

Exercise 7

March 23, 2022

0.0.1 Exercise 7

0.0.2 Index No: 190108X

0.0.3 Name: Chathuranga M.M.P.

```
[6]: # Question 1

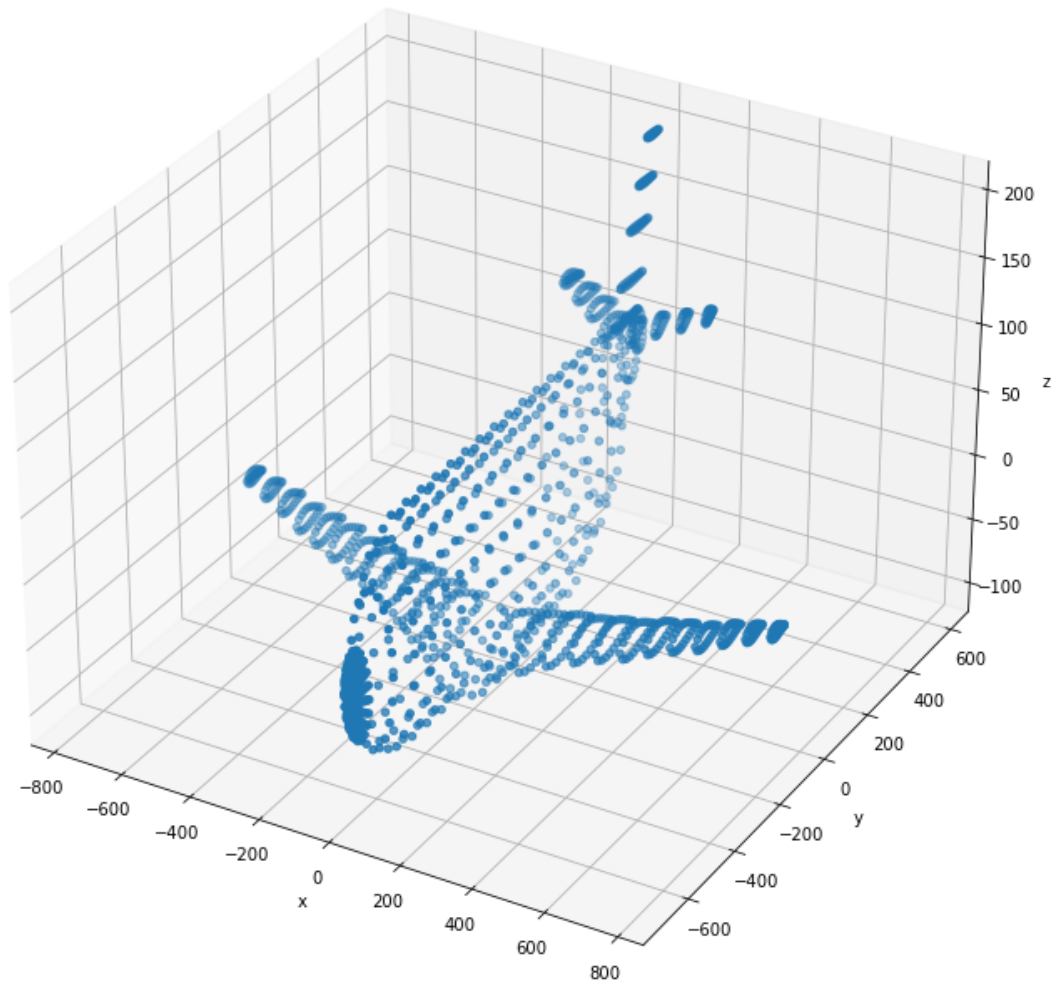
import numpy as np
from plyfile import PlyData, PlyElement
import matplotlib.pyplot as plt

pcd=PlyData.read(r'airplane.ply')
assert pcd is not None

points=np.concatenate((pcd['vertex']['x'].reshape(1,-1),pcd['vertex']['y'].
    ↳reshape(1,-1),pcd['vertex']['z'].reshape(1,-1)),axis=0)
points=points-np.mean(points,axis=1).reshape(3,1)

[7]: fig=plt.figure(figsize=(12,12))
ax=fig.add_subplot(111,projection='3d')
ax.scatter(points[0,:],points[1,:],points[2,:])
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
```

```
[7]: Text(0.5, 0, 'z')
```



```
[8]: ones=np.ones((1,points.shape[1]))
X=np.concatenate((points,ones),axis=0)

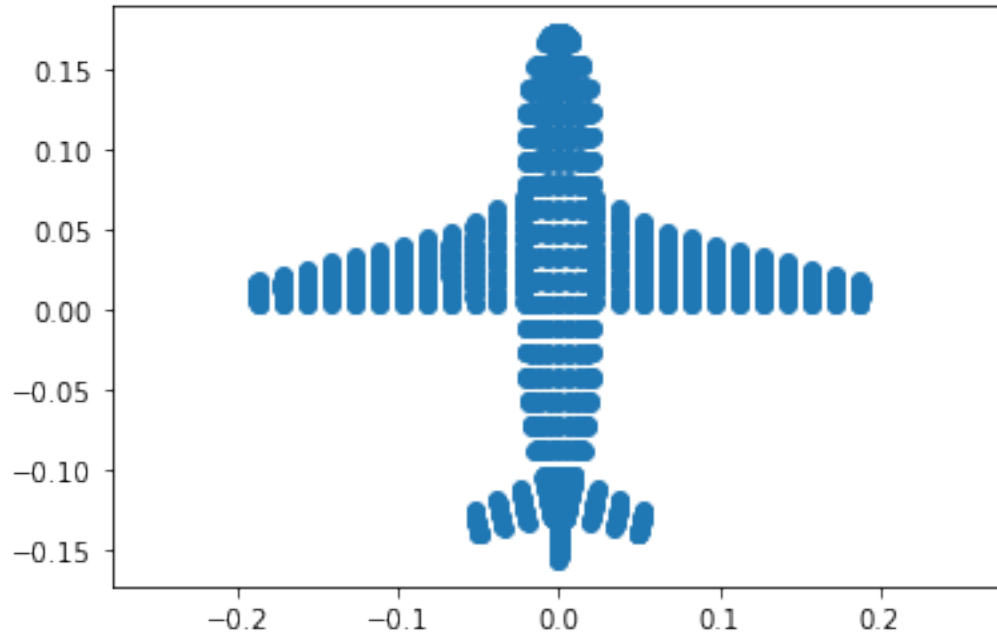
R=np.array([[1,0,0],[0,1,0],[0,0,1]])
K=np.array([[1,0,0],[0,1,0],[0,0,1]])
t=np.array([[0],[0],[-4000]])

P1=K@np.concatenate((R,t),axis=1)

x1=P1@X

x1=x1/x1[2,:]
```

```
[9]: fig,ax=plt.subplots(1,1,sharex=True,sharey=True)
ax.scatter(x1[0,:],x1[1,:])
ax.axis('equal')
plt.show()
```



