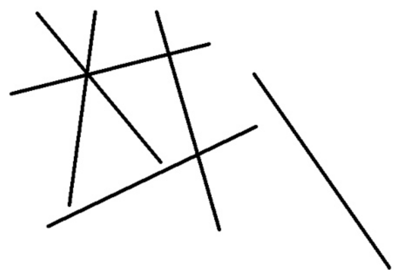
Naman Jain Vimal Kumar
1001670153

REPORT

PROBLEM 1:

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



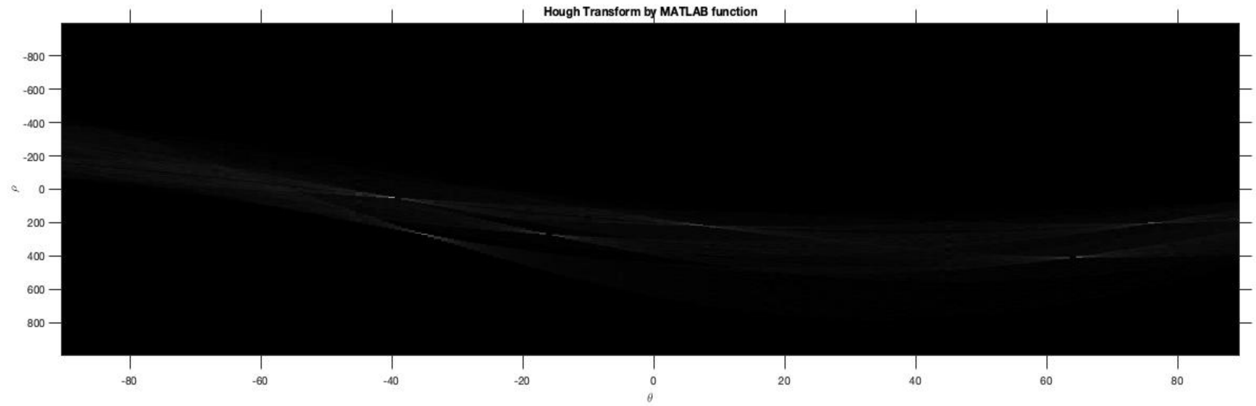
Write a MATLAB script that implements the Hough transform. It must compute the parameters for all the lines present in the accompanying image "line.jpg" (shown in Figure 1). You will need to do edge detection before you apply the Hough transform and may use the MATLAB command edge with the method of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

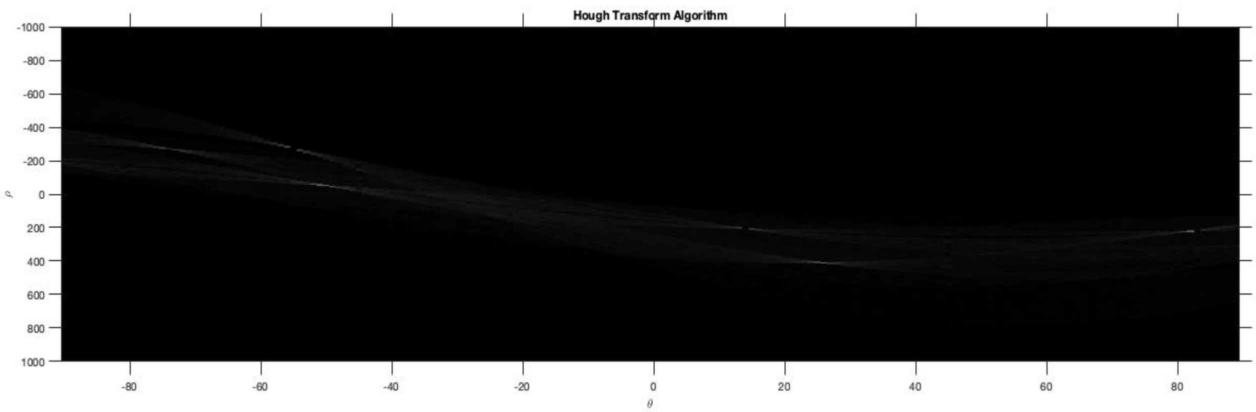
Procedure:

1. Convert the image to a grayscale image.
2. Perform an edge detection algorithm on the same.
3. Perform Hough Transform algorithm to find the accumulator array.
4. Find the peaks from the accumulator array based on a threshold.
5. Convert the peaks(theta, rho) to respective lines(x,y) based on the grayscale image.

