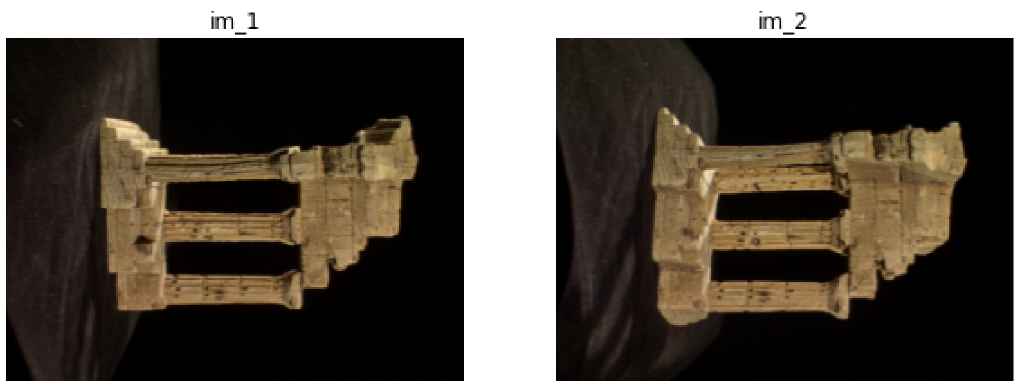
Name: C. J. Kurukulasuriya

Index No.: 190337X

Q1 and Q2

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv
```

In [ ]:
```
f = open(r'templeSparseRing/templeSR_par.txt', 'r')
assert f is not None

n = int(f.readline())

#first 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))

#second 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))

fig, ax = plt.subplots(1, 2, figsize = (10, 6))

im_1 = cv.imread(r'templeSparseRing/' + im1_fn, cv.IMREAD_COLOR)
im_2 = cv.imread(r'templeSparseRing/' + im2_fn, cv.IMREAD_COLOR)
ax[0].imshow(cv.cvtColor(im_1, cv.COLOR_BGR2RGB))
ax[0].set_title('im_1')
ax[1].imshow(cv.cvtColor(im_2, cv.COLOR_BGR2RGB))
ax[1].set_title('im_2')

for i in range(2):
    ax[i].axis('off')
plt.show()

#Compute P1 and P2
P1 = K1 @ np.hstack((R1, t1))
P2 = K2 @ np.hstack((R2, t2))
```



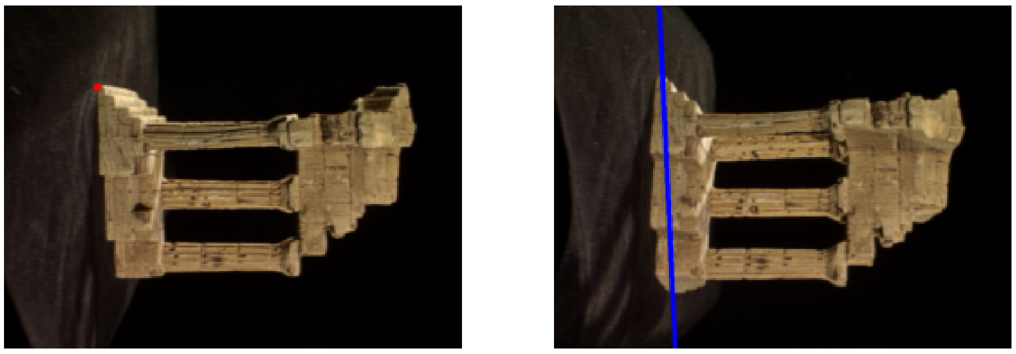
Q3

In [ ]:
```
from scipy.linalg import null_space
```

In [ ]:
```python
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)
F
```

Out[ ]:
```
array([[-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 [ ]:
```python
import matplotlib.patches as patches

x = np.array([130,115,1])
cv.circle(im_1,(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(im_2, (p1[0],p1[1]), (p2[0],p2[1]), (255,0,0), 5)

fig,ax = plt.subplots(1,2,figsize=(10,6))
ax[0].imshow(cv.cvtColor(im_1,cv.COLOR_BGR2RGB))
ax[1].imshow(cv.cvtColor(im_2,cv.COLOR_BGR2RGB))

for i in range(2):
    ax[i].axis('off')
plt.show()
```



Q4

In [ ]:
```python
def drawlines(im_1, im_2, lines, pts1, pts2):
    r,c = im_1.shape
    im_1 = cv.cvtColor(im_1, cv.COLOR_GRAY2BGR)
    im_2 = cv.cvtColor(im_2, cv.COLOR_GRAY2BGR)

    for r, pt1, pt2 in zip(lines, pts1, pts2):
        color = tuple(np.random.randint(0,255,3).tolist())
        x0, y0 = map(int, [0, -r[2]/r[1]])
        x1, y1 = map(int, [c, -(r[2]+r[0]*c)/r[1]])
        im_1 = cv.line(im_1, (x0,y0), (x1,y1), color, 1)
        im_1 = cv.circle(im_1, tuple(pt1), 5, color, -1)
        im_2 = cv.circle(im_2, tuple(pt2), 5, color, -1)

    return im_1, im_2
```

In [ ]:
```python
im_1 = cv.imread('templeSparseRing/'+im1_fn, 0)
im_2 = cv.imread('templeSparseRing/'+im2_fn, 0)
```

```python
sift = cv.SIFT_create()

keypoint1, descriptor1 = sift.detectAndCompute(im_1, None)
keypoint2, descriptor2 = sift.detectAndCompute(im_2, None)

FLANN_INDEX_KDTREE = 1
index_params = dict(algorithm = FLANN_INDEX_KDTREE, trees = 5)
search_params = dict(checks=50)
flann = cv.FlannBasedMatcher(index_params, search_params)
matches = flann.knnMatch(descriptor1, descriptor2, k = 2)
pts1 = []
pts2 = []

for i, (m,n) in enumerate(matches):
    if m.distance < 0.8 * n.distance:
        pts1.append(keypoint1[m.queryIdx].pt)
        pts2.append(keypoint2[m.trainIdx].pt)

pts1 = np.int32(pts1)
pts2 = np.int32(pts2)
F, mask = cv.findFundamentalMat(pts1, pts2, cv.FM_LMEDS)

pts1 = pts1[mask.ravel()==1]
pts2 = pts2[mask.ravel()==1]

lines1 = cv.computeCorrespondEpilines(pts2.reshape(-1,1,2), 2,F)
lines1 = lines1.reshape(-1,3)

im_5,im_6 = drawlines(im_1,im_2,lines1,pts1,pts2)
lines2 = cv.computeCorrespondEpilines(pts1.reshape(-1,1,2), 1,F)
lines2 = lines2.reshape(-1,3)

im_3, im_4 = drawlines(im_2, im_1, lines2, pts2, pts1)

plt.figure(figsize=(10, 6))

plt.subplot(121)
plt.imshow(im_5)
plt.axis('off')

plt.subplot(122)
plt.imshow(im_3)
plt.axis('off')

plt.show()
```