Name : Jegakumaran P.

Index Number : 190280N

In [ ]:

```
import numpy as np
f=open(r'./Images/templeSparseRing/templeSparseRing/templeSR_par.txt'
assert f is not None

n=int(f.readline())
l=f.readline().split()
img1_fn=l[0]

# Reading the information on the first image
K1=np.array([float(i) for i in l[1:10]]).reshape((3,3))
R1=np.array([float(i) for i in l[10:19]]).reshape((3,3))
t1=np.array([float(i) for i in l[19:22]]).reshape((3,1))

# Reading the information on the second image
l=f.readline().split()
img2_fn=l[0]
K2=np.array([float(i) for i in l[1:10]]).reshape((3,3))
R2=np.array([float(i) for i in l[10:19]]).reshape((3,3))
t2=np.array([float(i) for i in l[19:22]]).reshape((3,1))

print("K1 = ",K1)
print("R1 = ",R1)
print("t1 = ",t1)
print("K2 = ",K2)
print("R2 = ",R2)
print("t2 = ",t2)
```

```
K1 =  [[1.5204e+03 0.0000e+00 3.0232e+02]
 [0.0000e+00 1.5259e+03 2.4687e+02]
 [0.0000e+00 0.0000e+00 1.0000e+00]]
R1 =  [[ 0.02187598  0.98329681 -0.18068986]
 [ 0.99856708 -0.01266115  0.05199501]
 [ 0.04883878 -0.18156839 -0.9821648 ]]
t1 =  [[-0.07266377]
 [ 0.02233604]
 [ 0.61460485]]
K2 =  [[1.5204e+03 0.0000e+00 3.0232e+02]
 [0.0000e+00 1.5259e+03 2.4687e+02]
 [0.0000e+00 0.0000e+00 1.0000e+00]]
R2 =  [[-0.034722    0.98429285 -0.17309525]
 [ 0.93942193 -0.02695167 -0.3417017 ]
 [-0.34099974 -0.17447404 -0.92373047]]
t2 =  [[-0.0746307 ]
 [ 0.03381481]
 [ 0.60085057]]
```

In [ ]:

```
# Read the two images and show
import cv2 as cv
import matplotlib.pyplot as plt

img1=cv.imread(r'Images/templeSparseRing/templeSparseRing/' + img1_fn
img2=cv.imread(r'Images/templeSparseRing/templeSparseRing/' + img2_fn
assert img1 is not None
assert img2 is not None

fig,ax=plt.subplots(1,2,sharex='all',sharey='all',figsize=(15,15))
ax[0].imshow(cv.cvtColor(img1,cv.COLOR_BGR2RGB))
ax[0].set_title("Image1")
ax[0].set_xticks([]) , ax[0].set_yticks([])
ax[1].imshow(cv.cvtColor(img2,cv.COLOR_BGR2RGB))
ax[1].set_title("Image2")
ax[1].set_xticks([]) , ax[1].set_yticks([])
plt.show()
```

In [ ]:

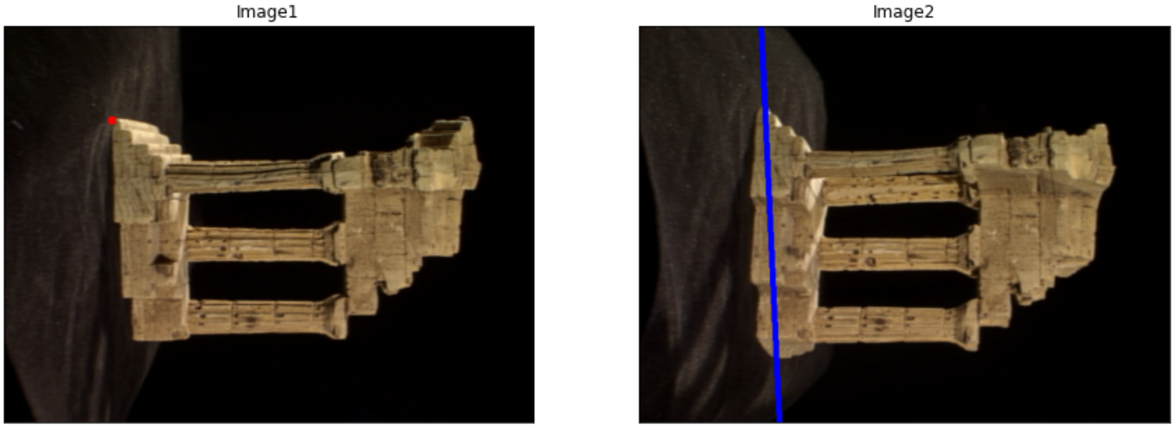
```
# Compute P1 and P2

P1=K1@np.hstack((R1,t1)) # P = K*[R|t]
P2=K1@np.hstack((R2,t2)) # P = K*[R|t]
print ("P1 = ", P1)
print ("P2 = ", P2)
```

```
P1 =  [[ 4.80251845e+01  1.44011271e+03 -5.71648932e+02  7.53293366e+
01]
 [ 1.53577034e+03 -6.41434324e+01 -1.63127843e+02  1.85810055e+02]
 [ 4.88387837e-02 -1.81568392e-01 -9.82164799e-01  6.14604846e-01]]
P2 =  [[-1.55882371e+02  1.44377186e+03 -5.42436214e+02  6.81806220e+
01]
 [ 1.34928131e+03 -8.41979541e+01 -7.49443961e+02  1.99929996e+02]
 [-3.40999743e-01 -1.74474039e-01 -9.23730472e-01  6.00850565e-01]]
```