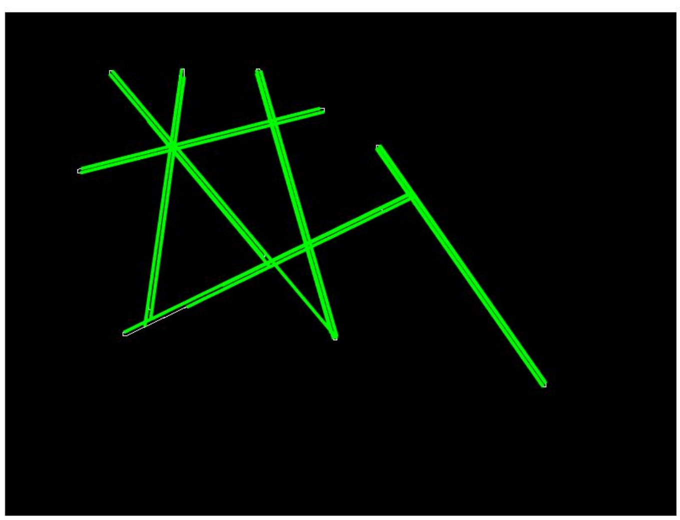
SOLUTION:

- Canny Edge Detection Algorithm has been used to find the edges(**I**) array for the lines.jpg image.
- Hough Transformation Algorithm is applied to find the accumulator array where **rho** ranges from $\left(-\sqrt{{I_x}^2 + {I_y}^2} : 1 : +\sqrt{{I_x}^2 + {I_y}^2}\right)$ and **theta** ranges from $(-90 : 1 : +89)$. Both Accumulator array with respect to MATLAB hough function and our algorithm implementation gives the following output.

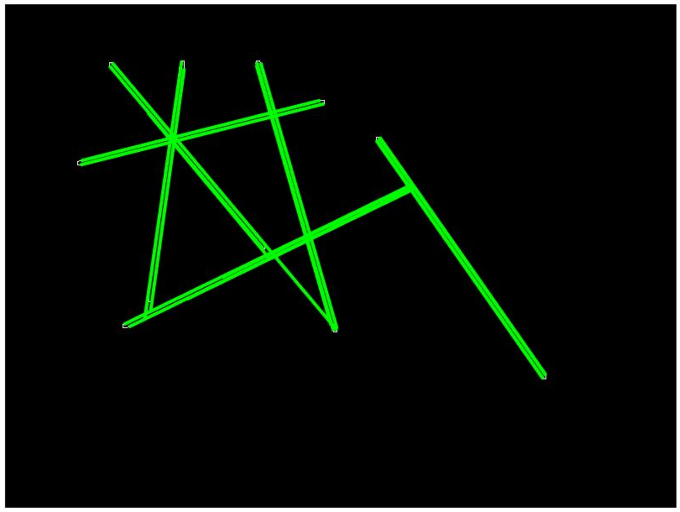




- To find the peaks Top-K Algorithm has been applied to kind highest ‘k’ element indexes are chosen. Thus we get ‘k’ peaks. [Note: In the program, I have used ‘p’ whose value is 20]
- For each peak value, a line object(rho, theta, point1, point2, last) has been defined which gets the required details to construct a line based on ‘I’ image.
- The output generated is highlighted in green and is given below when p = 17



- The output generated is highlighted in green and is given below when p = 18



PROBLEM 2:

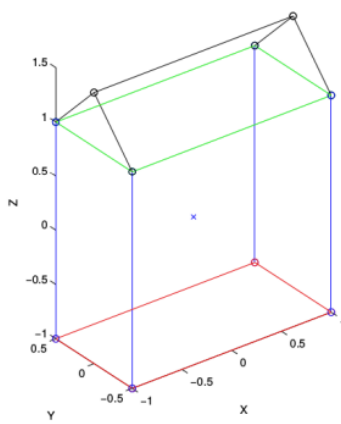


Figure 2: A wireframe house.

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 α = β = 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,

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

PROBLEM 2A:

Write a MATLAB function P C = project points(P_W, R, t) that takes as input an N × 3 vector of points with coordinates in the world frame and returns as output an N × 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.