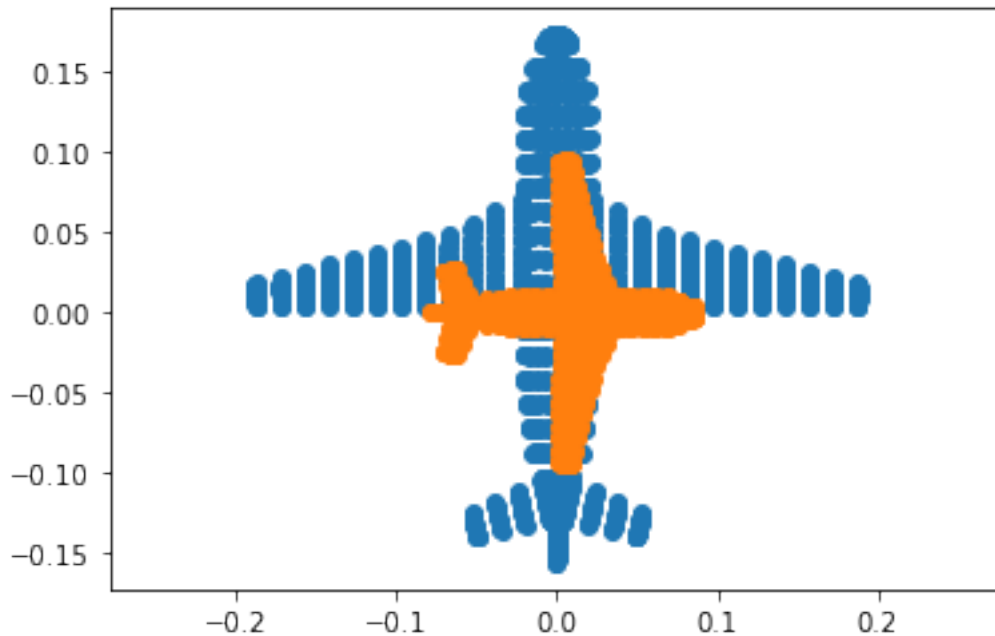
```
[18]: R=np.array([[0,1,0],[1,0,0],[0,0,1]])
K=np.array([[0.5,0,0],[0,0.5,0],[0,0,1]])
t=np.array([[0],[0],[-4000]])

P2=K@np.concatenate((R,t),axis=1)

x2=P2@X

x2=x2/x2[2,:]
```

```
[19]: fig,ax=plt.subplots(1,1,sharex=True,sharey=True)
ax.scatter(x1[0,:],x1[1,:])
ax.scatter(x2[0,:],x2[1,:])
ax.axis('equal')
plt.show()
```



```
[30]: # Question 3

import cv2 as cv
import numpy as np
img=cv.imread(r'earrings.jpg',cv.IMREAD_COLOR)
assert img is not None
hsv=cv.cvtColor(img,cv.COLOR_BGR2HSV)
th,bw=cv.threshold(hsv[:, :, 1],0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)

