**Assignment No:1**

**Name:Ravi Singi**

**RollNo: 2018900095**

1. Implement the DLT (Direct Linear Transformation) based calibration that we discussed in the class:

Ans:

Am using the below python script to view the image and find the image points physically.

import matplotlib.pyplot as plt

import cv2

import numpy as np

im = cv2.imread("IMG\_5455.JPG")

im\_resized = cv2.resize(im, (5472, 3648), interpolation=cv2.INTER\_LINEAR)

plt.imshow(cv2.cvtColor(im\_resized, cv2.COLOR\_BGR2RGB))

plt.show()

Now by using the measurements.jpg file am able to get the world points as shown below

worldpoints = [[0, 36, 0], [0, 0, 36], [36, 0, 0], [0, 72, 0], [36, 36, 0],

[36, 72, 0], [0, 0, 72], [36, 0, 36]]

Now am trying to create a Directlineartransformation method to use DLT approach and find the camera coordinates. Please find the code below

import numpy as np

def directlineartransformation(worldpoints, picpoints):

xyz = np.asarray(worldpoints)

uv = np.asarray(picpoints)

count\_points = np.shape(xyz)[0]

A = []

for i in range(count\_points):

x = xyz[i, 0]

y = xyz[i, 1]

z = xyz[i, 2]

u = uv[i, 0]

v = uv[i, 1]

A.append([x, y, z, 1, 0, 0, 0, 0, -u\*x, -u\*y, -u\*z, -u])

A.append([0, 0, 0, 0, x, y, z, 1, -v\*x, -v\*y, -v\*z, -v])

A = np.asarray(A)

U, S, V = np.linalg.svd(A)

H = V[11, :]

H = H.reshape(3, 4)

H = H/H[2, 3]

return H

def main():

worldpoints = [[0, 36, 0], [0, 0, 36], [36, 0, 0], [0, 72, 0], [36, 36, 0],

[36, 72, 0], [0, 0, 72], [36, 0, 36]]

picturepoints = [[4859.28, 1376.08], [4716.33, 2432.86], [3838.23, 2371.66], [

4900.12, 543.923], [4032.23, 1319.92], [4057.76, 508.16], [4649, 2780], [3700.39, 2698.33]]

outputmatrix = directlineartransformation(worldpoints, picturepoints)

print(outputmatrix)

main()

The output is as shown below:

[[-4.08200380e+01 8.35534496e+00 -1.24033708e+01 4.79754533e+03]

[-4.49065010e+00 -2.17921666e+01 1.89176180e+00 2.19251586e+03]

[-3.87994746e-03 1.41248177e-03 -2.21739626e-03 1.00000000e+00]]