SOLUTION:

Here, we have the following data given:

$$M_{Scaling\ Matrix} = diag[200, 200, 1, 1]$$

Thus we can define K as:

$$K = \begin{bmatrix} f & s & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * M_{Scaling\ Matrix}$$

where we know that $s_{skew} = 0, (p_x, p_y) = (50, 50)\ [Skew\ and\ image\ center]$.

Therefore

The required output can be

$$P_{C_i} = K * R * T * (P\_W_i)$$

Where R and T are homogenous (4*4) matrix

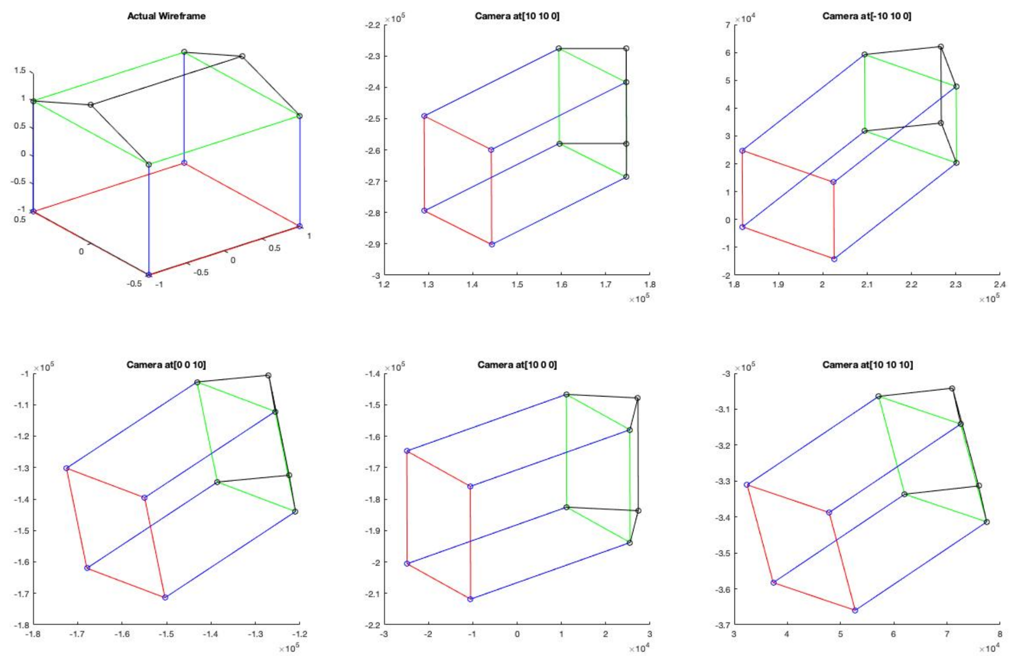
Note: The program takes in another parameter ‘C’ which defines the camera position.

PROBLEM 2B:

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 2 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.*

SOLUTION:

We get the following output:



PROBLEM 3:

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

SOLUTION:

We need to prove that Rotational Matrix is an orthogonal matrix. That is $R*R^T = I$

Let us check for a 2D plane.

$$R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

$$R * R^T = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} * \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}$$

$$R * R^T = \begin{bmatrix} ((cos\theta * cos\theta) + (-sin\theta * -sin\theta)) & ((cos\theta * sin\theta) + (-sin\theta * cos\theta)) \\ ((sin\theta * cos\theta) + (cos\theta * -sin\theta)) & ((sin\theta * sin\theta) + (cos\theta * cos\theta)) \end{bmatrix}$$

$$R * R^T = \begin{bmatrix} cos^2\theta + sin^2\theta & 0 \\ 0 & cos^2\theta + sin^2\theta \end{bmatrix}$$

$$R * R^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \qquad [Since, cos^2\theta + sin^2\theta = 1]$$

Therefore det(R) = ± 1

PROBLEM 4:

PROBLEM 4A:

Let R1 and R2 be two rotation matrices on the plane. Prove or disprove the following: R1R2 = R2R1.

