

## EN2550 Exercise 7

Name - Ekanayake E.M.S.S.N.

Index no - 190164M

```
In [ ]: import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
from plyfile import PlyData, PlyElement
```

01

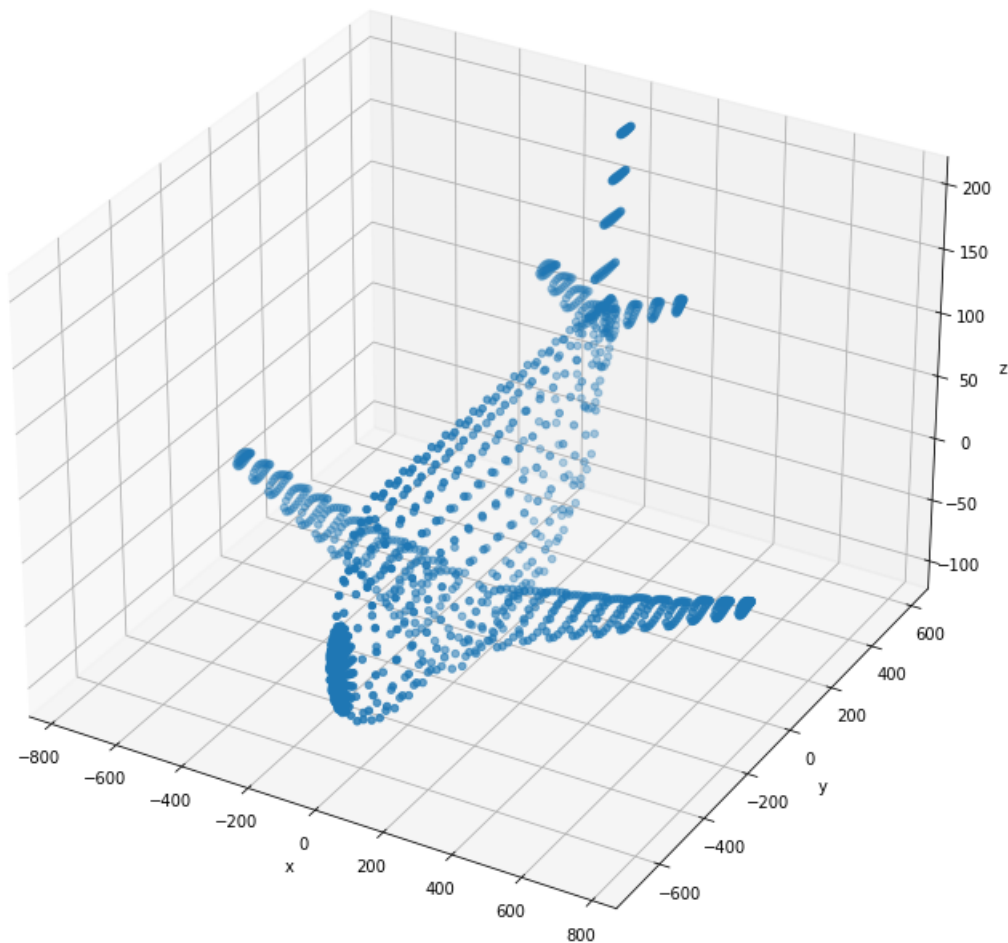
```
In [ ]: pcd = PlyData.read('Images/airplane.ply')
assert pcd is not None

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)

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')

plt.show()
```



```
In [ ]: 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],[-400]])

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

```

t = 30*np.pi/180
R=np.array([[np.cos(t),-np.sin(t),0],[np.sin(t),np.cos(t),0],[0,0,1]])
K=np.array([[0.8,0,0],[0,0.8,0],[0,0,1]])
t=np.array([[0],[0],[-4000]])