In [ ]:

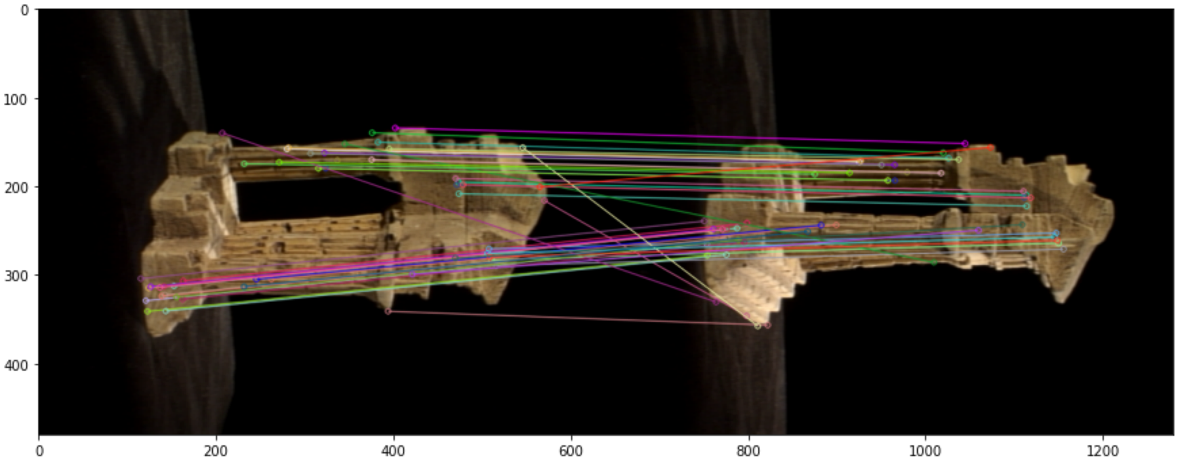
```
from scipy.linalg import null_space
def skew(x):
    x=x.ravel()
    return np.array([[0,-x[2],x[1]],[x[2],0,-x[0]],[-x[1],x[0],0]])

C= null_space(P1)
C=C*np.sign(C[0,0])

e2=P2@C
e2x=skew(e2)

F= e2x@P2@np.linalg.pinv(P1)
print('F = ',F)

x=np.array([130,115,1])
cv.circle(img1,(x[0],x[1]),5,(0,0,255),-1)
l2=F@x.T

p1=np.array([0,(l2[0]*0 + l2[2])/l2[1]]).astype(int)
p2=np.array([500,(l2[0]*500 + l2[2])/l2[1]]).astype(int)

cv.line(img2,(p1[0],p1[1]),(p2[0],p2[1]),(255,0,0),5)
fig,ax=plt.subplots(1,2,sharex='all',sharey='all',figsize=(15,15))
ax[0].imshow(cv.cvtColor(img1,cv.COLOR_BGR2RGB))
ax[0].set_title("Image1")
ax[0].set_xticks([]) , ax[0].set_yticks([])
ax[1].imshow(cv.cvtColor(img2,cv.COLOR_BGR2RGB))
ax[1].set_title("Image2")
ax[1].set_xticks([]) , ax[1].set_yticks([])
plt.show()
```

```
F =  [[-2.87071497e-04 -3.96261289e-02  2.94221686e+02]
 [-3.55039713e-02  1.65329260e-04  1.78860854e+01]
 [-2.76702814e+02  2.12942175e+01 -9.06669374e+03]]
```



In [ ]:

```
import cv2 as cv
import matplotlib.pyplot as plt


# read images
img1 = cv.imread(r'Images/templeSparseRing/templeSparseRing/templeSR0
img2 = cv.imread(r'Images/templeSparseRing/templeSparseRing/templeSR0
img1 = cv.cvtColor(img1, cv.IMREAD_GRAYSCALE)
img2 = cv.cvtColor(img2, cv.IMREAD_GRAYSCALE)

#sift
sift = cv.SIFT_create()

keypoints_1, descriptors_1 = sift.detectAndCompute(img1,None)
keypoints_2, descriptors_2 = sift.detectAndCompute(img2,None)

#feature matching
bf = cv.BFMatcher(cv.NORM_L1, crossCheck=True)

matches = bf.match(descriptors_1,descriptors_2)
matches = sorted(matches, key = lambda x:x.distance)

img3 = cv.drawMatches(img1, keypoints_1, img2, keypoints_2, matches[:
plt.figure(figsize=(15,8))
plt.imshow(cv.cvtColor(img3,cv.COLOR_BGR2RGB)),plt.show()
```



Out[ ]:

```
(<matplotlib.image.AxesImage at 0x23cfa57b640>, None)
```