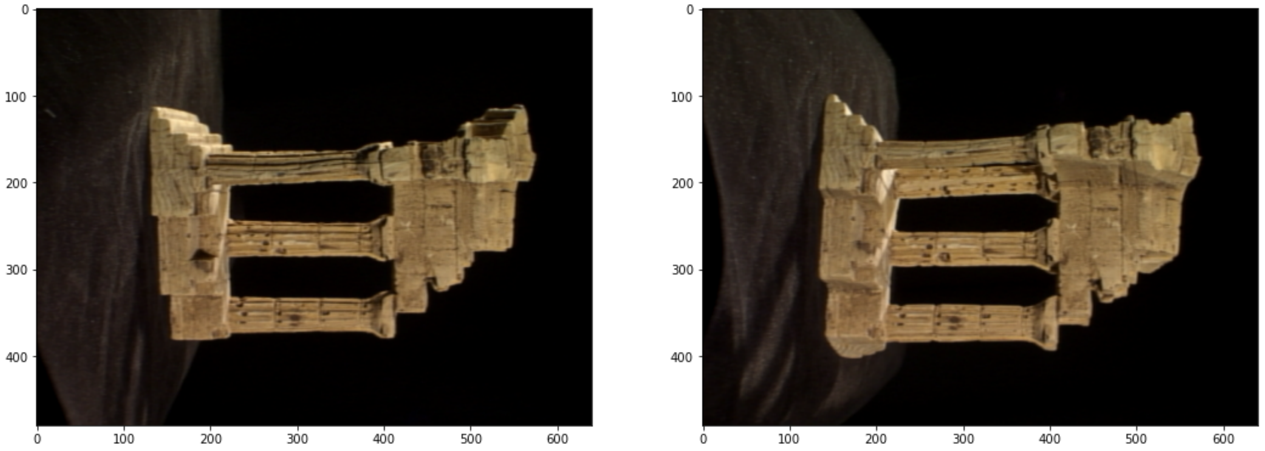
Exercise 9

Index No. : 190696U

Name : Wijegunawardana C.H.W.

In [ ]:

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
f=open(r'templeSparseRing\templeSR_par.txt','r')
assert f is not None
n= int(f.readline())

#reading the information of the 2nd image
l=f.readline().split()
im1_fn=l[0]

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 of the 2nd image

l=f.readline().split()
im2_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))

#read the two images and show
im1= cv.imread(r'templeSparseRing/' + im1_fn,cv.IMREAD_COLOR)
im2= cv.imread(r'templeSparseRing/' + im2_fn,cv.IMREAD_COLOR)
assert im1 is not None
assert im2 is not None

fig, ax = plt.subplots( 1, 2, figsize = (18, 8))

ax[0].imshow(cv.cvtColor(im1, cv.COLOR_BGR2RGB))
ax[1].imshow(cv.cvtColor(im2, cv.COLOR_BGR2RGB))
plt.show()
```

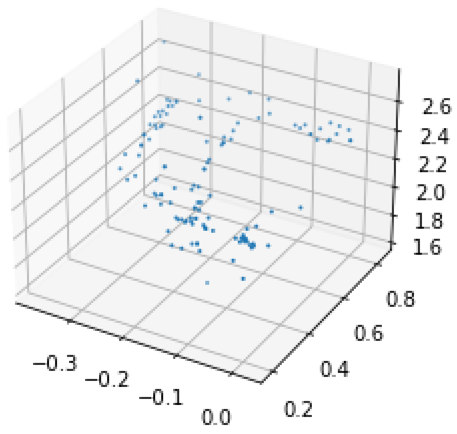


In [ ]:

```
sift = cv.SIFT_create()
kp1, decs1 = sift.detectAndCompute(im1, None)
kp2, decs2 = sift.detectAndCompute(im2, None)

FLANN_INDEX_KDTREE = 1
index_params = dict(algorithm =FLANN_INDEX_KDTREE, trees = 5 )
search_params = dict(checks=100)
flann = cv.FlannBasedMatcher(index_params, search_params)
matches = flann.knnMatch(decs1, decs2, k=2)

good = []
pts1 = []
pts2 = []
for i, (m,n) in enumerate(matches):
    if m.distance < 0.7*n.distance:
        good.append(m)
        pts1.append(kp1[m.queryIdx].pt)
        pts2.append(kp2[m.trainIdx].pt)

pts1 = np.array(pts1)
pts2 = np.array(pts2)

F,mask = cv.findFundamentalMat(pts1, pts2, cv.FM_RANSAC)
print ("F:\n",F)

E = k2.T @ F @ k1
print ("E:\n",E)

retval, R, t, mask = cv.recoverPose(E, pts1, pts2, k1)

R_t_1 = np.concatenate((R1, t1), axis =1) # 3 x 4
R2_ = R1 @ R
t2_ = R1 @ t
R_t_2 = np.concatenate((R2_, t2_), axis =1)

P1 = k1 @ np.hstack((R1, t1))
P2_ = k2 @ R_t_2

points4d = cv.triangulatePoints(P1, P2_, pts1.T, pts2.T)
points4d /= points4d[3, :]
import matplotlib.pyplot as plt
X = points4d[0, :]
Y = points4d[1, :]
Z = points4d[2, :]
```

```
fig = plt.figure(1)
ax = fig.add_subplot(111, projection='3d')

ax.scatter(X, Y, Z, s=1, cmap='gray')
plt.show()
```

```
F:
 [[ 5.89765040e-07  2.55977675e-06 -3.07847061e-02]
 [ 5.16444015e-06 -3.34348162e-07 -1.30738862e-03]
 [ 2.85158915e-02 -2.54311096e-03  1.00000000e+00]]
E:
 [[ 1.36331040e+00  5.93862667e+00 -4.55731937e+01]
 [ 1.19813894e+01 -7.78486500e-01  2.61515449e-01]
 [ 4.55650723e+01 -2.82563077e+00 -2.64674836e-02]]
```


2.6
2.4
2.2
2.0
1.8
1.6
0.8
0.6
0.4
0.2
−0.3
−0.2
−0.1
0.0