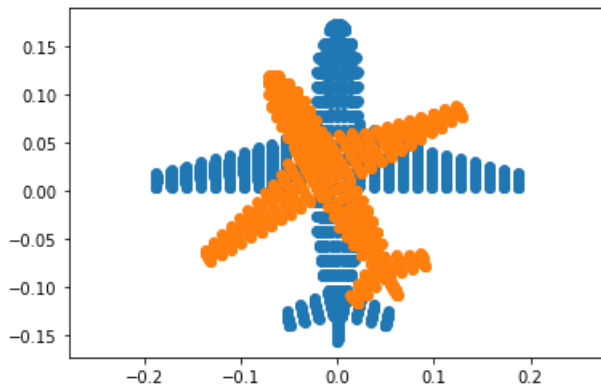
P2=K @ np.concatenate((R,t),axis=1)

x1=P1@X
x2=P2@X
x1=x1/x1[2,:]
x2=x2/x2[2,:]

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()

```



03

```

In [ ]: im=cv.imread('Images/earrings.jpg',cv.IMREAD_COLOR)
assert im is not None

hsv = cv.cvtColor(im, 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 # mm
f = 8 # mm
for i, s in enumerate(stats):
    if i != 0:
        print('Item ', i, ', area in pixels =', s[4])
        print('Item ', i, ', area in mmA2 =', s[4]*(2.2e-3)**2*(Z*Z)/(f*f))

fig,ax=plt.subplots(2,3,figsize=(20,12))
ax[0,0].imshow(cv.cvtColor(im,cv.COLOR_BGR2RGB))
ax[0,0].set_title("Original")

ax[0,1].imshow(cv.cvtColor(hsv[:, :, 1],cv.COLOR_BGR2RGB))
ax[0,1].set_title("hsv")

ax[0,2].imshow(cv.cvtColor(bw,cv.COLOR_BGR2RGB))
ax[0,2].set_title("bw")

ax[1,0].imshow(cv.cvtColor(opened,cv.COLOR_BGR2RGB))
ax[1,0].set_title("opened")

ax[1,1].imshow(cv.cvtColor(colormapped,cv.COLOR_BGR2RGB))
ax[1,1].set_title("colormapped")

[axi.set_axis_off() for axi in ax.ravel()]