w=5
kernel=np.ones((w,w),np.uint8)
opened=cv.morphologyEx(bw,cv.MORPH_CLOSE,kernel)

retval,labels,stats,centroids=cv.connectedComponentsWithStats(bw)
colormapped=cv.applyColorMap((labels/np.amax(labels)*255).astype('uint8'),cv.
    ↳COLORMAP_PARULA)

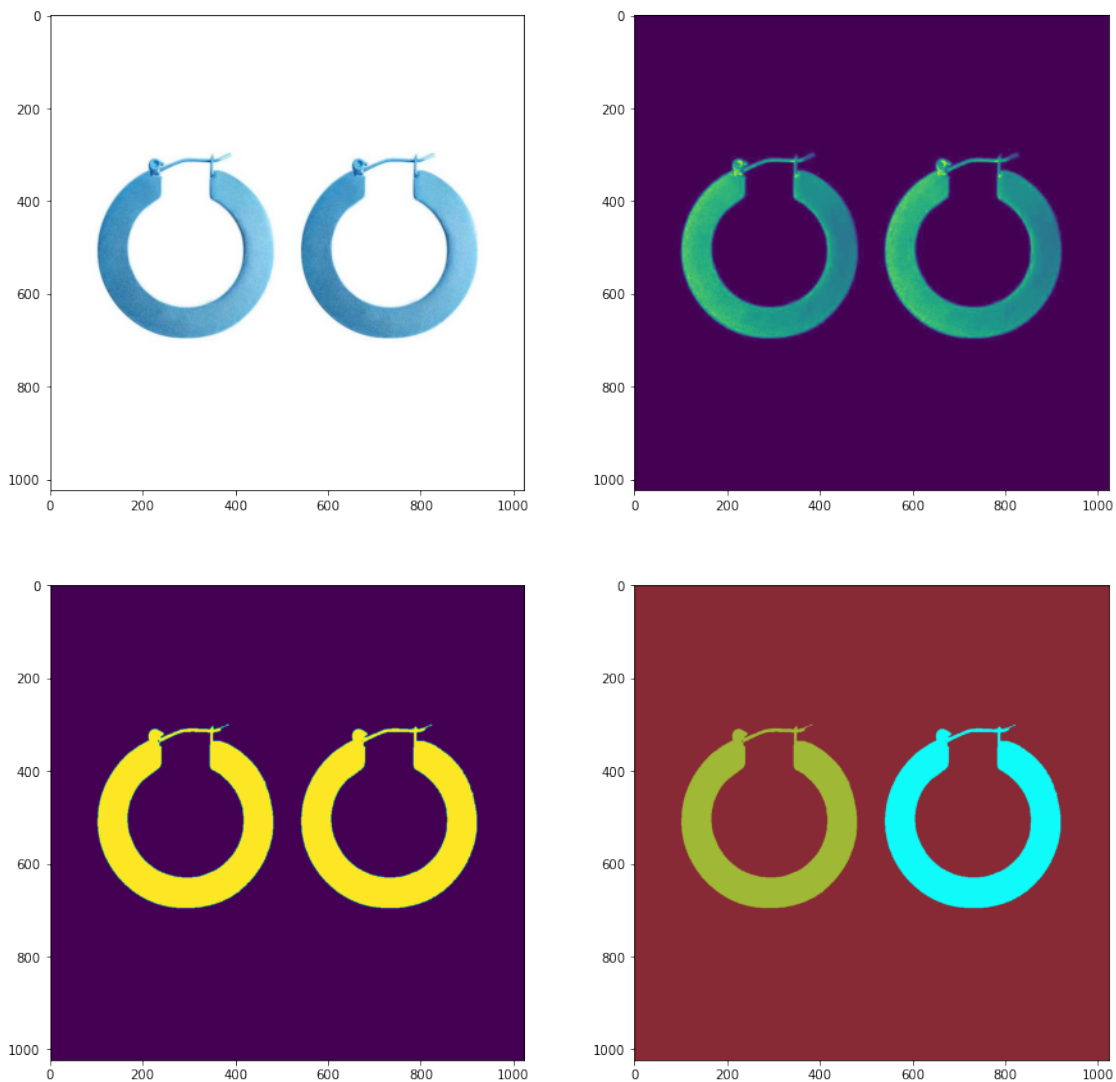
Z=720
f=8
for i,s in enumerate(stats):
    if i!=0:
        print('Item',i,'area in pixels',s[4])
        print('Item',i,'area in mm^2',s[4]*(2.2e-3)**2*(Z*Z)/(f*f))

fig,ax=plt.subplots(2,2,figsize=(15,15))
```

```
ax[0][0].imshow(img)
ax[0][1].imshow(hsv[:, :, 1])
ax[1][0].imshow(bw)
ax[1][1].imshow(colormapped)
```