SOLUTION:

Let

$$R_1 = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}, R_2 = \begin{bmatrix} cos\emptyset & -sin\emptyset \\ sin\emptyset & cos\emptyset \end{bmatrix}$$

Then

$$R_1 * R_2 = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} * \begin{bmatrix} cos\emptyset & -sin\emptyset \\ sin\emptyset & cos\emptyset \end{bmatrix}$$

$$R_1 * R_2 = \begin{bmatrix} ((cos\theta * cos\emptyset) + (-sin\theta * sin\emptyset)) & ((cos\theta * -sin\emptyset) + (-sin\theta * cos\emptyset)) \\ ((sin\theta * cos\emptyset) + (cos\theta * sin\emptyset)) & ((sin\theta * -sin\emptyset) + (cos\theta * cos\emptyset)) \end{bmatrix}$$

$$R_1 * R_2 = \begin{bmatrix} ((cos\theta * cos\emptyset) - (sin\theta * sin\emptyset)) & -((cos\theta * sin\emptyset) + (sin\theta * cos\emptyset)) \\ ((sin\theta * cos\emptyset) + (cos\theta * sin\emptyset)) & (-(sin\theta * sin\emptyset) + (cos\theta * cos\emptyset)) \end{bmatrix}$$

$$R_2 * R_1 = \begin{bmatrix} cos\emptyset & -sin\emptyset \\ sin\emptyset & cos\emptyset \end{bmatrix} * \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

$$R_2 * R_1 = \begin{bmatrix} ((cos\emptyset * cos\theta) + (-sin\emptyset * sin\theta)) & ((cos\emptyset * -sin\theta) + (-sin\emptyset * cos\theta)) \\ ((sin\emptyset * cos\theta) + (cos\emptyset * sin\theta)) & ((sin\emptyset * -sin\theta) + (cos\emptyset * cos\theta)) \end{bmatrix}$$

$$R_2 * R_1 = \begin{bmatrix} ((cos\emptyset * cos\theta) - (sin\emptyset * sin\theta)) & -((cos\emptyset * sin\theta) + (sin\emptyset * cos\theta)) \\ ((sin\emptyset * cos\theta) + (cos\emptyset * sin\theta)) & (-(sin\emptyset * sin\theta) + (cos\emptyset * cos\theta)) \end{bmatrix}$$

$$R_2 * R_1 = R_1 * R_2 \qquad [Thus\ Proved]$$

PROBLEM 4B:

Let R1 and R2 be two rotation matrices in 3D space. Prove or disprove the following: R1R2 = R2R1.

SOLUTION:

Let

$$R_1 = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_2 = \begin{bmatrix} cos\emptyset & 0 & -sin\emptyset \\ 0 & 1 & 0 \\ sin\emptyset & 0 & cos\emptyset \end{bmatrix}$$

Then

$$R_1 * R_2 = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} cos\emptyset & 0 & -sin\emptyset \\ 0 & 1 & 0 \\ sin\emptyset & 0 & cos\emptyset \end{bmatrix}$$

$$R_1 * R_2 = \begin{bmatrix} cos\theta * cos\emptyset & -sin\theta & -(cos\theta * sin\emptyset) \\ sin\theta * cos\emptyset & cos\theta & -(sin\theta * sin\emptyset) \\ sin\emptyset & 0 & cos\emptyset \end{bmatrix}$$

$$R_2 * R_1 = \begin{bmatrix} cos\emptyset & 0 & -sin\emptyset \\ 0 & 1 & 0 \\ sin\emptyset & 0 & cos\emptyset \end{bmatrix} * \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_2 * R_1 = \begin{bmatrix} cos\theta * cos\emptyset & -(sin\theta * cos\emptyset) & -sin\emptyset \\ sin\theta & cos\theta & 0 \\ sin\emptyset * cos\theta & -(sin\theta * sin\emptyset) & cos\emptyset \end{bmatrix}$$

$$R_2 * R_1 \neq R_1 * R_2 \qquad [Hence\ Disproved]$$

PROBLEM 5:

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.

SOLUTION:

The solution has been implemented for a (3x3) matrix.

References:

- http://www.aishack.in/tutorials/hough-transform-basics/
- http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/BMVA96Tut/node30.html
- https://www.khanacademy.org/partner-content/pixar/sets/rotation/v/set-7
- https://stackoverflow.com/questions/16974627/how-would-you-convert-x-y-points-to-rho-theta-for-hough-transform-in-c
- https://dsp.stackexchange.com/questions/30410/how-to-set-threshold-value-for-hough-transform
- http://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/geometry/geo-tran.html
- http://vlm1.uta.edu/~athitsos/courses/cse6367_spring2012/lectures/19_geometry/geometry.pdf