UNIVERSITY OF TEXAS AT ARLINGTON
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

6367 **COMPUTER VISION** **SPRING 2020**

ASSIGNMENT 2 (100 POINTS)
ASSIGNED: 2/11/2020 DUE: 2/27/2020

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)

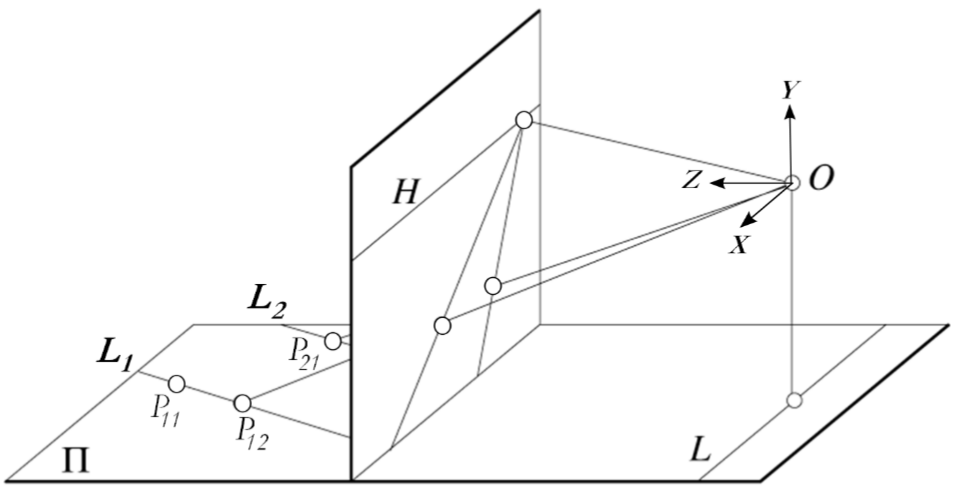


Figure 1: The projection of parallel lines.

(a) (15 points) 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) (15 points) 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) (5 points) 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) (15 points) 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.

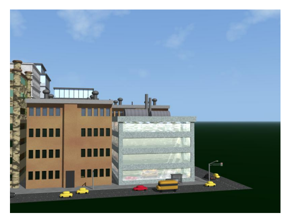
Submission Instructions: *Submit the following three files - project_point.m, find_intersection.m and pairwise_intersection.m along with any other m-files necessary to run your MATLAB code. In addition, embed all of the generated plots in the report. We must be able to run your code and produce the same 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)

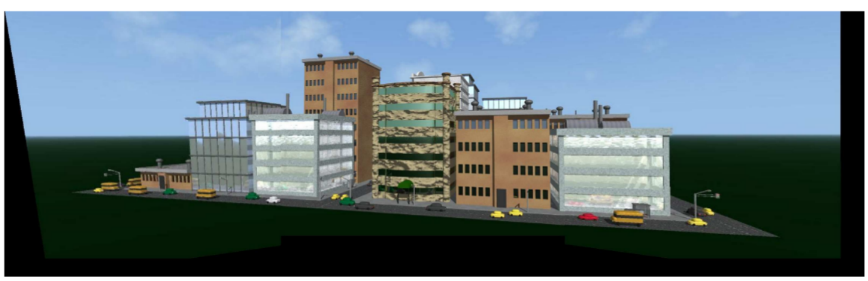
The three images "city1.jpg", "city2.jpg", and "city3.jpg" were taken from the same camera by pure rotation about the camera center. First, *understand* why the three image planes are related by a homography. In this problem, you are required to compute the homography between the three images and then stitch all three images together to create a composite image of the entire scene. Your final solution should produce an image similar to Figure 2d.

(a) city1.jpg

(b) city2.jpg



(c) city3.jpg



(d)

Figure 2: (d) is the composite image obtained by stitching together images (a)-(c).

(a) (25 points) Write a MATLAB function `H = compute_homography(pts1,pts2)` where `pts2 = H*pts1`, and `pts1` and `pts2` are the homogeneous coordinates of 4 corresponding points in two images. The dimensions of `pts1` should be 3×4 and those of `H` should be 3×3.

(b) (25 points) Use the `compute_homography` function to write a MATLAB script problem_1.m that creates a panoramic composite image from the three given images. Compute the homography between two pairs of images using 4 corresponding points. You can specify the corresponding points manually or interactively using `getpts`. Reproject the points from two images into the third. For finding non-integer pixel values, you can use bilinear interpolation. The pixels in the composite image for which no corresponding point exists in any of the three images must be kept black.

Submission Instructions: *Submit a MATLAB function compute_homography.m and a script file prob-*

lem_1.m that reads in the images and performs all the operations given above, along with any other files necessary for your code to run. Please take care to generate all the figures in new windows. Submit a brief, high-level description of the steps to your solution for part (b) and include the output composite image in the report. Do not submit any of the images separately.

Extra Credit (20 points)

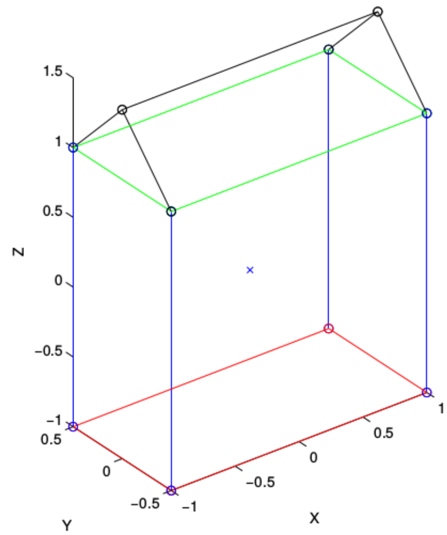


Figure 3: A wireframe house.

A simple wireframe house is shown in Figure 3. 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) (10 points) 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) (10 points) 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 3 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 report (do not submit the images separately). Clearly mark the camera location for each image in the report.*