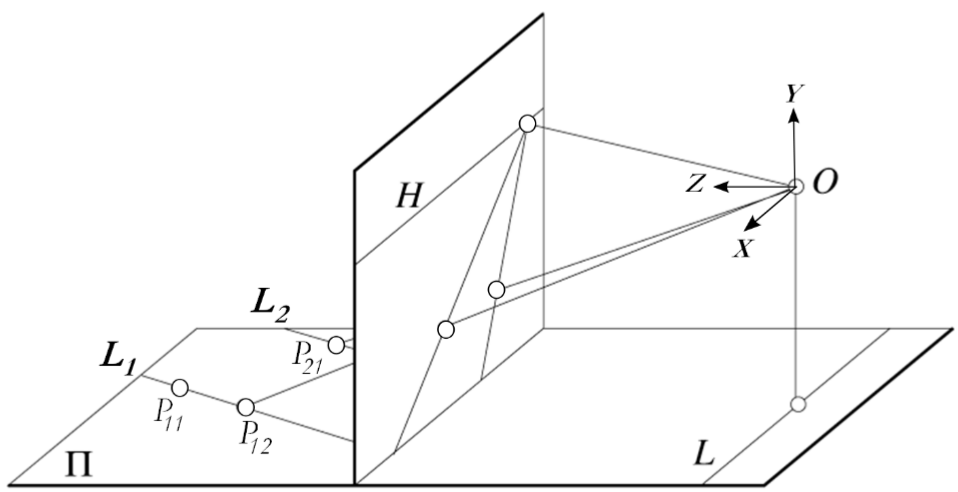

Figure 1: The projection of parallel lines.

(a) In Figure 1, the origin is co-located with the camera center at point O. The coordinate axes are as shown (the XZ plane is parallel to Π). The equation of plane Π is given as $y = -1$ while the equation of the image plane is $z = 1$. Π contains three parallel lines L_1, L_2 and L_3. The points $P_{11} = [-1, -1, 2]^T$ and $P_{12} = [-1, -1, 3]^T$ lie on line L_1, $P_{21} = [0, -1, 2]^T$ and $P_{22} = [0, -1, 3]^T$ lie on line L_2 and $P_{31} = [1, -1, 2]^T$ and $P_{32} = [1, -1, 3]^T$ lie on line L_3. Write a MATLAB function, `Q = project_point(P)`, that computes the projection Q on the image plane of a given point P. Apply this function to each of the given points in a MATLAB script. The center of projection is at the origin O. The projection can be computed by determining the point of intersection of the line joining the origin and the given point with the image plane.

(b) Using the function `project_point`, write a MATLAB function `Q = find_intersection(P11, P12, P21, P22)` that takes as input two points from each line, `L1` and `L2`, and computes the point of intersection Q of the lines projected on the image plane. Plot the given lines, their projections, and the point of intersection using MATLAB visualization functions (e.g. `plot3`).

(c) Verify that the point of intersection for each pair of parallel lines L_1, L_2, and L_3 is the same by applying the `find_intersection` function to each pair.

(d) Consider three pairs of parallel lines on the plane Π given by the following: (i) $x - 1 = 0$ and $x = 0$, (ii) $3x + 2z - 1 = 0$ and $3x + 2z - 2 = 0$, (iii) $5x - 2z - 1 = 0$ and $5x - 2z - 2 = 0$. Write a MATLAB script `pairwise_intersection` that uses `find_intersection` to determine the point of intersection for each pair. Verify that the three points found are collinear. Plot the given lines, their projections, and the points of intersection.

Submission Instructions: *Submit the following three files - project_point.m, find_intersection.m and pairwise_intersection.m along with any other m-files necessary to run your MATLAB code. In addition, submit a PDF report that embeds all of the generated plots. We must be able to run your code and produce the same output shown in the report. Please provide an appropriate title for each plot (do not submit the images separately). The MATLAB command* `print` *may be helpful in this regard.*

Problem 2 (50 points)

The objective of this problem is to introduce basic image processing operations in MATLAB. Load the accompanying "board.tif" image in MATLAB.

(a) Extract the rectangular block of the image between $(200, 90)$ and $(300, 180)$ corresponding to the crystal in the image. Display this block in a separate figure.

Solve the next two questions in the following three different ways: (i) using nested for loops and accessing each pixel individually, (ii) using MATLAB matrix operations, (iii) using built-in image processing functions `rgb2gray` and `im2bw`. Use the MATLAB profiler to compare the running times.

(b) Convert the image from RGB to grayscale. For each pixel, take the average of the R, G, and B values as the grayscale value.

(c) Convert the grayscale image to a binary image using the mean grayscale value as the threshold. Display both the grayscale and binary images in the same window.

(d) Smooth the grayscale image created above using a 7×7 averaging filter. This means that for each pixel at location (i, j), place a 7×7 window centered at (i, j) and replace the value of the pixel with the average of the values of the pixels in the window. Decide how you will handle pixels close to the image boundary. Solve this problem in the following two different ways: (i) using for loops, (ii) using the Matlab function `conv2`. Display the smoothed images.

Submission Instructions: *Submit a single MATLAB file, problem_2.m, that reads in the image and performs all of the operations stated above. Please take care to generate all the figures in new windows. You are also required to embedded all of the images in the PDF report (do not submit the images separately). The MATLAB command* `print` *may be helpful in this regard. Each sub-problem must be marked separately and clearly.*