Name: Janmejay Mohanty

CS 532-A

HW-3 Report

Source Code:

import numpy as np

from PIL import Image

import open3d as o3d

# Assigning and creating an array list.

P = np.array([[776.649963,-298.408539,-32.048386,993.1581875,132.852554,120.885834,-759.210876,1982.174000,0.744869,0.662592,-0.078377,4.629312012],

[431.503540,586.251892,-137.094040,1982.053375,23.799522,1.964373,-657.832764,1725.253500,-0.321776,0.869462,-0.374826,5.538025391],

[-153.607925,722.067139,-127.204468,2182.4950,141.564346,74.195686,-637.070984,1551.185125,-0.769772,0.354474,-0.530847,4.737782227],

[-823.909119,55.557896,-82.577644,2498.20825,-31.429972,42.725830,-777.534546,2083.363250,-0.484634,-0.807611,-0.335998,4.934550781],

[-715.434998,-351.073730,-147.460815,1978.534875,29.429260,-2.156084,-779.121704,2028.892750,0.030776,-0.941587,-0.335361,4.141203125],

[-417.221649,-700.318726,-27.361042,1599.565000,111.925537,-169.101776,-752.020142,1982.983750,0.542421,-0.837170,-0.070180,3.929336426],

[94.934860,-668.213623,-331.895508,769.8633125,-549.403137,-58.174614,-342.555359,1286.971000,0.196630,-0.136065,-0.970991,3.574729736],

[452.159027,-658.943909,-279.703522,883.495000,-262.442566,1.231108,-751.532349,1884.149625,0.776201,0.215114,-0.592653,4.235517090]])

# Declaring and assigning the values to variables, list, etc.

dict\_c = {}

dict\_s = {}

matrix\_p = np.zeros((3,4,8))

# Fetching the required image data files.

for i in range(8):

dict\_c[i] = np.asarray(Image.open("cam0" + str(i) + "\_00023\_0000008550.png"))

dict\_s[i] = np.asarray(Image.open("silh\_cam0" + str(i) + "\_00023\_0000008550.pbm"))

matrix\_p[:,:,i] = np.reshape(P[i],(3,4))

range\_x = 5

range\_y = 6

range\_z = 2.5

volume = range\_x\*range\_y\*range\_z

v\_num = 1000000 # Vox Number

v\_size = np.power((volume/v\_num),1/3) # Vox Size

v\_grid = []

s\_grid = [] # Surf Grid

d\_grid = []

for x in np.arange(-2.5, 2.5, v\_size):

for y in np.arange(-3, 3, v\_size):

min\_z = 3

max\_z = -3

for z in np.arange(0, 2.5, v\_size):

pass\_mat = np.zeros(8)

co\_ordinate = [x, y, z, 1]

for i in range(8):

pt = np.dot(co\_ordinate,np.transpose(matrix\_p[:,:,i]))

pt = pt/pt[2]

if((0<=pt[1]<582) and (0<=pt[0]<780)):

pass\_mat[i] = dict\_s[i][int(pt[1]),int(pt[0])]

if(np.sum(pass\_mat) == 8):

v\_grid.append([x,y,z])

if(z<min\_z):

min\_z = z

if(z>max\_z):

max\_z = z

if(min\_z!=3 and max\_z!=-3):

s\_grid.append([x,y,min\_z])

s\_grid.append([x,y,max\_z])

d\_grid.append('zbot')

d\_grid.append('ztop')

# Declaring a function for Grid Cleared.

def grid\_cleared(vox,surf):

for s in surf:

if(s in vox):

vox.remove(s)

grid\_cleared(v\_grid,s\_grid)

for z in np.arange(0, 2.5, v\_size):

for y in np.arange(-3, 3, v\_size):

min\_x = 3

max\_x = -3

for x in np.arange(-2.5, 2.5, v\_size):

if([x,y,z] in v\_grid):

if(x<min\_x):

min\_x = x

if(x>max\_x):

max\_x = x

if(min\_x!=3 and max\_x!=-3):

s\_grid.append([min\_x,y,z])

s\_grid.append([max\_x,y,z])

d\_grid.append('xleft')

d\_grid.append('xright')

grid\_cleared(v\_grid,s\_grid)

for z in np.arange(0, 2.5, v\_size):

for x in np.arange(-2.5, 2.5, v\_size):

min\_y = 3

max\_y = -3

for y in np.arange(-3, 3, v\_size):

if([x,y,z] in v\_grid):

if(y<min\_y):

min\_y = y

if(y>max\_y):

max\_y = y

if(min\_z!=3 and max\_z!=-3):

s\_grid.append([x,min\_y,z])

s\_grid.append([x,max\_y,z])

d\_grid.append('yback')

d\_grid.append('yfront')

grid\_color = []

dict\_d = {

"ztop": 6,

"zbot": 3,

"yback": 3,

"yfront": 0,

"xright": 2,

"xleft": 5

}

for i in range(0,len(d\_grid)):

view = dict\_d[d\_grid[i]]

co\_ordinate = s\_grid[i]+[1]

pt = np.dot(co\_ordinate,np.transpose(matrix\_p[:,:,view]))

pt = pt/pt[2]

rgb = dict\_c[view][int(pt[1]),int(pt[0])]

grid\_color.append([float(rgb[0])/255,float(rgb[1])/255,float(rgb[2])/255])

print(v\_size,len(grid\_color),len(s\_grid))

pcd = o3d.geometry.PointCloud()

pcd.points = o3d.utility.Vector3dVector(s\_grid)

pcd.colors = o3d.utility.Vector3dVector(grid\_color)

o3d.io.write\_point\_cloud("./dancer.ply", pcd)

o3d.visualization.draw\_geometries([pcd])