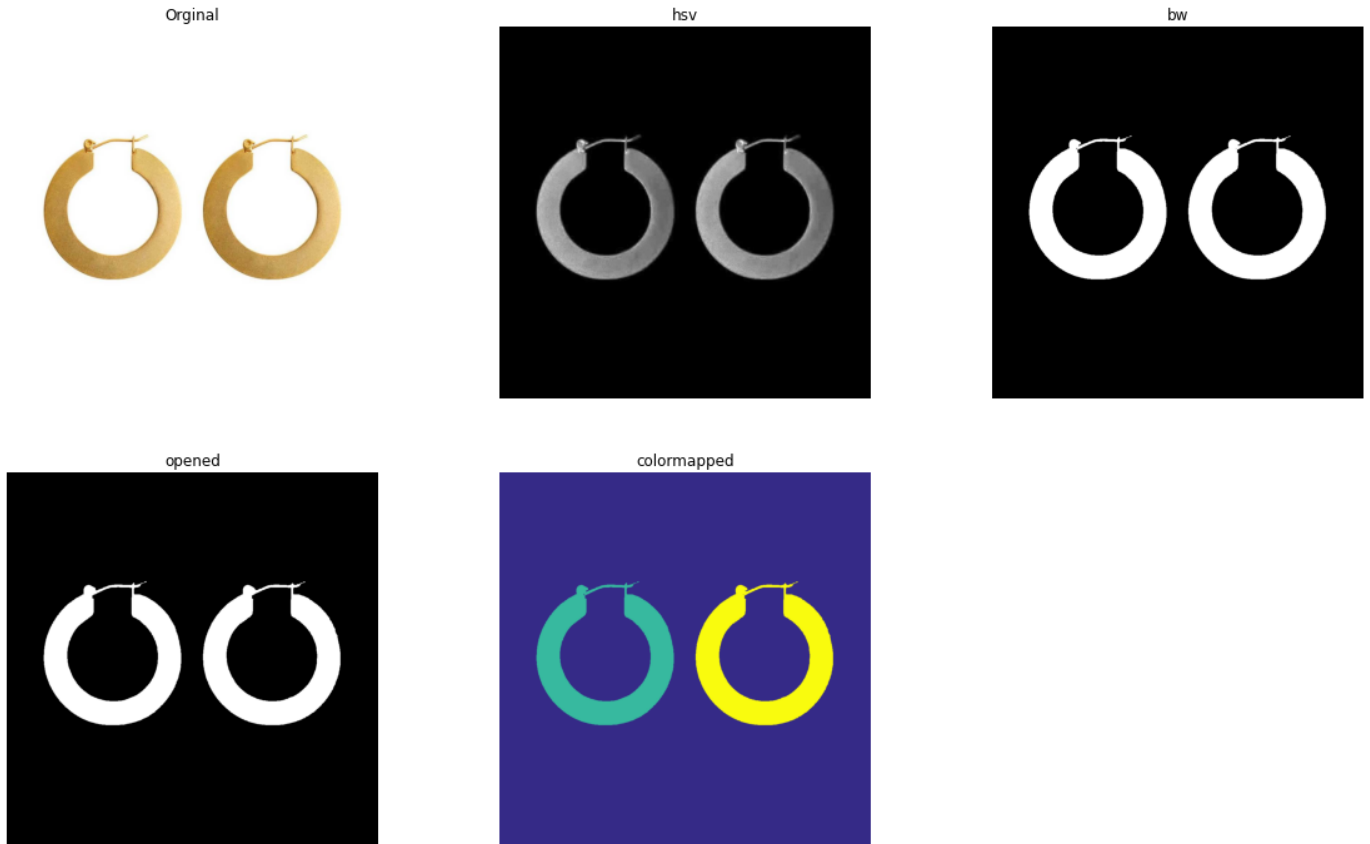
plt.show()

```

```

Item 1 , area in pixels = 59143
Item 1 , area in mmA2 = 2318.642172
Item 2 , area in pixels = 59211
Item 2 , area in mmA2 = 2321.3080440000003

```



04

```
In [ ]: file_name = 'Images/allenkeys.jpg'
im = cv.imread(file_name, cv.IMREAD_REDUCED_GRAYSCALE_2)
canny = cv.Canny(im, 50, 150)

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)

fig,ax = plt.subplots(1,3,figsize=(12,12))
im = cv.cvtColor(im,cv.COLOR_BGR2RGB)
ax[0].axis('off')
ax[0].imshow(im)
# ax[0].set_title("Original Image")

canny = cv.cvtColor(canny,cv.COLOR_BGR2RGB)
ax[1].axis('off')
ax[1].imshow(canny)
# ax[1].set_title("Black and White Image")

canny_color1 = cv.cvtColor(canny_color,cv.COLOR_BGR2RGB)
ax[2].axis('off')
ax[2].imshow(canny_color1)
# ax[2].set_title("Color Mapped Image")

r = cv.selectROI('Image' , canny_color , showCrosshair = True , fromCenter = False )
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

cv.line(canny_color, (0,int(c)) , (im.shape[0] , int (m*im.shape[0] + c )), (0,255,0), 2 , cv.LINE_AA)

fig,ax = plt.subplots()
canny_color = cv.cvtColor(canny_color,cv.COLOR_BGR2RGB)
```

```
ax.axis('off')
ax.imshow(canny_color)

dy = 1
y_sub_pixel = np.arange(0, im.shape[0]-1, dy)
f_sub_pixel = np.zeros_like(y_sub_pixel)
f_sub_pixel_nn = np.zeros_like(y_sub_pixel)

for i,y in enumerate(y_sub_pixel):

fig, ax = plt.subplots(figsize=(30,5))
ax.plot(f_sub_pixel_nn)
```