Item 1 area in pixels 59143
 Item 1 area in mm² 2318.642172
 Item 2 area in pixels 59211
 Item 2 area in mm² 2321.3080440000003

[30]: <matplotlib.image.AxesImage at 0x1b95b44fb20>



[25]: # Question 4

```

import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

file_name = 'allenkeys.jpg'
im = cv.imread ( file_name , cv.IMREAD_REDUCED_GRAYSCALE_2)
assert im is not None
canny = cv . Canny ( im , 50 , 150)

# Copy edges to the images that will display the results in BGR
canny_color = cv ..cvtColor ( canny , cv .COLOR_GRAY2BGR)
lines = cv . HoughLines ( canny , 1 , np . pi / 180 , 170 , None , 0 , 0)
if lines is not None:
    for i in range ( 0 , len ( lines ) ) :
        rho = lines [ i ] [ 0 ] [ 0 ]
        theta = lines [ i ] [ 0 ] [ 1 ]
        a = np . cos ( theta )
        b = np . sin ( theta )
        x0 = a * rho
        y0 = b * rho
        pt1 = ( int ( x0 + 1000*(-b ) ) , int ( y0 + 1000*(a ) ) )
        pt2 = ( int ( x0 - 1000*(-b ) ) , int ( y0 - 1000*(a ) ) )
        cv . line ( canny_color , pt1 , pt2 , ( 0 ,0 ,255) , 1 , cv .LINE_AA)

cv . namedWindow( 'Image' , cv .WINDOW_AUTOSIZE)
cv . imshow( 'Image' , im)
cv . waitKey ( 0 )
cv . imshow( 'Image' , canny )
cv . waitKey ( 0 )
cv . imshow( 'Image' , canny_color )
r = cv . selectROI ( 'Image' , canny_color , showCrosshair = True , fromCenter_
    ↪= False )

cv . waitKey ( 0 )
print ( r )
x0 , y0 = int ( r [ 0 ] + r [ 2 ] / 2 ) , int ( r [ 1 ] + r [ 3 ] / 2 )
m = b / a # Gradient
m = np . tan (np . median ( lines [ : , 0 , 1 ] ) )
c = y0 = m*x0 # Intercept
cv . line ( canny_color , ( 0 , int ( c ) ) , ( im.shape [ 0 ] , int (m*im._
    ↪shape [ 0 ] + c ) ) , ( 0 ,255 ,0 ) , 2 , cv .LINE_AA)
cv . imshow( 'Image' , canny_color )
cv . waitKey ( 0 )
cv . destroyAllWindows ( )
dy = 1
y_sub_pixel = np . arange ( 0 , im.shape [ 0 ] -1 , dy )
f_sub_pixel = np . zeros_like ( y_sub_pixel )

```


[]: